TaxaTreeMapper: A Novel Algorithm for Phylogenetic Ancestral State Reconstruction Using Set Theory

Osama A. Salman[1], and Gábor Hosszú[2]

***Abstract*—To determine evolutionary relationships, it is crucial to conduct phylogenetic ancestral state reconstruction. Although widely used, existing algorithms, such as Fitch's, are challenged by the computational demands of complex datasets. This study introduces the TaxaTreeMapper algorithm, which presents a streamlined approach that optimizes phylogenetic analysis. TaxaTreeMapper reduces computational time without compromising accuracy by performing ancestral state reconstruction in a single 'leaf-to-root' traversal. Our comparative study shows that TaxaTreeMapper correlates strongly with the Fitch algorithm and demonstrates superior efficiency, especially in identifying global minima in extensive datasets. This makes it significant in large-scale evolutionary studies.**

***Index Terms*—Algorithmic efficiency, Ancestral state reconstruction, Data processing in phylogenetics, Evolutionary tree optimization, Fitch algorithm, Machine learning applications in phylogenetics, Parsimony score, Phylogenetic analysis**

I. INTRODUCTION

THE quest for accurate ancestral state reconstruction in phylogenetics often encounters significant challenges, particularly with algorithms like Fitch's, which, while being intuitive and simple, may falter in cases of complex evolution or convergence [1]. Ancestral state reconstruction combines information about the evolutionary relationships of phylogenetic trees with the observed state of individual nodes. Each node represents a single taxon (taxonomic unit) [2]. Complex evolution in this context implies scenarios where evolutionary paths are shaped by multiple factors such as frequent mutations, horizontal gene transfer, genetic drift, or hybridization. These elements introduce intricacies in evolutionary histories, making accurate reconstruction a challenging endeavor.

A common limitation among many algorithms that seek to reconstruct common ancestors and determine the minimum parsimony score for a given tree is their reliance on a two-stage process: the 'leaf-to-root' followed by the 'root-to-leaf' traversal.

[1] Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics, and Doctoral School of Informatics, Department of Electron Devices, Budapest, Hungary (E-mail: osamaalisalman.khafajy@edu.bme.hu)

[2] Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics, Department of Electron Devices, Budapest, Hungary (E-mail: hosszu.gabor@vik.bme.hu)

DOI: 10.36244/ICJ.2025.5.2

Bidirectional tree traversal is a technique where the traversal progresses both from the root to the leaves and from the leaves to the root of a tree. This approach is beneficial in scenarios requiring information aggregation from both directions to make decisions at each node. A notable application of bidirectional tree traversal is in robot motion planning, where such a strategy enhances efficiency and avoids complex boundary value problems [3].

This two-pronged approach, while effective in certain contexts, often leads to increased computational complexity and may not always yield the most optimized results in terms of global minimum parsimony score.

1) Novel Contribution

This study presents a new method, TaxaTreeMapper, which is designed to traverse a given phylogenetic tree and determine its minimum parsimony score directly. This approach contributes to a more optimized method for identifying the global minimum. Our method seeks to address the limitations by reducing the computational process to a single-stage traversal. This not only simplifies the analysis but also reduces the computational effort required, making it a significant step forward in the pursuit of efficient phylogenetic analysis. By streamlining the process of ancestral state reconstruction, our approach aims to enhance the efficiency of phylogenetic tree evaluations, particularly in complex evolutionary scenarios.

While the TaxaTreeMapper algorithm enhances the efficiency of phylogenetic analysis by significantly reducing computational time, often less than that required by the Fitch algorithm, it is also designed to reliably identify all global minima in a given set of phylogenetic trees. However, it should be noted that alongside these global minima, TaxaTreeMapper may also occasionally include some local minima, erroneously presenting them as global. Despite this, the set of solutions provided by TaxaTreeMapper will contain all the true global minima, ensuring comprehensive coverage of the most parsimonious trees.

The article is structured as follows: First, it presents the concepts necessary for the theoretical background of the research, including phylogenetic inference methods, primarily the Fitch algorithm. Second, it presents the

developed method called TaxaTreeMapper algorithm, followed by the achieved results and their discussion. Finally, the article concludes with a summary of findings and a list of references cited in the literature.

II. BACKGROUND

1) Pattern systems, pattern evolution and scriptinformatics

A pattern system is a type of symbolic communication that includes symbols, syntax, and layout rules. Some pattern systems, like Morse code and Unicode, have evolved over time. The study of the evolution of pattern systems is called pattern evolution research. Human writing systems, or scripts, are a distinct type of pattern system. The study of the evolution of scripts is known as scriptinformatics, a subfield of pattern evolution. The TaxaTreeMapper phylogenetic inference algorithm was initially developed for evolutionary modeling of scripts, but it can also be applied to other taxa. TaxaTreeMapper has a broad scope beyond just scriptinformatics and belongs to any kind of evolutionary research.

Understanding the evolution and classification of different taxa has always been at the heart of scientific research. Various mathematical, computational, and heuristic models have been proposed over the years, aiming at creating a more structured and accurate representation of the evolutionary process. Significant advancements have been made in pattern systems, especially when applied to historical scripts [4] established a three-layer logical relationship for glyphs, further improved by the addition of a semantic layer in [5]. Recent contributions by [6] introduced the style layer, enhancing the depth of the analysis. Hosszú's emphasis on glyph complexity as a metric for reliability in comparing graphemes provides a foundation for comparative studies [7]. Such advancements have found applications in differentiating scripts using cluster analysis [8] and leveraging neural networks for glyph similarity studies [9].

In this study, we address the terminology used to describe taxonomic traits in phylogenetic analysis, a matter of great importance for ensuring clarity and precision. While 'character' is a term traditionally used in phylogenetics to denote the attributes or traits of organisms, in the field of scriptinformatics and in certain phylogenetic contexts, the term 'feature' is often utilized interchangeably. This duality in terminology is evident in recent research, such as the work [10] where features in evolutionary analysis of script variants are critically examined. Similarly, [11] employ the term in the context of phylogenetic analysis of script varieties, demonstrating its relevance and application. Further, their 2022 study on a phenetic approach to script variants also underscores the interchangeable use of these terms [12]. For the purposes of this article, we adopt this dual terminology, using 'feature' and 'character' interchangeably, with the understanding that both refer to taxonomic traits in our phylogenetic analysis. This approach aligns with broader scientific discourse and avoids potential ambiguity, particularly in scriptinformatics where 'character' might otherwise be confused with 'grapheme' or 'symbol'.

2) Phylogenetic inference methods

Phylogenetic methods have expanded their applicability beyond just biological evolution. For instance, its usage in linguistics has paved the way for constructing evolutionary trees for languages [13]. Phylogenetic analysis, especially with its parsimonious approach rooted in the Ockham's razor principle [14], has been paramount in creating hierarchical taxonomic structures. Delineation of synapomorphies further emphasizes the model's capability to account for a vast number of features in a simplistic manner [15].

Two significant comparative criteria, Maximum Parsimony [16] and Maximum Likelihood (ML), have emerged as primary techniques for tree optimization [11]. While MP revolves around the parsimony principle, ML uses probabilistic models to evaluate evolutionary event likelihoods. The Bayesian approach, exemplified by MrBayes software, further exemplifies the nuanced relationship between data and tree probabilities [17].

When exact and exhaustive searches are too costly or time-consuming, heuristic methods become necessary. While these approaches aim to approximate optimal solutions in the solution space, they cannot guarantee the identification of the globally optimal phylogenetic tree. To enhance heuristic search efficiency and improve upon the phylogenetic trees constructed, a branch-swapping algorithm, known as 'swapping' [10].

The Subtree Prune and Regraft (SPR), the Nearest Neighbor Interchange (NNI), and Tree Bisection and Reconnection (TBR), each come with their unique attributes, with TBR being the most computational but potentially offering the shortest tree [17, 18]. The present research focuses on phylogenetic inference methods, which involve searching for optimal phylogenetic trees. Only models where the evolutionary process can be estimated with a tree, rather than a network, are considered.

The search for the most realistic phylogenetic tree, despite its comprehensiveness, faces challenges with larger datasets [18]. Alternative heuristic methods like the Wagner method [19], the Branch and Bound technique [20] and Hill-Climbing [21] offer solutions with varied degrees of optimality and computational efficiency. Regarding Hill-Climbing algorithm is effective for finding local optima in phylogenetic trees by refining initial configurations, focusing on measures like parsimony. However, it falls short of guaranteeing the global optimum, often getting trapped in local optima. This underscores the necessity for supplementary methods to circumvent such limitations and achieve a more comprehensive search for the optimal phylogenetic tree [10, 11]. Additionally, visualization tools like histograms provide insights into the distribution of tree lengths, aiding in the discernment of optimal trees.

Matrix-based approaches in phylogenetics offer a variety of methods to derive evolutionary relationships among taxa. One classical group of methods, distance matrix methods, such as the Neighbor-Joining (NJ) and UPGMA, directly work with matrices that represent pairwise distances between taxa to infer a phylogenetic tree [22]. Alternatively, spectral methods exploit the eigenvalues and eigenvectors of matrices derived from genetic data. Cavender and Felsenstein's method, based on eigendecomposition of sequence similarities, is a prime example [23]. As another approach, quartet methods like the Q-method employ matrices showcasing relationships between quartet sets of taxa to infer broader trees [24]. Another avenue, character compatibility, creates a taxa by feature (aka character) matrix, checking feature compatibility to infer relationships [25]. Recent research has also highlighted the potential of algebraic statistics in phylogenetics, where algebraic techniques decode phylogenetic problems using matrix operations [26]. Lastly, phylogenetic networks, which encapsulate complex evolutionary patterns like hybridization, can be understood and analyzed using matrix representations [27].

In conclusion, the domain of phylogenetic inference has witnessed extensive research, with a multitude of evolutionary models and phylogenetic inference algorithms proposed. The ultimate objective remains the construction of accurate and representative evolutionary trees, aiding in a deeper understanding of taxa evolution. As computational power and methodologies continue to evolve, it's estimated that even more sophisticated models will emerge, bridging any existing gaps in the space of phylogenetics.

3) Fitch parsimony and algorithms

Fitch's contributions to the field of phylogenetics are evident through his development of distinct methods that address the reconstruction of evolutionary histories. One such method is the *Fitch parsimony*, which operates on a parsimony principle aiming to discern the evolutionary tree with the least number of feature state (aka character state) changes. Crucially, this method accommodates multistate features, allowing them to be disordered and unpolarized, meaning that transitions between any feature states are possible in a single evolutionary step. This principle is computationally manifested in the *Fitch algorithm*, which calculates the parsimony score, indicating the total number of feature state transitions for a specific tree topology [28]. The Fitch algorithm is a fundamental method in the field of phylogenetics. It was introduced by Walter Fitch in the 1970s and has since become a fundamental tool for ancestral state reconstruction based on parsimony principles [28]. On the other hand, Fitch, in collaboration with Margoliash, devised the *Fitch-Margoliash Phylogenetic Inference Algorithm*. Instead of feature states, this method is grounded on genetic distance data. Utilizing a weighted least squares clustering approach, it emphasizes the accuracy of genetic distances between species in the tree, giving more weight to closely related sequences. This method offers higher accuracy, albeit at the expense of computational efficiency when compared to alternatives like the neighbor-joining technique [29].

Fitch algorithm is a commonly used tool for ancestral state reconstruction based on parsimony methods. This algorithm works by minimizing the number of evolutionary changes [28] along the branches of a phylogenetic tree. Fitch algorithm, while not directly calculating the total length or Maximum Parsimony [16] score of a phylogenetic tree in a single computation, effectively minimizes the number of evolutionary changes across the tree. This minimization is achieved indirectly through the algorithm's two-pass process. In the first pass, the algorithm performs a bottom-up traversal of the tree, during which it identifies the possible feature states for each internal node without assigning specific branch lengths. In the second pass, a top-down traversal assigns definitive states to these nodes [30, 31].

During this process, the Fitch algorithm seeks to minimize the number of state changes at each step. The branch lengths, defined as the number of feature state changes between nodes, are indirectly determined through this process. The overall tree length, representing the sum of these branch lengths, is thus a result of the algorithm's optimization of state changes at each individual node and branch, rather than a direct calculation of the total tree length [30].

Fitch algorithm, originally developed for the parsimony-based reconstruction of phylogenies, is inherently designed to handle bifurcating or binary trees. Its two-phase traversal approach, involving postorder and preorder tree traversals, is optimized for dichotomous branching. When faced with polytomous trees, or trees with nodes having more than two descendants, the Fitch algorithm encounters challenges. Polytomies, which can be seen as either unresolved evolutionary relationships (soft polytomies) or simultaneous divergence events (hard polytomies), don't fit neatly into the binary framework of Fitch's method [32]. Adapting the algorithm to cater to these non-binary nodes introduces complexities and requires additional considerations or modifications. While some phylogenetic software tools have developed strategies to handle or resolve polytomies, the inherent limitation of Fitch's original design concerning polytomies remains a recognized challenge in the field of phylogenetics [16].

The Fitch algorithm employs a two-stage process for ancestral state reconstruction, beginning at the leaf nodes with known genetic states and moving toward the root to infer the most parsimonious common ancestor at each internal node as illustrated in Figure 1. For example, if taxa A and B both have a state of '1' for a particular characteristic, their common ancestor is presumed to also have the state of '1'. When discrepancies arise (e.g., A=1, B=0), the ancestor may inherit a set that includes

both states. In the next stage, the algorithm resolves these sets by working from the root back to the leaves, selecting states that minimize changes across the tree. While effective, the Fitch method can be computationally intensive for large datasets.

4) Pearson correlation

The Pearson correlation coefficient (also known as Pearson's r or simply the correlation coefficient) measures the linear relationship between two variables, typically denoted as X and Y. The formula for calculating the Pearson correlation coefficient is as follows:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}, \quad (1)$$

where n is the number of data points (the size of the dataset), X_i and Y_i are the individual data points of the variables X and Y, and finally $\bar{X}$ and $\bar{Y}$ are the mean (average) values of X and Y, respectively.

The incorporation of Pearson correlation in our study is integral to the assessment of phylogenetic relationships and evolutionary analysis. Pearson's correlation, a measure of the linear correlation between two variables, provides a quantifiable means to assess the degree of similarity or divergence between different taxa based on their phylogenetic profiles. This statistical tool is particularly effective in discerning the strength and direction of a linear relationship between two sets of data, which in our context, are the phylogenetic traits or features of different organisms. By applying Pearson correlation, we can systematically compare these traits to draw inferences about evolutionary patterns and relationships. Our approach aligns with contemporary advancements in phylogenetic profiling, where measures like Pearson correlation have been used to infer global protein-protein interactions and handle large genomic datasets effectively, as demonstrated in the study of Saccharomyces cerevisiae and Escherichia coli genomes [33]. This method's efficacy, especially in comparison to other measures such as mutual information and distance correlation, underscores its relevance and utility in our analysis.

III. METHOD

In this study, we introduce a novel methodological approach to determine the maximum parsimony in phylogenetic trees, distinguished by its efficient single-stage process. This approach is compared with the established Fitch algorithm, a keystone in phylogenetic analysis, known for its two-stage 'leaf-to-root' and 'root-to-leaf' traversal process. Notably, the Fitch algorithm, which has been widely used for phylogenetic tree reconstruction and serves as a benchmark in our comparative analysis.

Unlike conventional two-stage methods such as the Fitch algorithm, which operate through both 'leaf-to-root' and 'root-to-leaf' stages, our method simplifies the analysis by focusing exclusively on a 'leaf-to-root' traversal. This innovation effectively halves the computational steps typically required, as it eliminates the need for the subsequent 'root-to-leaf' stage. Conceptually, if Y represents the computational effort of a traditional method like Fitch's, then our method, X, can be said to operate at $\frac{Y}{2}$ in terms of computational, or mathematically, $f(X) = 2 \cdot X$ in terms of efficiency.

This significant enhancement not only accelerates the analytical process but also maintains the accuracy and robustness needed for phylogenetic studies. Our approach (TaxaTreeMapper) represents a substantial advancement in phylogenetic analysis, offering a more streamlined and time-efficient solution for uncovering evolutionary relationships. This methodology, with its single-stage focus, is not only a testament to the potential for innovation in phylogenetic analysis but also a practical solution that addresses the computational challenges often encountered in extensive biological datasets.

The TaxaTreeMapper algorithm applies set theory operations to clarify phylogenetic relationships. It starts with a 'leaf-to-root' assessment, as in Figure 1 where taxa *A*, *B*, and *C* are compared for their feature states. Discrepancies between taxa, like $A = 1$ and $B = 0$, lead to an interim ambiguous state $Z = 2$. The algorithm resolves this by checking the overlap with any resolved neighboring node states. If taxon *C* also has the state of 1, then the algorithm concludes the ancestral state *R* to be 1, through the intersection with the ambiguous state. Illustrated in Figure 1, this method streamlines the determination of the most likely internal node states, improving the precision of phylogenetic tree reconstruction.

The TaxaTreeMapper streamlines the process for ancestral state reconstruction into a single stage. It also starts at the leaves, but as it ascends the tree, it uses information from the subsequent ancestor (e.g., ancestor of A & B derives its state from C) to determine the states of intermediate ancestors directly refer to Figure 1. This approach not only simplifies the state determination process but also allows for simultaneous calculation of the tree length and the total number of changes, enhancing efficiency particularly for extensive datasets.

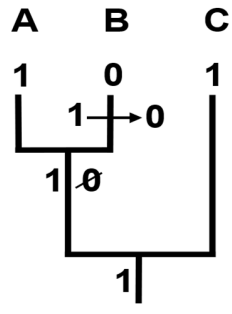

Figure 1: 'Leaf-to-Root' ancestral state reconstruction in TaxaTreeMapper

Algorithm 1: The main steps of the TaxaTreeMapper

Input: A node of a phylogenetic tree (root node to start), dataset
Output: The inferred state for the node, treeLength for the entire tree

Function TaxaTreeMapper(node, dataset):
1. If node is a leaf:
  1.1. Return the state of the leaf node from the dataset, and 0 as the treeLength.
2. Initialize Gab as an empty set for accumulating the node's inferred state.
3. Initialize localTreeLength = 0 to track state changes at this node.
4. For the first child of the node, establish a reference state (Ga):
  4.1. Ga, childTreeLength = TaxaTreeMapper(first child, dataset).
  4.2. Set Gab to Ga initially.
  4.3. Update localTreeLength += childTreeLength.
5. For each remaining child k (starting from the second child to the last):
  5.1. Gb, childTreeLength = TaxaTreeMapper(k, dataset).
  5.2. Update localTreeLength += childTreeLength.
  5.3. Perform intersection and union operations:
  5.4. Intersection: If Ga ∩ Gb is not empty, Gab = Gab ∩ Gb.
  5.5. Union with resolution: If Ga ∩ Gb is empty, then Gab = Gab ∪ Gb, but resolve `{2}` where possible:
  5.6. For each feature in Gab marked as `{2}`, if Gb has a known state, replace `{2}` in Gab with Gb's state.
  5.7. Conversely, for each `{2}` in Gb and known in Ga, update Gab accordingly.
  5.8. Determine unique changes: uniqG = (Ga Δ Gb) - `{2}` elements, where Δ represents the symmetric difference.
  5.9. Update localTreeLength for each unique change not involving `{2}`, as these represent evolutionary events.
  5.10. Mark unresolved differences as `{2}` in Gab for the next iteration.
6. After processing all children, the dataset is updated with the resolved state Gab for the internal node.
7. Return Gab as the node's state and localTreeLength.

Ambiguous states, denoted by a predetermined value within the dataset, are systematically managed by the TaxaTreeMapper algorithm, particularly in steps 5.4 and 5.6 These steps incorporate ambiguous states into the intersection operation, ensuring that uncertainties in data do not compromise the accuracy of the phylogenetic analysis.
Furthermore, the calculation of the symmetric difference between sets identifying features unique to each taxon occurs in step 5.8. This difference highlights the evolutionary divergence and is crucial for calculating the tree length, representing the extent of evolutionary adaptations since the taxa branched from their last common ancestor.
The algorithm then updates the dataset with a new set representing hypothetical ancestral taxa in step 6. This new set, a combination of intersected and unique elements, is pivotal for updating the feature set for each node in a 'leaf-to-root' traversal of the phylogenetic tree. The 'leaf-to-root' traversal, the primary focus of the TaxaTreeMapper algorithm, simplifies the analysis process and enhances the precision of phylogenetic inference by compiling shared and distinctive traits accurately. This is illustrated in steps 4 through 7 of the TaxaTreeMapper Algorithm 1.
Initially, the TaxaTreeMapper determines the number of taxa from the given dataset and initializes various variables and counters to their respective default values. The primary focus then shifts to traversing the phylogenetic tree sequence, where each feature is processed in sequence.

Beginning at the top of Algorithm 1, the algorithm initializes the necessary variables, including the dataset. The main input and output in this case the phylogenetic tree *node, dataset* as input and *treeLength* and inferred state for the phylogenetic tree as an output.

IV. RESULT AND DISCUSSION

This section presents a comparative analysis of the TaxaTreeMapper algorithm against Fitch algorithm, focusing on tree length determination, computational efficiency, and the handling of cladograms in phylogenetic analysis. We then discuss the inherent advantages of the TaxaTreeMapper algorithm, underpinned by the empirical results.

1) Comparative Analysis

In Figure 2, we illustrate the comparative analysis of tree lengths generated by the TaxaTreeMapper algorithm and the Fitch algorithm. The histogram in Figure 2.a, the sorted length curves in Figure 2.b, and the boxplot in Figure 2.c collectively highlight the similarity in tree length calculations and the distinct efficiencies between the two methods. Our findings suggest that TaxaTreeMapper consistently identifies the global minimum for maximum parsimony trees more efficiently than the Fitch process, which relies on a two-phase approach.

Figure 2 underscores the significant overlap in tree length evaluations between TaxaTreeMapper and Fitch across approximately 2.5 million diverse phylogenetic trees. This comparison validates the efficiency of TaxaTreeMapper in closely matching the established Fitch method while using a single-phase approach.

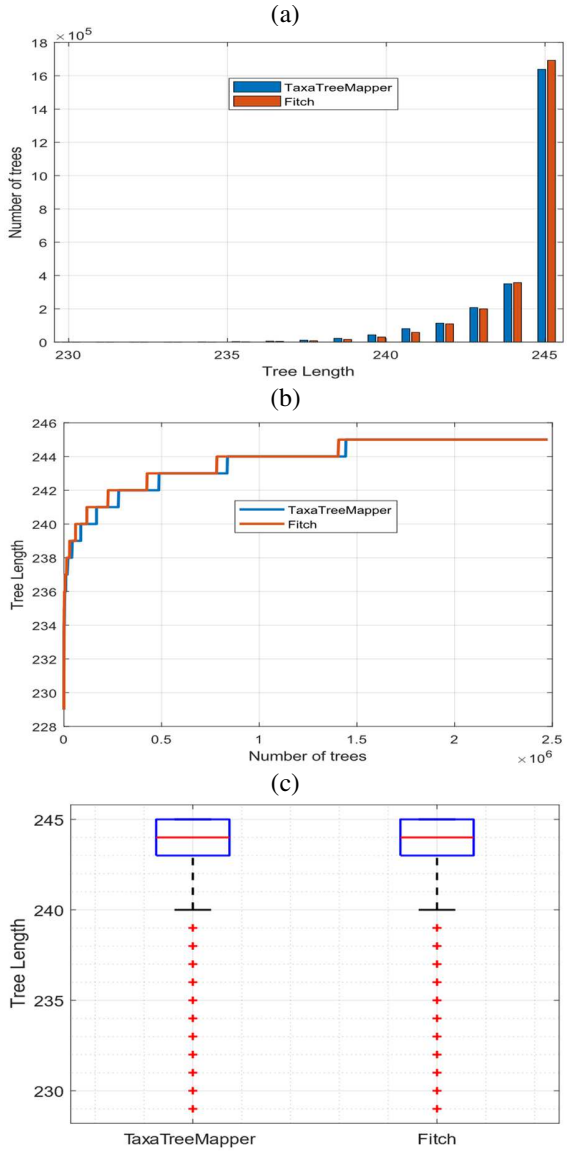

Figure 2: Comparative Efficiency of Tree Length Determination between TaxaTreeMapper and Fitch Algorithms

Despite the foundational differences in their operational stages—TaxaTreeMapper using a single-phase approach versus the two-phase process of Fitch—both methods consistently identified the global minimum tree length and exhibited a significant overlap in their evaluations of near-optimal tree lengths. This comparison not only demonstrates the algorithms' capability to accurately determine the most parsimonious tree but also validates the efficiency of TaxaTreeMapper in achieving results that align closely with the established Fitch method.

2) Pearson Correlation Coefficient Analysis

A Pearson correlation coefficient (r) analysis further substantiates the similarity between the algorithms. With r values of 0.91 for the same amount of trees that been tested in the comparative analysis. The analysis confirms a strong positive linear relationship between the tree lengths determined by TaxaTreeMapper and those by Fitch, indicating a convergence towards a global minimum by the TaxaTreeMapper algorithm.

It is noteworthy that when r equals 0.91, the estimated number of trees was approximately 2.5×10^6. Conversely, when r equals 0.956, the number of trees was precisely 2988. This observation provides a clear indication that the TaxaTreeMapper algorithm converges towards a global minimum.

3) Computational Efficiency

In Figure 3, we present of the running time of TaxaTreeMapper algorithm and Fitch algorithm. The elapsed time measurements clearly illustrate that TaxaTreeMapper outperforms the Fitch algorithm in terms of computational efficiency.

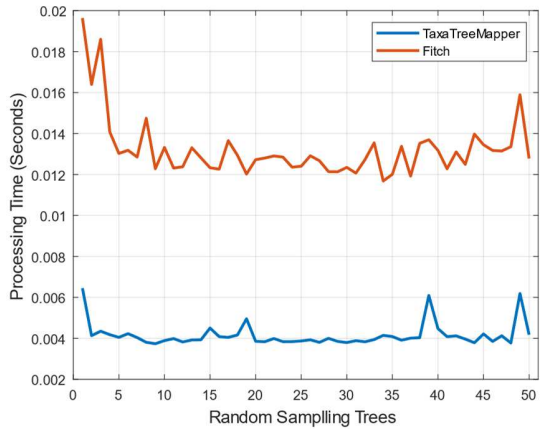

Figure 3: Tree processing time for TaxaTreeMapper algorithm verses Fitch algorithm.

The experimental evaluation was conducted on a system running Windows 10 Pro Version 22H2, equipped with an Intel(R) Core(TM) i7-2720QM CPU at 2.20GHz and 20GB of RAM, operating on a 64-bit architecture. The performance metrics for both the TaxaTreeMapper and Fitch algorithms were obtained using implementations coded in MATLAB R2023b. This hardware and software environment was chosen to ensure a consistent and controlled platform for benchmarking the computational efficiency of the phylogenetic analysis algorithms under investigation.

In Figure 4, the results show that TaxaTreeMapper generated four cladograms with identical minimum parsimony scores 229, also known as tree length. In contrast, the application of the Fitch algorithm yielded only two cladograms (a and b) as a global minimum. On other hands, the TaxaTreeMapper algorithm identified cladograms Figure 4.c and Figure 4.d as having a global minimum parsimony [16] score of 229. However, the Fitch algorithm attributed these same cladograms with higher tree lengths of 234. This divergence in scores initially suggests that TaxaTreeMapper incorrectly assess

these cladograms as optimal. While it may appear as a limitation, a closer examination of the Pearson correlation coefficient (r) between the tree lengths calculated by TaxaTreeMapper and Fitch reveals a high degree of correlation across the dataset. This indicates that, despite the identified discrepancies, the TaxaTreeMapper algorithm performs consistently with the Fitch algorithm for most cases.

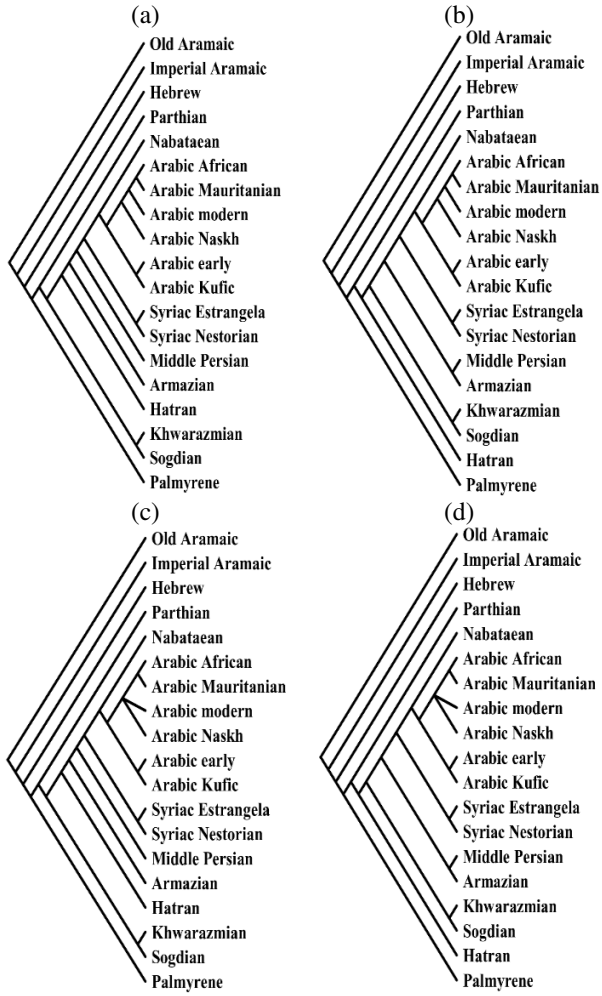

Figure 4: Comparative Cladograms of Semitic Scripts Generated by TaxaTreeMapper Algorithm

4) Empirical Validation

Table 1 quantifies the runtime efficiency of the TaxaTreeMapper algorithm, denoted as α, against the Fitch algorithm, denoted as β, across 50 random sampling trees. The use of α and β provides a simplified notation that facilitates the mathematical comparison of runtimes. The column labeled α to represent the runtime of TaxaTreeMapper, while the column labeled β for the Fitch algorithm's runtime. This symbolic representation streamlines the discussion and calculation of the efficiency metric, particularly in the subsequent computation of the Mean Squared Error (MSE).

The mean squared error (MSE), calculated is equal to 6.48×10^{-6}, is derived from the squared differences between α and $\frac{\beta}{2}$. A lower MSE corroborates the hypothesis that TaxaTreeMapper is about twice as fast as the Fitch algorithm.

TABLE I
COMPARATIVE RUNTIME ANALYSIS OF TAXATREEMAPPER (TTM) AND FITCH ALGORITHMS ACROSS PHYLOGENETIC TREES

	Runtime			$\left(\alpha - \frac{\beta}{2}\right)^2$
	α	β	$\frac{\beta}{2}$	
1	0.006	0.02	0.01	0.000016
2	0.004	0.016	0.008	0.000016
3	0.004	0.019	0.0095	0.00003025
4	0.004	0.014	0.007	0.000009
5	0.004	0.013	0.0065	0.00000625
6	0.004	0.013	0.0065	0.00000625
7	0.004	0.013	0.0065	0.00000625
8	0.004	0.015	0.0075	0.00001225
9	0.004	0.012	0.006	0.000004
10	0.004	0.013	0.0065	0.00000625
11	0.004	0.012	0.006	0.000004
12	0.004	0.012	0.006	0.000004
13	0.004	0.013	0.0065	0.00000625
14	0.004	0.013	0.0065	0.00000625
15	0.005	0.012	0.006	0.000001
16	0.004	0.012	0.006	0.000004
17	0.004	0.014	0.007	0.000009
18	0.004	0.013	0.0065	0.00000625
19	0.005	0.012	0.006	0.000001
20	0.004	0.013	0.0065	0.00000625
21	0.004	0.013	0.0065	0.00000625
22	0.004	0.013	0.0065	0.00000625
23	0.004	0.013	0.0065	0.00000625
24	0.004	0.012	0.006	0.000004
25	0.004	0.012	0.006	0.000004
26	0.004	0.013	0.0065	0.00000625
27	0.004	0.013	0.0065	0.00000625
28	0.004	0.012	0.006	0.000004
29	0.004	0.012	0.006	0.000004
30	0.004	0.012	0.006	0.000004
31	0.004	0.012	0.006	0.000004
32	0.004	0.013	0.0065	0.00000625
33	0.004	0.014	0.007	0.000009

34	0.004	0.012	0.006	0.000004
35	0.004	0.012	0.006	0.000004
36	0.004	0.013	0.0065	0.00000625
37	0.004	0.012	0.006	0.000004
38	0.004	0.014	0.007	0.000009
39	0.006	0.014	0.007	0.000001
40	0.004	0.013	0.0065	0.00000625
41	0.004	0.012	0.006	0.000004
42	0.004	0.013	0.0065	0.00000625
43	0.004	0.012	0.006	0.000004
44	0.004	0.014	0.007	0.000009
45	0.004	0.013	0.0065	0.00000625
46	0.004	0.013	0.0065	0.00000625
47	0.004	0.013	0.0065	0.00000625
48	0.004	0.013	0.0065	0.00000625
49	0.006	0.016	0.008	0.000004
50	0.004	0.013	0.0065	0.00000625

Figure 3 visually depicts the data presented in Table 1, illustrating the runtime comparison between the TaxaTreeMapper and Fitch algorithms for each of the 50 phylogenetic trees sampled. The graphical representation allows for an immediate visual grasp of the runtime dynamics where TaxaTreeMapper consistently outperforms Fitch, as indicated by the shorter processing times.

V. CONCLUSIONS

The TaxaTreeMapper algorithm employs set theory to enhance accuracy and efficiency in processing phylogenetic trees. By integrating with associated datasets, it simplifies analysis and accurately identifies evolutionary features. It adeptly handles complex relationships and large datasets, providing outputs such as tree length and hypothetical taxa.

By condensing the ancestral state reconstruction into a single traversal from leaf to root, TaxaTreeMapper not only simplifies the computational process but also proves to be computationally twice as efficient as the Fitch algorithm. This remarkable increase in efficiency does not come at the cost of accuracy, with TaxaTreeMapper demonstrating a strong correlation with Fitch's results in identifying global minima. The foundational principles of TaxaTreeMapper emphasize streamlining phylogenetic analysis, making it especially advantageous for handling large datasets where computational resources are at a premium.

The TaxaTreeMapper algorithm offers an innovative approach that enhances efficiency and reduces complexity. Its ability to quickly and accurately construct phylogenetic trees represents a substantial leap forward from the traditional, more time-intensive methods.

Though TaxaTreeMapper may occasionally yield false positives due to its heuristic approach diverging from Fitch's conservative estimations, its overall computational efficiency and ability to quickly converge on global minima present a compelling advantage. In extensive phylogenetic analyses, where computational resources are constrained, TaxaTreeMapper's speed and general accuracy provide a favorable balance between performance and resource utilization.

Acknowledging differences between TaxaTreeMapper and Fitch, it's crucial to weigh overall performance metrics. TaxaTreeMapper's emphasis on efficiency and speed makes it valuable in high throughput phylogenetic analysis. Thus, considering its performance profile and correlation with Fitch's results, TaxaTreeMapper stands as a robust alternative, especially in scenarios requiring rapid tree length estimations.

REFERENCES

[1] 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," in *International Journal of Plant Sciences*, 2016, vol. 177, no. 4, pp. 305–318.

[2] R. N. Randall, C. E. Radford, K. A. Roof, D. K. Natarajan, and E. A. Gaucher, "An experimental phylogeny to benchmark ancestral sequence reconstruction," in *Nature communications*, 2016, vol. 7, no. 1, p. 12 847.

[3] S. Nayak and M. W. Otte, "Bidirectional sampling-based motion planning without two-point boundary value solution," in *IEEE Transactions on Robotics*, 2022, vol. 38, no. 6, pp. 3636–3654.

[4] R. E. I. Pardede, L. L. Tóth, G. Hosszú, and F. Kovács, "Glyph Identification Based on Topological Analysis," in *Book Glyph Identification Based on Topological Analysis*, 2012, pp. 99–103.

[5] R. E. Pardede, L. L. Tóth, G. A. Jeney, F. Kovács, and G. Hosszú, "Four-layer grapheme model for computational paleography," in *Journal of Information Technology Research (JITR)*, 2016, vol. 9, no. 4, pp. 64–82.

[6] G. L. Hosszú, Scriptinformatics, in Book Scriptinformatics, Nap Kiadó, 2021.

[7] G. Hosszú, "A novel computerized paleographical method for determining the evolution of graphemes" in *Encyclopedia of Information Science and Technology, Third Edition* (IGI Global, 2015), pp. 2017–2031.

[8] L. L. Tóth, R. E. I. Pardede, G. A. Jeney, F. Kovács, and G. Hosszú, "Application of the cluster analysis in computational paleography" in *Handbook of Research on Advanced Computational Techniques for Simulation-Based Engineering* (IGI Global, 2016), pp. 525–543.

[9] S. Daggumati and P.Z. Revesz, "Convolutional Neural Networks Analysis Reveals Three Possible Sources of Bronze Age Writings between Greece and India," in *Information*, 2023, vol. 14, no. 4, p. 227.

[10] O. A. Salman, G. Hosszú, and F. Kovács, "A new feature selection algorithm for evolutionary analysis of Aramaic and Arabic script variants," in *International Journal of Intelligent Engineering Informatics*, 2022, vol. 10, no. 4, pp. 313–331.

[11] O. A. Salman and G. Hosszú, "Cladistic Analysis of the Evolution of Some Aramaic and Arabic Script Varieties," in *International Journal of Applied Evolutionary Computation (IJAEC)*, 2021, vol. 12, no. 4, pp. 18–38.

[12] O. A. Salman and G. Hosszú, "A Phenetic Approach to Selected Variants of Arabic and Aramaic Scripts," in International Journal of Data Analytics (IJDA), 2022, vol. 3, no. 1, pp. 1–23.

[13] L. L. Cavalli-Sforza, A. Piazza, P. Menozzi, and J. Mountain, "Reconstruction of human evolution: bringing together genetic, archaeological, and linguistic data," in *Proceedings of the National Academy of Sciences*, 1988, vol. 85, no. 16, pp. 6002–6006.

[14] P. D. Ashlock, "The uses of cladistics," in *Annual Review of Ecology and Systematics*, 1974, vol. 5, no. 1, pp. 81-99.
[15] I. J. Kitching, Cladistics: the theory and practice of parsimony analysis, (Oxford University Press, USA, 1998).
[16] C. Semple and M. Steel, Phylogenetics, (Oxford University Press on Demand, 2003).
[17] W. C. Wheeler, Systematics: a course of lectures, (John Wiley & Sons, 2012).
[18] E. O. Wiley and B. S. Lieberman, Phylogenetics: theory and practice of phylogenetic systematics, (John Wiley & Sons, 2011).
[19] J. S. Farris, "Methods for computing Wagner trees," in *Systematic Biology,* 1970, vol. 19, no. 1, pp. 83–92.
[20] E. J. Henley and R. A. Williams, Graph Theory in Modern Engineering: Computer Aided Design, Optimization, Reliability Analysis, (Academic Press, Inc., 1973).
[21] G. Ganapathy, V. Ramachandran, and T. Warnow, "Better hillclimbing searches for parsimony," in *Better hill-climbing searches for parsimony* (Springer, 2003), pp. 245–258.
[22] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," in *Molecular biology and evolution,* 1987, vol. 4, no. 4, pp. 406–425.
[23] J. A. Cavender and J. Felsenstein, "Invariants of phylogenies in a simple case with discrete states," in *Journal of classification*, 1987, vol. 4, pp. 57–71.
[24] G. F. Estabrook, F. McMorris, and C. A. Meacham, "Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units," in *Systematic Zoology,* 1985, vol. 34, no. 2, pp. 193–200.
[25] W. J. L. Quesne, "A method of selection of characters in numerical taxonomy," in *Systematic Zoology*, 1969, vol. 18, no. 2, pp. 201–205.
[26] B. Sturmfels and S. Sullivant, "Toric ideals of phylogenetic invariants," in *Journal of Computational Biology*, 2005, vol. 12, no. 4, pp. 457–481.
[27] D. H. Huson and D. Bryant, "Application of phylogenetic networks in evolutionary studies," in *Molecular biology and evolution*, 2006, vol. 23, no. 2, pp. 254–267.
[28] W. M. Fitch, "Toward defining the course of evolution: minimum change for a specific tree topology," in *Systematic Biology*, 1971, vol. 20, no. 4, pp. 406–416.
[29] W. M. Fitch and E. Margoliash, "Construction of phylogenetic trees: a method based on mutation distances as estimated from cytochrome c sequences is of general applicability," in *Science,* 1967, vol. 155, no. 3760, pp. 279–284.
[30] J. Felsenstein, Inferring phylogenies. Sunderland, Massachusetts: Sinauer Associates, 2004.
[31] D. Penny, M.D. Hendy, and M.A. Steel, "Progress with methods for constructing evolutionary trees," in *Trends in ecology & evolution,* 1992, vol. 7, no. 3, pp. 73–79.
[32] A. Purvis and T. Garland, "Polytomies in comparative analyses of continuous characters," in *Systematic Biology*, 1993, vol. 42, no. 4, pp. 569–575.
[33] G. Sferra, F. Fratini, M. Ponzi, and E. Pizzi, "Phylo_dCor: distance correlation as a novel metric for phylogenetic profiling," in *BMC bioinformatics,* 2017, vol. 18, no. 1, pp. 1–7.

Osama A. Salman completed his Bachelor's and Master's degrees in Computer Science at the University of Baghdad, in 2014 and 2018, respectively. He is currently a fourth-year doctoral student at the Budapest University of Technology and Economics (BME), Faculty of Electrical Engineering and Informatics.
The focus of his PhD research is on conducting a phylogenetic analysis of the evolution of writing systems including Scriptinformatics, Machine learning applications in phylogenetics, Computational biology, Data analytics and Data processing in phylogenetics. In MCS His research examines into Machine Learning, Neural Networks, and Deep Learning with a special emphasis on Behavioral Biometrics, and Data Science applications. His academic journey is marked by a strong foundation in both theoretical concepts and practical applications.

Gábor Hosszú received his M.E. degree in Electrical Engineering and his Ph.D. in Technical Sciences from the Budapest University of Technology and Economics in 1992. He also obtained an MSc in Law from Pázmány Péter Catholic University, Budapest in 2011. Currently, he is an associate professor at the Faculty of Electrical Engineering and Informatics at the Budapest University of Technology and Economics, where he has been a member since 1990. In 2013, he was awarded the title of Dr habil. in recognition of his contributions. His main activities involve statistical evaluation of bioelectronic signals and research on pattern evolution, including scriptinformatics. He has published over 250 technical papers.