УДК 378:004
DOI: 10.25686/978-5-8158-2474-4-2025-397-406

Visualization of phylogenetic self-similarity of genomic sequence

Kazem Forghani [1], Majid Forghani [2,3] and Mikhail Bolkov [4,*]
[1] Iran University of Science and Technology (IUST), Tehran, Iran
[2] Institute of Natural Sciences and Mathematics, Ural Federal University, Yekaterinburg, Russia
[3] N.N. Krasovskii Institute of Mathematics and Mechanics of the UB RAS, Yekaterinburg, Russia
[4] Institute for the Study of Aging, Russian Gerontological Scientific and Clinical Center, N.I. Pirogov Russian National Research Medical University of the Ministry of Health of the Russian Federation, Moscow, Russia

Abstract. The phylogenetic analysis is one of the fundamental analyses in bioinformatics, aiming at determining the evolutionary relationship between a set of taxa. While recent advances in sequencing technology have led to decreased costs and increased availability of whole-genome sequences, there are various studies focusing on employing a partial sequence as a genome representative in phylogenetic studies. This situation can often be encountered in the analysis of viral evolution, where the analysis is more biased towards antigen evolution and variation can be tracked through one or more coding subsequences of the genome. Such a representative may drastically decrease the cost of computation. The lack of a tool for finding a genome representative motivated us to investigate the application of distance metrics of tree space in order to highlight candidates for the role of phylogenetic representative. In this preliminary study, we propose a pipeline to compute the phylogenetic self-similarity of the genome by using a tree distance and visualizing the results in a 2D map. Though our approach is at an early stage of development, the obtained results indicate that the approach has potential as an exploratory tool in phylogenetic studies.

Keywords: visualization, phylogenetic tree, similarity, tree distance, genome.

Introduction

Phylogenetic analysis plays an important role in many studies within evolutionary biology [1]. It is one of the fundamental analyses in bioinformatics, aiming at determining the evolutionary relationship between a set of taxa. Reconstructing a phylogenetic tree from molecular data can lead to a better understanding of evolution among different groups of organisms [2]. Specifically, in the case of virus studies, it provides a valuable view of the evolutionary history of viruses and the transmission behavior of diseases, which is urgent for designing a strategy to combat them [3]. Recent advances in sequencing lead to a rapid growth of large-scale genomic data, which highlights that estimating phylogeny requires novel, fast, and accurate methods and pipelines [4].

Embedding genetic sequences into tree space is a good way of interpretation since the raw sequences are not human-readable in terms of phylogenetic characteristics. Although visualizing phylogenetic relationships as a rooted acyclic graph is simple, interpretable, and facilitates hypothesis formulation and testing, such representation cannot cover all evolutionary events [5]. Recombination, horizontal transfer, and hybridization are examples of events that are poorly described in the content of rooted acyclic graphs. Therefore, a generalization of tree structure called a phylogenetic network is employed to express the reticulate events [6, 7].

An efficient way to quickly gain overall insight about phylogeny is selecting phylogenetic representatives to approximate a phylogenetic tree of the full genome. Previous studies have also demonstrated that some regions of the genome have the potential to be selected as its phylogenetic representative. This can be visited especially in viruses, where the majority of the evolution is directed at events occurring in specific antigens. For example, a fragment of glycoprotein E (gE) from the Tick-Borne Encephalitis Virus (TBEV) complete genome is used for genotyping and classifying its isolates [8]. The fragment consists of 151 amino acid residues (aa positions 104–254), encoded by 454 nucleotides (positions 309–762) in gene E [9].

Similarity measurement deeply underpins many areas of bioinformatics, including classical topics, such as sequence alignment [10], as well as modern topics like protein function prediction and even protein folding [11]. Various types of similarity measures have been defined for genomic sequences. While some are directly related to phylogeny, others are taken from fields such as natural language processing (NLP) and fractal analysis. In this work, we focus on self-similarity, which can reflect the evolutionary behavior of a continuous fragment in relation to the entire genome. Here, we provide a brief review of fractal-based methods and those based on NLP techniques.

To the best of our knowledge, the first fractal-like visualization of genomic sequences is the Chaos Game Representation (CGR), proposed by Jeffrey [12]. The CGR method draws a unit square, with each corner representing a nucleotide (A, C, G, and T). The algorithm begins by setting the center of the square as the initial point. At each iteration, the algorithm selects the nucleotide at the position corresponding to the iteration index. It then assigns a new point that lies midway between the point associated with the previous iteration (i.e., the previous nucleotide) and the corner corresponding to the current nucleotide. A practical proof of the potential application of CGR for assessing self-similarity was provided by plotting the CGRs of 21 human chromosomes [13]. Hao et al. [14] have observed fractal-like patterns across long DNA sequences and employed them to develop a deterministic method for visualization of DNA in a 2D portrait format. As stated by the authors, the method can be used to highlight the evolutionary relatedness of species. The proposed visualization illustrates the regularities in DNA, which can be beneficial in measuring self-similarity. Recently, Durán-Meza et al. [13] studied the multifractal properties of DNA and proposed a visualization of long DNA sequences by plotting the multifractal spectra representing the level of multifractality in the genomic sequence. Although fractal-based approaches have typically been developed to measure similarity between sequences, for example, in classification tasks [15], we believe they also have potential for measuring self-similarity within genomic sequences. Despite these promising results, fractal-based approaches remain difficult to interpret.

Generally, many mathematical tools for measuring similarity require a numerical representation of data. A genomic sequence can be represented in a numerical vector space using encoding or embedding approaches. These methods typically break the sequence into k-mers while preserving their order and replace each k-mer with its numerical representation. A simple example of DNA encoding is one-hot encoding, where each nucleotide is represented by a binary vector. More advanced approaches are adapted from NLP, such as the Word2Vec framework [16]. Another popular method is positional embedding [17], a key component of transformer models. This type of embedding not only captures the relationships between words within the sequence context but also incorporates their positional information, thereby enhancing model performance.

Accordingly, various embeddings for genomic sequences have been proposed, such as ProtVec [18] and Gene2Vec [19]. These embeddings are typically trained on large datasets to capture general patterns within the data or are generated for specific tasks using specialized datasets, for example, immunological sequences. Since some species have a unique type of genome organization and specific genetic patterns, employing Word2Vec-based embeddings to identify a genome representative may not be adequate. Despite their promising results on similar tasks, these models also require training on genomic sequences of each species individually to effectively measure self-similarity within its genome. In other words, the model must be specifically retrained and adapted for each species. In addition, numerical vector embeddings of genomic sequences often increase the sequence length, leading to higher computational cost.

Our aim is to leverage phylogenetic self-similarity across a set of genome sequences to highlight potential regions of candidates for the role of genome representative. The absence of a dedicated tool motivates the development of our approach. To maintain simplicity and interpretability, the proposed method operates in the space of phylogenetic trees.

Our contribution in this paper is as follows:

- Proposing a pipeline for computing the phylogenetic self-similarity along the genome and visualizing the results in a 2D map.
- Experimentally investigating the extent to which phylogenetic representatives can approximate the phylogenetic behavior of the complete genome.

To the best of our knowledge, no such pipeline has been proposed so far. The pipeline consists of four main steps:

- Estimating a reference phylogenetic tree.
- Estimating partial phylogenetic trees based on fragmented genome sequences.
- Comparing the partial phylogenetic trees with the reference tree using a tree distance metric.
- Visualizing the resulting similarity measures along the genome.

The pipeline accepts an aligned *FASTA* file as its input, processes it, and generates the final output as a 2D visualization scene. We also investigate the phylogenetic characteristics of a coding (nucleotide) sequence of

glycoprotein E as a genome representative for tick-borne encephalitis virus (TBEV). This fragment serves as a typical sequence for determining the TBEV phylogeny and is the key element in the clusteron approach [8, 20] and its implementation in the TBEV Analyzer platform [21].

The remainder of this paper is organized as follows: Section 2 presents the methodology of the pipeline, where each step is described in detail. In Section 3, we outline the computational experiment setup and present the results. The pipeline is evaluated through numerical experiments and visualization of the TBEV genome. Finally, conclusions and directions for future work are provided in Section 4.

Methodology

Figure 1 illustrates the overall schema of the proposed pipeline. It consists of five main steps:

- Data preparation.
- Estimation of the reference phylogenetic tree from the complete sequences.
- Fragmentation of the complete sequences and construction of a phylogenetic tree for each fragment.
- Measurement of the distance between each fragment tree and the reference tree.
- Visualization of the results in a 2D map, where the x-axis represents the position of the fragment within the complete sequence, and the y-axis indicates the fragment length.

Each step of the pipeline is explained in detail in the following subsections.

1. Data preparation

The goal of this step is to prepare a FASTA-format file of genomic sequences that meet three criteria: they must be aligned, free of recombinant sequences, and free of duplicates. We begin with alignment. Sequence alignment is a crucial step in most evolutionary analyses. The purpose of alignment is to arrange genomic sequences by identifying regions of similarity that indicate the type of relationship, such as evolutionary history. The output of an alignment algorithm consists of sequences of equal length. For more details on sequence alignment, the interested reader is referred to [22]. It is worth noting that, although alignment remains a foundational technique in bioinformatics, there is a growing trend in the scientific community toward developing alignment-free methods for various bioinformatics tasks [23].

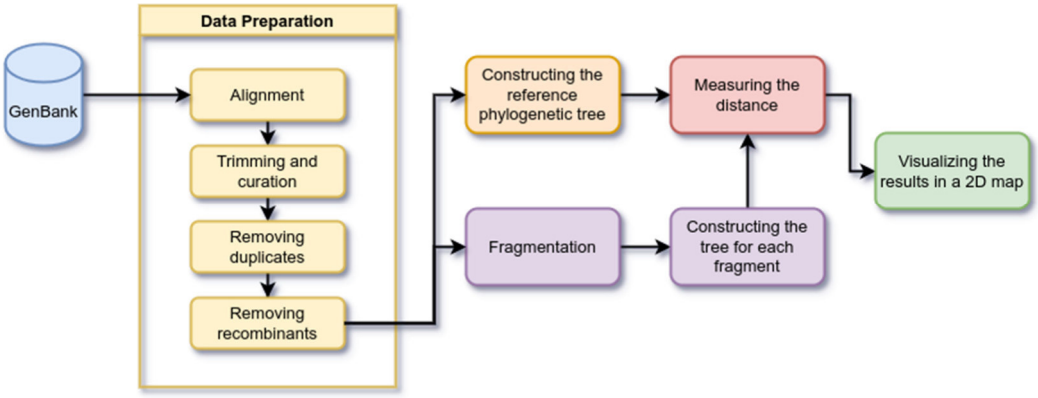

Figure 1. Schema of the proposed pipeline. The pipeline mainly consists of five steps: data preparation; construction of the reference phylogenetic tree; sequence fragmentation; construction of a phylogenetic tree for each fragment; measurement of the distance between the reference tree and each fragment tree; and visualization of the final results in a 2D map. Note that FastTree is used for phylogenetic tree construction

The output of an alignment process usually requires post-processing steps, such as trimming, to tackle gaps. Hence, alignment results often need manual correction and expert validation. Among the available alignment tools, we chose MAFFT (Multiple Alignment using Fast Fourier Transform) [24] due to its speed, accuracy, and efficiency in handling long sequences, large datasets, and gap-rich regions. Note that the quality of alignment significantly affects the accuracy and reliability of inferred phylogenetic trees [25].

After trimming and handling gaps in the aligned sequences, duplicates are identified and removed. As mentioned earlier, certain evolutionary events, such as recombination, cannot be accurately represented in a rooted acyclic graph. Therefore, it is necessary to exclude the sequences of recombinants. Recombination

events within the sequences are identified using the Recombination Detection Program (RDP) [26] along with other methods within RDP5 (GENECONV, BootScan, MaxChi, Chimaera, SiScan, and 3Seq), applying the default parameters and a significance threshold of p-value of 0.05. Sequences flagged as potential recombinants by at least two methods are excluded from further analysis. The final curated multiple sequence alignment (MSA) is then converted to *PHYLIP* format using the *Biopython* package [27].

2. Reference phylogenetic tree estimation

The second step involves estimating a reference phylogenetic tree from the aligned sequences obtained in the previous step. This inferred tree serves as a reference for evaluating the similarity between a selected fragment and the whole sequence. A fragment, in this context, refers to a continuous subsequence of the genome. Algorithms for reconstructing a phylogenetic tree from a set of aligned sequences can be grouped into three main categories: distance-based methods (e.g., neighbor-joining), character-based methods (such as maximum parsimony and maximum likelihood), and Bayesian inference methods [2]. Maximum likelihood (ML)-based methods are generally evaluated as the most accurate, but they are computationally intensive [4]. Since searching for the optimal tree under the ML framework is an NP-hard problem, many algorithms employ heuristics to manage the trade-off between speed and accuracy.

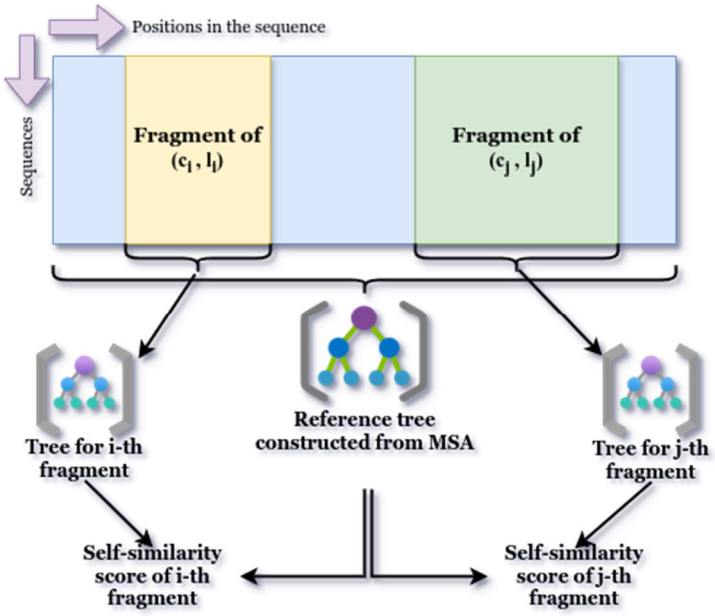

Figure 2. The process of fragmentation and measuring the phylogenetic tree. A fragment is a shifting window along the multiple sequence alignment. The fragment is varied by its center and length. The self-similarity score is obtained by measuring the distance between the fragment and reference trees

There are various widely used tools for phylogenetic tree inference based on maximum likelihood, e.g., RAxML [28], IQ-TREE2 [29], and FastTree 2 [30]. RAxML (Randomized Axelerated Maximum Likelihood) is one of the leading programs based on ML estimation and is capable of handling large-scale datasets through various strategies to accelerate computation. In terms of speed and performance, FastTree is recognized as one of the fastest tools for approximate phylogenetic tree inference using the ML approach, offering reasonable accuracy in topology estimation despite slightly lower accuracy in branch length estimation [4, 25, 3]. In our pipeline, we use FastTree 2 with its default parameters for phylogenetic tree construction due to its computational efficiency. For simplicity, we refer to FastTree 2 as FastTree throughout the remainder of this paper. We also include RAxML in our numerical experiments to demonstrate that FastTree provides a good approximation of RAxML's results. It is worth mentioning that all trees generated by these programs are subsequently rooted at the midpoint using the *ETE3 Toolkit* [31].

3. Fragmentation

A fragment is a multiple sequence alignment that represents a subset of the full-genome MSA. It consists of aligned subsequences defined by two parameters: center and length. Conceptually, a fragment can be viewed

as a sliding window along the complete sequences, where the window's position and size are determined by its center and radius (see Fig. 2). To measure self-similarity, we generate multiple fragments from the full-genome MSA by varying their center and length, construct a phylogenetic tree for each fragment, and compute the distance between the fragment tree and the reference tree. A parallel computation strategy is employed to accelerate the pipeline's speed during the distance measurement step. Notably, all settings used for phylogenetic tree construction in this step are identical to those used for estimating the reference tree.

Comparison of phylogenetic trees

Since a similarity measure can be viewed as the converse of a distance measure, we frequently switch between them in the rest of this paper. To evaluate the phylogenetic similarity between each fragment and the complete sequences, we use the topology of their corresponding phylogenetic trees. The genetic sequences of a fragment are embedded into tree space, and then the distances are calculated over this metric space. There are various metric distances defined for tree space according to tree clusters, splits, or sets of taxa (labels). The Robinson-Foulds (RF) distance [32] is one of the most common and simple metrics still used for phylogenetic tree comparison. Although it can be computed fast in linear time, it suffers from some shortcomings, e.g., imprecision compared to other methods, overestimation, and underestimation in some cases [33]. The distance is constructed based on two functions, α and α^{-1}, which are called contraction and decontraction operators, respectively. The α operator contracts an edge by merging its endpoints and creating a new vertex labeled by the union of the endpoint's labels. On the contrary, the α^{-1} performs the inverse operation by generating a new edge and converting a vertex into two vertices representing the endpoints of the new edge, while the label set of the old vertex is split in any fashion between the two new vertices. Given two trees, T_1 and T_2, defined on the same set of taxa, the α operator contracts all edges from T_1 that are not present in T_2 and results in $T_1 \wedge T_2$. The α^{-1} expands the $T_1 \wedge T_2$ structure by adding those edges that are in T_2 but are not included in T_1 to reconstruct T_2. The RF distance between two trees, T_1 and T_2, is defined as the minimum number of operations (α and α^{-1}) to convert T_1 into T_2) [32]. The distance ranges from zero to $2(n-3)$ for the case of unrooted trees with n taxa. Therefore, it can be normalized as follows:

$$d_{NRF}(T_1, T_2) = \frac{d_{RF}}{2(n-3)}.$$

There are various versions of RF distance by its generalization, normalization, or the method of calculating the similarity score between pairs of splits [34, 35, 36]. Smith [35] investigated the generalized RF metrics for comparing phylogenetic trees, in the results of which he recommends the application of clustering information (CI) distance due to intuitiveness and meaningfulness. The CI distance is a kind of entropy distance. Taking the same definition and notation provided by Smith, we explain the basics of this distance metric.

A split of a tree with taxa set X into a bipartition A and B leads to the formation of two disjoint clusters of taxa associated with A and B. The entropy of split S is calculated as follows:

$$E(S) = -P_{Cl}(A)\ log\ P_{Cl}(A) - P_{Cl}(B)\ log\ P_{Cl}(B),$$

where $P_{Cl}(A)$ is the probability that a randomly selected taxon (label) belongs to cluster A and is defined as $P_{Cl}(A) = |A| \div |X|$. The same holds for $P_{Cl}(B)$ accordingly. The CI distance corresponds to the mutual clustering information (MCI) concept. Given two splits, S_1 and S_2, the MCI measures the amount of information that is shared between splits. In other words, it reflects how knowledge about disjoint clusters A_1 and B_1 obtained from split S_1 decreases the uncertainty about disjoint clusters A_2 and B_2 in S_2. MCI is calculated as follows [34]:

$$\begin{aligned} I_{Cl}(S_1; S_2) = {} & P_{Cl}(A_1, A_2)\ log\ \frac{P_{Cl}(A_1, A_2)}{P_{Cl}(A_1)P_{Cl}(A_2)} + \\ & + P_{Cl}(A_1, B_2)\ log\ \frac{P_{Cl}(A_1, B_2)}{P_{Cl}(A_1)P_{Cl}(B_2)} + \\ & + P_{Cl}(B_1, A_2)\ log\ \frac{P_{Cl}(B_1, A_2)}{P_{Cl}(B_1)P_{Cl}(A_2)} + \\ & + P_{Cl}(B_1, B_2)\ log\ \frac{P_{Cl}(B_1, B_2)}{P_{Cl}(B_1)P_{Cl}(B_2)}, \end{aligned}$$

where $P_{Cl}(A_1, A_2) = |A_1 \cap A_2| \div |X|$ represents the probability of a point occurrence in both clusters A_1 and A_2 from S_1 and S_2, respectively. The metric reflects the degree of agreement between two trees on exhibiting the relationship of taxa by grouping them. The CI distance employed in this paper is based on MCI and normalized, so it ranges between zero and one. The interested reader is referred to Smith's paper [34] for a detailed description of the distance.

2D map visualization

The fragmentation yields a 2D map, with the X-axis and Y-axis representing the center and length of fragments, respectively. The map is constructed based on a grid, where each point corresponds to a fragment. The coordinates of each point are determined by striding over user-specified ranges for the center and length parameters, and the value at each point represents the distance between the fragment's phylogenetic tree and the reference tree. Thus, each fragment is associated with a 3D coordinate (x, y, z), where x and y denote the fragment's center and length, and z indicates the tree distance. To enhance the visualization quality, we apply interpolation and generate a contour plot, in which the color of each point reflects the similarity between the fragment and the full MSA. Colors are assigned using a blue-to-red gradient palette, where blue indicates high similarity and red indicates low similarity.

Experiments & Results

Here, we discuss the details of two experiments. In the first one, we reconstruct the 2D self-similarity map using the expensive method RAxML, the result of which is called the *reference map*. Although we believe this map has a very good approximation of self-similarity, we try to find a desirable approximation that has a better trade-off between the accuracy and cost. This new approximation is obtained by applying FastTree, and the resulting map is compared with the one obtained from RAxML. That is why we refer to the RAxML's map as the reference map.

We use the tick-borne encephalitis virus (TBEV) as a case study. This virus has an approximately 11-kilobase genome. We downloaded 493 records from the GenBank nuccore database [37] by setting the minimum length of sequence to nine kilobases. The obtained FASTA file from GenBank was fed into the MAFFT (v7.490) program to conduct alignment. The result of alignment was manually trimmed and curated in MEGA 11 (version 11.0.13) [38]. The curated FASTA file was checked for any duplicate sequence by the SeqKit toolkit [39]. We removed the recombinants using the RDP5 program [26], as stated in Subsection 2.1. The final FASTA file has 300 viruses that are further used for constructing the self-similarity map.

Before starting the experiments, we need to determine the grid, based on which the 2D map will be constructed. We set the minimum and maximum of the fragment length (L_{min} and L_{max}) to 300 and 600, respectively. The stride value for both length and center was set to three since a codon is defined by three nucleotides. The fragment center varies within the range $\left[\lfloor\frac{L_{\min}}{2}\rfloor, L - \lceil\frac{L_{\min}}{2}\rceil\right]$, where L is the length of the complete genome sequence. Considering these parameters, a total set of 334209 fragments was generated during each experiment.

To conduct the first experiment, it is required to construct the reference tree. This tree is obtained by applying RAxML to the whole genome sequences of TBEVs. We measure the similarity of a fragment tree according to the reference tree. Note that all parameters for generating the (reference and fragment) trees are fixed during each experiment. Once the reference tree was achieved, the fragments were fed into RAxML in a parallel fashion to construct their phylogenetic tree under the GTRGAMMA model. We employed the rpy2 package [40], which allows accessing R packages in Python, to use the TreeDist and ape packages for loading trees and measuring the distance between a fragment tree and the reference one. The distance was computed by the *ClusteringInfoDistance* function in a normalized fashion [34, 36]. Note that all trees were rooted at the midpoint using the *ETE3 Toolkit* [31] before distance measurement. The fragmentation forms a grid, the axes of which represent the length and center of fragments. Each row of the grid includes the fragments with the same length while the center is striding. To provide a contour visualization, we applied the unstructured triangular grid followed by interpolating to the grid using Matplotlib [41].

The only difference between the first and second experiments lies in the phylogenetic tree reconstruction program. The second experiment utilizes FastTree 2 instead of RAxML to boost the speed of the pipeline. To

evaluate the speed of FastTree in comparison to RAxML, we carried out a supplementary experiment in which 1000 out of 334209 fragments were randomly selected and given to both programs to construct the phylogenetic tree while we tracked the execution time for them. Figure 3 shows the distributions of tree generation for fragments. The result indicates that FastTree is at least 20 times faster than RAxML. Here, a question arises whether this acceleration gives a good approximation of the RAxML results. Figure 4 demonstrates the results of 2D map visualization for both programs.

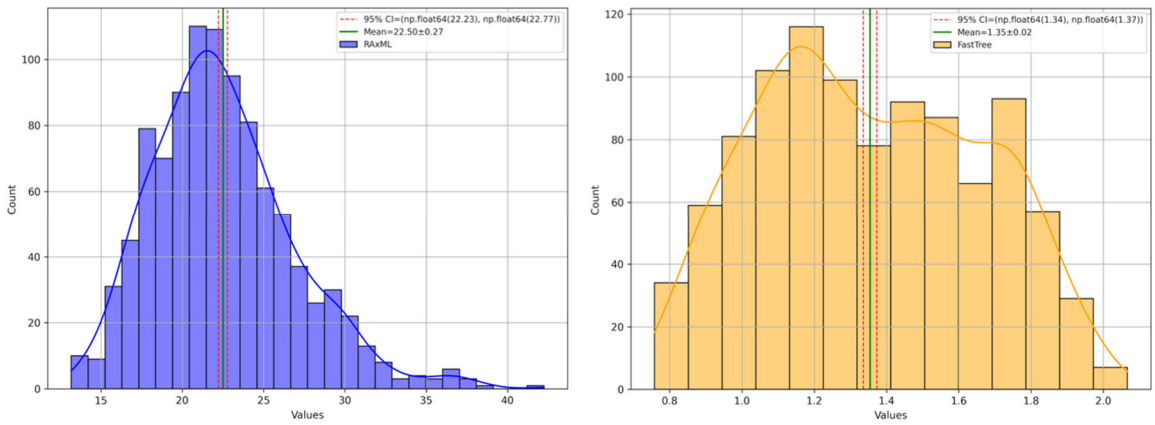

Figure 3. The execution time distribution for constructing the phylogenetic tree of randomly selected 1000 fragments. The left and right histograms are the results for RAxML and FastTree, respectively. FastTree outperforms RAxML in terms of speed by about 20 times. The X-axis represents time in seconds

In recent years, a platform for analyzing the characteristics of TBEV called TBEV Analyzer has been developed [21]. The platform utilizes a specific fragment of the coding (nucleotide) sequence of the gene E and returns the hierarchical phylogenetic characteristics of the query virus. This fragment includes 454 nucleotides (positions 309–762) encoding 151 amino acid residues (positions 104–254 aa) of the viral genome (gene E). As mentioned in [42], there are five reasons due to which this fragment is chosen to characterize TBEV:

- It includes both conservative and variable regions.
- It contains unique amino acid substitutions at positions 175, 206, and 234 that are critical for identifying the subtypes and phylogenetic lineages.
- The majority of the GenBank TBEV records cover this fragment.
- The fragment length is a compromise between sufficient information content and the possibility of its amplification, which allows studying the maximum number of virus samples obtained per season from natural foci of TBEV, almost in real-time mode.
- The results of phylogenetic analysis based on this fragment are quite comparable in informativeness with those of the complete genome sequence.

The last reason indicates that it has a role to be a representative for the TBEV genome. The green point in Fig. 4 indicates the position of this fragment along the grid. The overall distance of this fragment is about 0.3 and 0.25 for RAxML and FastTree 2, respectively. It seems that the fragment has a moderate degree of similarity. Obviously, there are several regions with dark blue indicating the high degree of similarity with the full genome. It should be better to point out that the 2D map is used to infer a set of candidates; however, choosing a representative requires additional information and verification that is beyond the pipeline's functionality.

Altogether, we recommend using our pipeline with FastTree to save time and resources. FastTree provides a good approximation of RAxML's results. Generally, the results of FastTree are smoother and more optimistic than those of RAxML. The pipeline mainly has two drawbacks. The first drawback is the resolution of the grid. Increasing the length of the fragment provides more information about the genome, leading to decreasing the fragment tree distance. We try to determine a proper resolution for the fragmentation grid. A high resolution in both center and length causes the number of fragments to increase in a quadratic manner, which adds more cost to computation. The appropriate choice is currently determined experimentally and depends on the length of the genome. Another drawback is that the choice of grid parameters requires prior knowledge about the

minimum and maximum length of the fragment. Indeed, the fragment should have enough length to reflect the phylogenetic variation.

To the best of our knowledge, this is the first time that a visualization of phylogenetic self-similarity of the genome has been proposed. The primary application of our pipeline is to identify a set of candidates of partial sequences aiming to search for a genome representative. The choice of candidates depends on the task in which the representative is employed and requires prior knowledge about, e.g., biological characteristics of candidates. Application of the genome representative may dramatically decrease the cost of computation in phylogenetic studies.

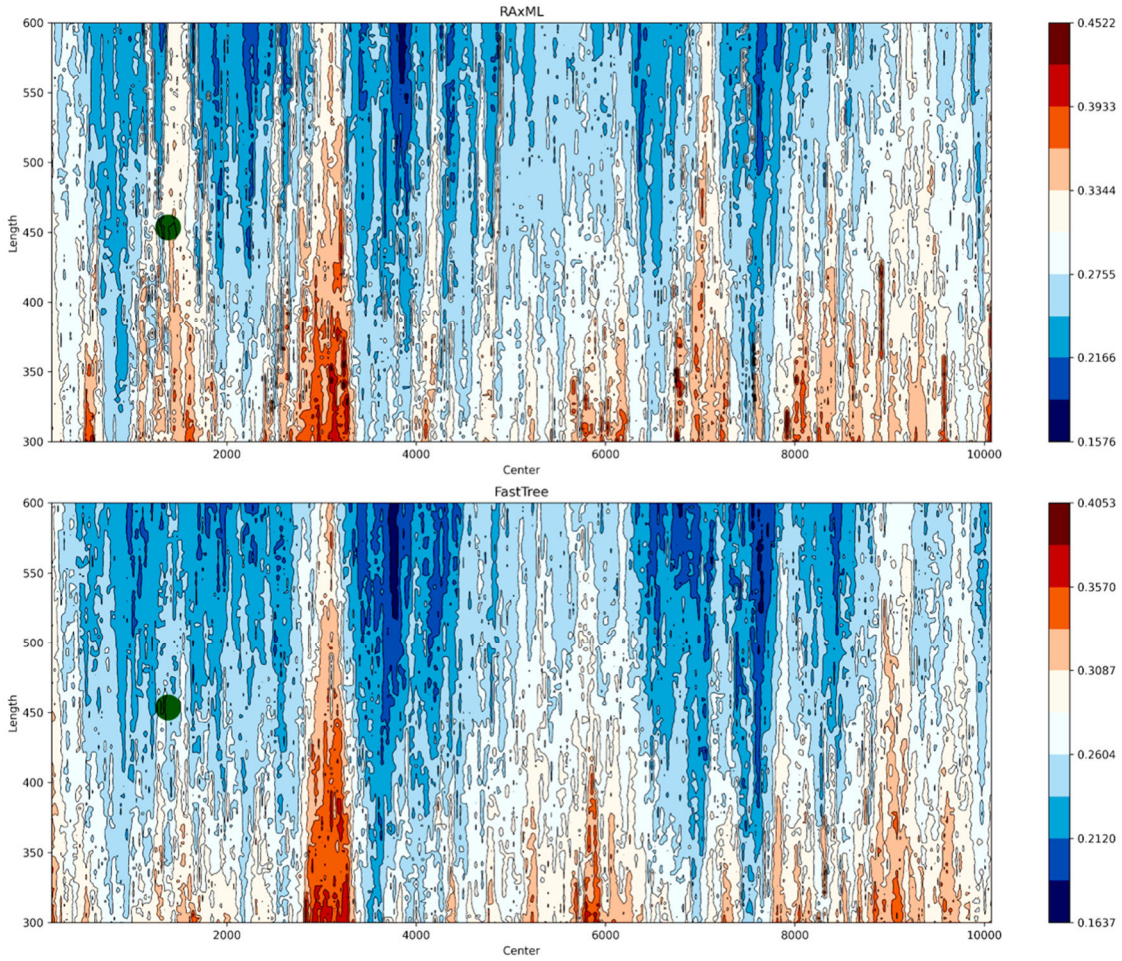

Figure 4. The 2D map visualization of phylogenetic self-similarity for the TBEV genome. The top plot is the results obtained from the RAxML program, while the bottom one is for FastTree. It is worth mentioning that FastTree gives a good approximation of the RAxML map while it is almost 22 times faster than RAxML within our experiments. The blue and red colors exhibit the low and high distances to the reference phylogenetic tree, respectively. The green point shows the result for the coding sequence of the gene E fragment used in the TBEV Analyzer platform

Conclusions

In this paper, we conducted a pilot study on computation and visualization of phylogenetic self-similarity for the genome. The main goal of the proposed pipeline is identifying the potential candidates that are more similar to the genome regarding the phylogenetic characteristics. The set of candidates can further be processed by an expert to select a representative of the genome. Such a representative may drastically decrease the cost of computation in phylogenetic studies. A desirable representative should be informative, biologically significant, and available in a public database. A good example of such a representative is the fragment of TBEV's gene E that is employed as input for the TBEV Analyzer platform. Besides its advantages, we demonstrated that this fragment has a moderate degree of similarity to the genome.

We plan to improve the pipeline speed by employing the VeryFastTree program [43] instead of FastTree. However, this requires comparing the map obtained from VeryFastTree with those of FastTree and RAxML

to determine the quality of approximation. Another improvement can be achieved by the application of heuristics for searching candidates in the grid space. This can avoid computing the similarity for the whole grid and lead to implementing an optimized algorithm. The recent advances in the field of constructing phylogenetic trees suggest the application of deep learning [44]. We expect that deep learning has the potential to facilitate the task of computing the phylogenetic self-similarity and speed up the pipeline while maintaining the quality of approximation in the visualization scene.

Acknowledgments

M. Forghani was supported by the Ministry of Science and Higher Education of the Russian Federation, project FEUZ-2023-0022.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] P. Kapli, Z. Yang, M. J. Telford, Phylogenetic tree building in the genomic age, Nature Reviews Genetics 21 (2020) 428–444. doi:10.1038/s41576-020-0233-0.

[2] R. Godini, H. Fallahi, A brief overview of the concepts, methods and computational tools used in phylogenetic tree construction and gene prediction. Meta Gene, 2019; 21: 100586, 2019. doi:10.1016/j.mgene.2019.100586.

[3] C. Young, S. Meng, N. Moshiri, An Evaluation of Phylogenetic Workflows in Viral Molecular Epidemiology, Viruses 14 (2022) 774. doi:10.3390/v14040774.

[4] P. Zaharias, T. Warnow, Recent progress on methods for estimating and updating large phylogenies, Philosophical Transactions of the Royal Society B 377 (2022) 20210244. doi:10.1098/rstb.2021.0244.

[5] D. H. Huson, D. Bryant, Application of phylogenetic networks in evolutionary studies, Molecular biology and evolution 23 (2006) 254–267. doi:10.1093/molbev/msj030.

[6] K. T. Huber, V. Moulton, T. Wu, Transforming phylogenetic networks: Moving beyond tree space, Journal of theoretical biology 404 (2016) 30–39. doi:10.48550/arXiv.1601.01788.

[7] M. Hellmuth, D. Schaller, P. F. Stadler, Clustering systems of phylogenetic networks, Theory in Biosciences 142 (2023) 301–358. doi:10.1007/s12064-023-00398-w.

[8] S. Kovalev, T. Mukhacheva, Reconsidering the classification of tick-borne encephalitis virus within the Siberian subtype gives new insights into its evolutionary history, Infection, Genetics and Evolution 55 (2017) 159–165. doi:10.1016/j.meegid.2017.09.014.

[9] M. Forghani, S. Kovalev, M. Bolkov, M. Khachay, P. Vasev, TBEV analyzer platform for evolutionary analysis and monitoring tick-borne encephalitis virus: 2020 update, Biostatistics & Epidemiology 6 (2022) 57–73. doi:10.1080/24709360.2021.1985392.

[10] M. S. Rosenberg, Sequence alignment: methods, models, concepts, and strategies, Univ of California Press, 2009.

[11] K. Yan, J. Wen, J.-X. Liu, Y. Xu, B. Liu, Protein fold recognition by combining support vector machines and pairwise sequence similarity scores, IEEE/ACM Transactions on Computational Biology and Bioinformatics 18 (2020) 2008–2016. doi:10.1109/TCBB.2020.2966450.

[12] H. J. Jeffrey, Chaos game visualization of sequences, Computers & Graphics 16 (1992) 25–33. doi:10.1016/0097-8493(92)90067-6.

[13] G. Durán-Meza, J. López-García, J. L. del Río-Correa, The self-similarity properties and multifractal analysis of DNA sequences., Applied Mathematics & Nonlinear Sciences 4 (2019). doi:10.2478/AMNS.2019.1.00023.

[14] B.-l. Hao, H.-C. Lee, S.-y. Zhang, Fractals related to long DNA sequences and complete genomes, Chaos, Solitons & Fractals 11 (2000) 825–836. doi:10.1016/S0960-0779(98)00182-9.

[15] Y. Li, B. Jiang, H. Chen, X. Yao, Symbolic sequence classification in the fractal space, IEEE Transactions on emerging topics in computational intelligence 5 (2018) 168–177. doi:10.1109/TETCI.2018.2876528.

[16] T. Mikolov, W.-t. Yih, G. Zweig, Linguistic regularities in continuous space word representations, in: Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies, 2013, pp. 746–751.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[18] E. Asgari, M. R. Mofrad, Continuous distributed representation of biological sequences for deep proteomics and genomics, PloS one 10 (2015) e0141287. doi:10.1371/journal.pone.0141287.

[19] J. Du, P. Jia, Y. Dai, C. Tao, Z. Zhao, D. Zhi, Gene2vec: distributed representation of genes based on co-expression, BMC genomics 20 (2019) 82. doi:10.1186/s12864-018-5370-x.

[20] S. Y. Kovalev, T. A. Mukhacheva, Clusteron structure of tick-borne encephalitis virus populations, Infection, Genetics and Evolution 14 (2013) 22–28. doi:10.1016/j.meegid.2012.10.011.

[21] M. Forghani, S. Kovalev, M. Khachay, E. Ramsay, M. Bolkov, P. Vasev, Identifying new clusterons: application of TBEV analyzer 3.0, Microorganisms 11 (2023) 324. doi:10.3390/microorganisms11020324.

[22] Y. Zhang, Q. Zhang, J. Zhou, Q. Zou, A survey on the algorithm and development of multiple sequence alignment, Briefings in bioinformatics 23 (2022) bbac069. doi:10.1093/bib/bbac069.

[23] A. Zielezinski, H. Z. Girgis, G. Bernard, C.-A. Leimeister, K. Tang, T. Dencker, A. K. Lau, S. Röhling, J. J. Choi, M. S. Waterman, et al., Benchmarking of alignment-free sequence comparison methods, Genome biology 20 (2019) 144. doi:10.1186/s13059-019-1755-7.

[24] K. Katoh, D. M. Standley, MAFFT multiple sequence alignment software version 7: improvements in performance and usability, Molecular biology and evolution 30 (2013) 772–780. doi:10.1093/molbev/mst010.

[25] J. A. Lees, M. Kendall, J. Parkhill, C. Colijn, S. D. Bentley, S. R. Harris, Evaluation of phylogenetic reconstruction methods using bacterial whole genomes: a simulation based study, Wellcome open research 3 (2018). doi:10.12688/wellcomeopenres.14265.2.

[26] D. P. Martin, A. Varsani, P. Roumagnac, G. Botha, S. Maslamoney, T. Schwab, Z. Kelz, V. Kumar, B. Murrell, RDP5: a computer program for analyzing recombination in, and removing signals of recombination from, nucleotide sequence datasets, Virus evolution 7 (2021) veaa087. doi:10.1093/ve/veaa087.

[27] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kau, B. Wilczynski, et al., Biopython: freely available python tools for computational molecular biology and bioinformatics, Bioinformatics 25 (2009) 1422. doi:10.1093/bioinformatics/btp163.

[28] A. Stamatakis, RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies, Bioinformatics 30 (2014) 1312–1313. doi:10.1093/bioinformatics/btu033.

[29] B. Q. Minh, H. A. Schmidt, O. Chernomor, D. Schrempf, M. D. Woodhams, A. Von Haeseler, R. Lanfear, IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era, Molecular biology and evolution 37 (2020) 1530–1534. doi:10.1093/molbev/msaa015.

[30] M. N. Price, P. S. Dehal, A. P. Arkin, FastTree 2–approximately maximum-likelihood trees for large alignments, PloS one 5 (2010) e9490. doi:10.1371/journal.pone.0009490.

[31] J. Huerta-Cepas, F. Serra, P. Bork, ETE 3: reconstruction, analysis, and visualization of phylogenomic data, Molecular biology and evolution 33 (2016) 1635–1638. doi:10.1093/molbev/msw046.

[32] D. F. Robinson, L. R. Foulds, Comparison of phylogenetic trees, Mathematical biosciences 53 (1981) 131–147. doi:10.1016/0025-5564(81)90043-2.

[33] H. Folkertsma, A. Mittal, Comparing phylogenetic trees: an overview of state-of-the-art methods, 16th SC@ RUG 2018-2019 (2019) 14.

[34] M. R. Smith, Information theoretic Generalized Robinson-Foulds metrics for comparing phylogenetic trees, Bioinformatics 36 (2020) 5007–5013. doi:10.1093/bioinformatics/btaa614.

[35] M. R. Smith, Robust analysis of phylogenetic tree space, Systematic Biology 71 (2022) 1255–1270. doi:10.1093/sysbio/syab100.

[36] M. R. Smith, TreeDist: Distances between Phylogenetic Trees. R package version 2.9.2, 2020. doi:10.5281/zenodo.3528124.

[37] D. A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, J. Ostell, K. D. Pruitt, E. W. Sayers, Genbank, Nucleic acids research 46 (2018) D41–D47. doi:10.1093/nar/gkx1094.

[38] K. Tamura, G. Stecher, S. Kumar, MEGA 11: molecular evolutionary genetics analysis version 11, Molecular biology and evolution 38 (2021) 3022–3027. doi:10.1093/molbev/msab120.

[39] W. Shen, B. Sipos, L. Zhao, SeqKit2: A swiss army knife for sequence and alignment processing, Imeta 3 (2024) e191. doi:10.1002/imt2.191.

[40] L. Gautier, rpy2, 2023. URL: https://github.com/rpy2/rpy2/releases/tag/RELEASE_3_5_11.

[41] J. D. Hunter, Matplotlib: A 2D graphics environment, Computing in Science & Engineering 9 (2007) 90–95. doi:10.1109/MCSE.2007.55.

[42] M. Forghani, S. Kovalev, P. Vasev, M. Bolkov, TBEV analyzer: A platform for evolutionary analysis of tick-borne encephalitis virus, in: 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), IEEE, 2019, pp. 0397–0402. doi:10.1109/SIBIRCON48586.2019.8958021.

[43] C. Piñeiro, J. M. Abuín, J. C. Pichel, Very Fast Tree: speeding up the estimation of phylogenies for large alignments through parallelization and vectorization strategies, Bioinformatics 36 (2020) 4658–4659. doi:10.1093/bioinformatics/btaa582.

[44] A. Suvorov, J. Hochuli, D. R. Schrider, Accurate inference of tree topologies from multiple sequence alignments using deep learning, Systematic biology 69 (2020) 221–233. doi:10.1093/sysbio/syz060.