Visualization of the Evolutionary Trajectory: Application of Reduced Amino Acid Alphabets and Word2Vec Embedding

Majid Forghani[1,2], Artyom Firstkov[1], Pavel Vasev[1] and Edward Ramsay[3]

[1]N.N. Krasovskii Institute of Mathematics and Mechanics of the Ural Branch of the Russian Academy of Sciences (IMM UB RAS), 16 S.Kovalevskaya St., Yekaterinburg, 620108, Russia

[2]Ural Federal University, 51 Lenina Ave., Yekaterinburg, 620075, Russia

[3]Saint Petersburg Pasteur Institute, 14 Mira St., Saint Petersburg, 197101, Russia

Abstract

Analysis of viral evolution is a key element of epidemiological surveillance and control. One of the fundamental tools which is widely used to illustrate evolutionary history is the phylogenetic tree. Recently, we have proposed an alternative visualization for the phylogenetic tree using the evolutionary trajectory of its taxa. An evolutionary trajectory is a path starting from a taxon and ending at the root of the tree. In this paper, we propose an embedding of tree nodes by encoding their genetic sequence using a reduced amino acid alphabet and employing the Word2Vec framework. The suggested visualization maintains the phylogenetic relationship between nodes, while their proximity in 3D space depends on three factors: the type of reduced amino acid alphabet; fixed-length genetic patterns used in Word2Vec; and the neighbor effect of adjacent signatures. The results of our experiments showed that the majority of evolutionary history can be described in the embedded space. Moreover, they suggest potential application of our approach as an explanatory tool in studying various aspects: evolutionary dynamics; evolutionary deviation of viral variants; and phylogenetic characteristics, such as formation of new clades. Besides the usual local analysis of point mutations, the developed framework enables studying these aspects based on a more comprehensive global context, including neighboring effects, genetic signatures.

Keywords

Visualization, Evolution, Word2Vec, Simplified Amino Acid Alphabet, Phylogenetic Tree.

1. Introduction

Study of the evolution of a species is a key tool for understanding their behavior, especially regarding determination of their evolutionary direction. Such knowledge plays a critical role in the surveillance and control of pathogens. Specialists investigate the evolutionary history of species through representation of their similarities and differences using various bioinformatic tools, such as phylogenetic trees and phylogenetic networks. The phylogenetic tree is a classical

GraphiCon 2022: 32nd International Conference on Computer Graphics and Vision, September 19–22, 2022, Ryazan State Radio Engineering University named after V.F. Utkin, Ryazan, Russia

forghani@imm.uran.ru (M. Forghani); firstk121@gmail.com (A. Firstkov); vasev@imm.uran.ru (P. Vasev); WarmSunnyDay@mail.ru (E. Ramsay)

0000-0002-9443-3610 (M. Forghani); 0000-0002-9636-9800 (A. Firstkov); 0000-0003-3854-0670 (P. Vasev); 0000-0001-7086-5825 (E. Ramsay)

and fundamental method aiming to embed the species from a complex genetic space into a tree space to represent their evolutionary proximity in a graphical, human-readable form [1].

Constructing a distance-based phylogenetic tree consists of performing two sequential procedures: computing a distance (or proximity) matrix; and inferring the tree topology from the matrix. The distance matrix is often obtained by employing a model of evolution, which estimates the genetic divergence between objects. Such models have been earlier developed based on the concept of the Markov model for sequence evolution and vary according to the type of genetic information (nucleotide, amino acid, codon, etc.) as well as substitution rate parameters [2, 3]. The obtained distance matrix is further passed into a hierarchical clustering algorithm to express the similarity or dissimilarity of objects. Traditionally, this can be carried out by applying one of the following well-known and widespread algorithms: neighbor-joining [4]; and unweighted pair group method with arithmetic mean (also called UPGMA) [5].

Although in most cases, the representation of evolutionary history in terms of point mutations is quite informative, sometimes it is required to consider a more complex genetic signature (or motif) to well describe a phenotype. As an example, most of the models for predicting antigenic evolution rely on complex patterns at antigenic sites [6]. Studies have shown that non-antigenic sites located in the vicinity of antigenic sites also impact antigenicity. This impact is known as the neighbor effect [7]. Thus, a comprehensive model should consider both genetic patterns and neighbor effects.

Recently, application of the simplified amino acid alphabets has become more popular, specifically in searching for a suitable space that can better describe a phenotype. A simplified amino acid alphabet (SAAA), also called reduced amino acid alphabet, is an alphabet in which the 20 standard amino acids are clustered and divided into groups. In this way, a mutation is redefined as a change between two amino acid groups of the simplified alphabets. For example, the following alphabet is achieved by grouping the standard amino acids based on their van der Waals volume: G1={G, A, S, C, T, P, D}, G2={N, V, E,Q, I, L}, and G3={M, H, K, F, R, Y, W}. In this content, any transition between the two groups (G1-G2, G2-G3, G1-G3) is a substitution. Since the volume and hydrophobicity of amino acids play a major role in substitution of amino acids [8], a representation using SAAA provides better insight into evolution and highlights significant features associated with a target phenotype from various perspectives including structural, biological, and physicochemical similarities.

As shown [9, 10], visualization of similarity/dissimilarity between viral strains is a key factor for predicting the current and future characteristics of the virus. When aiming to improve description and prediction quality for a phenotype, it is beneficial to take into account three factors for visualizing strain proximity: genetic pattern (instead of point mutation); neighbor amino acid effects; and application of simplified amino acid alphabets for redefining the substitution. We provide this kind of analysis and visualization by introducing a computational pipeline which employs simplified amino acid alphabets, and the Word2Vec framework.

As an extension of our previous work on phylogenetic tree visualization [1, 11], we propose a novel visualization in this paper. In it, the phylogenetic relationships between tree nodes are maintained, whereas the distance between two nodes is determined using Word2Vec encoding and SAAA. To our knowledge, this is the first time that the proximity of strains, in terms of higher-order genetic signatures, has been visualized along with their phylogenetic relationships.

Our contribution to this field is developing an approach for visualization of strain evolution

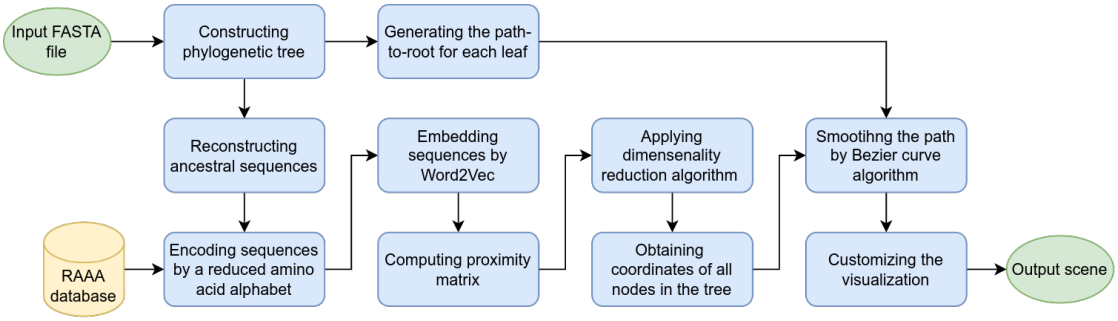

Figure 1: The overall schema of the proposed pipeline. The pipeline extracts information about the evolutionary path from the phylogenetic tree. The coordinate of each vertex in the path is generated by embedding genetic sequences using the simplified amino acid alphabet and the Word2Vec framework [12].

wherein their coordinates depend on the type of amino acid alphabet, on fixed-length genetic patterns, and on neighbor effects. We believe such a visualization can serve as an investigative tool capable of revealing more information than other approaches about hidden mechanisms of the evolutionary process at micro- and macro-scales. The rest of this paper is organized as follows. Section 2 explains the methodology in more detail. Section 3 describes the computational experiments and discusses their results. Finally, the conclusion is presented in Section 4.

2. Methodology

Our approach consists of four basic steps: reconstructing the phylogenetic relationships between strains; computing the sequence of ancestral (or inner) nodes located in the phylogenetic tree; embedding genetic sequences for all nodes and computing the distance matrix of strains; and finally visualizing strain relationships by utilizing their connections from the phylogenetic tree and their coordinates from the new embedding space. Figure 1 illustrates the overall schema of the proposed pipeline. We briefly describe each step in more detail in the following section.

As mentioned, the pipeline is an extended version of our recent work on visualization of the phylogenetic tree called PhyloTraVis [11]. PhyloTraVis focuses on visualization of evolutionary trajectories defined within a phylogenetic tree. An evolutionary trajectory is an individual unique path that connects a taxon (located in the tree leaf) into the root (the most common ancestor for all tree leaves). In PhyloTraVis, the distances between trajectory nodes are determined by embedding their genetic sequence into a 3D space using one-hot-encoding and the t-distributed Stochastic Neighbor Embedding (t-SNE) method [13].

The first and second steps of the pipeline are related to phylogenetic analysis and are carried out using Randomized Axelerated Maximum Likelihood (RAxML) [14], as described earlier [11]. Here, we focus on further steps related to the embedding and visualization.

The pipeline input must be an aligned FASTA file, including n amino acid sequences denoted by $\{l_1, l_2, ..., l_n\}$. By constructing their binary phylogenetic tree, $n - 1$ internal nodes connect n leaves, whereas each leaf represents a strain/sequence from the FASTA file. We denote an

internal node (ancestor) by a_i, where $i \in \{1, 2, ..., n-1\}$. Therefore, the phylogenetic tree T can be represented by graph notation as $T = (V, E)$, where V and E are set of vertices and edges, respectively.

$$V = \{l_i \mid i \in \{1, 2, ..., n\}\} \cup \{a_i \mid i \in \{1, 2, ..., n-1\}\} \tag{1}$$

The FASTA file includes amino acid sequences of all leaves, whereas the sequences of internal nodes need to be computed using the ancestral sequence reconstruction algorithm. Note that to reconstruct the ancestral sequences, the tree must be rooted. We perform this reconstruction by applying the flag '-f A' in the RAxML package. Here, we append the graphical representation by adding the genetic information of amino acid sequences. So the tree is expressed by triple (V, E, S), where S is defined as follow:

$$S = \{S_i \mid i \in \{1, 2, ..., 2n-1\}\} \tag{2}$$

where S_i is the amino acid sequence of length m for node $v_i \in V$:

$$S_i = \{s_{i,j} \mid j \in \{1, 2, ..., m\}\}, \quad i \in \{1, 2, ..., 2n-1\} \tag{3}$$

where $s_{i,j}$ is a standard amino acid or gap:

$$s_{i,j} \in \{gap(-), A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$$

PhyloTraVis shows the leaves' proximity through their path-to-root. Although it treats each path as an independent object for visualization, the path strongly depends on the tree topology and the sequences of tree nodes. Extracting the path-to-root of a leaf is carried out by parsing the tree using the module 'Phylo' from the Biopython package [15]. Since our visualization aims at representing such paths, we replace the edges set E in (V, E, S) with the set of paths P.

$$P = \{p_i \mid p_i \text{ is path-to-root for leaf } l_i,\ i \in \{1, 2, ..., n\}\} \tag{4}$$

Here, our graph definition changed to (V, P, S). The next step is embedding the genetic information located in S to generate the 2D or 3D coordinates of each node in the visualization space. In this step, we assume that there is no information about the connection of vertices (i.e., path-to-root). The proposed embedding approach consists of four sequential sub-steps, including encoding the amino acid sequences by a reduced amino acid alphabet; embedding the obtained sequences from the previous sub-step using the Word2Vec framework [12]; computing the distance matrix between all nodes by a metric; and finally applying the dimensionality reduction algorithm to obtain the 3D/2D coordinates from the distance matrix.

A reduced amino acid alphabet (RAAA) is an alphabet, in which the 20 standard amino acids are clustered into groups. In addition to earlier RAAA applications [16, 17], we have recently shown that RAAA is beneficial for increasing modeling accuracy regarding influenza virus antigenic evolution [18]. RAAA provides a point of view through which we can investigate a phynotype. Indeed, a substitution is redefined in the context of amino acid proximity. Currently, there are 41 amino acid alphabets, including the standard amino acid alphabet and 40 published RAAAs [19, 20, 21], and denoted by R_k, where $k \in \{1, 2, ..., 41\}$. In a RAAA, the first amino acid

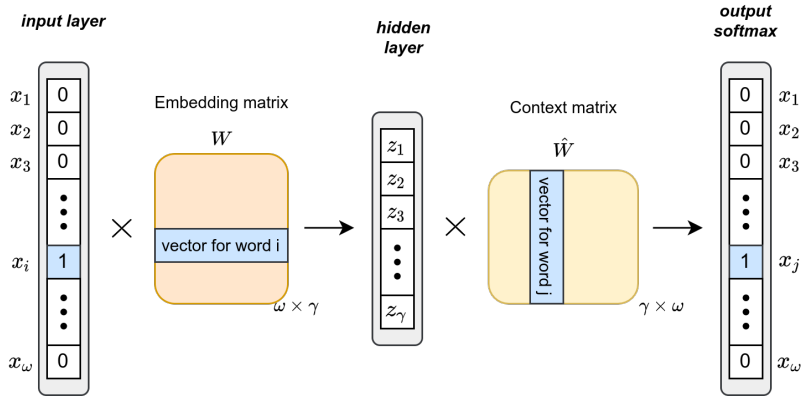

Figure 2: Skip-gram architecture. Once the network is trained, we employ matrix W for word embedding. Note that the vocabulary size is ω, while the hidden layer size or word embedding dimension size is γ.

of a group is considered its representative, e.g., the representative for group M, H, K, F, R, Y, W is M (Methionine). Therefore, we replace each standard amino acid in a protein sequence from S with its group representative.

If the amino acid sequence S_i (the sequence for vertex v_i) is encoded by RAAA R_k, the encoded sequence is denoted by S_{i,R_k}. Here, we replace the set of genetic sequences S by S_{R_k} (set of the encoded version of all sequences from S) in our graph and express it by (V, P, S_{R_k}). Note that S_{R_k} can have the repeated sequences.

The next procedure is representing sequences from S_{R_k} in a numerical space. Although the representation can simply be carried out by assigning a binary code to each group in RAAA R_k, numerical representation should reflect some aspects of the biological relationship. To do this, we employ the well-known and widespread framework of embedding for natural language processing, Word2Vec, proposed by Mikolov et al. [12, 22]. Word2Vec employs a neural network to learn word relationships in a defined neighborhood in a corpus of text. Once the network training process is accomplished, it can be used to obtain the numerical/vector representation of words. In this manner, semantically similar words have the high degree of cosine similarity. Word2Vec has shown promising applications in the encoding genetic sequences. Recently, several embeddings have been developed based on Word2Vec, including DNA2Vec [23], and Imuune2Vec [24]. Word2Vec is constructed based on one of the two architectures: continuous bag-of-words (also called CBOW); and skip-grams. We use skip-gram since it is more suitable for infrequent words (rare mutations in our case). The architecture of this network is illustrated in Figure 2. Suppose we use k-gram to split the sequence S_{i,R_k} in S_{R_k} into its words $W = (w_1, w_2, ..., w_h)$, while each k-gram has a vector representation of length γ. By concatenating the vector of words represented in W, we obtain a vector of length $h \times \gamma$ that represents S_{i,R_k} (Figure 3). We replace S_{R_k} in the graph representation (V, P, S_{R_k}) by $\Phi = \{\phi_i \mid i \in \{1, ..., 2n-1\}\}$, where ϕ_i is the vector representation for vertex $v_i \in V$.

In the previous work, we apply the method of dimensionality reduction to encoded sequences, which were obtained by one-hot-encoding. The one-hot-encoding generates a binary vector of $m \times 4$ (or $m \times 20$) for a DNA (or protein) sequence, where m is the length of the sequence. In the case of a Word2Vec embedding, the size of the vector that represents the genetic sequence

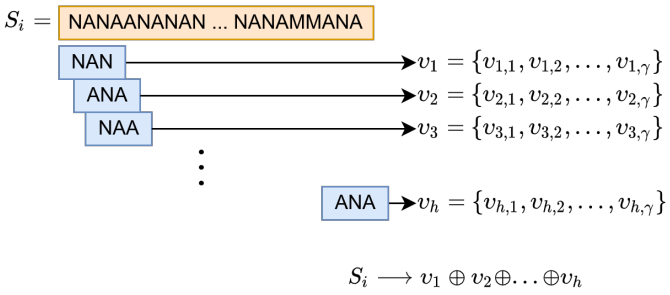

Figure 3: An example of generating the numerical representation for an encoded amino acid sequence. The blue rectangle includes the trigram, while its vector representation is given on the right side. The amino acid sequence S_i is represented by concatenating the vectors of its trigrams. Note that $\oplus$ indicates the concatenation operation on vectors.

is $\gamma \times (m - k + 1)$, where γ is the size of the word embedding vector, m is the length of the sequence, and k is the length of the word (k-gram). This may increase the computational complexity of the dimensionality reduction process if the length of sequence or the embedding size is relatively large.

In order to speed up the computation, we suggest computation of the distance matrix for embedded sequences and applying the dimensionality reduction method to the matrix. To do this, we apply a similarity measure or distance to compute the proximity between vertices using their vector representation in Φ. This generates a proximity/distance matrix, which is fed into a dimensionality reduction algorithm, e.g., t-SNE or multidimensional scaling (MDS) [25], to obtain the vertex coordinate. The choice of similarity measure can be varied since it depends on the target phenotype of interest to the specialist. One can even perform a Softmax operation for all vectors located in Φ to represent each of them as a probability distribution. Therefore, in addition to traditional measures of similarity such as Euclidean and Cosine distance, the final visualization can be customized using similarity measures between probability distribution, e.g., Jensen-Shannon distance.

We denote the coordinate of vertex $v_i \in V$ by c_i, where c_i can be a tuple or triple. Our final graph for visualization is defined by (V, P, C), where: V is the set of vertices or all nodes in the phylogenetic tree; P is the set of path-to-root for all leaves in the tree; and C is the set of coordinates for vertices. Indeed, C gives the coordinates of vertices of V, while P determines how vertices are connected in the space. Before visualization, all paths are smoothed by the Bezier curve algorithm, which has been described in more detail elsewhere [11]. The smoothed path information is passed into Vrungel to visualize the graph.

Vrungel is a visualization technology, consisting of a programming language and its interpreter, developed by one the authors of this paper. It facilitates visualization through a relatively brief description of tree objects. Visualizations are displayed in a web browser, and the representation can be customized by setting parameters in a two-dimensional interface. In addition, Vrungel is capable to enter to virtual reality (VR) mode thanks to WebVR technology. We consider VR as a usable option for data investigation both for detailed and general view. VR specifically becomes handy in the case of working with a large data set. However additional

user interaction techniques should be developed to more efficiently use VR mode in Vrungel. Vrungle is publicly available at https://github.com/viewzavr/vrungel.

In the next section, we present a visualization example using a sample set of influenza virus strains.

3. Experiments & Results

To demonstrate the application of our approach, we visualized evolutionary trajectories for a data set of influenza viral strains. Currently, there are three subtypes of influenza virus which pose a significant threat to the public health: H1N1, H3N2, and H5N1. It is known that the H3N2 subtype is more genetically variable than other influenza virus subtypes. That is the reason why we choose this subtype for our experiment. The experimental data set includes HA1 sequences (hemagglutinin protein) for 335 strains of H3N2 subtype, collected during 1968-2007, as described [26]. Since the input file needs to be aligned, an alignment process may be required before submitting the data set into the pipeline for performing phylogenetic analysis. The prepared input file included sequences with a length of 329 amino acids.

We constructed the phylogenetic tree through four steps: generating an initial tree by FastTree software [27]; constructing the middle tree using the PROTCATFLU model by RAxML from the initial tree; constructing the final tree using the PROTGAMMAFLU model by RAxML from the middle tree; and finally rooting the tree by RAxML by setting '-f I' and the PROTGAMMAFLU model. Our approach relies on knowledge about the genetic sequence of each node in the tree. Therefore, we conduct ancestral sequence reconstruction to compute marginal ancestral states of the internal nodes (i.e., ancestors). We perform it by RAxML using the '-f A' flag under the PROTGAMMAFLU model.

As mentioned, the connection between tree nodes in the final visualization is based on evolutionary trajectory. We extract the information of all trajectories by applying the 'Phylo' module from the Biopython package [15]. The proposed approach treats each trajectory as an individual object for visualization, while the coordinates of its nodes are determined based on the genetic sequence embedding. The definition of amino acid substitution can be customized by applying a reduced amino acid alphabet. The choice of alphabet depends on the subject of the analysis. For example, our previous studies [18] have shown that 3 alphabets are advantageous in modeling the antigenic evolution of the H3N2 subtype. They are Hyrphobicity alphabets ({RKEDQN,GASTPHY,CVLIMFW}); Polarity alphabet ({LIFWCMVY,PATGS,HQRKNED}); and Cannata 2002 alphabet ({D,E,N,KR,Q,ST,G,P,H,A,C,W,Y,F,ML,IV}). In this experiment, we use these alphabets to encode the genetic information of all tree nodes before embedding them by the skip-gram model.

To train the skip-gram network, there are two options for choosing the training data set. The training process can be carried out on the experimental data set, or we can use a larger data set which reflects a more comprehensive history of subtype evolution. To enhance embedding quality, we trained the network on a large data set (H3N2 subtype) featuring 36,434 unique hemagglutinin protein sequences. Note that the training data set must also be encoded by the RAAA. Since applying a RAAA leads to a decrease in the variation of genetic sequence mutations, some sequences may be repeated in the encoded training data set. Thus, we created

a training data set with unique sequences by removing redundant sequences using the SeqKit toolkit [28]. The training was carried out in the PyTorch framework [29] by setting the following hyperparameters: k-gram size to three; context window size to 20 words; hidden size to 100; learning rate to 0.001; epoch number to 100, k-negative sampling size to 5 words; and finally optimizer to Adam. This setup was obtained by performing several computational experiments and assessing the quality of embedding using the cosine similarity between adjacent and distant words. Note that the word size of embedding includes three amino acids, i.e., each word is a trigram. The process of generating words from a sequence is shown in Figure 3. The stride for word generation is set to one amino acid.

Our approach can serve as an exploratory tool to investigate the connection between a phenotype (represented in the embedding space) and evolution. A key factor to better visualize this connection is customizing the concept of similarity between strains. Hence, we offer a modified version of the skip-gram architecture by adding a softmax function to the output of the hidden layer. This allows us to consider each embedding vector in the word embedding matrix as a probability distribution (see matrix W in Figure 2). As mentioned in Section 2, vector representation of a sequence is obtained through concatenation of the vector representation of its words. In the case of the modified skip-gram, the sum of sequence vector elements is equal to the number of its words (since the sum vector elements of each word is equal to one). To treat the sequence vector representation as a probability distribution, we scale each element by the total number of words in the sequence. This provides us with a wide choice of distance, including those are used for measuring the similarity between two probability distributions, to score the similarity between sequences in the embedding space.

In this experiment, we considered 3 distances (Euclidean, cosine, Jensen-Shannon). In order to estimate semantic similarity between words in Word2Vec, researchers compute the cosine similarity between the vector representation of words. To extend the definition of similarity between sequences, we used the Jensen-Shannon distance, which is the square root of the Jensen-Shannon divergence. We generated a distance matrix for each metric using the 'distance' module in the Scipy library [30]. The matrix was further utilized to extract the coordinates of tree nodes. This was conducted by applying a dimensionality reduction algorithm, t-SNE, to the distance matrix. In result, we have the coordinate of every node in each evolutionary trajectory. We smoothed every trajectory by applying the Bezier curve algorithm to increase the visualization quality. A sample visualization of evolutionary trajectories for the experiment data set is illustrated in Figure 4.

Considering the mentioned three alphabets (Polarity, Hydrophobicity, Cannata 2002), we applied three distances for each alphabet and obtained nine distance matrices. To assess how well the derived embeddings can preserve the evolutionary history of strains, we calculated the pairwise evolutionary distances between original genetic sequences of strains and compared them with the distance matrices from the embeddings. To do this, we calculated distance matrices for three evolutionary distances: uncorrected distance; Jukes-Cantor distance [3]; and Kimura distance [2]. To measure the correlation between matrices, we conducted the Mantel test [31]. Table 1 shows the results of the test.

Among the selected RAAAs, the Cannata 2002 alphabet has 16 groups of amino acids [32], while the other two alphabets include 3 groups. The Polarity and Hydrophobicity alphabets provide more compact representations of the standard amino alphabet than the Cannata 2002

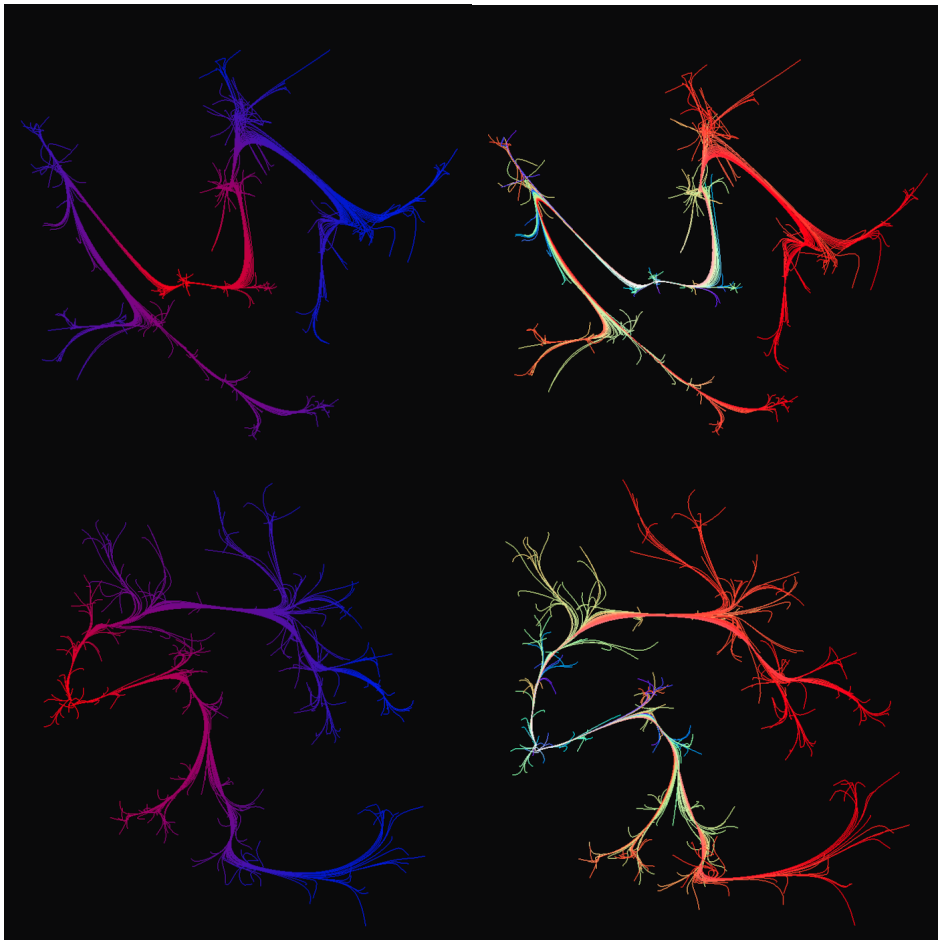

Figure 4: A sample visualization of our experiment data set. The data set included 335 strains collected during 1968-2007. Top images obtained by encoding protein sequences using Hydrophobicity alphabet and applying cosine distance to measure the proximity of sequences in the embedding space. The below images are the visualization of same phylogenetic relationship as top images by encoding using Cannata 2002 alphabet and employing the Jensen-Shannon distance. Left side images are colored based on the node distance to the root, while the color in right side images expresses the isolation year of strain.

alphabet. This is why the Cannata 2002 alphabet has the highest correlation for each distance in Table 1. The Hydrophobicity alphabet shows slightly more correlation than Polarity does with evolutionary distance matrices. Moreover, the distance matrix obtained based on Hydrophobicity alphabet encoding with cosine distance, surprisingly, achieved the highest correlation with Jucke-cantor and Kimura distance matrices, yet it describes only 3 possible transitions in the encoded genetic sequences.

In summary, Table 1 shows that the majority of evolutionary history is preserved in the proposed embedding space. Moreover, the cosine distance outperforms the other two distances in terms of correlation coefficients, while the Euclidean distance achieves slightly less correlation than the Jensen-Shannon distance. Taken together, the results indicate that the proposed

Table 1
Mantel test results for assessing the correlation between distance matrices. The Mantel test was calculated using the Pearson method, 1000 permutations, and a two-sided test. All reported correlation coefficients have been achieved with p_{value} of 1e-03.

Metric	RAAA	Uncorrected	Jukes-Cantor	Kimura
Jensen–Shannon	Hydrophobic	0.940	0.935	0.933
	Polarity	0.933	0.926	0.924
	Cannata 2002	0.955	0.948	0.946
Euclidean	Hydrophobic	0.935	0.930	0.928
	Polarity	0.933	0.927	0.925
	Cannata 2002	0.938	0.931	0.928
Cosine	Hydrophobic	0.962	0.960	0.959
	Polarity	0.958	0.956	0.955
	Cannata 2002	0.963	0.959	0.958

approach can serve as an auxiliary tool for exploring and studying the evolutionary process and phylogenetic characteristics which can be described in terms of amino acid properties and higher order genetic signatures.

4. Conclusions

In this paper, we propose an alternative visualization of the phylogenetic tree by maintaining the phylogenetic relationship between strains and modifying their coordinates. The coordinate of each node is computed by applying the Word2Vec framework to encoded genetic sequences. The encoding is carried out using a reduced amino acid alphabet (RAAA), with redefinition of the mutation as a change between its groups.

The results indicate that encoding the genetic sequences by RAAA, along with Word2Vec embedding, preserves the majority of evolutionary history in the resultant visualization. One prominent advantage of our approach is highlighting of abrupt changes in the direction of evolution. Such a phenomenon represents a significant change in a specific amino acid property that has been introduced by the RAAA in encoding genetic sequences. The Word2Vec framework allows us to incorporate long signatures in measuring the similarity between objects, instead of comparing them by considering point mutations. Additionally, such an embedding considers the neighbor effects of adjacent signatures through learning the word in its context.

We plan to generate Word2Vec embedding for hemagglutinin protein sequences of the influenza virus subtypes using other simplified alphabets and to study the impact of hyperparameters on the training performance. To assess the quality of embedding, it may be better to compute the distortion of the embedding instead of the correlation coefficient between distance matrices. Additionally, embedding can be conducted by implementing more than one RAAA. In this way, an amino acid is considered as a multidimensional object. Thus, we achieve a vector representation of a protein sequence for each alphabet. The final vector of the sequence can be obtained by concatenating all its representations. We believe our approach can be beneficial for providing a better insight into the evolutionary process and revealing factors that drive the evolution of species.

5. Acknowledgments

The reported study was funded by Russian Foundation for Basic Research (RFBR), project number 19-31-60025. Artyom Firstkov was funded by the Ural Mathematical Center with the financial support of the Ministry of Education and Science of the Russian Federation (Agreement number 075-02-2022-874)

References

[1] M. Forghani, P. Vasev, V. Averbukh, I. Ras, Three-dimensional visualization for phylogenetic tree, Scientific Visualization 9 (2017) 59–66. doi:10.26583/sv.9.4.06.

[2] M. Kimura, A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences, Journal of molecular evolution 16 (1980) 111–120. doi:10.1007/bf01731581.

[3] T. Jukes, C. Cantor, Evolution of protein molecules. in 'mammalian protein metabolism'.(ed. hn munro.) pp. 21–132, Academic Press, New York) 1 (1969) 504–511. doi:10.1016/B978-1-4832-3211-9.50009-7.

[4] N. Saitou, M. Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees., Molecular biology and evolution 4 (1987) 406–425. doi:10.1093/oxfordjournals.molbev.a040454.

[5] R. R. Sokal, A statistical method for evaluating systematic relationships., Univ. Kansas, Sci. Bull. 38 (1958) 1409–1438.

[6] M. Forghani, M. Khachay, Convolutional neural network based approach to in silico non-anticipating prediction of antigenic distance for influenza virus, Viruses 12 (2020) 1019. doi:10.3390/v12091019.

[7] X. Xia, Z. Xie, Protein structure, neighbor effect, and a new index of amino acid dissimilarities, Molecular biology and evolution 19 (2002) 58–67. doi:10.1093/oxfordjournals.molbev.a003982.

[8] K. Tomii, M. Kanehisa, Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins, Protein Engineering, Design and Selection 9 (1996) 27–36. doi:10.1093/protein/9.1.27.

[9] K. Ito, M. Igarashi, Y. Miyazaki, T. Murakami, S. Iida, H. Kida, A. Takada, Gnarled-trunk evolutionary model of influenza a virus hemagglutinin, PloS one 6 (2011) e25953. doi:10.1371/journal.pone.0025953.

[10] R. A. Neher, T. Bedford, R. S. Daniels, C. A. Russell, B. I. Shraiman, Prediction, dynamics, and visualization of antigenic phenotypes of seasonal influenza viruses, Proceedings of the National Academy of Sciences 113 (2016) E1701–E1709. doi:10.1073/pnas.1525578113.

[11] M. Forghani, P. Vasev, M. Bolkov, E. Ramsay, A. Bersenev, Phylotravis: A new approach to visualization of the phylogenetic tree, Programming and Computer Software 48 (2022) 215–226. doi:10.1134/S0361768822030045.

[12] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013). doi:10.48550/arXiv.1301.3781.

[13] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (2008).

[14] 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.

[15] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, et al., Biopython: freely available python tools for computational molecular biology and bioinformatics, Bioinformatics 25 (2009) 1422–1423. doi:10.1093/bioinformatics/btp163.

[16] L. Nanni, A. Lumini, A genetic approach for building different alphabets for peptide and protein classification, BMC bioinformatics 9 (2008) 1–10. doi:10.1186/1471-2105-9-45.

[17] Y.-C. Zuo, Q.-Z. Li, Using reduced amino acid composition to predict defensin family and subfamily: Integrating similarity measure and structural alphabet, Peptides 30 (2009) 1788–1793. doi:10.1016/j.peptides.2009.06.032.

[18] M. Forghani, M. Khachay, M. M. AlyanNezhadi, The impact of amino acid encoding on the prediction of antigenic variants, in: 2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), IEEE, 2020, pp. 1–5. doi:10.1109/ICSPIS51611.2020.9349560.

[19] J. D. Stephenson, S. J. Freeland, Unearthing the root of amino acid similarity, Journal of molecular evolution 77 (2013) 159–169. doi:10.1007/s00239-013-9565-0.

[20] X.-Y. Yang, X.-H. Shi, X. Meng, X.-L. Li, K. Lin, Z.-L. Qian, K.-Y. Feng, X.-Y. Kong, Y.-D. Cai, Classification of transcription factors using protein primary structure, Protein and Peptide Letters 17 (2010) 899–908. doi:10.2174/092986610791306670.

[21] R. C. Edgar, Local homology recognition and distance measures in linear time using compressed amino acid alphabets, Nucleic acids research 32 (2004) 380–385. doi:10.1093/nar/gkh180.

[22] 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.

[23] P. Ng, dna2vec: Consistent vector representations of variable-length k-mers, arXiv preprint arXiv:1701.06279 (2017). doi:10.48550/arXiv.1701.06279.

[24] M. Ostrovsky-Berman, B. Frankel, P. Polak, G. Yaari, Immune2vec: Embedding B/T cell receptor sequences in $\mathbb{R}^N$ using natural language processing, Frontiers in immunology (2021) 2706. doi:10.3389/fimmu.2021.680687.

[25] M. A. Cox, T. F. Cox, Multidimensional scaling, in: Handbook of data visualization, Springer, 2008, pp. 315–347. doi:10.1007/978-3-540-33037-0_14.

[26] P. Wang, W. Zhu, B. Liao, L. Cai, L. Peng, J. Yang, Predicting influenza antigenicity by matrix completion with antigen and antiserum similarity, Frontiers in microbiology 9 (2018) 2500. doi:10.3389/fmicb.2018.02500.

[27] 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.

[28] W. Shen, S. Le, Y. Li, F. Hu, Seqkit: a cross-platform and ultrafast toolkit for fasta/q file manipulation, PloS one 11 (2016) e0163962. doi:10.1371/journal.pone.0163962.

[29] A. Paszke, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).
[30] P. Virtanen, et al., Scipy 1.0: fundamental algorithms for scientific computing in python, Nature methods 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.
[31] N. Mantel, The detection of disease clustering and a generalized regression approach, Cancer research 27 (1967) 209–220.
[32] N. Cannata, S. Toppo, C. Romualdi, G. Valle, Simplifying amino acid alphabets by means of a branch and bound algorithm and substitution matrices, Bioinformatics 18 (2002) 1102–1108. doi:10.1093/bioinformatics/18.8.1102.