УДК 378:004
DOI: 10.25686/978-5-8158-2474-4-2025-385-396

Low-dimensional embedding for exploring the phylogenetic characteristics of Tick-borne Encephalitis Virus

Majid Forghani[1,2], Kazem Forghani[3] and Mikhail Bolkov[4,*]
[1] N.N. Krasovskii Institute of Mathematics and Mechanics of the UB RAS, St, Yekaterinburg, Russia
[2] Institute of Natural Sciences and Mathematics, Ural Federal University, Yekaterinburg, Russia
[3] Iran University of Science and Technology (IUST), Tehran, Iran
[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 development of a model for predicting virus evolution requires an accurate determination of virus characteristics. TBEV Analyzer is an analytical platform for characterizing the tick-borne encephalitis virus, which provides the evolutionary history of the virus in a hierarchical graph form called the clusteron structure. In this work, we explore three-fold characteristics of phylogenetic analysis, including subtype-lineage-clusteron, represented by clusteron structure through embedding the viruses into a 2D numerical space. The embedding is carried out by computing genetic distances and projecting viruses from genetic sequence space into the two-dimensional numerical vector space using dimensionality reduction techniques. Further, a phylogenetic tree is mapped into 2D space to describe the relationship between viruses. The 2D coordinates of the tree's inner nodes are optimized by gradient descent. To characterize each clusteron, we estimate and visualize its distribution in the final low-dimensional space. To the best of our knowledge, this is the first time that such a visualization has been made for tick-borne encephalitis virus that includes viruses, clusteron distribution, and a phylogenetic tree in one scene. Our preliminary results indicate that the approach has the potential to serve as an exploratory and complementary tool in studying and modeling the virus evolution in TBEV surveillance.

Keywords: visualization, embedding, TBEV Analyzer, phylogenetic tree, t-SNE.

Introduction

Viruses are an integral part of human life, some of which can lead to serious and severe diseases. During the evolution, they change their characteristics, part of which is due to evading the immune response of the host. The tick-borne encephalitis virus (TBEV) is a pathogen that can cause serious damage to the central nervous system, leading to disability or death in humans. The geographical distribution of TBEV forms a belt that includes the Russian Federation. Generally, TBEV includes four subtypes, namely, TBEV-FE (Far Eastern), TBEV-Bkl (Baikalian), TBEV-Sib (Siberian), and TBEV-Eu (European) [1]. The general distribution of TBEV is patchy. The distribution is locally restricted to a geographical area known as the focus, an environment in which a specific kind of TBEV circulates. This leads to the formation of micro- and macro-foci, which strongly depend on various factors, including the TBEV characteristics [2].

In 2019, an analytical platform titled TBEV Analyzer was introduced [3] that provides the phylogenetic characteristics of TBEV at three hierarchical levels, namely, subtype, lineage, and clusteron. The platform can directly fetch a query from GenBank [4] and analyze it, supply an interactive map for visualization of TBEV distribution, and offer an application programming interface (API) [5]. The recent version, TBEV Analyzer 3.0, has been announced, followed by theoretical and practical enhancements, including updating the phylogenetic model called clusteron structure (CS) (see Fig. 1), which is the core of the platform, by adding eleven novel clusterons [6]. The results of previous studies indicate that the platform can accurately determine the phylogenetic characteristics of a query virus.

The TBEV Analyzer relies on the clusteron approach (CA), which provides the overall picture of TBEV evolution in a hierarchical representation in graph form. The term “clusteron” stems from “cluster” and “clone.” Indeed, it is the smallest unit of viral population in the CA framework. A clusteron refers to the group of viruses featured by the three following conditions [7]:

- First, the viruses have an identical amino acid signature of the glycoprotein E fragment.
- Second, they are phylogenetically related.
- Third, they are characterized by a specific type of spatial distribution.

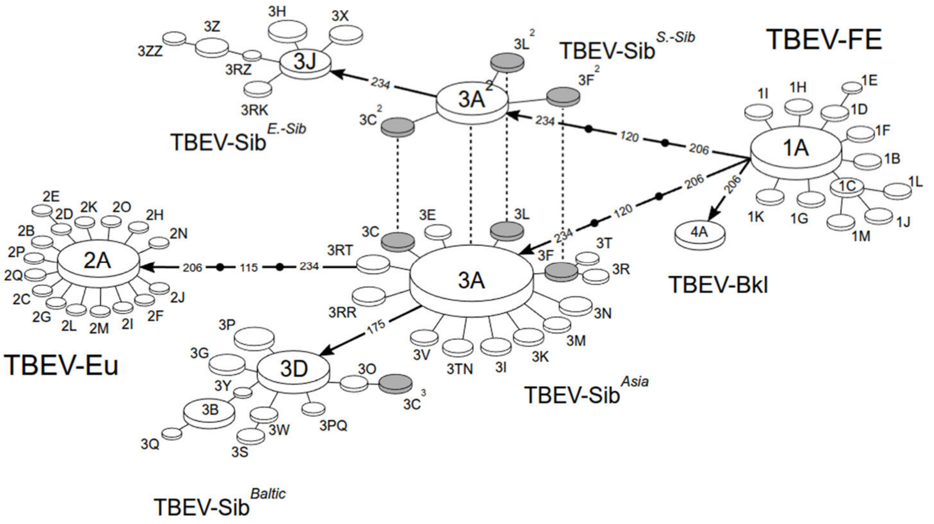

Figure 1. The current clusteron structure of TBEV. The homoplastic clusterons are connected with dashed lines. The connected clusterons have the same amino acid signature, but they belong to different subtypes/lineages. The CS has four dominant subtypes, among which the Siberian features four lineages

The computational pipeline of the platform mainly consists of preprocessing and verification of the query; construction of the phylogenetic tree inference for determining the subtype/lineage of the query; and finally, the search for matches based on clusteron-specific amino acid signatures. The pipeline requires specific configurations and carefully selected algorithms to obtain reliable results. This is the basic motivation behind the development of a unified analytical platform for TBEV.

The traditional way to represent the evolutionary history of a set of taxa is by constructing the phylogenetic tree. The tree accounts for the similarities and differences between genetic sequences to describe the sophisticated evolutionary process in a graphical, human-readable fashion. There are various applications of phylogenetic trees in viroinformatics. As an example, the information of phylogenetic trees can directly be incorporated in modeling the influenza antigenicity [8]. Previously, we suggested several approaches to visualize the phylogenetic tree of the influenza virus in a 3D numerical vector space. In [9], we suggested clustering taxa according to the physicochemical properties of amino acids of protein sequences. By incorporating the additional dimension into the 2D representation of the tree, the information of taxa clusters enriches the visualization, where each cluster can be independently studied according to its phylogeny.

Inspired by Rubik's cube solving algorithms [10], we proposed a visualization of phylogenetic trees by plotting the leaf-to-root path [11]. Since only the sequences of tree leaves are known, the algorithm of reconstructing the ancestral sequences was employed to obtain the sequence of every inner node. Further, the genetic sequences were encoded by one-hot encoding and embedded by multidimensional scaling (MDS) [12] or t-Distributed Stochastic Neighbor Embedding (t-SNE) [13] into 2D or 3D space for visualization. This approach was enhanced by applying reduced amino acid alphabets (RAAA) to embed other aspects of evolution along the 2D tree representation and embedding by Word2Vec [14] instead of evolutionary distance embedding [15, 16].

Considering the aforementioned works, a question may arise: "How can the relationships between viruses in the clusteron structure be represented in a low-dimensional vector space?" Perhaps the answer to this question may help us better understand and explore the evolutionary process. In addition, this may facilitate understanding those boundaries that form a clusteron in terms of the physicochemical properties of genetic sequence, which may reveal the hidden mechanism of the evolutionary process.

The aim of this paper is to generate a visualization of TBEV viruses by employing a two-step embedding from the space of genetic sequence into the 2D numerical space. Our contribution to this paper is twofold:

- Performing the clusteron approach analysis of all TBEV viruses registered in GenBank, determining their characteristics using the TBEV Analyzer platform, and creating a unified dataset of TBEV.
- Proposing a new visualization approach for TBEV by embedding the viruses from genetic space into 2D numerical vector space. This visualization includes three components: viruses, the phylogenetic

tree of clusteron representatives, and finally supplying the kernel density estimation (KDE) plot for each clusteron in the 2D visualization scene to explore the underlying structure of clusterons.

To the best of our knowledge, this is the first attempt to visualize relationships between TBE viruses by combining three components: viruses, a phylogenetic tree, and clusteron distributions.

The remaining parts of this paper are organized as follows: Section 2 explains the materials and methodology in more detail. Section 3 presents the computational experiment setup and the results. Finally, we give conclusions in Section 4.

Methodology

The overall schema of the proposed approach is illustrated in Fig. 2. The approach consists of the following steps:

- Analyzing GenBank records for TBEV and determining three-fold characteristics of viruses.
- Computing the genetic distance matrix of viruses collected in the previous step using both coding (nucleotide) and protein sequences.
- Embedding the viruses by applying dimensionality reduction techniques to the genetic distance matrix and obtaining the coordinate of each virus in the 2D space.
- Constructing the phylogenetic tree for representatives of clusterons.
- Embedding the tree by optimizing the coordinates of the tree's inner nodes in 2D space through gradient descent.
- Computing the distribution of clusteron members.
- Visualizing the obtained results, including all viruses, the distribution of clusterons, and the phylogenetic tree in 2D space.

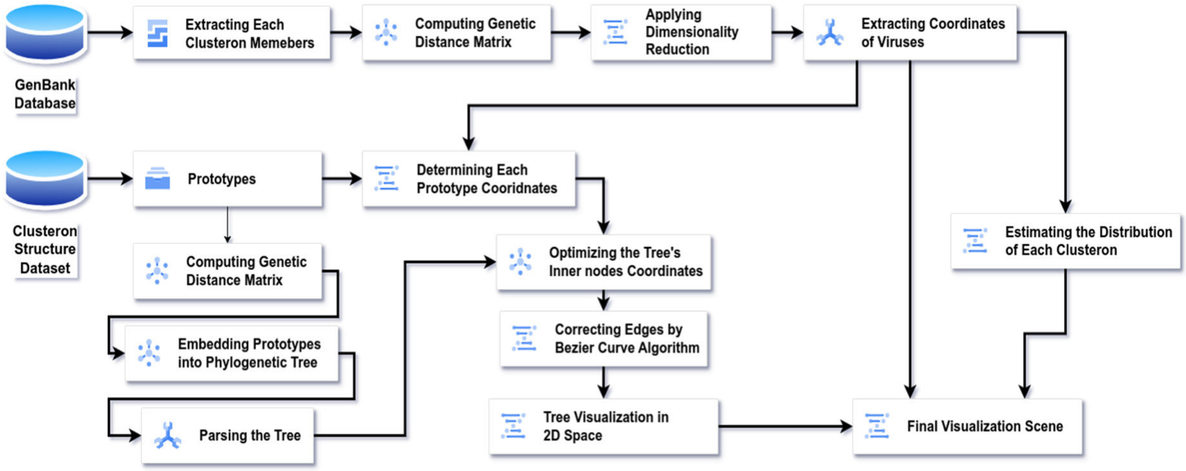

Figure 2. The overall schema of our pipeline. The procedure provides two-step embedding. The 2D coordinates of viruses are obtained by computing the pairwise distance matrix and applying a dimensionality reduction technique. Further, a phylogenetic tree is constructed on prototype sequences and embedded into the 2D space. Finally, we estimate the distribution of each clusteron and plot it in the visualization scene

1. Data Collection

Typically, the TBEV Analyzer requires both coding (nucleotide) and amino acid sequences to infer the phylogenetic characteristics of a query virus. On the one hand, the coding sequence is essential to determine the virus characteristics at the global scale, i.e., subtype and lineage. On the other hand, the amino acid sequence is required to search for a match regarding the clusteron-specific amino acid signature within the lineage or subtype, in the result of which the platform detects the type of clusteron for the query. Algorithmically, the TBEV Analyzer works with two types of data: coding and amino acid sequences. In practice, the platform requires a fragment of the genetic sequence of glycoprotein E as input data. The fragment includes 454 nucleotides (positions 309–762) in gene E. Since the amino acid sequence can easily be achieved by translating the nucleotide sequence codons, we restrict our work to the coding sequences.

In the TBEV Analyzer framework, a clusteron has three hierarchical characteristics: subtype, lineage, and clusteron. A clusteron in the framework has three elements: a name (or label), a prototype, and a specific amino acid signature. The prototype coding sequence is a key to identifying the phylogenetic lineage or subtype of a query, whereas the specific amino acid signature is the primary component to determine the type of clusteron. A prototype is the representative of a clusteron. It plays a role similar to the centroid of a cluster, where this centroid is assigned from the biological aspect.

To compile the database for our experiments, we extract viruses using the query "TBEV" or "tick-borne encephalitis virus" in FASTA format from the nucleotide core (nuccore) of the GenBank. The records are then fed into the TBEV Analyzer, in the results of which three data sets are obtained (see Fig. 3). A virus record can have one of the following cases:

- The record does not have the target fragment sequence of gene E completely. Such a query is assigned to the "invalid set."
- The record has the complete sequence of the target fragment, and it belongs to a known clusteron within the CA framework. Such a record is assigned to the "clusteron members set," denoted by S_c.
- The final case happens when the record has the complete sequence of the target fragment, but it does not belong to any known clusteron. The collection of such records is called the "unique set."

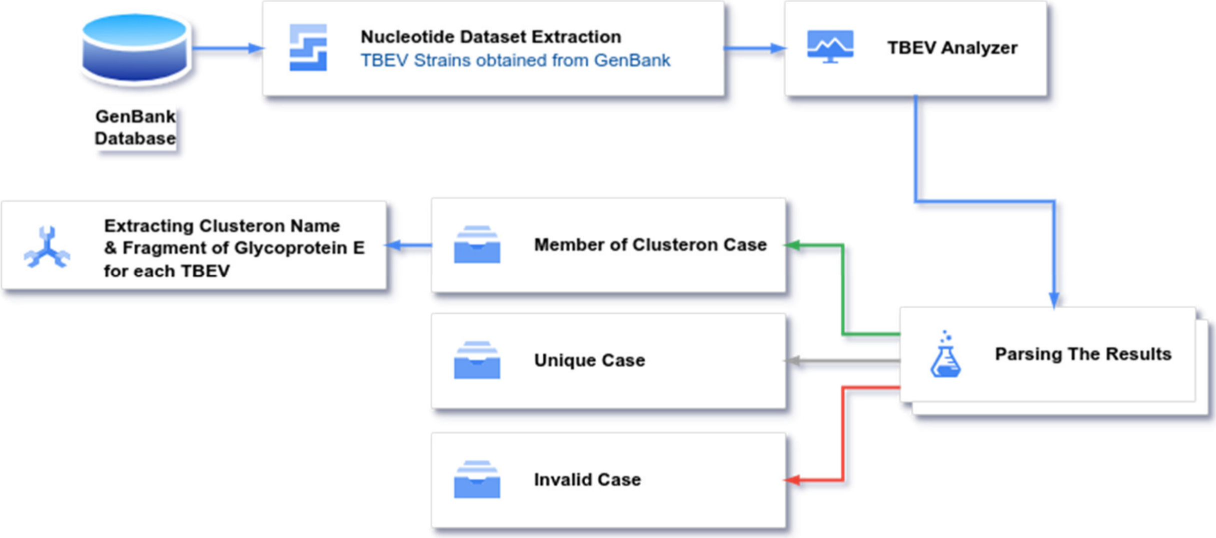
GenBank Database
Nucleotide Dataset Extraction
TBEV Strains obtained from GenBank
TBEV Analyzer
Extracting Clusteron Name & Fragment of Glycoprotein E for each TBEV
Member of Clusteron Case
Unique Case
Invalid Case
Parsing The Results

Figure 3. Database preparation for the proposed pipeline. We search GenBank by query "TBEV" or "tick-borne encephalitis virus." All obtained records are sent to the TBEV Analyzer. After parsing the results of analysis, the members of each clusteron are determined. Each entry of the final collected dataset includes three fields: the GenBank ID, the specific fragment of the coding sequence of glycoprotein E, and the label (i.e., the clusteron name)

After parsing TBEV Analyzer's results, we focus on the clusteron members set, where each member has its own label and the sequence of the target fragment. Thanks to the TBEV Analyzer, all members have sequences of the same length, acquired from the results of the platform analysis. Therefore, there is no need for preprocessing steps such as alignment.

In addition to the set of clusteron members, we require information about the CS prototypes. As stated earlier, each prototype also contains the coding sequence of the target fragment. Hence, each member of the prototype set has a label (or the name of the clusteron) and a corresponding coding sequence. This set is referred to as the "prototype set" and denoted by S_p. Since each prototype represents a virus sequence taken from GenBank, it follows that $S_p \subset S_c$.

2. Computing genetic distance

There are several genetic distance metrics, each of which has its own advantages and disadvantages. For the sake of simplicity, we employ p-distance, or uncorrected distance. The distance is computed using the following formula for both coding and protein sequences:

$$d(S1, S2) = \frac{diff(S1, S2)}{L},$$

where $S1$ and $S2$ are two sequences with the same length L, $d(S1, S2)$ is their distance, and $diff(S1, S2)$ is the number of sites with different nucleotides/amino acids. We compute the pairwise distance matrix for the set of all viruses in S_c in two formats: coding and protein sequences. These two matrices are further combined through a weighted average to leverage the information of both coding and protein sequences. The obtained pairwise distance matrix for the set of clusteron members, i.e., S_c, is denoted by D_c.

3. Embedding GenBank viruses

A key aspect of our approach is the representation of three types of information (viruses, a tree, and clusterons) in the low-dimensional space to facilitate the exploration of viral relationships in the 2D space. To achieve this, the t-SNE technique is applied to the distance matrix D_c to obtain the coordinates of each virus within S_c in the low-dimensional space. t-SNE is a non-linear dimensionality reduction technique that employs the distances between objects in high-dimensional space, representing them in conditional probabilities, and minimizes Kullback-Leibler divergence between distributions of similarity in high- and low-dimensional spaces. The embedding can mainly be controlled by perplexity, which is related to the number of nearest neighbors used in the algorithm, and early exaggeration, which affects the space between clusters.

We apply t-SNE and obtain the coordinate of each virus in the low-dimensional space. In such a manner, the coordinates of each prototype in the embedding space can be extracted, and the set of these coordinates is denoted by X_{pro}.

4. Reconstructing the phylogenetic tree

Algorithms for reconstructing a phylogenetic tree from a set of aligned sequences can be categorized into three main groups: distance-based methods (e.g., neighbor-joining), character-based methods (such as maximum parsimony and maximum likelihood), and Bayesian inference methods. Among these, the neighbor-joining (NJ) algorithm is widely used in molecular evolution studies that can provide a rough phylogenetic tree from a pairwise distance matrix of sequences. While we adopt this algorithm for its simplicity, our approach works with any tree reconstruction method that provides branch lengths.

A binary phylogenetic tree is constructed by applying the NJ algorithm to the pairwise distance matrix of prototypes (denoted by D_p). The prototypes are located in the leaves of the tree, while the inner nodes express the inferred similarity between them. As another component of the final visualization, we embed this phylogenetic tree into the low-dimensional space gained in Subsection 2.3.

5. Embedding phylogenetic tree in low-dimensional space

Here, we embed the tree structure in the low-dimensional space. To achieve this, we first need to establish a correspondence between the objects in the tree structure and those in the low-dimensional space. This bridge can be depicted by prototypes. Subsection 2.3 gives the coordinates of prototypes, i.e., X_{pro}, in the low-dimensional space, while prototypes are located in the tree leaves. Accordingly, in order to embed the tree, we need to optimize the position of inner nodes in the low-dimensional space in such a way that the edge lengths of the tree are preserved as much as possible.

Zhang et al. [17] proposed a visualization algorithm for genetic sequences called PhyloMap. In their method, they optimized the inner node positions using the gradient descent and the following objective function, which is very similar to “Sammon's mapping”:

$$E = \frac{1}{\sum_{i<j} s.d^*_{i,j}} \sum_{i<j} \frac{\left(s.d^*_{i,j} - d_{i,j}\right)^2}{s.d^*_{i,j}}, \tag{1}$$

where $d^*_{i,j}$ and $d_{i,j}$ are the distances between objects i and j in the tree and the low-dimensional space, respectively. The parameter s in Eq. 1 acts as a scaling factor and compensates for the differences between spaces when the distances in the phylogenetic tree and low-dimensional space come from different natures. Their algorithm uses the gradient descent on the inner nodes in order to minimize error expressed by Eq. 1. To optimize the inner node coordinates, we use the central difference formula for approximating the partial derivative of the objective function numerically. Given that our phylogenetic tree obtained from the previous step is binary and rooted, each inner node (excluding the root) is connected to three nodes (two children and one ancestor). Consequently, we consider only these three distances in the objective function for each inner node. In the case of the root node, we use only two edges of its children.

A phylogenetic tree may not be perfectly embedded in the low-dimensional space. For instance, the straight-line distance between two nodes in the low-dimensional space can be shorter than their corresponding edge length in the tree. Zhang et al. tackled this issue by introducing the application of the Bezier curve [18], where the straight edge is replaced by a smooth curve. In this case, the $d_{i,j}$ in Eq. 1 is replaced by $d_{i,j}^{b}$, which is the length of the Bezier curve between nodes i and j. Here, we employ the quadratic Bezier curve, which requires at least three points to draw a curve. The curve points are computed as follows:

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + \ t^2 P_2,$$

where P_0, P_1, and P_2 are the control points, and $t \in [0,1]$. The interval [0,1] is discretized to m values, where m determines the resolution of the curve.

6. Estimating the distribution of clusteron members

As an additional layer of information in our visualization, we compute and plot the distribution of clusteron members for each clusteron individually. One critical component of the clusteron definition is its specific geographical distribution. Since we do not have access to the exact geographical location of the sample for some viruses in S_c, we use the 2D coordinates from Subsection 2.3 to obtain an estimation of distribution for each clusteron in the embedding space. The distribution is estimated using the kernel density estimation (KDE) method [19]. Given a set of n points $X = \{x_1, x_2, \ldots, x_n\}$, the KDE value for a point x can be computed by:

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

where $h > 0$ is the bandwidth parameter that controls the smoothness of the estimated distribution, and $K(.)$ is the kernel function. In our case, a Gaussian kernel is employed that produces a smooth and continuous density estimate without artifacts participating. The coordinates of the clusteron members in the low-dimensional space are employed by the KDE method to compute the distribution, which is then visualized as a part of the final 2D scene.

7. Visualization scene

In this study, the dimension of the visualization scene is set to two. The final scene comprises three main components:

- A 2D visualization of all viruses within the set S_c, where each member is associated with a specific clusteron.
- A phylogenetic tree, in which each leaf corresponds to a clusteron prototype. Since $S_p \subset S_c$, the coordinates of the tree leaves are fixed, while the positions of inner nodes are achieved by solving optimization tasks through gradient descent.
- The KDE-based spatial distribution plot for each clusteron that represents the density of its members in the embedding space.

The algorithmic overview of the approach is presented in Algorithm 1.

Algorithm 1. Visualization of TBEV

```
Input: The set of sequences for TBEVs from GenBank: $S$, the set of prototype sequences: $S_p$, error: $e$, the number of iterations: $iter\_num$.
Output: A 2D map of viruses and their characteristics.
1 Filtering the set $S$ by the TBEV Analyzer and obtaining the set $S_c$.
2 Computing the pairwise distance matrix $D_c$ for objects in $S_c$.
3 Embedding the objects of $S_c$ into 2D space by applying t-SNE to $D_c$.
4 Determining the coordinates of prototypes, i.e., $X_{pro}$, from step 3.
5 Computing pairwise distance matrix $D_p$ for $S_p$.
6 Constructing the phylogenetic tree $T$ from $D_p$.
7 Fixing the coordinates of leaves in $T$ using $X_{pro}$.
8 Initializing the 2D coordinates of inner nodes in $T$.
```

```
9  while $E_i < \delta$ or $i < iter\_num$ do
10     for inner node $x$ do
11         if $i\%5 == 0$ then
12             Update the coordinate of $x$ by gradient descent.
13         else
14             Update the coordinate of $x$ by gradient descent when at least one of the edges connected to the $x$
               has the condition $d_{i,j} > s.d^*_{i,j}$.
15     $E_i = error\ by\ Eq. 1$

16 for each edge with $d_{i,j} < s.d^*_{i,j}$ do
17     Correct the edge by computing the Bezier curve between nodes $i$ and $j$ such that $d_{i,j} \approx s.d^*_{i,j}$.

18 for each clusteron $C_k$ do
19     Determine its members from $S_c$.
20     Obtain the coordinates of elements in $S_c$ from the results of step 3. The set of coordinates is denoted by
       $X_{c_k}$.
21     Compute the distribution for $X_{c_k}$.
22     Plot $X_{c_k}$ and its distribution in the 2D scene.

23 Plotting the edges of the tree in the 2D scene.
```

Experiments & Results

We downloaded all TBE viruses from the nucleotide core of the GenBank database. After analyzing them with the TBEV Analyzer (available at tbev.viroinformatics.com), a total of 895 viruses passed the analysis, which formed S_c. Other viruses were excluded from the experiment due to lack of a complete sequence (of the target fragment) or of a known clusteron. Thus, each member of S_c has the coding sequence of the target fragment of the gene E and its label, i.e., the clusteron to which it belongs. As mentioned earlier, we applied the p-distance to compute the distance matrices of coding and protein sequences. A weighted average was applied to obtain the pairwise distance D_c, where the weight for coding and protein distance matrices was 0.4 and 0.6, respectively. The distance matrix was then fed into the t-SNE algorithm, available via the scikit-learn package [20], to map viruses into the 2D numerical space. After embedding, each member of S_c is characterized by three properties: a label, a coding sequence, and a 2D coordinate.

We used the Kimura 2-parameter distance [21] to compute D_p and construct the phylogenetic tree by applying the NJ algorithm. The tree was built from the sequences of prototypes S_p, using MEGA software version 11.0.13 [22]. MEGA facilitates the tree parsing by allowing the tree to be saved in a tabular format, providing access to each edge along with its length and endpoints. Since distances $d^*_{i,j}$ and $d_{i,j}$ in Eq. 1 have different natures, we employed distance scaling parameter s to embed the tree in low-dimensional space. Each prototype (located at a leaf of the tree) is also a member of S_c, and its 2D coordinate was available. To preserve edge length in the 2D space, we optimized the coordinates of the inner nodes by numerically approximating the partial derivative and applying gradient descent to minimize the objective function defined in Eq. 1. A quadratic Bezier curve was further used to correct the edges on the 2D map.

Totally, there are 68 clusterons within the CA framework. The study of each clusteron as a significant unit of CA is essential. We customized the visualization of each clusteron by a unique color. We plotted all objects of S_c in the 2D visualization scene, while each object is colored regarding its label. The S_c may contain duplicate entries with identical coding sequences of the target fragment, leading to the same 2D coordinates. To highlight the number of viruses with the identical sequence, the radius of the plotted marker in the 2D scatter plot can be adjusted by the number of duplicates. We labeled each prototype point in the scene with the name of its clusteron. The set of edges, refined using Bezier curve correction, was also plotted into the scene. The last component is the distribution of the clusteron members. The distribution was computed by the function *kdeplot* from the Python package Seaborn [23]. The visualization was enhanced by plotting the contours of the 2D density for each clusteron, coloring it according to the clusteron label, and decreasing the alpha value to see overlaps between clusterons.

The final results are presented in Fig. 4. Here, we clearly observe four distinct subtypes: TBEV-FE (Far-Eastern) in the bottom, TBEV-Sib (Siberian) with its lineages, TBEV-Bkl (Baikalian) at the right side of the figure, and TBEV-Eu (European) in the center of the figure. An interesting observation arises with the clusteron founders, where their distributions are clearly larger than derivative clusterons. Note that a TBEV can be characterized at three phylogenetic hierarchical levels: subtype, lineage, and clusteron. The lineage is currently defined only for the Siberian subtype. Both the subtype and lineage of a query virus can be simply inferred from the coding sequence of the target fragment of gene E. In contrast, the clusteron is determined using a clusteron-specific amino acid signature. If the query's signature matches that of a known clusteron within the query's subtype and lineage, the virus belongs to that clusteron. As observed in Fig. 4, some clusterons have a wide distribution while others are more compact. Such dispersion in distribution is the result of the embedding using nucleotide-level information. The t-SNE algorithm controls the visualization through two parameters: perplexity and early exaggeration. Since the perplexity is related to the number of nearest neighbors, it is expected that leveraging more viruses for derivative clusterons can improve their distribution and visualization quality.

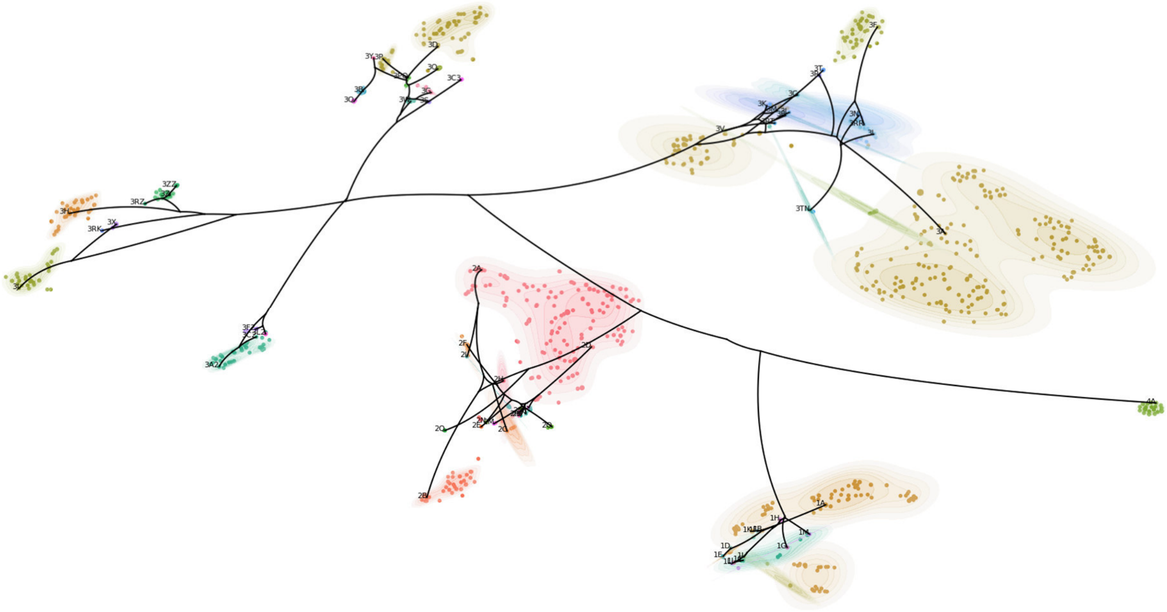

Figure 4. The visualization of TBE viruses, accepted by the framework of the clusteron approach in the 2D numerical space. This scene demonstrates three key elements: the viruses, their clusteron distribution, and the phylogenetic tree of prototypes. Each point in the plot is colored according to its clusteron label, and the same color scheme is used for the corresponding clusteron distribution. The location of each prototype is labeled by its clusteron name

In our previous work [16], we suggested the visualization of the leaf-to-node path as a unified curve. This can be advantageous when working with a large number of viruses, in studying pivotal events of evolution, checking the variability among viruses, observing the main trend, and monitoring the direction of evolution. Taking this idea, we generated a smoothed version for visualizing the evolution of TBEV (see Fig. 5).

A decisive factor of our approach is the distance metric, which participated in the computation of 2D coordinates. The definition of distance can be customized by application of reduced amino acid alphabets [24] or by introducing the amino acid physicochemical properties [25]. As a supplementary experiment, we used both the coding and amino acid sequences of each virus and embedded them by Word2Vec into a numerical vector space. Further, the objects were mapped from this high-dimensional space into 2D space by t-SNE. Fig. 6 shows the visualization of viruses leveraging the information of coding sequences. Fig. 7 illustrates the visualization of the same viruses by adding information on amino acid sequences along with the coding sequences. The subtype and lineage can be clearly recognized in Fig. 6. In Fig. 7, we customized the marker and color of viruses to provide a unique visual shape for members of each clusteron. It can be seen that clusterons with a higher number of members achieve better representation. On the contrary, the clusterons with a lower number of members are close together and more compact in the region, which makes their recognition difficult. This may also be due to the t-SNE settings. Unfortunately, replacing t-SNE by MDS does not generate a desirable result. We believe that controlling the influence of coding and amino acid sequence information involved in embedding calculations using a weighted average technique has a direct impact on the quality of visualization. Note that although the embedding of high-dimensional vectors into a 2D space using t-SNE is not isometric, it effectively captures the structure of virus clusterons at a coarse level and provides visually meaningful separation for most clusterons at a fine level. It should be better to mention that we also applied other dimensionality reduction techniques such as MDS, Isomap, and Locally Linear Embedding, but the best results were obtained by t-SNE. However, the choice of distance metric and dimensionality reduction technique strongly affects the final visualization, which requires further investigation.

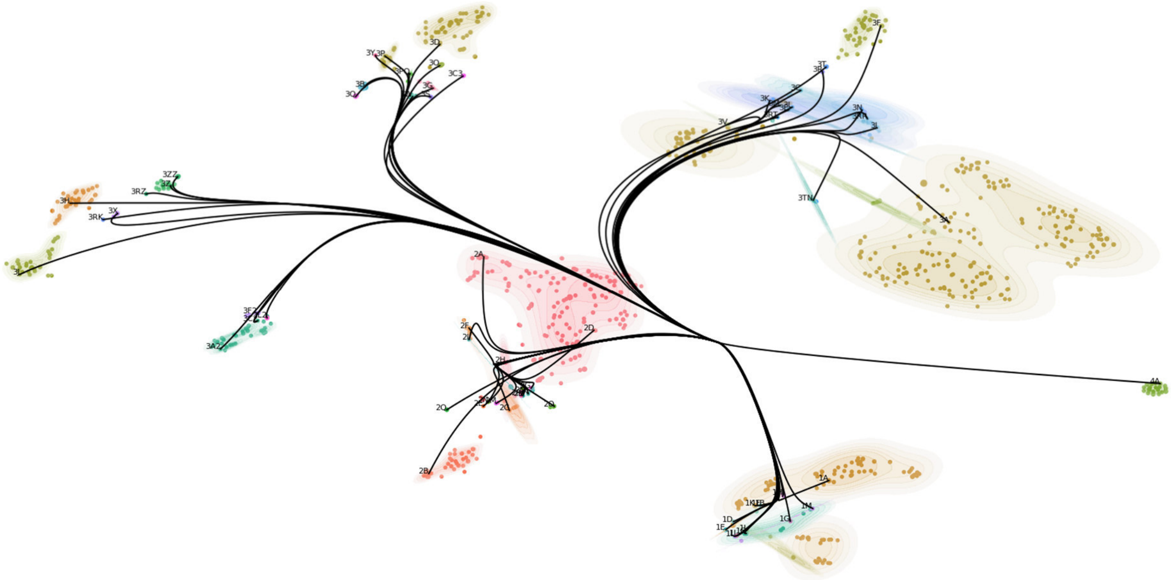

Figure 5. The visualization of the leaf-to-root path for TBEVs. Each path begins at a leaf node, passes through inner nodes, and terminates at the root. The leaf, root, and their inner nodes along the path are employed as control points to generate a smooth curve by the Bezier curve

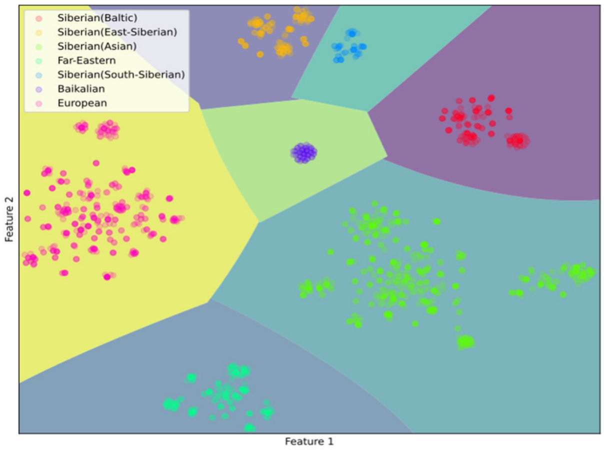

Figure 6. Visualization of the TBEV subtype and lineage. The visualization was obtained by the coding sequence embedding of viruses into a high-dimensional numerical vector space, followed by applying t-SNE to generate 2D coordinates. Virus colors are customized based on its subtype/lineage. Note that the boundary approximation was obtained using a support vector machine classifier

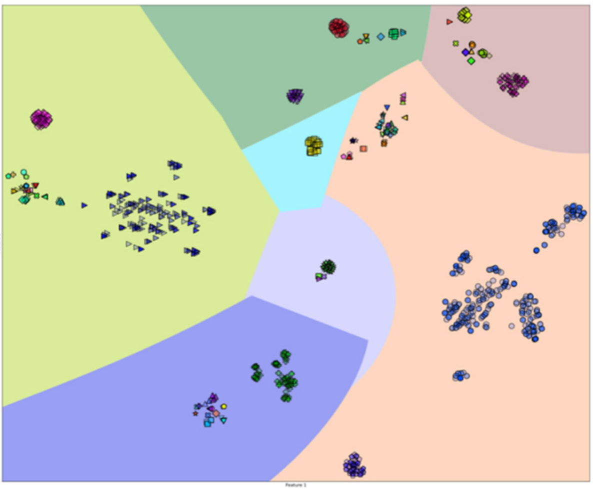

Figure 7. Visualization of TBEV's threefold phylogenetic characteristics. The visualization was obtained by embedding both coding and amino acid sequences of viruses, computing the distance matrix, and then applying t-SNE. Virus colors and markers are customized based on their clusteron. Regions are highlighted according to subtype/lineage. Note that the boundary approximation was obtained using a support vector machine classifier

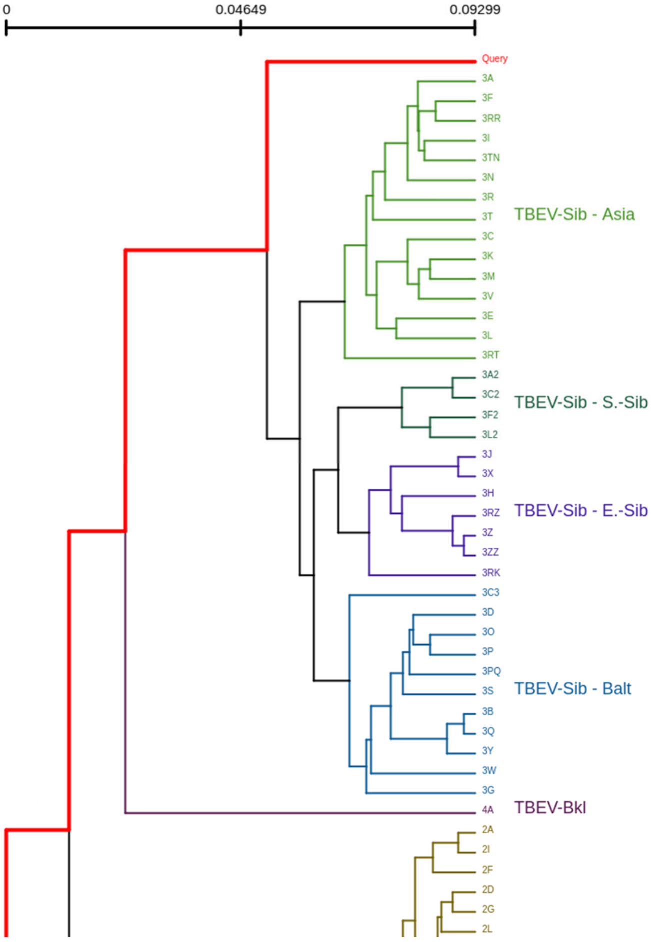

Figure 8. An example of a TBEV virus recognized as unique by the TBEV Analyzer. In this partial visualization, the virus appears to have a close relationship with the Siberian subtype within the phylogenetic tree. Recognizing such unique sequences in our visualization scene requires defining boundaries for the formation of clusterons

Considering the proposed visualization, a challenge may arise when the model encounters viruses that are recognized as unique by the TBEV Analyzer. The design and development of algorithms in the TBEV Analyzer ensure that a unique virus maintains a distant relationship with known clusterons within the phylogenetic framework. This guarantees that a virus falling outside the scope of known lineages/subtypes is definitively tagged as unique (see Fig. 8). While this can be clearly expressed in the context of genetic distance by the CA framework, there is no simple boundary yet in the proposed visualization that defines when a virus is unrelated or unique. This challenge is critical for designing a robust model to characterize TBEV and to define thresholds or boundaries for the identification and formation of a clusteron, which also provides room for further investigation.

Conclusions

In this pilot study, we attempt to explore and characterize the TBEV clusterons by embedding them in a 2D numerical space. To the best of our knowledge, this is the first time that such a customized visualization has been generated for TBEV. The main advantage of the proposed approach is embedding three types of information into the 2D visualization scene: the spatial representation of viruses on the map, the distribution of clusterons, and the relationship between clusteron representatives in a phylogenetic tree. A desirable fact in Fig. 4 is the distinct representation of subtypes/lineages in the visualization scene with dominant distribution of clusteron founders, i.e., clusterons $1A$, $2A$, $3A$, $3A^2$, $3J$, $3D$, and $4A$. It should be noted that this approach does not claim to be a replacement for classical representation of phylogenetic trees.

From a technical perspective, a drawback of our approach is handling a large number of viruses or embedding a large phylogenetic tree in the 2D space, causing a complex visualization scene. This issue becomes particularly serious when the distance for obtaining the 2D coordinates and that for constructing the phylogenetic tree are different. Switching from 2D to 3D space may help refine and simplify the visualization. In comparison with our previous work, the coordinate of an inner node in the phylogenetic tree is achieved through an optimization task instead of applying the ancestral reconstruction algorithms and encoding the inner node sequence directly. Although we use the t-SNE technique to map from high-dimensional to 2D spaces, it can easily be replaced by any dimensionality reduction technique. Also, the phylogenetic tree algorithm can be replaced by any algorithm that generates a tree with branch lengths.

Our results indicate that such embedding can be advantageous for characterizing TBEV at the coarse level by providing its subtype/lineage only using the information obtained from the coding sequence. The three-fold phylogenetic characteristics (i.e., subtype-lineage-clusteron) of a virus cannot be determined without incorporating the amino acid signature. Adding the information of amino acid sequences improves the quality of visualization. This can be observed for clusteron founders $3A$ and $3A^2$. While these founders have the same clusteron-specific amino acid signature, they belong to different phylogenetic lineages. That is why their representations are far from each other in Fig. 4.

Further investigation is required to develop a reliable algorithm to accurately identify the phylogenetic characteristics of TBEV, especially in the case of unique viruses (see a unique case in Fig. 8). This helps us determine the physicochemical and phylogenetic boundaries that define a clusteron in the CS. Moreover, the search for an appropriate distance metric is of interest. During the data collection process, we have identified several unique viruses with the same signature that can be analyzed for forming new potential clusterons in the CA framework, which will be reported to the TBEV Analyzer team. Taken all together, our results reveal that the approach can serve as an exploratory and complementary tool in studying and modeling evolution in TBEV surveillance. We expect that extending the genetic database of TBEV will lead to an expansion of the clusteron member set and improve our understanding of TBEV evolution. We plan to take advantage of the ZADU package [26] to assess embeddings using various metrics. Further research can focus on investigating the correlation between geographical, phylogenetic, and embedding distances in order to extract key insights about hidden mechanisms underlying viral evolution.

Acknowledgments

The work was performed as part of research conducted in the Ural Mathematical Center with the financial support of the Ministry of Science and Higher Education of the Russian Federation (Agreement number 075-02-2025-1549).

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] 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.

[2] A. Wallenhammar, R. Lindqvist, N. Asghar, S. Gunaltay, H. Fredlund, A. Davidsson, S. Andersson, A. K. Overby, M. Johansson, Revealing new tick-borne encephalitis virus foci by screening antibodies in sheep milk, Parasites & Vectors 13 (2020) 1–12. doi:10.1186/s13071-020-04030-4.

[3] 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.

[4] 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.

[5] 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

[6] 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.

[7] 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.

[8] W. T. Harvey, D. J. Benton, V. Gregory, J. P. Hall, R. S. Daniels, T. Bedford, D. T. Haydon, A. J. Hay, J. W. McCauley, R. Reeve, Identification of low-and high-impact hemagglutinin amino acid substitutions that drive antigenic drift of influenza a (h1n1) viruses, PLoS pathogens 12 (2016) e1005526. doi:10.1371/journal.ppat.1005526.

[9] M. Forghani, P. Vasev, V. Averbukh, Three-dimensional visualization for phylogenetic tree, Scientific Visualization 9 (2017) 59–66.

[10] C. A. Steinparz, A. P. Hinterreiter, H. Stitz, M. Streit, Visualization of Rubik's cube solution algorithms., in: EuroVA@ EuroVis, 2019, pp. 19–23.

[11] M. Forghani, P. Vasev, E. Ramsay, A. Bersenev, Visualization of the evolutionary path: an influenza case study, in: CEUR Workshop Proc.–CEUR-WS, volume 3027, 2021, pp. 358–368.

[12] M. A. A. Cox, T. F. Cox, Multidimensional Scaling, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 315–347. doi:10.1007/978-3-540-33037-0_14.

[13] L. v. d. Maaten, G. Hinton, Visualizing data using t-SNE, Journal of machine learning research 9 (2008) 2579–2605.

[14] 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.

[15] M. Forghani, A. Firstkov, P. Vasev, E. Ramsay, Visualization of the evolutionary trajectory: Application of reduced amino acid alphabets and word2vec embedding, in: Graphicon-Conference on Computer Graphics and Vision, volume 32, 2022, pp. 275–287.

[16] 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.

[17] J. Zhang, A. M. Mamlouk, T. Martinetz, S. Chang, J. Wang, R. Hilgenfeld, Phylomap: an algorithm for visualizing relationships of large sequence data sets and its application to the influenza A virus genome, BMC bioinformatics 12 (2011) 248. doi:10.1186/1471-2105-12-248.

[18] L. Shao, H. Zhou, Curve fitting with Bezier cubics, Graphical models and image processing 58 (1996) 223–232. doi:10.1006/gmip.1996.0019.

[19] Y.-C. Chen, A tutorial on kernel density estimation and recent advances, Biostatistics & Epidemiology 1 (2017) 161–187. doi:10.1080/24709360.2017.1396742.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825– 2830.

[21] 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.

[22] K. Tamura, G. Stecher, S. Kumar, MEGA11: molecular evolutionary genetics analysis version 11, Molecular biology and evolution 38 (2021) 3022–3027. doi:10.1093/molbev/msab120.

[23] M. L. Waskom, Seaborn: statistical data visualization, Journal of Open Source Software 6 (2021) 3021. doi:10.21105/joss.03021.

[24] M. Forghani, A. Firstkov, M. Alyannezhadi, D. Danilenko, A. Komissarov, Reduced amino acid alphabet-based encoding and its impact on modeling influenza antigenic evolution, Russian Journal of Infection and Immunity 12 (2022) 837–849. doi:10.15789/2220-7619-RAA-1968.

[25] S. Kawashima, P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, M. Kanehisa, AAindex: amino acid index database, progress report 2008, Nucleic acids research 36 (2007) D202–D205. doi:10.1093/nar/gkm998.

[26] H. Jeon, A. Cho, J. Jang, S. Lee, J. Hyun, H.-K. Ko, J. Jo, J. Seo, ZADU: A python library for evaluating the reliability of dimensionality reduction embeddings, in: 2023 IEEE Visualization and Visual Analytics (VIS), IEEE, 2023, pp. 196–200. doi:10.1109/VIS54172.2023.00048.