ISSN 0361-7688, Programming and Computer Software, 2022, Vol. 48, No. 3, pp. 215–226. © Pleiades Publishing, Ltd., 2022.
Russian Text © The Author(s), 2022, published in Programmirovanie, 2022, Vol. 48, No. 3.

PhyloTraVis: A New Approach to Visualization of the Phylogenetic Tree

M. Forghani[a,b,*], P. A. Vasev[a,], M. A. Bolkov[c,***], E. S. Ramsay[d,****], and A. Y. Bersenev[a,*****]**

[a] *N.N. Krasovsky Institute of Mathematics and Mechanics, Yekaterinburg, 620108 Russia*

[b] *Ural Federal University Named After the first President of Russia B. N. Yeltsin, Yekaterinburg, 620002 Russia*

[c] *Institute of Immunology and Physiology UB RAS, Yekaterinburg, 620049 Russia*

[d] *Smorodintsev Research Institute of Influenza, St. Petersburg, 197376 Russia*

**e-mail: forghani@imm.uran.ru*

***e-mail: vasev@imm.uran.ru*

****e-mail: mbolkov@iip.uran.ru*

*****e-mail: warmsunnyday@mail.ru*

******e-mail: bay@hackerdom.ru*

Received December 18, 2021; revised January 11, 2022; accepted January 20, 2022

Abstract—The study of evolution is an essential task in predicting the variability of species, especially for pathogens such as viruses. One of the main stages of evolutionary analysis is constructing a phylogenetic tree. This work is devoted to developing a new approach for visualization of the phylogenetic tree, which is based on reconstructing the evolutionary trajectory of a taxon in three-dimensional space. An evolutionary trajectory is a path that connects a particular taxon and the root of the tree. By reconstructing ancestral sequences and applying one-hot-encoding, each tree node is represented as a multidimensional object, then mapped into three-dimensional space using the embedding method, due to which, evolutionary paths from leaves to the tree's root are generated. This approach makes it possible to visualize rapid changes in evolutionary direction, both locally and globally. The results are based on the experiments on visualization of the evolutionary trajectory of the H3N2 influenza virus and the development of a publicly available web platform called PhyloTraVis. They suggest the application of our approach for early detection of changes in the direction of evolution, the study of evolutionary dynamics, evaluation of emerging novel virus variants, and modeling of possible antigenic diversity, which are important tasks in computational virology.

DOI: 10.1134/S0361768822030045

INTRODUCTION

Molecular evolution is the process underlying molecular changes in DNA, RNA, and/or amino acid sequences between generations. Molecular evolution can be viewed as an algorithmic process that includes three sequential stages: at the genotype level, a generation is formed with random mutations; then natural selection works at the phenotypic level; while in the last stage, reproduction of newly divergent variants occurs [1]. By accumulating mutations in the genetic sequence, a generation may increase its functionality and gain superiority over other variants.

The study of evolution makes it possible to predict the direction of the variability of living beings, which becomes an essential task concerning epidemically dangerous pathogens. It is a necessary step in the development of effective drugs and vaccines, especially when there is a requirement to predict conserved areas that are not subject to variability [2]. In addition, the evolution of pathogens such as viruses, and viroids is linked to the evolution of other species; it expands our understanding of the evolution of all life on planet Earth [3].

With regard to viruses, study of evolutionary trajectory is of great importance. Basically, changes in the viral population occur due to mutations during the replication process. Viral polymerases are error-prone, resulting in mutations in the viral genome. This can lead to the formation of a complex population of related, but non-identical, genomes called quasi-species [4].

Viruses are one of the simplest models to study evolution. The small sizes of viral genomes and proteins make it possible to take into account not only significant changes in viruses, but also minor amino acid substitutions, which have turned out to be critical for the antigenic evolution of viruses [5]. Sometimes, even a single amino acid substitution can lead to a change in the antigenic cluster, as happened between the two antigenic clusters BE89 and SI87 of the influenza virus [6]. For most viruses, variability occurs in two classical ways: antigenic drift, the gradual accumulation of changes; and antigenic variation (shift) or reassortment, which is sudden and significantly changes the viral genome. The process of evolution causes changes in the antigenic characteristics of the virus, which leads to the virus escaping from the host's immune response, thereby reducing the effectiveness of vaccines [7].

Evolution can be analyzed at different levels. For example, in the case of the influenza virus, there are evolutionary models based on various factors (phylogenetic and population genetics, antigenic relationships, epidemiological data, protein structure data) [8]. The underlying, classic model for representing the evolutionary relationships between different species is construction of a phylogenetic tree. A *phylogenetic tree* is a bioinformatics data structure in which differences and similarities between species (e.g., in genetic space) are shown in a compact form similar to a dendrogram. Indeed, the tree transforms complex evolutionary relationships into a human-readable graphic [9, 10].

Visualization tasks are an integral part of most scientific research, including those in the field of bioinformatics [11]. Generally speaking, the transformation of a set of nucleotide or amino acid sequences into a visualized tree requires two sequential steps: assessment of the phylogenetic relationship between species using phylogenetic analysis (with methods divided into phenetic and cladistic); and visualization of the phylogenetic analysis results. In this case, the visualization aims to: facilitate a better understanding of genetic divergence; monitor or discover new variants; and validate, receive, or understand insights from data [12].

Here, we mentioned three prominent approaches developed for evolution modeling, in which visualization plays a crucial role. In 2011, Ito et al. [13] proposed a model to predict the future direction of evolutionary changes in the influenza virus. Their model is based on the application of the well-known multidimensional scaling (MDS) method [14] on distances of amino acid sequences of previous isolated strains. The 3D visualization they present shows the direction of evolution and the sequence phylogeny simultaneously, which cannot be provided through traditional phylogenetic analysis. Their approach achieves a recall of about 0.70, which indicates the ability of the model to predict amino acid substitution.

Another example of using visualization to represent evolution graphically is antigenic cartography proposed by Smith et al. [6]. The main idea behind this approach is to apply a modified MDS method in antigenic distance to locate antigens and antisera on a map. Antigenic mapping is still one of the traditional tools for demonstrating the antigenic evolution of pathogens with high antigenic variability. Sometimes, visualization is an intermediate step to model evolution. For example, as suggested by [5], [15], some features or variables of antigenic evolution model can be extracted from a phylogenetic tree. Despite all the above approaches, modeling and visualization of evolution are still relevant tasks in bioinformatics.

A phylogenetic tree can also be achieved by employing a model of evolution. In fact, such a model is a tool to estimate the evolutionary distance, a measure of genetic divergence, from observed differences between species. There are various models, most of which are based on the Markov model for sequence evolution. These models vary depending on the type of genetic data (DNA, protein, codon) and parameters that describe the substitution rate. For example, it is known that the rate of transitions, that is, the replacement of a purine base by a purine ($A \leftrightarrow G$), or a pyrimidine by another pyrimidine ($C \leftrightarrow T$), is higher than the rate of transversions (replacement of a purine base by a pyrimidine, and vice versa). The Kimura model, known as K80, assigns an individual rate to each of the transitions and transversions, representing the rates as model parameters [16].

In parallel with the rapid progress in modeling and approaches for inferring the tree's topology, visualization techniques and corresponding packages have also developed. Examples of such packages are TREEVIEW [17], PHYLO_WIN [18], FIGTREE [19], iTOL [20], Phylo.io [21], PhyloExplorer [22], and Treeio [23].

Extending our previous attempts to visualize the phylogenetic tree [10], [24], we focus in this work on representing the new 2D and 3D visualization by incorporating ancestral node information. The main idea of the presented approach is inspired by visualization of Rubik's cube solving algorithms [25]. In Rubik's cube visualization, a solution path is initialized by a random state and ends in the final state (the complete solution of the cube). The solution path is visualized using the one-hot-encoding and t-distributed stochastic neighbor embedding (t-SNE) [26]. Considering a taxon, located in a leaf, as the initial and the tree's root as the final states, we represent a reverse evolutionary path similar to that presented in the Rubik's Cube visualization. The construction of such a path requires the presence of genetic sequences of the tree's internal nodes, which can be obtained using algorithms for reconstructing the ancestral sequences.

Our contribution to this work is creating a new approach for visualization of the phylogenetic tree and implementing this approach in the form of an online

platform called *PhyloTraVis* (Phylogenetic Trajectory Visualization). The platform is publicly available at phylotravis.viroinformatics.com. The proposed approach can be applied in various studies, including modeling of viral antigenic evolution. The rest of this paper is organized as follows. Section 2 describes the methodology and associated algorithms in more detail. Section 3 presents experimental setup and results. Finally, the conclusion is given in Section 4.

METHODOLOGY

PhyloTraVis is a platform developed using FLASK [27], Biopython [28] and Scikit-learn [29]. It consists of two parts: phylogenetic analysis and visualization. In the first stage, an aligned FASTA file, including the genetic sequences of taxa, is transferred into a matrix, which is further embedded in three-dimensional space by the algorithms of the second stage. The overall workflow of our approach is illustrated in Fig. 1.

The input file must be in FASTA format and aligned. Moreover, since constructing a phylogenetic tree requires all genetic sequences to be unique, there must be no duplicate sequences in the input file. Otherwise, the platform returns an exception, informing the user. Phylogenetic analysis is conducted by employing two packages: FastTree [30] and RAxML [31]. Randomized Axelerated Maximum Likelihood (RAxML) is a maximum likelihood phylogenetic analysis program with high accuracy. Stochastic models (such as maximum likelihood) are more practical in biological research, providing many desirable statistical properties, but they often suffer from low computational efficiency. For this reason, we use FastTree, one of the fastest maximum likelihood estimation methods, to create an initial tree. The initial tree is fed into RAxML to generate the final phylogenetic tree. RAxML provides a wide range of options, including the substitution model. RAxML is advantageous in working with large-scale data sets since it is equipped with parallel computation for estimating the best maximum likelihood score.

The final output of phylogenetic analysis depends on the topology of the tree, as well as the genetic sequences of all internal nodes and the root, as the most common ancestor. At the same time, a prerequisite for calculating ancestral sequences in RAxML is the presence of a root. A rooted tree can simply be obtained from an unrooted tree using RAxML employing the '-f I' flag. The tree's root is located on the branch that best balances the subtree lengths for the left and right subtrees.

Ancestral sequence reconstruction is a technique that employs maximum likelihood or Bayesian methods to infer the genetic sequence of ancestral nodes statistically. The computation of ancestral sequences is highly dependent on topology and phylogeny. The reconstruction of ancestral sequences can be achieved by utilizing the '-f A' flag in the RAxML package. As a result, RAxML will generate several files, including the ancestral probabilities and states, which are used in the next step.

At the visualization stage, the genetic information of all nodes and the topology of the tree are combined for its embedding into a new three-dimensional representation. Incorporation of the ancestor nodes' information into the visualization is conducted by defining a phylogenetic trajectory (or path). A phylogenetic trajectory is a non-directional graph path starting at the taxon and ending at the root of the tree (Fig. 2). Such a path represents the evolutionary history of the target derived taxon from the most common ancestor (or root), while the genetic similarity between two nodes determines the length of the path between them. The path is generated from the tree using the module 'phylo' from the Biopython package [28]. The module allows extracting the child-parent relationship between tree nodes, which is necessary to reconstruct the evolutionary trajectory. Note that the total number of paths equals the total number of taxa (the number of species in the input FASTA file).

Although the formation of the path depends on the genetic information of all taxa, at the stage of visualization, the path can be considered as an independent object. The main task of the visualization is to embed the evolutionary path in 3D/2D space. To do this, genetic sequences of path nodes are employed; therefore, the task turns into an embedding procedure to project each node from genetic space into 3D/2D space. Nevertheless, the genetic data must be represented in a numerical format before embedding since most mathematical embedding techniques work with numerical vectors.

There are various methods for representing a genetic sequence in numerical space. Inspired by the visualization of the Rubik's Cube solution [25], we use one-hot-encoding to encode both amino acids and nucleotides. This encoding does not consider the priority between the building blocks of the genetic sequence. Table 1 shows an example of a one-hot-encoding for nucleotides. A similar table can be drawn up for a set of amino acids.

By applying one-hot-encoding, the genetic sequences for all nodes (including internal and terminal) are represented in the form of a binary matrix. Each row of the matrix is a binary sequence associated with a specific tree node, representing the node in a multidimensional binary space. The matrix is fed into the embedding technique to reduce the dimension into 2D or 3D. Note that, unlike our previous approach for visualizing the phylogenetic tree [24], in which each path is individually embedded in 3D space, PhyloTraVis embeds the binary matrix in low-dimensional space by considering all nodes together.

Currently, the platform provides two well-known embedding techniques: MDS and t-SNE. While MDS

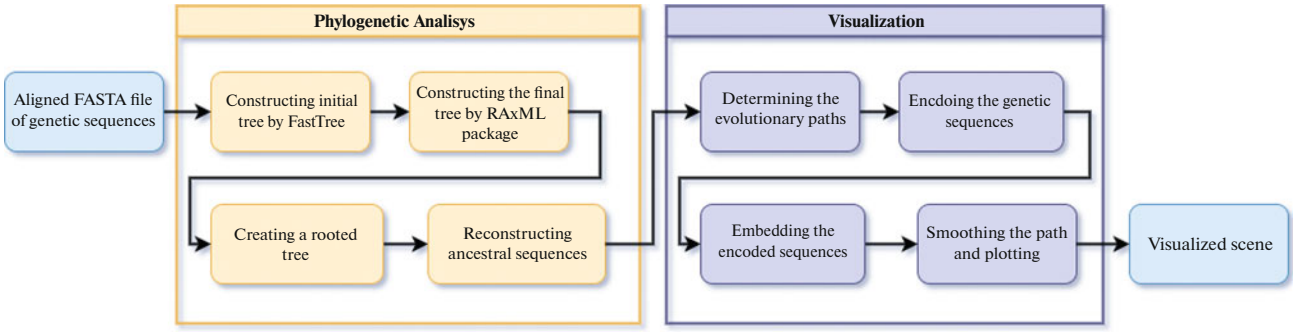

Fig. 1. The figure shows the overall scheme of the proposed approach. It consists of two sequential steps: phylogenetic analysis and visualization. Note that the workflow input is an aligned FASTA file.

aims to preserve distances between objects as much as possible, t-SNE converts the objects into the joint probability and tries to minimize the Kullback-Leibler (KL) divergence between the probabilities in high- and low-dimensional spaces. In other words, given n points $X = \{x_1, x_2, \ldots, x_n\}$, $x_i \in \mathbb{R}^p$, and their affinity (similarity) matrix D, MDS aims to minimize the following objective function

$$\min_Y \sum_{i=1}^{n} \sum_{j=1}^{n} (d_{i,j} - \hat{d}_{i,j})^2$$

where $Y = \{y_1, y_2, \ldots, y_n\}$, $y_i \in \mathbb{R}^q$ are points in low-dimensional space ($q < p$), and $\hat{D}$ is their affinity matrix in the new space.

The t-SNE method uses the distance between objects in high- and low-dimensional spaces to define a conditional probability that determines whether two points belong to the same group or not. If $p_{j|i}$ and $q_{j|i}$ are conditional probabilities for points i, j representing the similarity of data point x_j to data point x_i in high- and low-dimensional spaces, respectively, then the objective function, i.e. KL divergence, can be defined as:

$$KL(P \parallel Q) = \sum_{i \neq j} p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

which expresses the total cost of representing objects in a low-dimensional space. The divergence can be minimized by the gradient descent algorithm.

Table 1. One-hot-encoding for representing an alphabetical sequence of nucleotides in a numerical binary space

Nucleotide	One-hot-code
A	(1, 0, 0, 0)
C	(0, 1, 0, 0)
G	(0, 0, 1, 0)
T	(0, 0, 0, 1)
'-' Gap	(0, 0, 0, 0)

After embedding nodes in 3D/2D space and before final plotting, we apply a smoothing technique called the Bezier curve algorithm to improve the representation of the path in low-dimensional space. A Bezier curve is a parametric curve defined by a set of points called control points. The number of points minus one represents the order of the curve. For example, given two distinct points p_0 and p_1, a linear Bezier curve is defined as follows [33]:

$$B_{p_0,p_1}(t) = p_0 + (p_1 - p_0)t = (1-t)p_0 + tp_1$$

where $0 \leq t \leq 1$.

Generally speaking, a Bezier curve with degree n can be recursively expressed as a linear combination of two Bezier curves of degree $n-1$, as follows:

$$Bp_0(t) = p_0$$

$$Bp_0, \ldots, p_n(t) = (1-t)Bp_0, \ldots, p_{n-1}(t) + tBp_1, \ldots, p_n(t)$$

where $0 \leq t \leq 1$, $Bp_0, \ldots, p_{n-1}(t)$ and $Bp_1, \ldots, p_n(t)$ are Bezier curves of order $n - 1$ for the set of points $p_0, \ldots, p_{n-1}$ and $p_1, \ldots, p_n$, respectively.

After smoothing, all evolutionary paths are transformed into Bezier curves, which are visualized in three-dimensional space. By employing the Bezier curve algorithm, all internal nodes participate in constructing the final path. The nodes that create the path, including the ancestors, serve as control points for the curve. Since the coordinates of these points are obtained by embedding these objects from the multi-dimensional sequence space into 3D paths, the resulting final curves reflect a complete taxa history based on the presented phylogenetic tree.

As a result of trajectory calculations, a set of curves in three-dimensional space is formed. Each curve corresponds to some taxon whose phylogenetic trajectory is represented by the curve. The curves are converted into polylines and written in a CSV text file with columns $(N, X, Y, Z, R, G, B, TEXT)$, as shown in Fig. 3.

This (and only this) data is the input to the program for plotting on the screen. The program works in a web browser and displays the given polylines and text using

3D graphics technologies. A user can interactively change camera direction in the view, enable or disable the display of text labels of taxa, and manage other parameters.

The program is implemented using the *Vrungel* technology [34], developed by one of the authors of this paper. The technology consists of a programming language and its interpreter running in a web browser. The language allows a relatively brief description of a tree of objects that forms: a) a three-dimensional scene; b) a two-dimensional interface; and c) additional calculations. The code of the visualization program is presented in Fig. 4.

Each object is described by a structure like "name: main-feature parameters... features... {nested_objects...}". The object name is optional, and its parameters are like other programming languages. Features are identifiers of the environments that will be added to the created object (an analogy is "mixin" in the Ruby language). The primary feature is the main environment of the object (an analogy is "classes" in other programming languages).

The tree, nodes of which are the created objects, is generated using records of the objects. The parent-child relationship can further be used in various semantics. For example, in the case of the two-dimensional user interface, the relationship is employed to calculate the position of interface elements on the screen.

Constructions, such as `param1=@obj->param2`, mean a link. Any change in the value of `@obj->param2` causes a copy of the value into the current object in the parameter `param1`. By changing parameter values, the object's event handlers are called. Therefore, it can be asserted that the presented code has the features of reactive programming. Constructions, like `object1 | object2 ... | objectN`, mean a pipeline wherein input and output objects are chained together with links (i.e. `object2 input=@object1->input`, and so on). For example, the construction `@dat | linestrips` creates a `linestrips` object that is responsible for generating a 3D polyline representation; its input is the values of `@dat->output`. Definition of object features can be carried out in JavaScript or in the visualization program language. For example, a new feature named `"rescale_rgb"` is defined in `register_feature name="rescale_rgb"` in Fig. 4.

Three-dimensional representation in Vrungel is implemented using the ThreeJS library [35]. The feature `view3d` specifies a two-dimensional output area. Feature `render3d` generates a rendering loop with output into the specified area. The parent-child relationship specifies the list of objects that will be rendered using `render3d`. The `linestrips` and `text3d` objects form the necessary three-dimensional objects: polylines and text, respectively. They

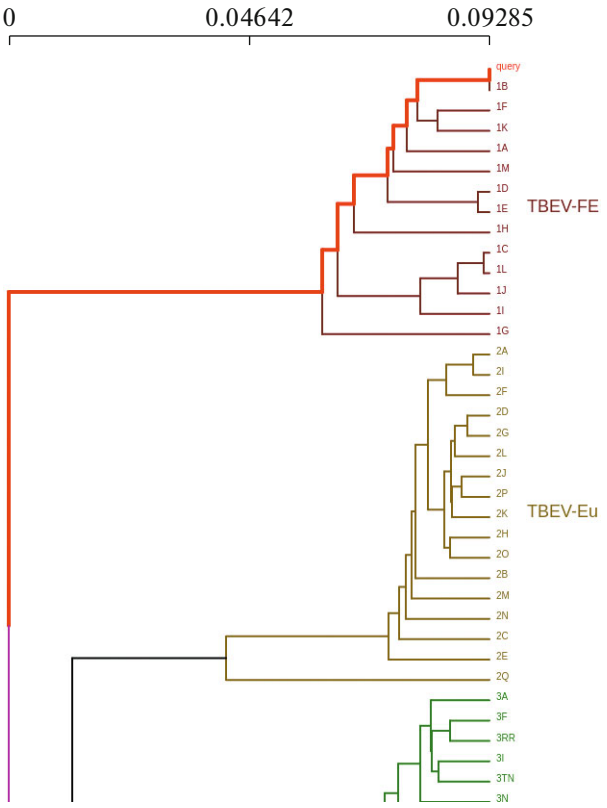

Fig. 2. A phylogenetic tree is usually represented as a dendrogram. The evolutionary trajectory is the path starting from a taxon and ending at the tree's root. An example of such a path is highlighted in red. PhyloTraVis treats each path as a separate visualization object. Here, only a part of the tree, obtained from the TBEV-Analyzer platform [32], a platform for tick-borne encephalitis virus, is presented.

are children of `render3d` and fed into it for rendering.

The written program is passed to the interpreter included in the Vrungel project. It loads the program in the web browser. The process is further determined according to the created object tree. For example, the following steps are carried out for the program presented in Fig. 4:

1. The object `get_query_param` reads the query parameter from the browser's URL string; the parameter name is `csv_file`.

2. The obtained value of the parameter `csv_file` is passed to the object `load-file`.

3. In response to a value change, `load-file` reads the file located at the address specified in the value.

4. The content of the file fed into the object `parse_csv` and then to the object `rescale_rgb`. The result is written to the object `dat` in the field `output`.

5. Data from `dat` goes to `linestrips`, where a `ThreeJS` object is formed to render polylines. Also,

```
N,X,Y,Z,R,G,B,TEXT
1,-0.4047768712043762,-15.214411735534668,-6.602959156036377,58,0,197,HK/1/68
1,-1.2805455923080444,-15.827953338623047,-6.8607497215271,53,0,202,
1,-1.6616933345794678,-16.126724243164062,-6.948936939239502,51,0,204,
1,-1.8214616775512695,-16.22327995300293,-6.913727283477783,50,0,205,
1,2.3749938011169434,12.389556884765625,2.3549633026123047,129,0,126,
1,2.185904026031494,12.525310516357422,2.4929850101470947,129,0,126,
1,2.014712333679199,12.572115898132324,2.7459797859191895,130,0,125,
1,1.8646876811981201,12.506885528564453,3.1399433612823486,132,0,123,
1,1.738439679145813,12.305143356323242,3.6987006664276123,136,0,119,
1,1.6378767490386963,11.942933082580566,4.440480709075928,142,0,113,
1,1.564407229423523,11.399391174316406,5.373449802398682,149,0,106,
1,1.5194145441055298,10.6599702835083,6.490389347076416,159,0,96,
1,1.5048946142196655,9.720165252685547,7.762887001037598,170,0,85,
1,1.5240517854690552,8.590001106262207,9.136513710021973,183,0,72,
2,-0.9473636746406555,-16.751054763793945,-8.015810012817383,42,0,213,HK/107/71
2,-1.0571842193603516,-16.987211227416992,-7.431650638580322,44,0,211,
2,-1.1745434999465942,-16.97091293334961,-7.110812187194824,46,0,209,
2,-1.3054088354110718,-16.821640014648438,-6.894881248474121,48,0,207,
2,-1.4556471109390259,-16.60980224609375,-6.689837455749512,50,0,205,
2,-1.6283318996429443,-16.37251091003418,-6.438336372375488,52,0,203,
2,-1.8214434385299683,-16.12669563293457,-6.109073162078857,55,0,200,
2,-2.027768611907959,-15.878969192504883,-5.691685676574707,59,0,196,
2,-2.2363240718841553,-15.631830215454102,-5.192210674285889,62,0,193,
2,-2.434361696243286,-15.387199401855469,-4.628213405609131,66,0,189,
2,-2.609226703643799,-15.147857666015625,-4.02383279800415,71,0,184,
......
```

Fig. 3. Input data file screenshot. Column *N* identifies polyline. *X*, *Y*, *Z* define coordinates of node in polyline. *R*, *G*, *B* define the node color. Nodes with the same value of *N* describe the same polyline. The name of the taxon is presented in the TEXT column.

`dat` is passed to the filter for empty lines and further to `text3d` forming a `ThreeJS` text rendering object.

6. The generated `ThreeJS` objects are collected by the `render3d` object and rendered 60 frames per second with output to `view3d`.

7. The user can control the camera as well as two additional parameters: checkbox to turn text display on/off and `select_color` to customize its color.

An interesting feature of Vrungel is an experimental debugger that depicts the entities of the loaded program in three-dimensional space (Fig. 5). The debugger uses the molecule metaphor [36], which allows observing the program locally and globally. Information about the Vrungel project and its source codes are publicly available at https://github.com/viewzavr/vrungel.

EXPERIMENTS AND RESULTS

In order to demonstrate the ability of our approach, we visualized a phylogenetic tree for the influenza virus. We conduct two visualizations. In the first case, we show how a phylogenetic tree with a relatively high number of strains can be visualized. The second visualization shows the ability of the proposed method's application in mathematical modeling of phenotype, in this case, antigenic evolution. Each step is described in more detail in the following subsections.

DATA PREPARATION

Two datasets of influenza virus H3N2 subtype were selected for experiments. The reason for choosing this subtype is its high variability compared to other influenza virus subtypes. The first set of strains included 512 hemagglutinin (HA) protein sequences collected during 1968–2007, taken from [37]. Although the file was already aligned, the alignment process can be performed using programs such as the widely used MUltiple Sequence Comparison by LogExpectation (also known as MUSCLE) [38]. MUSCLE has significant speed and accuracy compared with other programs such as Multiple Alignment using Fast Fourier Transform (MAFFT) [39]. However, in the case of a large database, MAFFT offers high speed through its parallel computing.

For the second experiment, a described dataset of strains [6] was used. After extracting strain sequences from public databases, since PhyloTraVis demands an input file without any duplicate sequence, we cleaned the extracted database and obtained 153 sequences. Note that if a sequence has a duplicate in the dataset, we remove both the original strain and the strains with the same sequence. The reason for this filtering is that we need to assign to the sequence its coordinates from the antigenic cartography described [6]. Since duplication of sequences means that there are multiple coordinates for a single sequence, we cannot decide which one should be assigned to it. Therefore, we excluded them from our experiment. The HA sequences in both files were aligned and consist of 329 amino acids. PhyloTraVis requires that the user specifies the type of genetic sequence (nucleotide/amino acid) before uploading a file.

PARAMETER SETUP

After successful uploading of the input FASTA file, the system proceeds to the second stage: parameter setting. All parameters, for both phylogenetic analysis and visualization, can be set at this stage. It is worth

```
/// Loading modules, including definition of used features
load files="lib3dv3 csv params io gui render-params df misc scene-explorer-3d";

/// loading and preparing the data
pq:  get_query_param name="csv_file";
dat: load-file file=@pq->output | parse_csv | rescale_rgb;

/// scene content and rendering
render3d target=@view
{
    @dat | linestrips;

    @dat
      | df_filter code="(line) => line.TEXT?.length > Ø"
      | text3d size=0.2 visible=@cb1->value color=@titlecol->value;
};

/// user interface
screen auto-activate {

  column padding="1em" {
    cb1: checkbox text="Show titles";
    titlecol: select_color value=[1,1,1];
  };

  view: view3d;
};

/// extra functions
/// rescale_rgb - divides data located in R,G,B columns by 255-converting them to 0..1 range
register_feature name="rescale_rgb" {
  df_div column="R" coef=255 | df_div column="G" coef=255 | df_div column="B" coef=255;
};
```

Fig. 4. Simplified version of the visualization program code. The full version is available at github.com/viewzavr/vr-flu-evolution/blob/main/main.cl.

noting that the system can provide a two-stage reconstruction of the phylogenetic tree. In other words, the initial tree generated by FastTree is fed into the first model. The output tree from the first model is called the middle tree, and it serves as the input to the second model for fine-tuning. The final unrooted phylogenetic tree is the output of the second model. The use of the second model is optional and can be activated by the user. The setup parameters for both experiments are presented in Table 2.

VISUALIZATION

After submitting a job, the web page periodically checks its status and reports it to the user. When the job is accomplished, the system generates a CSV file containing the coordinates of the smoothed trajectories. The CSV file is passed to Vrungel to plot the constructed evolutionary trajectories in three-dimensional space. Representative visualizations, with the aforementioned setup for both datasets, are shown in Figs. 6 and 7. They are also available online (https://github.com/viewzavr/vr-flu-evolution).

RESULTS

Unlike our recently developed approach [24], in which each path is individually embedded in the low-dimensional space, PhyloTraVis considers all the nodes of the phylogenetic tree and embeds them together in the low-dimensional space. The resulting path coordinates are affected in three levels. The tree topology is defined through RAxML using a substitution model at the first level. Accordingly, the hierarchy of the ancestor nodes determines the order of control points for the Bezier curve after embedding in low-dimensional space. The second level of path modification depends on the embedding technique, which maps objects from high-dimensional to low-dimensional space. The final level of path modification occurs during the smoothing process when the control points are involved in constructing the Bezier curve. Generally speaking, the coordinate of each control point depends, in one way or another, on all nodes of the tree. Therefore, any change in the tree's topology affects the final visualization.

Since the visualization is based on genetic sequence, the proposed approach tends to preserve the genetic similarity between the strains. As expected, genetically-close strains are also close in 3D space,

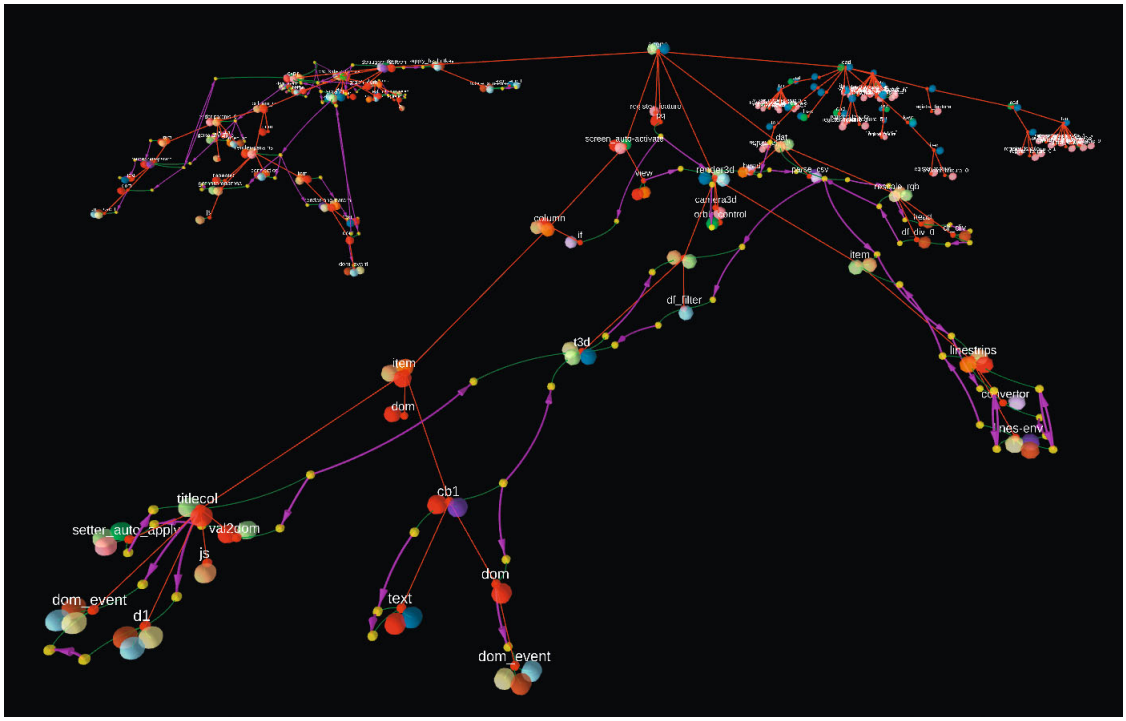

Fig. 5. A presentation of the tree of objects generated by the visualization program. Small red spheres are objects. Yellow spheres are object parameters, and large red spheres are features ('mixin') of objects. Links colored in red indicate the parent-child relationship, while green links show object-parameter. Purple arrows indicate passing of parameter values by link.

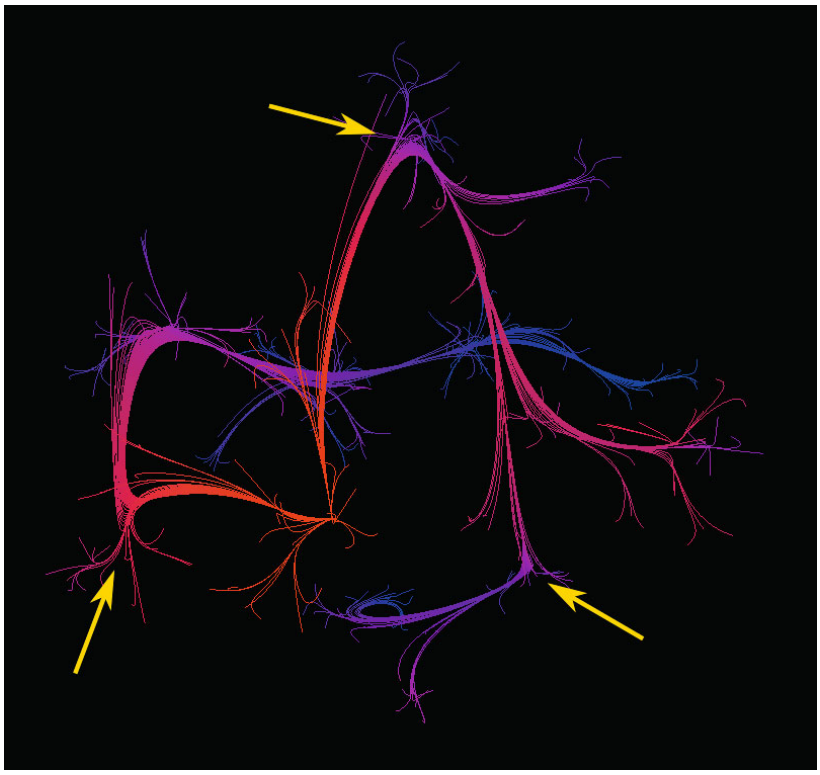

Fig. 6. Visualization of evolution for a sample of 512 strains of influenza virus H3N2 subtype isolated during 1986–2007. For clarity, strain labels are not displayed. Color indicates the Euclidean distance to the root. The shortest distance to the root is shown in red, while blue indicates the longest distance. Yellow arrows show significant changes in genetic content during evolution.

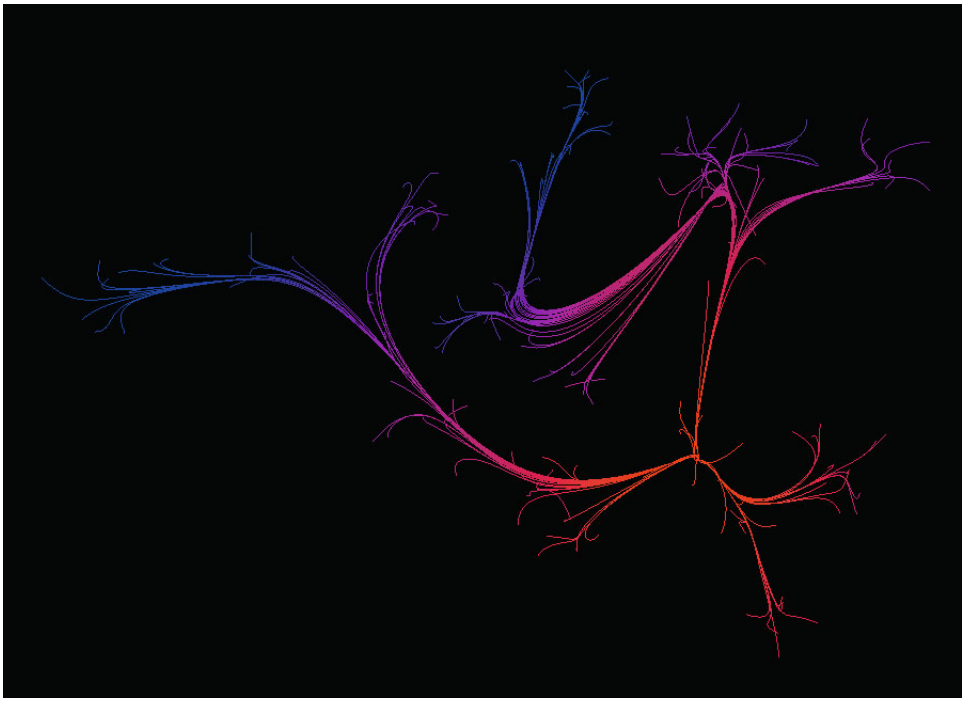

Fig. 7. Visualization of a small sample (153) of influenza H3N2 strains collected during 1968–2004 [6]. The Euclidean distance between strain coordinates in the embedded space has a moderate correlation of 0.61 (p-value=1e-05) with the Euclidean distance between their positions on the antigenic map.

while genetically-far strains have a correspondingly more pronounced distance in space. One of the distinguishing features of our approach is its customization. As mentioned, coordinates are influenced by encoding and embedding techniques. Thus, a user can customize the visualization through these two processes. Encoding determines how genetic information is reflected in numerical space, while embedding determines the similarity between objects in low-dimensional space. In our experiment, there is no priority between amino acids, but it can be taken into account by encoding them using AAindex [40]. Moreover, the concept of similarity can be defined based on the embedding technique. This provides the user with a wide range of choices to explore taxonomic relationships. Our experiments show that t-SNE provides better visualization quality than MDS by expressing smoother paths.

In order to demonstrate the effectiveness of the proposed approach, we compared the distance between objects in three-dimensional space with their genetic distance, as well as with the distance on the described antigenic map [6]. We first conducted an experiment with 153 HA proteins of the H3N2 influenza virus. By employing PhyloTraVis, we achieved new coordinates of each strain in 3D space. We further calculated the pairwise Euclidean distance between strains and created a distance matrix. Similarly, the Hamming distance was calculated for each pair of strains. In order to compare the two distance matrices, we use the Mantel test [41]. The Mantel test is a non-parametric test that assesses the significance of the correlation between two distance matrices by permutation of rows and columns. The test yields a correlation coefficient that is between -1 and 1. A coefficient close to 1 indicates a strong positive correlation, while close to -1 indicates a strong negative correlation. The coefficient about zero exhibits the absence of correlation between the matrices.

Table 3 illustrates the results of the Mantel test. Interestingly, the results show that Euclidean distance matrices of strains in embedded 3D and antigenic cartography spaces express a moderate correlation of 0.61 ($p = 1e\text{-}05$). This highlights that the visualization can be used in modeling of antigenic evolution. Moreover, Table 3 shows that there is a moderate correlation between the Hamming distance matrix and the Euclidean distance matrix of embedded space. This indicates that most genetic variation is reflected in the embedded space. Based on the results of the presented visualization, it can be concluded that this approach can serve as an auxiliary tool for phylogenetic analysis. Moreover, the results can, in turn, be employed in phenotype modeling.

CONCLUSIONS

Generally, the experiments show that our approach, and the PhyloTraVis platform, do not replace the classical representation of the phylogenetic

Table 2. Setup parameters for visualization of the influenza virus hemagglutinin (HA) protein in our experiments. Here, we employ a two-stage phylogenetic tree reconstruction

Process	Parameters
First RAxML model	-m PROTCATGTR -p 12345 -e 0.01
Second RAxML model	-m PROTGAMMAGTR -p 12345 -e 0.01
Rooting	-f I -m PROTCATGTR -p 12345 -e 0.01
Ancestral reconstruction	-f A -m PROTCATGTR -p 12345 -e 0.01
Encoding	binary (currently, one-hot-encoding is available)
Embedding	t-SNE perplexity=30.0 early_exaggeration=12.0 learning_rate =200.0 n_iter=1000 random_state=1234

Table 3. Mantel test results for correlations between the Euclidean distance matrix (for strains in 3D space) and: the Hamming distance matrix; and the matrix of Euclidean distances (between strains) on the antigenic map. The Mantel test was calculated using the Pearson method, 100,000 permutations, and a two-sided test

Distance matrix	Hamming distance matrix	Matrix of Euclidean distances between strains on the antigenic map
Matrix of Euclidean distances between strains in our 3D space	0.63 (p-value=1e-05)	0.61 (p-value=1e-05)

tree. Rather, they serve as additional research and analytical tools for studying and modeling evolution, including viral evolution. Thanks to such 3D visualization, one can recognize abrupt changes, i.e., significant sequence changes (protein or nucleic acid), in the direction of evolution at specific points in phylogeny.

Understanding pivotal events in the evolution of infectious agents is essential for predicting mutations, or even preventing the formation of circumstances that accelerate their variability. By conducting a retrospective analysis of the environmental conditions at these points in which such changes occurred, it is possible to identify environmental pressure factors that could become critically significant for macrovariability. In addition, understanding patterns in pathogen evolution helps identify particularly variable or conserved (protein and/or genome) regions, which is necessary for developing drugs and vaccines.

From a technical perspective, the only significant drawback of our approach is the computational complexity of building a phylogenetic tree and embedding algorithms for a large number of taxa. Although only the amino acid sequence of the hemagglutinin (HA) protein was employed to visualize influenza viral evolution, however, it is a partial reflection of evolution. A deeper insight into its evolution could be obtained using analysis of full genomes available in public databases. In addition, this approach is not limited to virological research. It can be applied to analyze the evolution of other species of living beings.

Moreover, amino acids can be considered in terms of different biological or physicochemical properties if they are encoded using the well-known AAindex database [40]. This provides an opportunity to visualize and study evolution from different points of view, such as hydrophilicity or lipophilicity. Besides AAindex, it is also possible to apply simplified amino acid alphabets to our visualization. Future work needs to be done on encoding methods, and associated criteria (metrics, thresholds, etc.), to automatically recognize taxonomic variants and clusters from the provided visualization.

ACKNOWLEDGMENTS

The reported study was funded by Russian Foundation for Basic Research (RFBR), project number 19-31-60025.

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Orengo, C., Jones, D., and Thornton, J., *Bioinformatics: Genes, Proteins and Computers,* Taylor & Francis, 2003.

2. Xu, X., Zhang, Q.Y., Chu, X.Y., Quan, Y., Lv, B.M., and Zhang, H.Y., Facilitating antiviral drug discovery using genetic and evolutionary knowledge, *Viruses,* 2021, vol. 13, no. 11, p. 2117.
3. Moelling, K. and Broecker, F., Viruses and evolution – viruses first? A personal perspective, *Front. Microbiol.,* 2019, vol. 10, p. 523.
4. Novella, I.S., Presloid, J.B., and Taylor, R.T., RNA replication errors and the evolution of virus pathogenicity and virulence, *Curr. Opin. Virol.,* 2014, vol. 9, pp. 143–147.
5. Harvey, W.T., et al., Identification of low-and highimpact hemagglutinin amino acid substitutions that drive antigenic drift of influenza A (H1N1) viruses, *PLoS Pathog.,* 2016, vol. 12, no. 4, p. e1005526.
6. Smith, D.J., Lapedes, A.S., De Jong, J.C., Bestebroer, T.M., Rimmelzwaan, G.F., Osterhaus, A.D., and Fouchier, R.A., Mapping the antigenic and genetic evolution of influenza virus, *Science,* 2004, vol. 305, no. 5682, pp. 371–376.
7. Forghani, M. and Khachay, M., Convolutional neural network based approach to in silico non-anticipating prediction of antigenic distance for influenza virus, *Viruses,* 2020, vol. 12, no. 9, p. 1019.
8. Klingen, T.R., Reimering, S., Guzm'an, C.A., and McHardy, A.C., In silico vaccine strain prediction for human influenza viruses, *Trends Microbiol.,* 2018, vol. 26, no. 2, pp. 119–131.
9. Jordan, G.E. and Piel, W.H., Web-based visualizations for the tree of life, *Bioinformatics,* 2008, vol. 24, no. 14, pp. 1641–1642.
10. Forghani, M., Vasev, P., and Averbukh, V., Threedimensional visualization for phylogenetic tree, *Sci. Visualization,* 2017, vol. 9, no. 4, pp. 59–66. http://sv-journal.org/2017-4/06/.
11. Averbukh, V.L., Semiotics and foundations of the theory of computer visualization, *Online Sci. J. Philos. Probl. IT Cyberspace,* 2013, no. 1, pp. 26–41. http://www.cv.imm.uran.ru/e/3241413.
12. Wang, C., Feng, Y., Bodik, R., Cheung, A., and Dillig, I., Visualization by example, *Proc. ACM Program. Lang.,* 2019, vol. 4, no. POPL, pp. 1–28.
13. Ito, K., Igarashi, M., Miyazaki, Y., Murakami, T., Iida, S., Kida, H., and Takada, A., Gnarledtrunk evolutionary model of influenza A virus hemagglutinin, *PloS One,* 2011, vol. 6, no. 10, p. e25953.
14. Cox, M.A. and Cox, T.F., *Multidimensional Scaling, Handbook of Data Visualization,* Berlin, Heidelberg: Springer, 2008, pp. 315–347.
15. Neher, R.A., Bedford, T., Daniels, R.S., Russell, C.A., and Shraiman, B.I., Prediction, dynamics, and visualization of antigenic phenotypes of seasonal influenza viruses, *Proc. Nat. Acad. Sci.,* 2016, vol. 113, no. 12, pp. E1701–E1709.
16. Kimura, M.A.,Simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences, *J. Mol. Evol.,* 1980, vol. 16, no. 2, pp. 111–120.
17. Page, R.D., Tree View: an application to display phylogenetic trees on personal computers, *Bioinformatics,* 1996, vol. 12, no. 4, pp. 357–358.
18. Galtier, N., Gouy, M., and Gautier, C., SEAVIEW and PHYLO_WIN: two graphic tools for sequence alignment and molecular phylogeny, *Bioinformatics,* 1996, vol. 12, no. 6, pp. 543–548.
19. FigTree. http://tree.bio.ed.ac.uk/software/figtree/.
20. Letunic, I. and Bork, P., Interactive Tree Of Life (iTOL): an online tool for phylogenetic tree display and annotation, *Bioinformatics,* 2007, vol. 23, no. 1, pp. 127–128.
21. Robinson, O., Dylus, D., and Dessimoz, C., Phylo.io: interactive viewing and comparison of large phylogenetic trees on the web, *Mol. Biol. Evol.,* 2016, vol. 33, no. 8, pp. 2163–2166.
22. Ranwez, V., Clairon, N., Delsuc, F., Pourali, S., Auberval, N., Diser, S., and Berry, V., PhyloExplorer: a web server to validate, explore and query phylogenetic trees, *BMC Evol. Biol.,* 2009, vol. 9, no. 1, pp. 1–13.
23. Wang, L.G., et al., Treeio: an R package for phylogenetic tree input and output with richly annotated and associated data, *Mol. Biol. Evol.,* 2020, vol. 37, no. 2, pp. 599–603.
24. Forghani, M., Vasev, P., Ramsay, E., and Bersenev, A., Visualization of the evolutionary path: an influenza case study, *CEUR Workshop Proc. – CEUR-WS,* 2021, vol. 3027, pp. 358–368.
25. Steinparz, C.A., Hinterreiter, A.P., Stitz, H., and Streit, M., Visualization of Rubik's cube solution algorithms, *Proc. EuroVis Workshop on Visual Analytics,* Porto, 2019, pp. 19–23.
26. Van der Maaten, L. and Hinton, G., Visualizing data using t-SNE, *J. Mach. Learn. Res.,* 2008, vol. 9, no. 11.
27. Grinberg, M., *Flask Web Development: Developing Web Applications with Python,* O'Reilly Media, 2018.
28. Cock, P.J., et al., Biopython: freely available Python tools for computational molecular biology and bioinformatics, *Bioinformatics,* 2009, vol. 25, no. 11, pp. 1422–1423.
29. Pedregosa, F., et al., Scikit-learn: machine learning in Python, *J. Mach. Learn. Res.,* 2011, vol. 12, pp. 2825–2830.
30. Price, M.N., Dehal, P.S., and Arkin, A.P., FastTree 2 – approximately maximum-likelihood trees for large alignments, *PloS One,* 2010, vol. 5, no. 3, p. e9490.
31. Stamatakis, A., RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies, *Bioinformatics,* 2014, vol. 30, no. 9, pp. 1312–1313.
32. Forghani, M., Kovalev, S., Bolkov, M., Khachay, M., and Vasev, P., TBEV analyzer platform for evolutionary analysis and monitoring tick-borne encephalitis virus: 2020 update, *Biostat. Epidemiol.,* 2021, pp. 1–17.
33. Baydas, S. and Karakas, B., Defining a curve as a Bezier curve, *J. Taibah Univ. Sci.,* 2019, vol. 13, no. 1, pp. 522–528.
34. Vasev, P., Vrungel. https://github.com/viewzavr/vrungel.
35. Dirksen, J., *Learning Three. js: the JavaScript 3D library for WebGL,* Packt Publ. Ltd., 2013.
36. Averbukh, V.L., Baidalin, A.Yu., Ismagilov, D.R., Kazantsev, A.Yu., and Timoshpolsky, S.P., Using 3D metaphors of visualization, *Proc. 14th Int. Conf. on Computer Graphics and Vision GraphiCon,* Moscow,

Sept. 6–10, 2004, pp. 295–298. http://www.cv.imm.uran.ru/e/3549.

37. Wang, P., Zhu, W., Liao, B., Cai, L., Peng, L., and Yang, J., Predicting influenza antigenicity by matrix completion with antigen and antiserum similarity, *Front. Microbiol.,* 2018, vol. 9, p. 2500.
38. Edgar, R.C., MUSCLE: multiple sequence alignment with high accuracy and high throughput, *Nucl. Acids Res.,* 2004, vol. 32, no. 5, pp. 1792–1797.
39. Rozewicki, J., Li, S., Amada, K.M., Standley, D.M., and Katoh, K., MAFFT-DASH: integrated protein sequence and structural alignment, *Nucl. Acids Res.,* 2019, vol. 47, issue W1, pp. W5–W10. https://doi.org/10.1093/nar/gkz342
40. Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., and Kanehisa, M., AAindex: amino acid index database, progress report 2008, *Nucl. Acids Res.,* 2007, vol. 36, suppl. 1, pp. D202–D205. https://www.genome.jp/aaindex/.
41. Mantel, N., The detection of disease clustering and a generalized regression approach, *Cancer Res.,* 1967, vol. 27, no. 2, part 1, pp. 209–220.