Molecular Phylogenetics and Evolution 200 (2024) 108181

Contents lists available at ScienceDirect

Molecular Phylogenetics and Evolution

journal homepage: www.elsevier.com/locate/ympev

Machine learning can be as good as maximum likelihood when reconstructing phylogenetic trees and determining the best evolutionary model on four taxon alignments

Nikita Kulikov [a,b,*], Fatemeh Derakhshandeh [b,c], Christoph Mayer [b]

[a] Molecular Evolutionary Biology, Department of Biology, Hamburg University, Germany
[b] Leibniz Institute for the Analysis of Biodiversity Change (LIB), Germany
[c] Medical Faculty, Heidelberg University, Germany

ARTICLE INFO

Keywords:
Phylogeny
Machine learning
Neural network
Phylogenetic tree reconstruction
Substitution model selection

ABSTRACT

Phylogenetic tree reconstruction with molecular data is important in many fields of life science research. The gold standard in this discipline is the phylogenetic tree reconstruction based on the Maximum Likelihood method. In this study, we present neural networks to predict the best model of sequence evolution and the correct topology for four sequence alignments of nucleotide or amino acid sequence data. We trained neural networks with different architectures using simulated alignments for a wide range of evolutionary models, model parameters and branch lengths. By comparing the accuracy of model and topology prediction of the trained neural networks with Maximum Likelihood and Neighbour Joining methods, we show that for quartet trees, the neural network classifier outperforms the Neighbour Joining method and is in most cases as good as the Maximum Likelihood method to infer the best model of sequence evolution and the best tree topology. These results are consistent for nucleotide and amino acid sequence data. We also show that our method is superior for model selection than previously published methods based on convolutionary networks. Furthermore, we found that neural network classifiers are much faster than the IQ-TREE implementation of the Maximum Likelihood method. Our results show that neural networks could become a true competitor for the Maximum Likelihood method in phylogenetic reconstructions.

1. Introduction

In parallel with the development of sequencing technologies, many algorithms have been developed to reconstruct phylogenetic trees from molecular sequence data (Yang, 2014). Tree reconstruction methods fall into two main categories: distance-based methods such as UPGMA (Sneath & Sokal, 1973), Neighbour Joining (NJ) (Saitou & Nei, 1987) or minimum evolution (ME) (Edwards & Cavalli-Sforza, 1964), and character-based methods like Maximum Parsimony (MP) (Farris, 1970; Fitch, 1971), Maximum Likelihood (ML) (Cavalli-Sforza & Edwards, 1967; Felsenstein, 1981) and Bayesian inference methods (Ronquist & Huelsenbeck, 2003). These methods differ not only in methodology and underlying assumptions but also in crucial aspects such as statistical consistency, efficiency, and robustness (Penny et al., 1992). The capacity of a statistical method to accurately infer the correct phylogenetic tree, when the amount of data approaches infinity, is known as statistical consistency. A method is called efficient if the amount of data necessary to find the correct solution with a high probability is small and it is called robust if it finds the correct solution even if some of its underlying assumptions are slightly violated.

Today the most widely used method in phylogenetic reconstructions is the ML method. This method is under realistic conditions superior to distance-based and the MP method (Xia, 2018). For a given phylogenetic hypothesis=(tree + model + branch lengths + model parameters) and for given input sequence data, the likelihood of the hypothesis is the probability of observing the sequence data by chance under that hypothesis (Felsenstein, 1981).

The tree and model that have the highest likelihood of producing the data set should be preferred and the aim of the ML methods is to find this model and tree. The need to assume models of sequence evolution is occasionally criticised (Abadi et al., 2019), although they are inherently needed for all probability-based statistical methods. The ML approach

* Corresponding author.
E-mail address: n.kulikov@leibniz-lib.de (N. Kulikov).

https://doi.org/10.1016/j.ympev.2024.108181
Received 30 September 2023; Received in revised form 15 July 2024; Accepted 26 August 2024
Available online 30 August 2024

has been proven to be statistically consistent (Graur & Li, 1997; Truszkowski & Goldman, 2016) under the premise that the evolutionary process (i.e. the tree and model of sequence evolution) that created the data are among the evolutionary processes considered in the phylogenetic tree reconstruction. Beyond consistency, it has been shown that the ML method is efficient as well as robust for a range of possible model violations (Yang, 2014).

Models of sequence evolution specify the probabilities that nucleotides or amino acids are substituted by other residues in given intervals of time. A wide range of mostly time reversible models of sequence evolution are used today (Felsenstein, 2004; Yang, 2014). For nucleotides, time reversible models range from the simplest possible model, the Jukes Cantor (JC) model (Jukes & Cantor, 1969) which assumes equal base frequencies and substitution rates for all nucleotides, to the most general time reversible (GTR) model (Tavaré, 1986). For amino acids, usually models are used with empirical amino acid frequencies and substitution rates since estimating all parameters of the most general time reversible model with 20 residues is very time consuming (Whelan & Goldman, 2001; Keane et al., 2006). During a tree reconstruction, the model and its parameters are typically estimated along with the tree. Alternatively, if they are at least partially known, they can be specified in advance.

There is theoretical evidence that no method can outcompete the ML method when the amount of available data approaches infinity. This follows from the Cramér-Rao lower bound theorem (Rao, 1945; Cramér, 1946) which provides an asymptotic lower bound for the achievable variance of consistent and unbiased estimators as the amount of data approaches infinity, together with the observation that the consistent and unbiased maximum likelihood estimator takes on this lower bound under relatively mild conditions asymptotically (Stuart et al., 1999, Chapter 18; Yang, 2014). In particular this means that no other estimator exists that is more efficient, i.e. requires less data, than the maximum likelihood estimator used in phylogenetic reconstructions. Altogether our goal cannot be to find a method that has statistical properties that are better than those of the ML method, but a method that is equally good and computationally more efficient.

The traditional tree reconstruction methods have different advantages and disadvantages. Distance based methods normally require fewer computational resources but are less efficient and therefore less accurate than the ML method under realistic conditions (Huelsenbeck, 1995), while the ML method is computationally intensive but efficient and statistically consistent.

In recent years, machine learning has become an important method not only in many fields of data science in general but also in biology (Borowiec et al., 2022). Machine learning refers to a wide range of methods that can be used mainly to classify data or solve regression problems. Apart from Neural Networks (NNs), this includes popular methods such as decision trees, Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting (GB) and k-means clustering (Mahesh, 2020). As machine learning methods are particularly good at solving classification problems, they should be candidates to classify/predict models of sequence evolution and topologies for given alignments of molecular sequence data. Indeed, machine learning has been proposed for phylogenetic tree reconstruction as early as in 2012 (Halgaswaththa et al., 2012). Several studies concentrated on inferring the correct topology of quartet trees by analysing the image created from aligned sequences, i.e. after representing the nucleotides with different colours and applying image recognition techniques to these images using Convolutional Neural Networks (CNNs, LeCun et al., 1989) with a large number of convolutional layers (Suvorov et al., 2020; Zou et al., 2020; Suvorov & Schrider, 2024). Image classification of alignments using deep CNNs has also been suggested for determining the best evolutionary model of sequence evolution (Burgstaller-Muehlbacher et al., 2023). Furthermore, Pinheiro et al. (2022) used a RF algorithm for predicting missing sequence regions and using this information to reconstruct phylogenetic trees. However, the accuracy of the RF method could hardly come close to that of the Neighbour Joining method.

Machine learning has also been proposed for phylogenetic tree reconstruction in combination with alignment free methods. Zhu & Cai (2021) used k-mer frequencies to represent biological sequences and used them to train a reinforcement learning model. Unfortunately, they compared their results with distance-based methods such as the UPGMA method rather than the ML method. Similarly, Gamage et al. (2020) used k-mer frequencies as a proxy for phylogenetic distances in combination with RF algorithm to infer the tree topology. The results were compared to the NJ method.

Several machine learning algorithms proposed so far are inferior to the ML method and only as good as distance-based methods (Zhu & Cai, 2021; Pinheiro et al., 2022). Suvorov et al. (2020) and Zou et al. (2020) showed that image-based alignment classifications with CNNs can yield results similar to those of the maximum likelihood method. Burgstaller-Muehlbacher et al., 2023 have shown that CNNs analysing images of nucleotide alignments can compete with the ML method when it comes to model prediction and computing times.

In this study we propose to use site pattern frequencies of four taxon alignments as input to neural networks to predict the best model of sequence evolution and the most likely tree topology that created the sequences (see Fig. 1). This novel approach can be applied to nucleotide and amino acid data sets.

Site pattern frequencies contain the full information of an alignment, if alignment sites are treated as independent and identically distributed as assumed in the ML method for all commonly used models of sequence evolution. Instead of computing the likelihood function for the set of alignment sites, we trained the NNs such that they can predict the best model and topology from the site pattern frequency distributions. If sufficiently complex NNs are trained with data sets that sample the full variety of possible data with sufficient density, there is no reason this method could not achieve the accuracy of the ML method. We expect higher reconstruction accuracies when topology prediction networks were specifically trained for the different nucleotide and amino acid evolutionary models. Therefore, we trained topology prediction networks separately for a set of evolutionary models of sequence evolution. Analogously to the ML method, this implies that the best model has to be estimated prior to the final topology inference. An interesting property of NNs based on dense layers is that after the pattern frequencies have been determined, a prediction requires constant computational resources. Considering how complex it is to evaluate the likelihood function even for a small number of taxa, the number of operations needed to determine the response of the neural network for a single input should be much smaller than the number of operations needed to compute the likelihood function. Therefore NNs should be computationally more efficient than the ML method for model and topology predictions using site pattern frequencies.

In parallel and independent of us, Suvorov & Schrider (2024) have developed NNs that use the site pattern frequency distribution of four nucleotide sequences for training NNs to predict branch lengths. They used images of four taxon alignments for topology classifications and site pattern frequencies of four taxon alignments for branch length estimates and found that these estimates are about as good as those of the ML method. Leuchtenberger et al. (2020) proposed site pattern frequencies in combination with NNs to select the best tree reconstruction method for the Farris and Felsenstein Zone.

A limitation of using machine learning for phylogenetic reconstruction, whether we analyse site pattern frequencies or images of alignments, is that each possible classification has to be trained with sufficient training data, i.e. for all topologies we need to consider all combinations of branch lengths, models and model parameters. Therefore, we restrict our analyses to the four taxon case in which only three topologies are possible (Huelsenbeck & Hillis, 1993).

We describe a set of six NN architectures and how they can be trained to predict the best model of sequence evolution and the best tree topology for four taxon alignments of nucleotide or amino acid sequences.

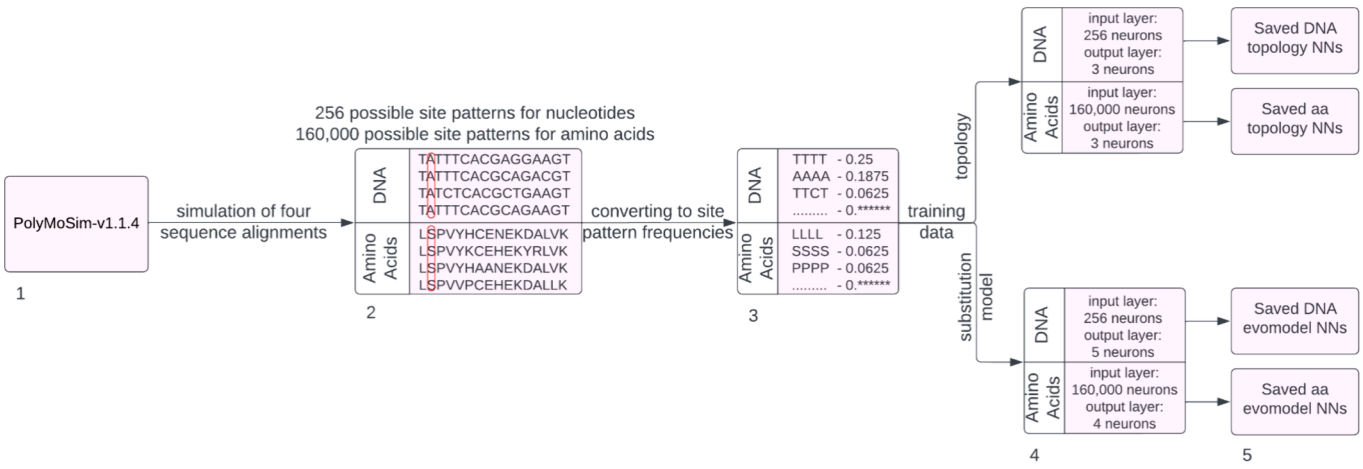

Fig. 1. Schematic flowchart for simulating data sets and training NNs. (1) Alignment simulation. (2) Storing correct classification and site pattern frequencies. (3–4) Training of NNs with known classification and site pattern frequencies data. (5) Saving trained NNs. Note that site patterns are defined as the ordered strings of all nucleotide or amino acid residues at given alignment positions. Site pattern frequencies are the relative frequencies with which site pattern strings occur in the alignment. In the shown nucleotide alignment the site pattern "TTCT" occurs at alignment position 4 and has a frequency of 1/16.

Since we use site pattern frequencies to classify models and topologies, the input layers of the NNs have a number of neurons equal to the number of different site patterns in four taxon alignments, i.e. $256 = 4^4$ for nucleotides and $160{,}000 = 20^4$ for amino acids (Table 1). The NNs are trained by presenting them a large number of pattern frequency vectors together with the correct classification, i.e. the correct model of sequence evolution or the correct topology under which the four taxon alignment was created. Finally, the number of neurons in the output layer has to be equal to the number of different classes in the prediction problem. Specifically, the output values of these neurons are equal to the probabilities the NN assigns to the different possible outcomes. For the training we use simulated alignments that evolved under known hypotheses, since only for these the true topology and model are certainly known and only for these a sufficient number of training data sets is available that sample the full variety of each prediction class of our classification problem.

As one of the main results we present a software package DeepNNPhylogeny (https://github.com/cmayer/DeepNNPhylogeny) consisting of training and prediction scripts as well as the trained neural networks to predict the best substitution model and tree topology for four sequence alignments of nucleotide and amino acid sequences. We for the first time propose site pattern frequencies in combination with neural networks for model and topology predictions and we tested their performance on nucleotide and amino acid data sets. Furthermore, we tested non-neural network classifiers such as nearest neighbours, SVM or GB for predicting the best tree topology using site pattern frequencies. These classifiers are commonly used to classify frequency distributions (see e.g. Hastie et al., 2009) and could be a potential alternative to NN classifiers for our prediction tasks.

Finally, we compared the prediction success and prediction times of the NNs we proposed with the success and prediction times of the ML + BIC (Bayesian Information Criterion; Schwarz, 1978) method for predicting the best model of sequence evolution as well as that of the ML and Neighbour Joining methods for predicting the best tree topology. We also compare the model prediction success and run times with those of the recently published ModelRevelator software (Burgstaller-Muehlbacher et al., 2023). Prediction success was compared using simulated data as well as an empirical data set.

2. Methods

In this study, we trained NNs to predict the model of sequence evolution and the unrooted tree topology that most likely created a given four-taxon alignment of nucleotide or amino acid sequences.

2.1. Neural network architectures and implementation

The training and prediction scripts and the NNs were implemented in Python using the TensorFlow library (Abadi et al., 2016). Workflows for training the NNs and for using trained NNs for predictions are illustrated in Figs. 1-2. Distributions, such as site pattern frequencies, are best classified with a series of fully connected layers, called dense layers in TensorFlow (Khoussi et al., 2021). We designed six different NN architectures, which all consist of dense, normalisation and dropout layers, but with different numbers of branches leading from the input to the output layer and one network with interconnections. The NNs we

Table 1
Sizes of input and output layers and information on the size and characteristics of the NN training data sets.

	Nucleotide model prediction	Nucleotide topology prediction	Amino acid model prediction	Amino acid topology prediction
Neurons in input layer	256	256	160,000	160,000
Neurons in output layer, i.e. number of classes in prediction	5	3	4	3
Length of training alignments	30,000 bp	100,000 bp	1,000,000 aa	1,000,000 aa
Alignments with normal internal branch length	450,000	300,000	150,000	30,000
Alignments with short internal branch length	450,000	300,000	150,000	30,000
Models included	JC, K80, F81, HKY, GTR	JC, K80, F81, HKY, GTR	JTT, LG, WAG, Dayhoff	JTT, LG, WAG, Dayhoff
Size of training data set array	(900,000; 256)	(600,000; 256)	(300,000; 160,000)	(60,000; 160,000)
Number of NN architectures	6	6	6	6
Total number of trained NNs	6 architectures	6 architectures * 5 models	6 architectures	6 architectures * 4 models

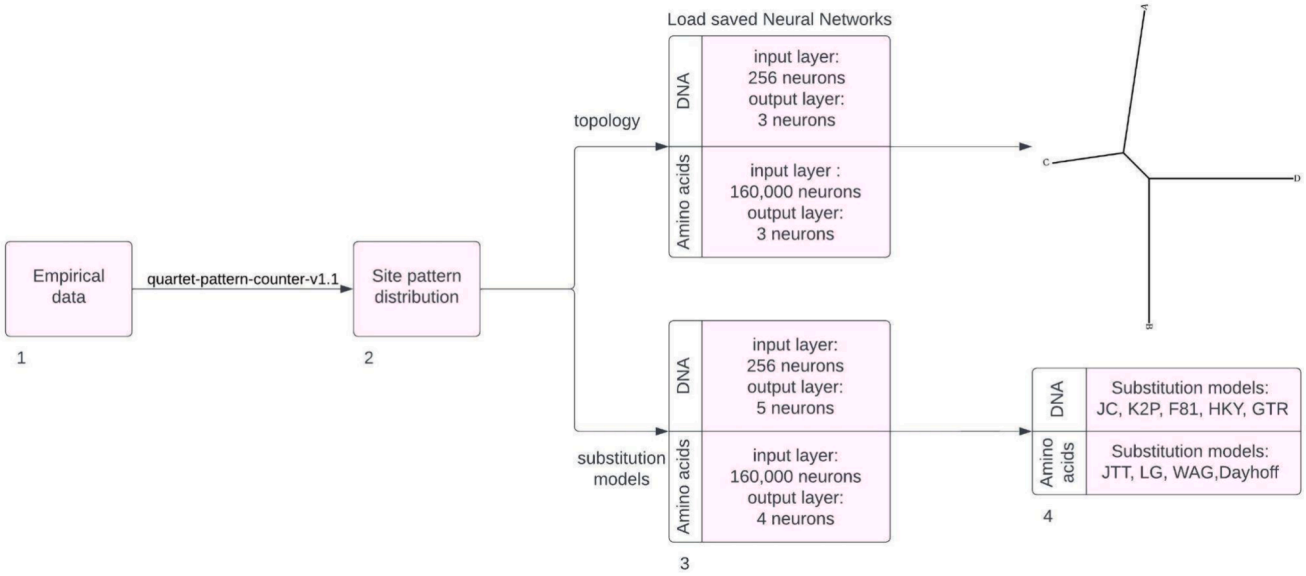

Fig. 2. Schematic flowchart for model and topology predictions of empirical data sets using trained and saved NNs. (1) Loading alignment which shall be classified. (2) Determining pattern frequencies with Quartet-pattern-counter-v1.1 program. (3) Loading trained NNs. (4) Classifying alignment according to classification task. Model selection with subsequent topology classification using the selected model is also possible.

implemented are described in detail in the supplementary materials Section 1 and in Figs. S1-S2. We refer to the six NN architectures as B1, B2, B3, B10, U, CU, where the n in Bn, stands for the number of independent branches in the network that lead from the input to the output layer, U stands for a U-shaped network and CU for a U-shaped network with internal connections. The U and CU models were introduced with the intention to test networks with a large number of free parameters in a non-trivial architecture in the hope that they are able to store more information about our complex frequency distributions. With its interconnections, the CU model shares some distant similarity with the U-Net proposed by Ronneberger et al., (2015) for image segmentation. However descending and ascending branches serve very different tasks here and it was merely introduced since we found that this architecture trains very fast and efficiently compared to the complexity and the large number of free parameters it has. The architecture is explained in more detail in the supplementary materials, sections 1.3–1.4. We tested the same six NN architectures for all classification tasks so that for each task the architecture with the most appropriate complexity can be chosen.

The following hyperparameters were used in all neural networks:

The Adaptive Moment Estimation (Adam) was chosen as the optimiser (Kingma & Ba, 2014) and the hyperbolic tangent (tanh) as the activation function for the dense hidden layers, while we chose softmax as the activation function for the output layer (Bridle, 1990; Guo et al., 2017). In dropout layers we used a dropout rate of 30 %, except for the B10 model where we used 30 % and 50 % in the different dropout layers. Dropout has been shown to greatly improve the results (Wager et al., 2013). All NNs were trained for 500 epochs, except for the amino acid model selection NNs which were only trained for 50 epochs due to the long training time and a very good convergence after this number of epochs. Categorical-Cross-Entropy was selected as the loss function and accuracy as the validation metric. More details are given in the supplement. Trained NNs can be downloaded with the link given in the Data Availability section below.

2.2. Creating the training data sets

Supervised training of NNs requires extensive training data sets with many thousand alignments for which the correct model and topology are known. Alignments were simulated by starting from a random sequence with nucleotide or amino acid frequencies given by equilibrium frequencies of the substitution model. Sequences evolved under the stochastic process of a given model of sequence evolution along the given tree with branch lengths. In the present work we used the PolyMoSim-v1.1.4 software (https://github.com/cmayer/PolyMoSim) to simulate alignments of four sequences for a given tree and a specified model of sequence evolution for nucleotides or amino acids. PolyMoSim offers a wide range of output formats, including a format that directly outputs pattern frequencies of simulated alignments, which tremendously speeds up the process of creating training data sets.

We generated training data sets for the following prediction tasks:

(I) Prediction of the best fitting nucleotide substitution model for a given four sequence alignment: Models included in the prediction are the JC+I+G (Jukes & Cantor, 1969), K2PI+G (Kimura, 1980), F81+I+G (Felsenstein, 1981), HKY+I+G (Hasegawa et al., 1985), GTR+I+G (Tavaré, 1986) models. For all models a variable amount of invariant sites (+I) was considered and for all other alignment sites, gamma distributed site rates (+G) were assumed in the training and later in the verification data.

For each of the five nucleotide substitution models we simulated 90,000 alignments with what we call “normal” internal branch lengths (Table 2) and 90,000 alignments with particularly short internal branch lengths, resulting in a training data set of 900,000 alignments (Table 1). Alignment lengths were 30,000 bp, which is sufficient for sampling the 256 possible site pattern frequencies. The tree topology each alignment evolved on was chosen randomly among the three possible topologies for four taxa. Branch lengths were drawn randomly from uniform distributions in intervals given in Table 2. Model parameters of the corresponding substitution models were chosen randomly from the distributions and ranges as given in Table 2.

(II) Prediction of the best fitting amino acid substitution model for a given four taxon alignment: Models included in the prediction are the JTT+I+G (Jones et al., 1992), LG+I+G (Le & Gascuel, 2008), WAG+I+G (Whelan & Goldman, 2001), Dayhoff+I+G (Dayhoff et al., 1978) models. Again, invariant sites (+I) and a site rate heterogeneity governed by a gamma distribution (+G) have been used in the simulations. Since computing and storing amino acid site pattern frequencies is computationally expensive, we decreased the number of simulated data sets to 37,500 per substitution model for normal and the same number for short internal branch lengths, resulting in a training data set of 37,500*4*2 = 300,000 alignments (Table 1). The sequence length was

Table 2
Parameter ranges and distributions used in data simulation. The minimum and maximum values for the proportion of invariant sites, the shape parameter of the gamma distribution and the CG content have been chosen after inspecting a large number of real gene alignments.

Parameter	Range of values	Distribution	Substitution models involved
Proportion of invariant sites	0.0 to 0.5	Uniform	all
Shape parameter of gamma distribution	0.01 to 4.0	Uniform	all
Transition/ transversion ratio	1.0 to 3.0	Uniform	K2P,and HKY
Base frequencies: We choose CG randomly. Then the base frequencies are $\pi_A = \pi_T = (1\text{-}CG)/2$, $\pi_C = \pi_G = CG/2$	0.4 to 0.6	Normal distribution with mean = 0.5 and standard deviation = 0.03. Random numbers are redrawn until inside the range.	F81, HKY, and GTR
Relative rate of each substitution (six parameters)	0.1 to 1.0	Uniform	GTR
Terminal branch lengths	0.1 to 0.5	Uniform	all
Internal branch lengths (two subsets)	0.1 to 0.5 (normal) 0.001 to 0.02 (short)	Uniform	all

increased to 1,000,000 aa in order to sample the 160,000 different site patterns more comprehensively. Tree topologies, branch lengths and model parameters were chosen randomly as in the case of the nucleotide model from distributions and ranges given in Table 2.

(III) Prediction of the most likely topology for a given four taxon alignment of nucleotide sequences. As mentioned in the introduction, we trained NNs specific for the nucleotide substitution models (i.e., JC+I+G, K2P+I+G, F81+I+G, HKY+I+G, GTR+I+G). Each model specific NN was trained with 200,000 alignments of length 100,000 bp for each of the three topologies, i.e. 100,000 for normal and short internal branch lengths (see Table 1). Branch lengths and model parameters were chosen randomly from the distributions and ranges given in Table 2.

(IV) Prediction of the most likely topology for a given four taxon alignment of amino acid sequences: Again, we trained NNs specific for the amino acid substitution models (i.e. LG+I+G, JTT+I+G, WAG+I+G, Dayhoff+I+G). Each model specific NN was trained with 20,000 alignments of length 1,000,000 aa for each of the three topologies, i.e. 10,000 for normal and 10,000 short internal branch lengths (see Table 1). Branch lengths and model parameters were chosen randomly from the distributions and ranges given in Table 2.

The numbers of NNs we trained for each of the prediction tasks are given in Table 1. All trained NNs can be downloaded with the link provided in the Data Availability Section below.

2.3. Creating verification data sets for model prediction

Verification data sets used to test the trained NNs, and to compare prediction accuracies and computation times with other methods/programs have been generated independently but in the same way the training data was generated. Only different alignment lengths have been used in the different comparisons. To verify the substitution model classification, we simulated 1,000 alignments for each substitution model. For the five nucleotide models 5,000 alignments of length 30,000 bp and for the four amino acid models 4,000 alignments of length 10,000 aa were generated. The prediction successes of the NNs were compared to the model prediction of the ModelFinder software (Kalyaanamoorthy et al., 2017) implemented in IQ-TREE version 2.1.3 (Minh et al., 2020). To conduct a fair comparison, the set of models among which ModelFinder chooses the best model was restricted to the models available in the NN prediction by specifying the command line parameter "-mset JC,K2P,F81,HKY,GTR −mrate I+G" for nucleotide and "-mset JTT,LG,WAG,Dayhoff −mrate I+G" for amino acid data sets. In ModelFinder the BIC was used to select the best model. Two-tailed binomial tests with the standard Wilson score interval and with continuity correction (Wilson, 1927; Crawley, 2014) have been carried out to determine whether the ModelFinder or the NN predictions were significantly better (p-value < 0.05). For nucleotide models, the prediction accuracies of the correct base model have also been compared to the ModelRevelator software of Burgstaller-Muehlbacher et al. (2023). In all model selection comparison tests the model was classified as correctly chosen if the base model was equal to the base model used in the simulation. Model prediction accuracies of DeepNNPhylogeny and ModelRelevator were also benchmarked against models determined by ModelFinder on a COI data set. Methods and results of this comparison are presented in supplementary Section 2.4.

2.4. Creating verification data sets for topology prediction

To verify the topology prediction, we simulated 1,000 alignments of the lengths 1,000 and 10,000 nucleotide bases and amino acids for each of the three topologies. Two separate verification data sets of the given sizes were created for normal and short internal branch lengths. The topology predictions have been compared with the IQ-TREE implementation of the ML method and the BioNJ method (Gascuel, 1997), an improved implementation of the Neighbour Joining algorithm. IQ-TREE produces this tree as a side product when invoked with the parameters given in supplementary Section 1.5. The topologies of the BioNJ trees were then taken from the ".bionj" files created by IQ-TREE. For the ML tree inference, IQ-TREE was restricted to the correct substitution models (−m basemodel+I+G option) for a better comparison with the NN method. Again, the same binomial test was used to determine whether the NN method was significantly better or worse than the ML and/or the BioNJ methods.

2.5. Computing times

Computing times of our NNs have been compared with the ML and BioNJ methods as implemented in IQ-TREE and with the ModelRevelator software as described in the supplement Section 1.5.

2.6. Alternative machine learning algorithms

Besides NNs, a number of machine learning classifiers exist that are well suited for frequency distribution classifications (see e.g. Hastie et al., 2009). To evaluate the performance of different machine learning algorithms, including the Gaussian process classifier, several implementations of the RF classifier, the Light Gradient-Boosting Machine (LGBM) classifier and the SVM classifier, the Lazypredict Python library modelwas utilised (Pandala, https://github.com/shankarpandala/lazypredict). A detailed description of how the alternative machine learning algorithms were trained is given in the supplementary methods section 1.6.

2.7. Data visualisation

Prediction and computation time results for NNs were visualised with the Matplotlib library (Hunter, 2007) using Python3. Lazypredict results were visualised using the Seaborn library (Waskom et al., 2017).

3. Results

3.1. Substitution model prediction success

For nucleotide and amino acid alignments, the model prediction accuracy of the best NN architecture B10 is shown in Fig. 3. The mean prediction accuracy of the B10 nucleotide NN was 92.6 %, which was considerably higher than the prediction accuracy of the ModelFinder + BIC and the CNN based ModelRevelator model inferences, which showed accuracies of 87.1 % and 63 % respectively. DeepNNPhylogeny was significantly better than the ML method and the ModelRevelator software (p-value < 2e-16).

The amino acid NN prediction accuracy of 99.8 % obtained with the B10 network is only marginally, but significantly (p-value = 0.0076) worse than the 100 % accuracy obtained with ModelFinder for our test conditions. A comparison of the prediction accuracies of different NN architectures for nucleotide and amino acid evolutionary models is presented in supplementary materials Tables S1 and S2.

3.2. Topology prediction success

For nucleotide and amino acid alignments, the topology prediction accuracies of the best NN architectures and their comparison with other methods are shown in Figs. 4 to 7. We have trained substitution model specific NNs for the topology prediction, which means that a model has to be selected prior to conducting a topology classification. This is analogous to what we do when using the ML method. Therefore, we tested the NNs with alignments that evolved under the same evolutionary model as the alignments that were used to train the models, but with independently chosen model parameters and branch lengths.

Topology prediction accuracies using NNs, the ML method and the BioNJ method for nucleotide alignments of length 1,000 bp and 10,000 bp and inner branch lengths in the range 0.1 < v < 0.5 are shown in Fig. 4.

The accuracy of NN topology predictions for nucleotide alignments of length 1,000 bp ranged from 99.4% to 99.6 % (Tables S1; Fig. 4). No statistically significant differences were found between NNs and the ML method (p-values ranged from 0.42 to 1.0).

For alignments lengths of 10,000 bp, our NN predictions achieved high accuracies of 99.97 % for the HKY model and 100.0 % for all other models (see Supplementary Tables S1 and Fig. 4), which did not differ significantly from the reconstruction success of the ML method (p-value = 1.0). In contrast, the BioNJ algorithm showed significantly lower prediction accuracy than the NN classifiers (p-values < 2e-16). A comparison of the six NN architectures is shown in the supplementary materials. See Tables S2 for details.

The topology reconstruction success of the ML, BioNJ and NN methods were also compared for alignments that were simulated on trees with particularly short internal branches in the range 0.001 < v < 0.02 (Fig. 5). For sequences of length 1,000 bp, we observed a poor accuracy for both, NNs (ranging from 47.6 % to 48.3 %) and the ML method (ranging form 47.6 % and 48.6 %), with no significant differences (p-value between 0.72 and 1.0). The BioNJ method's prediction success rate was 37.8 %–43.3 %, which is better than choosing a topology randomly (33.3 %). For alignment lengths of 10,000 bp and for short internal branch lengths the ML method (with accuracies ranging from 71.8 % and 73.5 %) slightly outperformed the NNs for the K2P+I+G, the F81+I+G, and the HKY+I+G models (p-value between 0.014 and 0.04), but not for the other models (p-value between 0.07 and 0.08). Accuracies are given for the best performing NN architecture. A detailed comparison of the accuracies for different NN architectures is given in Supplementary Table S2.

For amino acid alignments of lengths 1,000 aa and 10,000 aa that evolved on trees with inner branch lengths in the ranges 0.1–0.5 and 0.001–0.02, respectively, topology prediction success rates are shown in Fig. 6 and Fig. 7. For normal internal branch lengths and an alignment length of 1,000 aa, NNs achieved prediction accuracies of 97.5 %–98.3 % (Table S1; Fig. 6), while the ML method achieved accuracies of 99.90 %–99.97 %, showing a significantly better prediction success (p-value < 2e-16). The BioNJ method had accuracies of 93.5 %–95.3 %, which was significantly lower than that of the NNs (p-value between 9.7e and 11 and < 2e-16). For alignments of length 10,000 amino acid residues, the binomial distribution test did not find any statistically significant difference between NNs and the ML method for the JTT+I+G, LG+I+G,

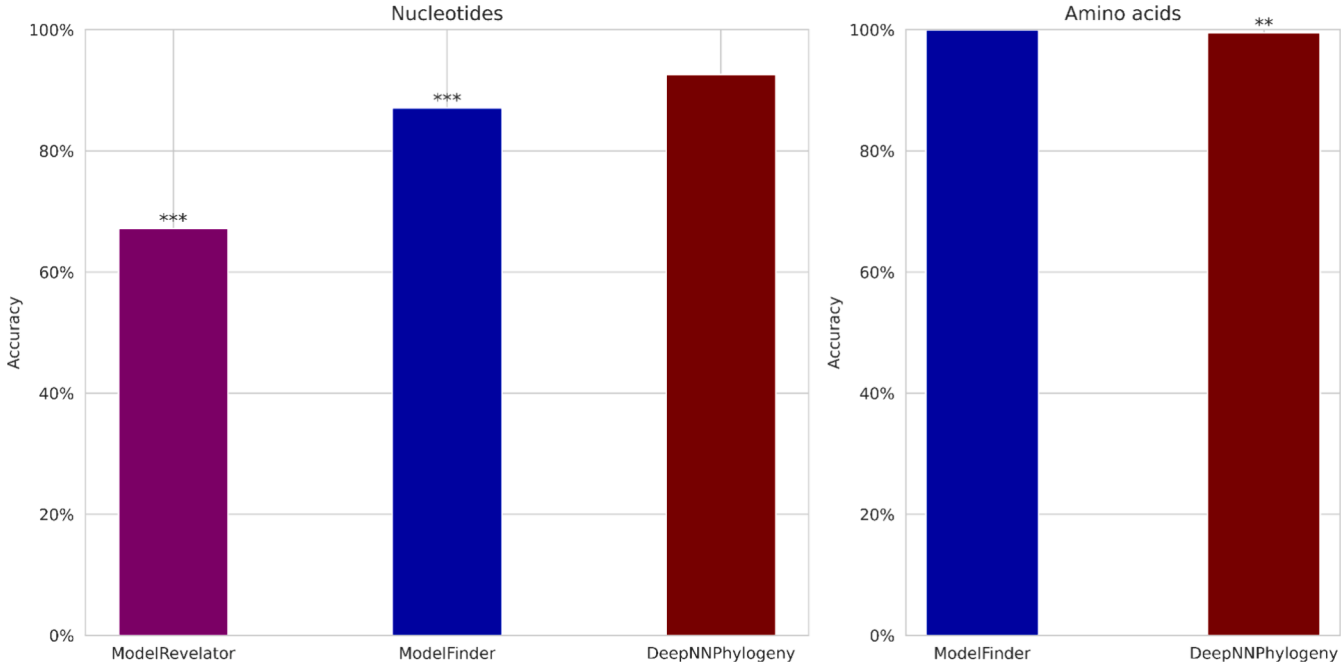

Fig. 3. Substitution model prediction accuracy of DeepNNPhylogeny, ModelRevelator, and ML + BIC on test data sets with alignments of length 30,000 bp and 10,000 aa (with branch lengths ranging from 0.1 to 0.5). p-values with 0.001 < p ≤ 0.01 and p ≤ 0.001 are flagged with "**" (medium significance) and "***" (high significance), respectively.

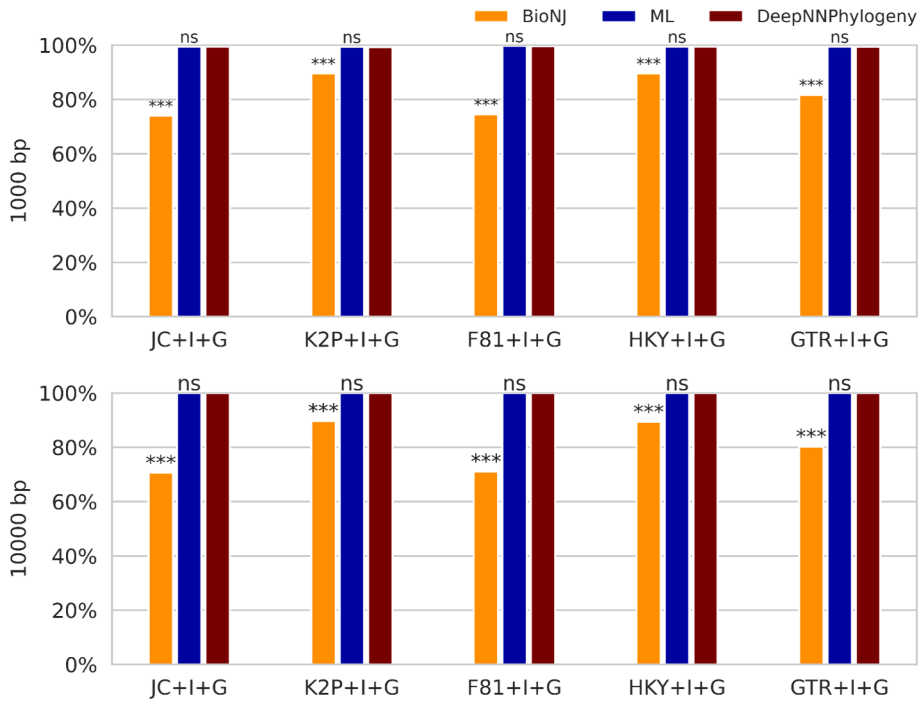

Fig. 4. Accuracy of topology reconstruction, with branch lengths ranging from 0.1 to 0.5, using NNs, ML + BIC and BioNJ for 1,000 bp (top figure) and 10,000 bp (bottom figure) long nucleotide alignments. The significance test was conducted between BioNJ and NN and between ML and NN. p-values with $p > 0.05$, and $p \leq 0.001$ are flagged with "ns" (non-significant), and "***" (high significance), respectively.

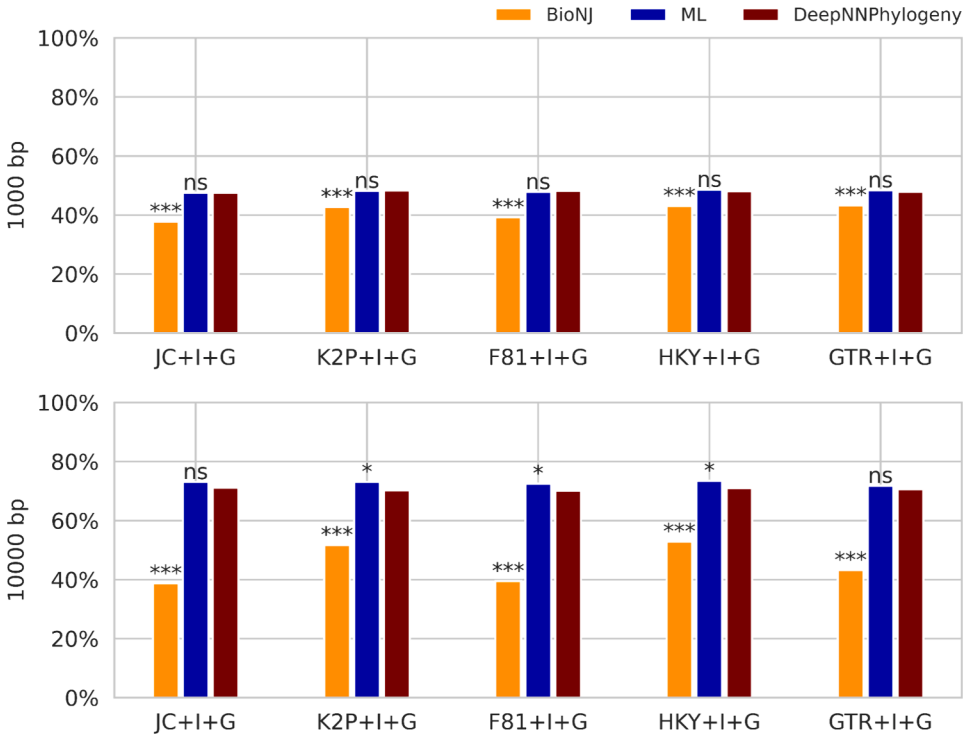

Fig. 5. Accuracy of topology reconstruction with internal branch lengths in the range 0.001 to 0.02 using NNs, ML + BIC and BioNJ on 1,000 bp and 10,000 bp long nucleotide alignments. Significance tests were conducted between BioNJ and NN, and between ML and NN. p-values with $p > 0.05$, $0.01 < p \leq 0.05$, $0.001 < p \leq 0.01$ and $p \leq 0.001$ are flagged with "ns" (non-significant), "*" (low significance), "**" (medium significance), and "***" (high significance), respectively.

and WAG+I+G models (p-value from 0.074 to 0.48), but for the Dayhoff+I+G model (p-value = 0.023). Accuracies achieved with BioNJ were significantly lower for all evolutionary models (p-value < 2e-16).

For amino acid alignments of length 1,000 aa with short internal branch lengths, the NNs achieved accuracies of 41.2 %–43.5 % (Fig. 7). They were significantly outperformed by the ML and BioNJ methods (p-values between 3e and 05 and < 2e-16).

For longer alignments with 10,000 aa, the NNs achieved average

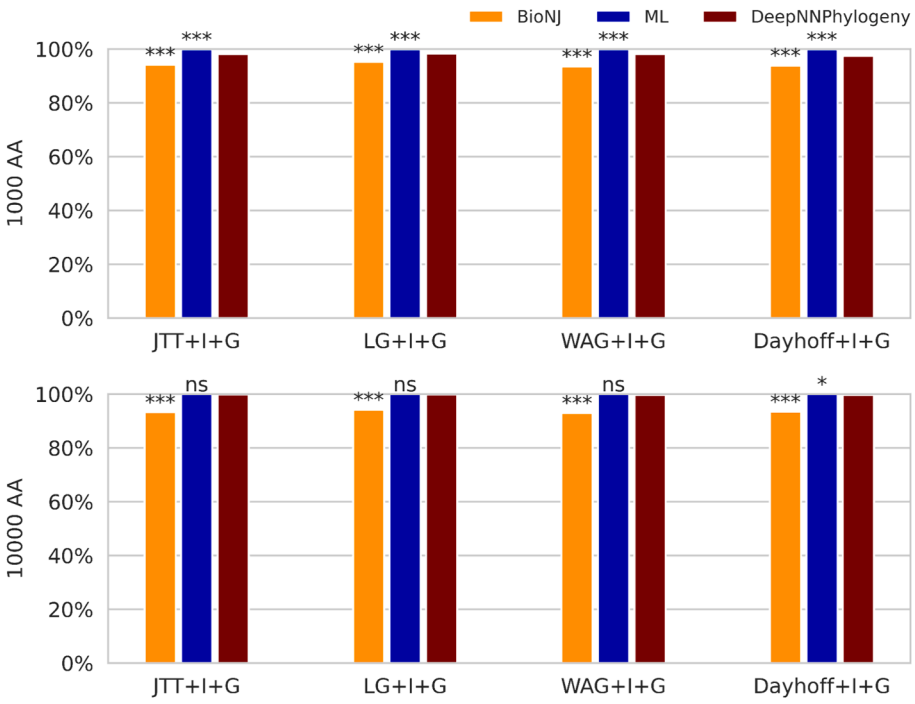

Fig. 6. Accuracy of topology reconstruction (with internal branch lengths in the range 0.1 to 0.5) using NNs, ML + BIC and BioNJ on 1,000 and 10,000 long amino acid alignments. The results were obtained for the sequences that evolved under four different substitution models: JTT+I+G, LG+I+G, WAG+I+G, Dayhoff+I+G. Significance tests were conducted between BioNJ and NN, and between ML and NN. p-values with $p > 0.05$, $0.01 < p \leq 0.05$, $0.001 < p \leq 0.01$ and $p \leq 0.001$ are flagged with "ns", "*", "**", and "***", respectively.

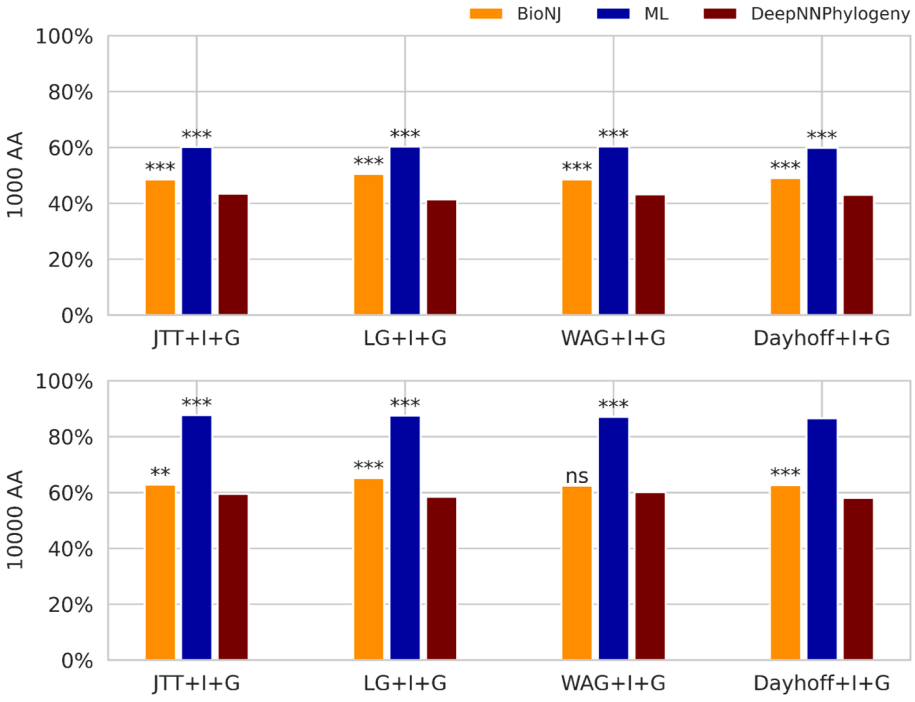

Fig. 7. Accuracy of topology reconstruction with inner branch lengths ranging from 0.001 to 0.02 using NNs, ML + BIC and BioNJ on 1,000 (top figure) and 10,000 (bottom figure) long amino acid alignments. Significance tests were conducted between NN and BioNJ as well as between NN and ML. p-values with $p > 0.05$, $0.01 < p \leq 0.05$, $0.001 < p \leq 0.01$ and $p \leq 0.001$ are flagged with "ns", "*", "**", and "***", respectively.

accuracies of 58.1 %–60.2 %. They are again outperformed by the ML method (p-value < 2e-16) but were as accurate as BioNJ for the WAG evolutionary model (with p-values of 0.08), though they remained inferior for the rest of the models (p-value between 0.009 and 1e-07).

3.3. *Alternative machine learning algorithms for topology prediction*

For classifiers trained and tested on nucleotide alignments of length 100,000 bp that evolved under the K2P+I+G evolutionary model on trees with normal and short internal branch lengths, we obtained the

highest prediction accuracy of 96 % with the SVM classifier which is marginally better than the best NN (B2) with 95.4 %. Detailed results are provided in the supplementary materials. For amino acid alignments of length 1,000,000 aa that evolved under the Dayhoff+I+G evolutionary model the highest prediction accuracy of 98 % was obtained with the LGBM classifier which is a bit lower than the prediction accuracy of the best NN (B10) with 99.1 %. We found that nine out of 25 machine learning algorithms achieved a topology prediction accuracy of 90 % or higher in predicting the correct tree topology for nucleotide data, whereas 12 out of 26 machine learning algorithms achieved a comparable accuracy for amino acid data sets. See supplementary material section 2.2 and Figs. S4 for details.

3.4. Prediction times

We found that prediction times for the ML method are at least a factor of 3.5 higher than the prediction times of the NNs for nucleotide alignments of length 1,000 bp and a factor of 16 higher for amino acid alignment of length 1,000 aa. For increasing alignment lengths, the relative advantage of the NNs over the ML method increases. Detailed results are presented in the supplementary results and Figs. S3. The ModelRevelator program which analyses images of nucleotide alignments to predict the best substitution model was a factor of 290 slower for alignments of length 1.000 bp and a factor of about 370 slower for alignments of length 100.000 bp. Furthermore, the ModelRevelator software was much slower than ModelFinder implemented in IQ-TREE.

4. Discussion and comparison of results

Predicting the model of sequence evolution and the topology, which most likely created an alignment, is an essential task in molecular biology. Today, the ML method is the gold standard for these inference tasks, since it has been shown that the ML method converges on the correct result as the amount of available data approaches infinity if certain conditions are met. Disadvantages of the ML method are comparatively long computation times as well as a complex procedure to implement new evolutionary scenarios such as new models or complex mixture models. Alternative methods to the ML method are not only of theoretical interest. They could offer advantages in computational efficiency and facilitate the implementation of more complex evolutionary scenarios in the future.

Our results show that NNs are well suited to predict the best fitting model of sequence evolution for the nucleotide and amino acid models included in our test. For the nucleotide model prediction on simulated data, our NNs were not only significantly more accurate than the ModelRevelator software, but also than the ModelFinder result using ML + BIC (see Fig. 3 and Table S1 for details). A result better than the ML method would not be expected and we conjecture that the lower prediction accuracy of ModelFinder is linked to the fact that it does not select the model with the highest likelihood, but instead the model with the best BIC score, which is the most widely used model selection criterion and the default in ModelFinder. A detailed comparison of the NN model predictions with different model selection criteria implemented in ModelFinder would be interesting but is beyond the scope of this paper. For nucleotide alignments, our NNs, ModelFinder and ModelRevelator yielded model prediction accuracies well below 100 %, which is expected since many substitution models are nested, i.e. a considerable part of the model parameter space of more complex models can mimic parameter values of a simpler model. For these model parameter combinations, the correct behaviour is to choose the simpler model. The model selection accuracies of DeepNNPhylogey and ModelReveltor on an avian CO1 data set were very similar (see supplementary Section 2.4).

In the case of amino acid model predictions, the ML method was marginally but significantly better than the NN (100 % vs 99.8 %). Since amino acid models are not nested, accuracies close to 100 % are achievable. During the course of the project, we have already increased the size of the amino acid training data from a total of 60,000 to 300,000 training alignments, which increased the model prediction accuracy from 88.0 % (data not shown) to 99.8 %. We expect that an even larger training data set yields a NN that is not significantly worse than ModelFinder. It should be noted that for amino acid alignments of four sequences, we have 160,000 different site patterns. Even for a length of the training sequences of 1,000,000 aa the average number of patterns found per pattern class is only 1,000,000/160,000 = 6.25. Clearly, this alignment length does not allow us to simulate highly precise pattern frequencies. In order to sample the pattern frequencies sufficiently for all model parameter and branch length combinations, it is expected that (i) longer sequence lengths and (ii) a larger number of training alignments is required. We expect that with more computational resources it should be possible to train amino acid NNs such that they become as accurate as the ML method in predicting the correct substitution model. Indeed, the size of the training data set was limited by the available computational resources required for creating and storing the training data set. The problem is illustrated by the fact that the amino acid training data set we generated requires 160,000*300,000*4 Byte = 192 GB of computer RAM, if 32-bit floating point numbers are utilised.

Machine learning has also been used by Abadi et al. (2020) and Burgstaller-Muehlbacher et al. (2023) to choose suitable nucleotide substitution models. Abadi et al. (2020) used a RF classifier to choose the model of sequence evolution that yields the best branch length estimates in a phylogenetic tree reconstruction that follows the model selection step. They have shown that their approach is better than the standard model selection criteria to select a substitution model that recovers the correct branch lengths.

Burgstaller-Muehlbacher et al. (2023) used their ModelRevelator software and analysed images of alignments of 8, 16, 64, 128, 256 and 1024 taxa with CNNs to determine the best model of sequence evolution and to estimate the shape parameter of the gamma distribution. For alignment lengths of 1,000 bp to 100,000 bp, they obtained model prediction accuracies that are comparable to those of ModelFinder. A feature of the ModelRevelator software is that it was trained on alignments with empirical branch length distributions. Furthermore it was trained to distinguish a set of nucleotide base models with and without assuming gamma distributed site rates (+G). For our branch length distributions and a site rate heterogeneity including invariant sites and gamma distributed site rates (+I+G) we find that ModelRevelator has a much lower prediction accuracy for the base model (i.e. neglecting the model extension) than ModelFinder and DeepNNPhlyogeny. Since a +I+G rate heterogeneity is common, it should be included in the training of the ModelRevelator. An advantage of ModelRevelator is that it is directly applicable to data sets with a higher number of sequences. An important disadvantage is that computations times are a factor of 290 to about 370 longer for alignments of length 1000 to 100.000 bp.

Our results also show that NNs are highly suitable to predict the best tree topology for nucleotide and amino acid alignments. Topology prediction accuracies of NNs for nucleotide alignments were highly similar to those of the ML method. The ML method was marginally but significantly better only for alignments of length 10,000 bp that evolved on a tree with short internal branch lengths (0.001 to 0.02) and for the evolutionary model K2P+I+G, and F81+I+G if the level of significance is used without Bonferroni correction. Since we have conducted multiple tests, some tests are expected to have a p-value < 0.05 by chance. With a Bonferroni correction that corrects for multiple tests, the ML method is not significantly better in any of the nucleotide topology predictions. Furthermore, we expect that more training data could remove any difference we have found. It should be noted that short internal branch lengths pose a problem to all tree reconstruction methods, since for an alignment of length of 10,000 bp and a short internal branch length in the range 0.001 to 0.02 we only expect a total of 10–200 substitutions along the inner branch and along the whole alignment. Therefore, the site pattern frequencies of alignments that evolved on different

topologies will differ only marginally if the inner branch length is very short. The high number of misclassifications of the ML and NN method for short internal branch lengths are most likely attributed to cases in which site pattern frequencies of alignments favour the wrong topology by chance. In this case only longer sequences can help to reconstruct the correct topology.

Suvorov (2020) showed that CNNs which analyse images of alignments can be as good as ML and other methods for alignments that evolved on trees in the so-called Farris- and Felsenstein zones. For alignments of length 1,000 bp that evolved on trees with internal branches in the range 0 to 0.5, their CNNs showed an accuracy of 82 %, which is as accurate as the ML method (82 %) for the same data set. In our study, NNs showed accuracies above 99 % for internal branches in the range 0.1 to 0.5 for the same alignment length. For alignments of length 1,000 bp that evolved on trees with an internal branch length in the range of 0.0 to 0.05, their CNNs achieved an accuracy of 52 %, while the ML method achieved 57 %. Our NNs also showed accuracies around 48 %, with internal branch lengths in the range 0.001 to 0.02. A direct comparison with their software was unfortunately not possible, since despite considerable efforts we were not able to load trained CNNs into their software.

Also Zou et al. (2020) proposed CNNs to analyse images of alignments that showed accuracies on quartet trees that are comparable to the ML and Bayesian methods. Their approach showed accuracies in the range from 90 % to 98 % when alignment lengths increased from 100 to 10,000 bp. The branch lengths were in the range from 0.02 to 2, which implies a considerably larger mean branch length than in our simulations and should be the reason for the higher accuracies in their reconstructions.

For amino acid alignments the ML topology predictions were always significantly better than NNs topology predictions for alignments of the length 1,000 aa and for alignments generated with short internal branch lengths. For alignment lengths of 10,000 aa and normal internal branch lengths, the differences between ML and NNs would not be significant if the level of significance would be adjusted for multiple testing using the Bonferroni correction procedure. As in the case of the amino acid model selection, the size of the training data set was limited by the available computational resources. Our initial training data set was even smaller with only 6,000 training alignments of length 1,000,000 aa. For the final amino acid NNs we used 60,000 training alignments of length 1,000,000 aa. When testing the trained NNs with alignments of length 1,000 aa and normal internal branch lengths the prediction accuracy for the four amino acid models improved considerably from the range 94.0 % to 95.5 % (data not shown) to a range of 97.5 % to 98.3 % (Table S2). This suggests that by further increasing the size of the training data set, it is possible to achieve higher accuracies. Altogether we have found prediction accuracies as good as or almost as good as the prediction accuracies obtained with the ML method.

We want to emphasise that we trained all neural networks on much longer alignments than the typical alignment lengths that are used in predictions. We have found that increasing the alignment lengths in the training data increases the prediction accuracies for all alignment lengths used in the predictions. Our networks classify models and topologies using site pattern frequency distributions and these distributions are sampled more accurately if the training alignments are longer. In particular, alignment lengths should be sufficiently long so that (almost) all patterns are sampled at least a few times.

Site pattern frequencies have previously been used by Leuchtenberger et al. (2020) to choose the best tree reconstruction method and by Suvorov & Schrider (2024) to predict branch lengths. In contrast, Zou et al. (2020) and Suvorov et al. (2020), Suvorov and Schrider (2024) suggested analysing images of alignments with CNNs to predict the best tree topology. Due to the width of the convolutional kernel, this approach combines the information about neighbouring alignment sites, which are consequently not treated as independent and identically distributed. It is possible that this extracts information about the co-evolution of neighbouring alignment sites from an alignment, but the influence of this approach has not been investigated yet. Presumably, these CNNs would be specific for genomic regions since their convolutional kernels will include information about co-evolving neighbouring alignment sites with the feature or drawback that different NNs would need to be trained for different genomic regions. A classification with site pattern frequencies will not include information about co-evolving neighbouring alignment sites.

A problem we faced during training of NNs was the need to normalise the site pattern frequencies before passing them to the first dense layer for some of the prediction tasks. Even though the site pattern frequencies are all numbers in the range of 0 to 1, we found that a normalisation is indispensable for some of the NNs. In particular the amino acid NNs often did not train at all if non-normalised data was used, resulting in training accuracies below 50 % in the limit of long training times or in an oscillation of the training accuracies. When normalising input data before training and predictions, this problem did not occur. The methods we used to normalise the data is described in detail in supplementary materials section 1.2.

In contrast to the ML method, which evaluates an optimality criterion, the machine learning methods learn the site pattern frequency distributions that are associated with certain substitution models and tree topologies. By training a classifier, one circumvents optimising the likelihood function, which can be computationally more efficient. In order for a machine learning method to be able to predict the correct model and topology, the training data must sample site pattern frequencies with sufficient density for all combinations of the substitution model parameters and branch lengths that shall be classified, in particular for the boundary cases.

From a mathematical point of view one can say that machine learning methods are trained to learn the probability distributions which are hardcoded in maximum likelihood methods. Since it is more simple to generate training data for more complex evolutionary scenarios than implementing the maths to incorporate them in the maximum likelihood framework, the methods proposed here should make it more simple to study the effect of what is referred to as model violations in the maximum likelihood framework. Studying model violations and the robustness of the proposed machine learning methods should be explored in future research. This could include investigating machine learning based approaches to identify violations to assumptions made by the maximum likelihood method.

If NNs are used for the classification, they have to be designed such that they are sufficiently complex to be able to “memorise” the frequency distributions they were trained with. For this, the network architecture, the number of layers and neurons as well as other hyperparameters, such as the activation function and the optimizer have to be optimised for the specific classification task. In this project we tried to optimise the number of layers, the number of neurons and other hyperparameters such as the activation function, the optimizer and the dropout rate for all six architectures by conducting grid searches for selected parameters. Since the number of possible hyperparameter combinations is huge, it is possible that other hyperparameter combinations lead to even higher prediction accuracies.

In this study we compared six NN architectures ranging from a simple sequential NN to NNs with multiple branches and a NN with interconnections. The NNs with more neurons and more trainable parameters are also able to memorise more information. For some of the classification tasks the best NN architecture is significantly better than all other architectures (see supplementary Tables S1 and S2 for details). Altogether, the B10 architecture was the best for both substitution model prediction tasks. For most topology predictions, the B3 architecture is preferred and for the few cases for which B3 is not the best architecture, it was not significantly worse. Therefore, the NN architecture implemented in B3 is a good starting point when designing NNs for topology predictions. The most complex architecture CU was the best architecture for several topology prediction tasks, but it could not

outperform the other architectures significantly. While we have a significant difference in the number of features used in the nucleotide and amino acid data sets, we see no general trend that the simpler architectures are better suited for nucleotide data or more complex architectures are better suited for amino acid data.

We have also compared the NN classifiers with alternative machine learning classifiers such as the SVM and LGBM classifiers. For selected test cases we found that the best performing alternative classifiers are as good or almost as good as the best NNs. Certainly, these classifiers should be explored in more detail in future studies. Their main advantages are faster training and predictions and that the trained models have much smaller file sizes and would be much easier to use.

The main advantage of DeepNNPhylogeny is its high speed in determining the best model of sequence evolution and topology compared to ML method, but also to the ModelRevelator software. For short nucleotide alignments, the NNs are about 3.5 times faster than the ML method and the advantage of the NNs increases for longer alignments and amino acid sequences. We have mentioned in the supplement that an even faster prediction would be possible if multiple predictions are combined or if a GPU is used.

Currently, the main disadvantage of using machine learning methods for phylogenetic tree reconstruction is that these methods are difficult to extend beyond a few taxa. The reason is that classifiers have to be trained with a large number of data sets for each possible topology, which quickly becomes infeasible as the number of sequences increases. We are convinced that this problem can be overcome in the future, e.g. by combining quartet trees using quartet puzzling (Stimmer and von Haeseler 1996) or related methods to obtain trees with many taxa. Therefore we expect that our method can be extended to alignments with more taxa and will be useful in future research. Another limitation is that we currently can only analyse alignment for which gap or ambiguity containing alignment columns have been removed. We are confident that including gaps in the site pattern will be possible in future updates of the software.

Despite the fact that NN classifiers are currently limited to four taxa, there are a number of potential direct applications of our method. Beside their computational advantage, the following applications are conceivable:

(i) The PartitionFinder program (Lanfear et al., 2012; Lanfear et al., 2017) spends most of its computing time on determining the best model of sequence evolution for a large number of subsets of a data set. Using NNs could tremendously speed up the process of finding *meta*-partitions that best evolve under the same substitution model. (ii) Massive computations of quartet trees are used e.g. in approaches such as Quartet Puzzling (Strimmer & von Haeseler, 1996), Likelihood mapping (Strimmer & von Haeseler, 1997), and Sliding window phylogenetic analyses to detect introgression (see e.g. Hibbins & Hahn, 2022). These methods could benefit from a fast topology prediction method for four sequences.

Recommendations for choosing the best neural network architecture.

- For model predictions, the B10 architecture had the highest prediction success for nucleotide as well as for amino acid substitution model predictions. The B3 model was only insignificantly less accurate. (See Tables S1 and S2.)
- For topology predictions, we recommend the B3 model corresponding to the substitution model predicted by our model predictor. The B3 model is the best model for the majority of the studied cases (Tables S1 and S2). In all cases in which it was not the best model, it was not significantly worse making it a good general choice independent of the model, the alignment length and the typically unknown internal branch length.
- In summary, we propose the B3 model as a universally good choice for model selection and topology classification problems with site pattern frequencies.

5. Conclusion

In this study we have shown that NNs trained on site pattern frequencies of four taxon alignments are as accurate or close to being as accurate as the ML method to select the best fitting nucleotide or amino acid evolutionary model as well as the tree topology on which the sequences in the alignment evolved. In cases NNs are not as accurate as the ML method, we provide evidence that a larger training data set should be able to close the accuracy gap. The main advantage of our NNs is that they are much faster than the ML method and the ModelRevelator software that predicts nucleotide models by analysing alignment images. For the ML method, the speed difference ranged from a factor of about 3.5 on short nucleotide alignments to more than 1000 for the long amino acid alignments analysed in this study. Furthermore, we have shown that other machine learning classifiers such as SVM and the LGBM classifier are about as accurate as the NNs.

CRediT authorship contribution statement

Nikita Kulikov: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Fatemeh Derakhshandeh:** Methodology, Investigation, Software, Writing – review & editing. **Christoph Mayer:** Conceptualization, Investigation, Methodology, Project administration, Supervision, Software, Validation, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The software for the NNs implementation as well as scripts that have been used in this study can be found at GitHub (https://github.com/cmayer/DeepNNPhylogeny). The PolyMoSim-v1.1.4 program that is needed for training data generation can be obtained at (https://github.com/cmayer/PolyMoSim). The trained machine learning models can be downloaded from the DryAd repository with the DOI https://doi.org/10.5061/dryad.ksn02v783.

Acknowledgement

We cordially thank Marie Brasseur for valuable suggestions on the manuscript. Furthermore, we are thankful to the anonymous reviewers for their helpful comments. The project used the high-performance computing resources of the LIB and the University of Bonn.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ympev.2024.108181.

References

Abadi, S., Azouri, D., Pupko, T., et al., 2019. Model selection may not be a mandatory step for phylogeny reconstruction. Nat Commun 10, 934. https://doi.org/10.1038/s41467-019-08822-w.

Abadi, S., Avram, O., Rosset, S., Pupko, T., Mayrose, I., 2020. ModelTeller: model selection for optimal phylogenetic reconstruction using machine learning. Mol. Biol. Evol. 37 (11), 3338–3352.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al., 2016. TensorFlow: a system for Large-Scale machine learning. In: 12th USENIX symposium on operating systems design and implementation (OSDI 16), pp. 265–283.

Borowiec, M.L., Dikow, R.B., Frandsen, P.B., McKeeken, A., Valentini, G., White, A.E., 2022. Deep learning as a tool for ecology and evolution. Methods Ecol. Evol. 13 (8), 1640–1660.
Bridle, J.S., 1990. Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In: Soulié, F.F., Hérault, J. (Eds.), Neurocomputing: Algorithms, architectures and applications. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 227–236. https://doi.org/10.1007/978-3-642-76153-9_28.
Burgstaller-Muehlbacher, S., Crotty, S.M., Schmidt, H.A., Reden, F., Drucks, T., von Haeseler, A., 2023. ModelRevelator: Fast phylogenetic model estimation via deep learning. Mol. Phylogenet. Evol. 188, 107905 https://doi.org/10.1016/j.ympev.2023.107905.
Cavalli-Sforza, L.L., Edwards, A.W., 1967. Phylogenetic analysis. Models and estimation procedures. Am. J. Hum. Genet. 19 (3.1), 233.
Cramér, H., 1946. Mathematical Methods of Statistics. Princeton Univ. Press, Princeton, NJ.
Crawley, M.J., 2014. Statistics: an introduction using R. John Wiley & Sons.
Dayhoff, M., Schwartz, R., Orcutt, B., 1978. 22 a model of evolutionary change in proteins. Atlas of Protein Sequence and Structure 5, 345–352.
Edwards, A.W.E., Cavalli-Sforza, L.L., 1964. Reconstruction of evolutionary trees. Phenetic and phylogenetic classification. Systematics Association, London, pp. 67–76.
Farris, J.S., 1970. Methods for computing Wagner trees. Syst. Biol. 19 (1), 83–92. https://doi.org/10.1093/sysbio/19.1.83.
Felsenstein, J., 1981. Evolutionary trees from DNA sequences: A maximum likelihood approach. J Mol Evol 17, 368–376. https://doi.org/10.1007/BF01734359.
Felsenstein, J., 2004. Inferring Phylogenies. Sinauer Associates Inc., Sunderland (MA).
Fitch, W.M., 1971. Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. Syst. Biol. 20 (4), 406–416. https://doi.org/10.1093/sysbio/20.4.406.
Gamage, G., Gimhana, N., Perera, I., Bandara, S., Pathirana, T., Wickramarachchi, A., Mallawaarachchi, V., 2020. Phylogenetic Tree Construction Using K-Mer Forest-Based Distance Calculation. International Association of Online Engineering.
Gascuel, O., 1997. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. Mol. Biol. Evol. 14 (7), 685–695. https://doi.org/10.1093/oxfordjournals.molbev.a025808.
Graur, D., Li, W.H., 1997. Molecular evolution. Sinauer Associates, Sunderland, MA.
Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q., 2017. On Calibration of Modern Neural Networks. https://doi.org/10.48550/arXiv.1706.04599.
Halgaswaththa, T., Atukorale, A.S., Jayawardena, M., Weerasena, J., 2012. Neural network based phylogenetic analysis. In: 2012 International Conference on Biomedical Engineering (ICoBE). Presented at the 2012 International Conference on Biomedical Engineering (ICoBE). IEEE, Penang, Malaysia, pp. 155–160. https://doi.org/10.1109/ICoBE.2012.6178974.
Hasegawa, M., Kishino, H., Yano, Ta., 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J Mol Evol 22, 160–174. https://doi.org/10.1007/BF02101694.
Hastie, T., Tibshirani, R., Friedman, J.H., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
Hibbins, M.S., Hahn, M.W., 2022. Phylogenomic approaches to detecting and characterizing introgression. Genetics 220, 173. https://doi.org/10.1093/genetics/iyab173.
Huelsenbeck, J.P., 1995. Performance of phylogenetic methods in simulation. Syst. Biol. 44 (1), 17–48.
Huelsenbeck, J.P., Hillis, D.M., 1993. Success of Phylogenetic Methods in the Four-Taxon Case. Syst Biol 42, 247–264. https://doi.org/10.1093/sysbio/42.3.247.
Hunter, J.D., 2007. Matplotlib: A 2D Graphics Environment. Comput. Sci. Eng. 9, 90–95. https://doi.org/10.1109/MCSE.2007.55.
Jones, D.T., Taylor, W.R., Thornton, J.M., 1992. The rapid generation of mutation data matrices from protein sequences. Bioinformatics 8 (3), 275–282. https://doi.org/10.1093/bioinformatics/8.3.275.
Jukes, T.H., Cantor, C.R., 1969. Evolution of protein molecules. Mammalian Protein Metabolism 3, 21–132.
Kalyaanamoorthy, S., Minh, B., Wong, T., et al., 2017. ModelFinder: fast model selection for accurate phylogenetic estimates. Nat Methods 14, 587–589. https://doi.org/10.1038/nmeth.4285.
Keane, T.M., Creevey, C.J., Pentony, M.M., Naughton, T.J., Mclnerney, J.O., 2006. Assessment of methods for amino acid matrix selection and their use on empirical data shows that ad hoc assumptions for choice of matrix are not justified. BMC *Evol Biol*. 24, 6–29. https://doi.org/10.1186/1471-2148-6-29.
Khoussi, S., Heckert, A., Battou, A., Bensalem, S., 2021. Neural networks for classifying probability distributions (No. NIST TN 2152). National Institute of Standards and Technology (U.S.), Gaithersburg, MD. https://doi.org/10.6028/NIST.TN.2152.
Kimura, M., 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. J Mol Evol 16, 111–120. https://doi.org/10.1007/BF01731581.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. https://doi.org/10.48550/arXiv.1412.6980.
Lanfear, R., Calcott, B., Ho, S.Y., Guindon, S., 2012. PartitionFinder: combined selection of partitioning schemes and substitution models for phylogenetic analyses. Mol. Biol. Evol. 29 (6), 1695–1701.
Lanfear, R., Frandsen, P.B., Wright, A.M., Senfeld, T., Calcott, B., 2017. PartitionFinder 2: new methods for selecting partitioned models of evolution for molecular and morphological phylogenetic analyses. Mol. Biol. Evol. 34 (3), 772–773.
Le, S.Q., Gascuel, O., 2008. An improved general amino acid replacement matrix. Mol. Biol. Evol. 25, 1307–1320. https://doi.org/10.1093/molbev/msn067.
LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L. D., 1989. Backpropagation applied to handwritten zip code recognition. Neural Comput. 1 (4), 541–551.
Leuchtenberger, A.F., Crotty, S.M., Drucks, T., Schmidt, H.A., Burgstaller-Muehlbacher, S., von Haeseler, A., 2020. Distinguishing Felsenstein zone from Farris zone using neural networks. Mol. Biol. Evol. 37 (12), 3632–3641.
Mahesh, B., 2020. Machine learning algorithms-a review. International Journal of Science and Research (IJSR). 9, 381–386.
Minh, B.Q., Schmidt, H.A., Chernomor, O., Schrempf, D., Woodhams, M.D., von Haeseler, A., et al., 2020. IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era. Mol Biol Evol. 37, 1530–1534.
Penny, D., Hendy, M.D., Steel, M.A., 1992. Progress with methods for constructing evolutionary trees. Trends Ecol. Evol. 7 (3), 73–79.
Pinheiro, D., Santander-Jiménéz, S., Ilic, A., 2022. PhyloMissForest: a random forest framework to construct phylogenetic trees with missing data. BMC Genomics 23 (1), 1–21.
Rao, Calyampudi Radakrishna, 1945. Information and the accuracy attainable in the estimation of statistical parameters. Bulletin of the Calcutta Mathematical Society. 37: 81–89. MR 0015748.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. https://doi.org/10.48550/arXiv.1505.04597.
Ronquist, F., Huelsenbeck, J.P., 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19 (12), 1572–1574. https://doi.org/10.1093/bioinformatics/btg180.
Saitou, N., Nei, M., 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4 (4), 406–425.
Schwarz, G., 1978. Estimating the Dimension of a Model. Ann. Stat. 6, 461–464. https://doi.org/10.1214/aos/1176344136.
Sneath, P.H., Sokal, R.R., 1973. Unweighted pair group method with arithmetic mean. Numerical Taxonomy 230–234.
Strimmer, K., Von Haeseler, A., 1996. Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. Mol. Biol. Evol. 13 (7), 964–969.
Strimmer, K., von Haeseler, A., 1997. Likelihood-mapping: A simple method to visualize phylogenetic content of a sequence alignment. Proc Natl Acad Sci USA 94, 6815–6819.
Stuart, A., Ord, K., Arnold, S., 1999. Kendall's Advanced Theory of Statistics. Arnold, London.
Suvorov, A., Hochuli, J., Schrider, D.R., 2020. Accurate Inference of Tree Topologies from Multiple Sequence Alignments Using Deep Learning. Syst. Biol. 69, 221–233. https://doi.org/10.1093/sysbio/syz060.
Suvorov, A., Schrider, D.R., 2024. Reliable estimation of tree branch lengths using deep neural networks. PLOS Computational Biology 20 (8). https://doi.org/10.1371/journal.pcbi.1012337.
Tavaré, S., 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. Lectures on Mathematics in the Life Sciences 17 (2), 57–86.
Truszkowski, J., Goldman, N., 2016. Maximum Likelihood Phylogenetic Inference is Consistent on Multiple Sequence Alignments, with or without Gaps. Syst Biol 65, 328–333. https://doi.org/10.1093/sysbio/syv089.
Wager, S., Wang, S., Liang, P.S., 2013. Dropout training as adaptive regularization. In: Advances in Neural Information Processing Systems, p. 26.
Waskom, M. et al., 2017. Seaborn, https://github.com/mwaskom/seaborn.
Whelan, S., Goldman, N., 2001. A General Empirical Model of Protein Evolution Derived from Multiple Protein Families Using a Maximum-Likelihood Approach. Mol. Biol. Evol. 18 (5), 691–699. https://doi.org/10.1093/oxfordjournals.molbev.a003851.
Wilson, E.B., 1927. Probable inference, the law of succession, and statistical inference. J. Am. Stat. Assoc. 22 (158), 209–212.
Xia, X., 2018. Maximum Likelihood in Molecular Phylogenetics. In: Bioinformatics and the Cell, pp. 381–395. https://doi.org/10.1007/978-3-319-90684-3_16.
Yang, Z., 2014. Molecular evolution: a statistical approach. Oxford University Press.
Zhu, T., Cai, Y., 2021. Applying Neural Network to Reconstruction of Phylogenetic Tree. In: 2021 13th International Conference on Machine Learning and Computing, pp. 146–152.
Zou, Z., Zhang, H., Guan, Y., Zhang, J., 2020. Deep residual neural networks resolve quartet molecular phylogenies. Mol. Biol. Evol. 37 (5), 1495–1507.