Harnessing machine learning to boost heuristic strategies for phylogenetic-tree search

Dana Azouri
Tel Aviv University https://orcid.org/0000-0003-0620-2626
Shiran Abadi
Tel Aviv University https://orcid.org/0000-0002-3932-6310
Yishay Mansour
Tel-Aviv University
Itay Mayrose
Tel Aviv University
Tal Pupko (talp@tauex.tau.ac.il)
Tel Aviv University https://orcid.org/0000-0001-9463-2575

Article

Keywords: phylogenetic tree, evolutionary studies, machine-learning approaches, heuristic tree searches, tree search methodology

Posted Date: August 6th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-48247/v1

Version of Record: A version of this preprint was published at Nature Communications on March 31st, 2021. See the published version at https://doi.org/10.1038/s41467-021-22073-8.

Abstract

Inferring a phylogenetic tree, which describes the evolutionary relationships among a set of organisms, genes, or genomes, is a fundamental step in numerous evolutionary studies. With the aim of making tree inference feasible for problems involving more than a handful of sequences, current algorithms for phylogenetic tree reconstruction utilize various heuristic approaches. Such approaches rely on performing costly likelihood optimizations, and thus evaluate only a subset of all potential trees. Consequently, all existing methods suffer from the known tradeoff between accuracy and running time. Here, we train a machine-learning algorithm over an extensive cohort of empirical data to predict the neighboring trees that increase the likelihood, without actually computing their likelihood. This provides means to safely discard a large set of the search space, thus avoiding numerous expensive likelihood computations. Our analyses suggest that machine-learning approaches can make heuristic tree searches substantially faster without losing accuracy and thus could be incorporated for narrowing down the examined neighboring trees of each intermediate tree in any tree search methodology.

Introduction

One of the most fundamental goals in biology is to reconstruct the evolutionary history of all organisms on earth. The obtained phylogeny is of interest to many downstream analyses concerning evolutionary and genomics research. For example, by using extant sequences and the phylogenetic relationships among them, it is possible to infer the most plausible ancestral sequences from which they were derived, to infer the divergence dates of various lineages, and to infer the selection regime characterizing genes and genomes. The genomic revolution driven by next generation sequencing methodologies has revolutionized the field, by providing vast amounts of genomics and metagenomics sequence data. Thus, while until recently most studies focused on a few to several dozens of sequences, current phylogenomic studies analyze longer sequences (up to entire genomes) and include a greater diversity (hundreds and even thousands of lineages), consequently challenging the ability of computational resources to handle these amounts of data.

Current approaches for phylogeny reconstruction rely on probabilistic evolutionary models that describe the stochastic processes of nucleotide, amino-acid, and codon substitutions[1]. Given an evolutionary model, a tree topology with its associated branch lengths, and a multiple sequence alignment, the likelihood of the data is efficiently computed using Felsenstein's pruning algorithm[2]. While the alignment is usually assumed to be known, parameters of the evolutionary model, the tree topology and its associated branch lengths are usually inferred by maximizing the likelihood of the data. Thus, for a specific evolutionary model with fixed parameter values, each tree inference algorithm visits a large number of candidate tree topologies and for each such a topology it searches for the optimal set of branch lengths. Notably, the number of possible tree topologies increases super-exponentially with the number of sequences. Moreover, the computational search for the best tree topology was shown to be NP-hard, both when using the maximum-parsimony approach and when the more advanced method of maximum likelihood was utilized[3]. Optimizing the set of branch lengths for each candidate tree is

computationally intensive, adding another layer of complexity to this endeavor. Thus, all current algorithms for phylogenetic tree reconstruction use various heuristics to make tree inference feasible.

The general approach for a heuristic search is to begin either with a random starting tree or a starting tree obtained by rapid and often inaccurate methods such as Neighbor Joining[4]. The score of this initial tree is its log-likelihood, which is based on the specified probabilistic model. Next, a set of alternative topologies is considered, each of which is a small modification of the current tree topology (each such topology is considered to be a "neighbor" of the current topology). The neighbor with the highest score is selected and used as an initial tree for the next step. The process proceeds iteratively until none of the alternative trees produces a higher score compared to the current one. Various algorithms differ in their definition of a neighbor. In this study, we focus on subtree pruning and regrafting (SPR)[5]. An SPR neighbor is obtained by pruning a subtree from the main tree and regrafting it to the remaining tree, as illustrated in Fig. 1. Several improvements to the basic heuristic scheme described above have been suggested. These improvements include better exploration of the tree space and introduction of shortcuts in order to substantially reduce running time with little to no influence on inference accuracy. Notable examples include: (1) proceeding with the first neighbor that improves the likelihood score without examining the remaining neighbors[6]; (2) avoiding optimization of the entire branch lengths by optimizing only those in the vicinity of the regrafted subtree[6]; (3) discarding neighbors whose estimated sum of branch lengths highly deviates from that of the current tree[7]; (4) genetic algorithms and simulated annealing versions of the heuristic search[8,9]. In addition, a common practice is to apply the bootstrap procedure that provides a measure of confidence for each split in the obtained tree. This is done by executing the tree search on bootstrapped data at least 100 times. This time-consuming step further emphasizes the need for an efficient heuristic[3,10]. To date, machine-learning tools have not been employed for enhancing the heuristic tree search.

In this study we use a diverse set of thousands of empirical datasets to train a supervised machine-learning regression model, specifically a random forest learning algorithm, in order to predict the optimal move for a single step in a phylogenetic tree search. The output of this learner, trained on a succinct collection of twenty features, is a numerical value for each possible SPR move that represents its propensity to be the highest-scoring neighbor. Our results show that this procedure yields very high agreement between the true and inferred ranking, indicating the high predictive power of the developed machine-learning framework. Furthermore, we demonstrate that by using the learning framework it is sufficient to evaluate the costly likelihood score for a small subset of all possible neighbors. This study thus establishes a comprehensive proof-of-concept that methodologies based on artificial intelligence can substantially accelerate tree search algorithms without sacrificing accuracy.

Results

A machine-learning algorithm for accelerating the maximum-likelihood tree search

Our goal is to rank all possible SPR neighbors of a given tree according to their log-likelihood without actually computing the likelihood function. To this end, we rely on a set of features that capture essential information regarding an SPR tree rearrangement and that can be efficiently computed. Specifically, we trained a random forest for regression machine-learning algorithm to predict the ranking of all possible SPR modifications, according to their effect on the log-likelihood score. The algorithm was trained on a large set of known examples (data points). In our case, each data point is a pair (V, L). V is an array which includes the starting tree, the resulting tree following an SPR move, and the set of features, while L is a function of the log-likelihood difference between the starting and the resulting tree (see Methods). The regression model learns the association between V and L. Given a trained algorithm and a starting tree topology, an array V is computed for each possible SPR move. The trained machine-learning algorithm provides the ranking of all possible SPR moves according to their predicted L values. A perfect machine-learning model would predict the optimal SPR neighbor and would thus eliminate the need for expensive likelihood computations. A sub-optimal predictor may also be highly valuable if the vast majority of the SPR moves can be safely discarded without computing their likelihoods.

The machine-learning algorithm was trained on 16,991,941 data points, one data point for each possible SPR move of 4,300 different empirical phylogenies. The empirical alignments varied in terms of the number of sequences (7 to 70), the number of positions (62 to 50,000) and the extent of sequence divergence. The number of neighbors of each tree is affected by the number of sequences and by the tree topology and ranges between a few dozens to over ten thousand. We chose to analyze empirical rather than simulated data, as it is known that reconstructing the best tree is more challenging for the former[11–13]. The learning was based on 20 features, extracted from each data point (Table 1). The first eight features were extracted from the starting and resulting trees, e.g., the lengths of the branches in the pruning and regrafting locations. The remaining features were generated based on the subtrees that were induced by the SPR move, e.g., the sum of branch lengths of the pruned subtree (Fig. 1).

Performance evaluation

We evaluated the performance of our trained learner in a ten-fold cross-validation procedure. Namely, the empirical datasets were divided to ten subsets, such that in each of the ten training iterations, the induced data points of nine folds were used for training the model, and the remaining data points were used for testing. We first evaluated the accuracy of the learner in ranking alternative SPR moves. The Spearman rank correlation coefficient (ρ) was thus computed between the true ranking, inferred through a full likelihood-optimization, and the predicted ranking, based on the machine-learning predictions. The mean ρ, averaged over all 4,300 samples was 0.93 (Fig. 2a), suggesting that the machine-learning algorithm successfully discriminates between beneficial and unfavorable SPR moves.

Notably, the Spearman correlation quantifies the prediction performance when all SPR neighbors are considered. However, in a typical hill-climbing heuristic, the single best SPR neighbor is chosen as the starting tree for the next step. It is thus interesting to estimate the ability of the algorithm to predict this

best neighbor. Accordingly, we measured the performance of the trained algorithm by two additional metrics: (1) the rank of this best move in the predicted ranking; (2) the rank of the predicted best move within the true rank, as obtained according to the full likelihood optimization. Our results indicated that in 63% and 91% of the datasets the best move was among the top 10% and 25% predictions (Fig. 2b). In 89% and 97% of the datasets, the top-ranked prediction was among the top 10% and 25% SPR moves (Fig. 2c). Moreover, in 99.99% of the cases, the top prediction resulted in higher likelihood compared to the starting tree, suggesting that an improvement is typically obtained. In contrast, a random move increased the likelihood score in only 3.7% of the datasets. These results strengthen the evident potential of the machine-learning algorithm to direct the tree search to a narrow region of the tree space, thus avoiding numerous expensive likelihood calculations.

We next evaluated the trained model on entirely different datasets than the training data (see Methods). Unlike the training data, the machine-learning algorithm was not optimized on these data, not even in cross-validation, thus negating possible overfitting effects. When applied to these validation data, the performance of the trained model was very similar to that reported above using cross validation ($\rho = 0.91$; Supplementary Fig. 1), suggesting that the machine-learning algorithm is well generalized for various datasets, evolved under an array of evolutionary scenarios.

To gain further insight into factors affecting the prediction accuracy, we analyzed whether the predictions accuracy is affected by: (1) the number of taxa; (2) the level of divergence as measured by the sum of branch lengths; (3) the six databases used (four for training and two for validation); (4) the amount of the training data. No significant correlation was observed between ρ and the number of taxa or between ρ and the level of sequence divergence (Supplementary Figures 2a and 2b). Among the six databases, predictions were most accurate for Selectome, with mean ρ of 0.95, and least accurate for ProtDBs, with mean ρ of 0.83 (Supplementary Fig. 2c). Finally, our results imply that increasing the number of trained samples above 4,300 does not significantly increase the accuracy (P value > 0.24 for ANOVA test comparing 4,300 datasets to 6,000; Supplementary Fig. 3).

Performance evaluation on an example dataset: the SEMG2 gene in primates

We exemplify the application of the machine-learning algorithm on a specific dataset, consisting of 28 primate sequences encoding the SEMG2 reproductive gene (see Methods). We reconstructed a starting tree (Fig. 3a), generated all its 2,345 SPR neighbors, and ranked them according to their log-likelihoods. We then compared this ranking to the ranking predicted by the trained machine-learning algorithm. The Spearman rank correlation (ρ) between the true and the predicted ranking was 0.91, which is similar to the average ρ reported for both the training and validation data. Indeed, the best move was among the top 6% predictions, and the best SPR move predicted by the model was the second best possible move. Furthermore, the best SPR move and the predicted best SPR move led to rearrangements in the same clade of the phylogenetic tree, containing nine species (Fig. 3).

As explained in the Methods section, our algorithm ranks neighboring trees by predicting their log-likelihoods. While the ultimate goal is to predict the ranking of the possible SPR moves in order to limit the search space, focusing on one example enables the inspection of the actual predicted change in log-likelihood between each potential resulting tree and the starting tree. For this example, a Pearson correlation (R^2) of 0.88 between the predicted and true change in log-likelihood was observed (the full list of the predicted and true log-likelihood differences for all 2,345 single-step SPR moves is given in Supplementary Data 1). The predicted best move improved the log-likelihood of the initial tree by 22.8, similar to a log-likelihood improvement of 23.4 obtained by the best SPR move. Moreover, according to our model, 97 and 2,248 SPR moves were predicted to increase and decrease the log-likelihood, respectively, and these predictions were true for 66% and 92% of these cases. These results corroborate the potential of the machine-learning approach to correctly discard many irrelevant SPR neighbors.

In addition, we measured the running time for evaluating the 2,345 neighboring trees for this example. The computation of the features and the application of the trained model for each neighbor took $1.1x10^{-3}$ seconds on average. The likelihood computation took 40.49 seconds on average for each neighbor, roughly 35,000 times longer compared to the machine-learning algorithm.

We next examined whether the high performance of the trained model is maintained when applied to other intermediate trees in the chain towards the maximum-likelihood tree. When applied to the second phase of the search, i.e., starting from the best possible neighbor of the initial tree, the trained model yielded results that are highly similar to those reported for the initial tree (Spearman correlation coefficient of $\rho = 0.91$). Prominently, the best move according to the predictions increased the true log-likelihood score by 0.039, implying that the likelihood improvement is maintained following additional SPR steps. Finally, we examined the performance of the algorithm when the initial tree is one step away from the maximum-likelihood tree. To this end, we selected a tree that is a neighbor of the maximum-likelihood tree by applying a random SPR move to the latter. When applied to this tree, the model predicted the maximum-likelihood tree as the best possible neighbor, with a log-likelihood improvement of 0.007 ($p = 0.92$).

Feature importance

The prediction capabilities of the random forest algorithm are determined by the provided features. In turn, the relative contribution of each of these features to the predictive model, termed ‘the feature importance’, can be extracted. In our implementation, the feature that contributed most to the prediction accuracy was the sum of branch lengths along the path between the pruning and the regrafting locations, while the second best feature was the number of nodes along that path (for the importance values of all features, see Supplementary Table 1). These findings provide some justification for the common practice of considering only local changes in various tree search heuristics[6,8,14]. The next two features were the length of the pruned branch and the longest branch in the pruned subtree. The fifth feature was the

estimated sum of branch lengths of the resulting tree, a feature that was previously suggested as a single filtering criterion by Hordijk and Gascuel[7].

Many common tree search heuristics utilize a single feature to limit the scope of inspected neighbors. We thus exploited the devised framework to examine whether the use of a single feature leads to similar performance. To this end, we trained 20 random forest models on the training set, such that each model accounted for a single feature. The performance of each of these models provided a measure of the predictive power of each feature, independent of the others. The best single-feature model obtained a Spearman correlation coefficient of $\rho = 0.706$ on average across the training set, and was based on the number of nodes in the path between the pruning and the regrafting locations, a feature that was ranked second when the entire set of features was used for training (Supplementary Table 1). The average ρ for each of the remaining features was below 0.47 (Supplementary Table 2). These observations, together with the substantial increase in average ρ when comparing the usage of a single feature to using the entire set of features, combined (average ρ of 0.93), highlights the benefit of relying on a large set of features that together provide more informative prediction.

Discussion

Inferring a phylogenetic tree is of central importance in numerous evolutionary studies. As follows, methods for tree reconstruction are widely used by the biological research community. Still, since such methods incur complex computations, all existing methods attempt to reduce running time at the expense of accuracy, being dependent on heuristics to overcome the feasibility problem. Here we developed a machine-learning framework, trained to rank neighboring trees according to their propensity to increase the likelihood. The evident high predictive power of this framework demonstrates that the computationally-intensive step of likelihood evaluation can be limited to a small set of potential neighbors, substantially reducing the running time without jeopardizing accuracy. By boosting tree inference, our study directly impacts efforts of downstream analyses, such as molecular dating[15], inference of positive selection[16], protein fold recognition[17], identification of functionally divergent protein residue[18], recombination detection[19], and ancestral sequence reconstruction[20]. Furthermore, our research could grant the development of richer and more realistic substitution models, which are currently too computationally intensive to be considered within a tree-search procedure (e.g., a covarion model[21] for codon characters).

Ranking of neighboring trees to speed up the tree search was previously suggested, albeit with the use of a single attribute and without learning from large training data. For example, Hordijk and Gascuel[7] proposed testing only neighbors for which their estimated total sum of branch lengths does not substantially differ from the starting tree. Our methodology advances over previous approaches, as we use multiple features instead of one, and utilize machine learning to optimally combine these features based on extensive training. Notably, recent studies suggested the use of deep neural networks to infer unrooted four-taxa topologies from multiple sequence alignments[22,23]. Suvorov et al.[22] and Zou et al.[23]

utilized residual and convolutional neural networks, respectively, to infer unrooted four-taxa topologies from multiple sequence alignments. While their devised method performs well, it can currently be applied to infer topologies of four taxa only. In addition,S in order to reconstruct the true generating topology, Suvorov et al. were required to rely on simulated datasets, which were previously shown to be easier to interpret and infer[11–13]. The objective of our study, narrowing the search space in a single step towards a final, faster, convergence of the maximum likelihood, enabled us to rely on empirical datasets for training and testing.

How can our machine-learning algorithm be used in practice? One trivial application would be to start evaluating the log-likelihoods of the neighboring trees, starting from the top-ranked predicted neighbor. If this neighbor obtains a log-likelihood score that is higher than the starting tree, proceed with that tree as the starting tree, iteratively repeating this procedure. If this neighbor obtains a log-likelihood score that is lower than the starting tree, evaluate the next ranked neighbor. End the iterative chain of tree search when no improvement is obtained. Clearly, more sophisticated tree search schemes can be considered. For example, one could progress a few steps, based on the best predictions only, without evaluating the likelihoods, expecting the obtained tree to have higher log-likelihood compared to the starting tree. Furthermore, our approach can be integrated within existing maximum-likelihood or Bayesian frameworks, which are already implemented in the leading tree search algorithms, such as RevBayes[24], RAxML[25], PhyML[26] and IQtree[27]. For example, in IQtree a set of trees is kept and the algorithm samples from this set. Such an approach to sample within a subset of more likely neighbors can easily be combined with our machine-learning approach that allows sampling the most promising trees while rapidly traversing large regions of the tree space. Further developments of the proposed methodology towards a complete search are possible. For example, we have not put effort in assessing the branch lengths associated with the inferred topology. It is also interesting to study how our approach generalizes to more complex models of evolution, such as amino-acid codon models and partition models[28,29]. In addition, our algorithm was implemented using SPR moves only. The benefit of using additional types of tree rearrangement moves, such as nearest neighbor interchange (NNI)[30,31] and tree bisection and regrafting (TBR)[32] should be evaluated.

To conclude, we provide a methodology that can substantially accelerate tree-search algorithms without sacrificing accuracy. We believe that harnessing artificial intelligence to the task of phylogenomics inference has the potential to substantially increase the scale of the analyzed datasets and, potentially, the level of sophistication of the underlying evolutionary models.

Methods

Empirical and validation data

We assembled a training data composed of 4,300 empirical alignments from several databases: 3,810 from TreeBase[33], 233 from Selectome[34], 78 from protDB[35] and 179 from PlantDB[36]. TreeBase is a

repository of user-submitted phylogenies; Selectome database includes codon alignments of species within four groups (Euteleostomi, Primates, Glires, and Drosophila); protDB includes genomic sequences that were aligned according to the tertiary structure alignments of the encoded proteins published in BALIBASE[37]; and PlantDB contains alignments with sequences belonging to a single plant genus and a potential outgroup. We randomly selected datasets with 7 to 70 sequences and more than 50 sites, excluding alignments with a majority of gapped or missing sites, and alignments that contained sequences that are entirely composed of gapped or missing characters.

To test the predictive power of our model also over unseen validation data that were neither used for training our model nor for cross validation, we gathered a database encompassing 1,000 multiple sequence alignments, collected from two databases that were not used to generate the training set: 500 datasets from PANDIT[38], which includes alignments of coding sequences, and 500 datasets from OrthoMaM[39], a database of orthologous mammalian markers.

Example dataset

The example dataset was composed of 28 primates coding sequences of the Semenogelin-2 (SEMG2) protein, obtained from NCBI Nucleotide and aligned using revTrans[40], following procedures detailed in Halabi et al.[41]. Sites with more than 90% gaps were removed from the alignment.

Starting trees reconstruction, SPR neighbors generation, and likelihood estimation

The starting tree for each alignment was reconstructed using the BioNJ[42] tree reconstruction as implemented in PhyML 3.0[26], assuming the GTR+I+G model. For each starting tree, we computed all SPR neighbors, i.e., all trees that can be obtained by pruning and regrafting any pair of the starting tree branches. Next, we used PhyML to optimize the branch lengths and the substitution rate parameters for the starting trees and each of their SPR neighbors. We recorded the log-likelihoods of the optimized trees assuming the GTR+I+G model.

A machine-learning algorithm for ranking neighboring trees

Random forest for regression, as implemented in python scikit-learn module[43], was applied using 70 decision trees. In each split of the tree, a random subset of one third of the total number of features was considered. The target value of the machine-learning training was computed as $target = \frac{LL_{neighbor} - LL_{starting\ tree}}{LL_{starting\ tree}}$, namely, the log-likelihood difference between the neighbor and its starting tree, divided by the log-likelihood of the starting tree. Notably, these ratios are log distributed across the training set and may lead to unbalanced decision trees in the random-forest training. Therefore, the training outcomes were transformed according

to $f(target) = 2^{target+1}$ to generate a distribution that is more uniform (Supplementary Fig. 4). The same function was applied to the predicted values accordingly.

The learning scheme we implemented in this study is a random forest regression algorithm. This model was chosen over four other alternative supervised-machine-learning regression algorithms we implemented, as it outperformed all others: Support vector machine, Bayesian Ridge, Lasso, and K-Nearest-Neighbors (Supplementary Table 3).

Predictive features

The learning was based on extracting 20 features from each data point (Table 1). The computation of all features was implemented in Python and required $O(n^2 \log n)$ operations for all the pruning and regrafting locations of a single tree, n being the number of sequences (see Supplementary information for feature extraction details). The first eight features were extracted from the starting and resulting trees (Fig. 1a, d; Table 1 features 1-8). The remaining features rely on the following definition of four intermediate subtrees: the two subtrees induced by splitting the starting tree at the pruning location and the two subtrees induced by splitting the starting tree at the regrafting location (Fig. 1). For each of these four subtrees we calculated five features, resulting in a total of twelve features (Table 1; features 9-20).

To examine whether the feature set could be reduced to enhance computational performance, we applied a backward stepwise elimination procedure[44]. To this end, we began with the full set of 20 features. We then removed the feature with the minimal importance score and trained the random forest algorithm for the remaining features, to compute the ρ metric. We repeated this procedure, successively eliminating an additional feature with the minimal importance score. The best ρ value was obtained when all the features were included, and therefore the results are presented for the full set of features.

Data availability

The datasets contained within the empirical set have been deposited in Open Source Framework (OSF) with the identifier DOI 10.17605/OSF.IO/B8AQJ[45].

Declarations

Acknowledgements

D.A. is supported by fellowships from The Council for Higher Education program for excellent PhD students in Data Sciences and the Fast & Direct Ph.D Program by the Argentinean Friends of TAU. S.A. was supported by the Rothchild Caesarea Foundation and by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel Aviv University. Y.M. was supported in part by a grant of the Israel Science

Foundation (ISF) 993/17. I.M. was supported by an Israel Science Foundation grant 961/17. T.P. was supported by an Israel Science Foundation grant 802/16.S

Author Contributions

D.A., S.A., Y.M., I.M. and T.P. designed the study, helped in interpreting the results, and provided inputs on the draft. D.A. performed the analyses and prepared the manuscript. I.M. and T.A. supervised this work and revised the manuscript.

Competing interests

The authors declare no competing interests.

References

1. Thorne, J. L. Models of protein sequence evolution and their applications. *Current Opinion in Genetics and Development* 10, 602–605 (2000).
2. Felsenstein, J. Evolutionary trees from DNA sequences: A maximum likelihood approach. *J. Mol. Evol.* 17, 368–376 (1981).
3. Chor, B. & Tuller, T. Maximum likelihood of evolutionary trees: Hardness and approximation. *Bioinformatics* 21, i97-106 (2005).
4. Saitou, N. & Nei, M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* 4, 406–425 (1987).
5. Felsenstein, J. Inferring phylogenies. *Am. J. Hum. Genet.* 74, 1074 (2004).
6. Stamatakis, A. P., Ludwig, T. & Meier, H. A fast program for maximum likelihood-based inference of large phylogenetic trees. in *Proceedings of the ACM Symposium on Applied Computing* 1, 197–201 (2004).
7. Hordijk, W. & Gascuel, O. Improving the efficiency of SPR moves in phylogenetic tree search methods based on maximum likelihood. *Bioinformatics* 21, 4338–4347 (2005).
8. Stamatakis, A. An efficient program for phylogenetic inference using simulated annealing. in *Proceedings - 19th IEEE International Parallel and Distributed Processing Symposium* (2005).
9. Helaers, R. & Milinkovitch, M. C. MetaPIGA v2.0: maximum likelihood large phylogeny estimation using the metapopulation genetic algorithm and other stochastic heuristics. *BMC Bioinformatics* 11, 379 (2010).
10. Stamatakis, A., Hoover, P. & Rougemont, J. A rapid bootstrap algorithm for the RAxML web servers. *Syst. Biol.* 57, 758–771 (2008).
11. Abadi, S., Azouri, D., Pupko, T. & Mayrose, I. Model selection may not be a mandatory step for phylogeny reconstruction. *Nat. Commun.* 10, 934 (2019).

12. Huelsenbeck, J. P. Performance of phylogenetic methods in simulation. *Syst. Biol.* 44, 17–48 (1995).
13. Edwards, A. W. F., Nei, M., Takezaki, N. & Sitnikova, T. Assessing molecular phylogenies. *Science* 267, 253–255 (1995).
14. Stewart, C. A. *et al.* Parallel implementation and performance of fastdnaml-a program for maximum likelihood phylogenetic inference. in *Supercomputing, ACM/IEEE Conference* 32 (2001).
15. Lartillot, N., Lepage, T. & Blanquart, S. PhyloBayes 3: A Bayesian software package for phylogenetic reconstruction and molecular dating. *Bioinformatics* 25, 2286–2288 (2009).
16. Nielsen, R. & Yang, Z. Likelihood models for detecting positively selected amino acid sites and applications to the HIV-1 envelope gene. *Genetics* 148, 929–936 (1998).
17. Choi, S. C., Hobolth, A., Robinson, D. M., Kishino, H. & Thorne, J. L. Quantifying the impact of protein tertiary structure on molecular evolution. *Mol. Biol. Evol.* 24, 1769–1782 (2007).
18. Gaston, D., Susko, E. & Roger, A. J. A phylogenetic mixture model for the identification of functionally divergent protein residues. *Bioinformatics* 27, 2655–2663 (2011).
19. Pond, S. L. K., Posada, D., Gravenor, M. B., Woelk, C. H. & Frost, S. D. W. Automated phylogenetic detection of recombination using a genetic algorithm. *Mol. Biol. Evol.* 23, 1891–1901 (2006).
20. Ashkenazy, H. *et al.* FastML: A web server for probabilistic reconstruction of ancestral sequences. *Nucleic Acids Res.* 40, W580-584 (2012).
21. Galtier, N. Maximum-likelihood phylogenetic analysis under a covarion-like model. *Mol. Biol. Evol.* 18, 866–873 (2001).
22. Suvorov, A., Hochuli, J. & Schrider, D. R. Accurate Inference of Tree Topologies from Multiple Sequence Alignments Using Deep Learning. *Syst. Biol.* 69, 221–233 (2020).
23. Zou, Z., Zhang, H., Guan, Y., Zhang, J. & Liu, L. Deep residual neural networks resolve quartet molecular phylogenies. *Mol. Biol. Evol.* 37, 1495–1507 (2020).
24. Hohna, S. *et al.* RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Syst. Biol.* 65, 726–736 (2016).
25. Stamatakis, A. RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* 30, 1312–1313 (2014).
26. Guindon, S. *et al.* New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst. Biol.* 59, 307–321 (2010).
27. Nguyen, L. T., Schmidt, H. A., Von Haeseler, A. & Minh, B. Q. IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Mol. Biol. Evol.* 32, 268–274 (2015).
28. Yang, Z., Nielsen, R., Goldman, N. & Krabbe Pedersen, A.-M. Codon-substitution models for heterogeneous selection pressure at amino acid sites. *Genetics* 155, 431-449 (2000).
29. Marshall, D. C. Cryptic failure of partitioned bayesian phylogenetic analyses: Lost in the land of long trees. *Syst. Biol.* 59, 108–117 (2010).
30. Robinson, D. F. Comparison of labeled trees with valency three. *J. Comb. Theory, Ser. B* 11, 105–119 (1971).

31. Moore, G. W., Goodman, M. & Barnabas, J. An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets. *J. Theor. Biol.* 38, 423–457 (1973).
32. Allen, B. L. & Steel, M. Subtree Transfer operations and their Induced metrics on evolutionary trees. *Ann. Comb.* 5, 1–15 (2001).
33. Piel, W. H. *et al.* TreeBASE v. 2: A Database of phylogenetic knowledge. in *e-BioSphere* (2009).
34. Moretti, S. *et al.* Selectome update: Quality control and computational improvements to a database of positive selection. *Nucleic Acids Res.* 42, (2014).
35. Carroll, H. *et al.* DNA reference alignment benchmarks based on tertiary structure of encoded proteins. *Bioinformatics* 23, 2648–2649 (2007).
36. Glick, L., Sabath, N., Ashman, T.-L., Goldberg, E. & Mayrose, I. Polyploidy and sexual system in angiosperms: Is there an association? *Am. J. Bot.* 103, 1223–1235 (2016).
37. Thompson, J. D., Koehl, P., Ripp, R. & Poch, O. BAliBASE 3.0: Latest developments of the multiple sequence alignment benchmark. *Proteins Struct. Funct. Genet.* 61, 127–136 (2005).
38. Whelan, S., de Bakker, P. I. W. & Goldman, N. Pandit: A database of protein and associated nucleotide domains with inferred trees. *Bioinformatics* 19, 1556–1563 (2003).
39. Ranwez, V. *et al.* OrthoMaM: A database of orthologous genomic markers for placental mammal phylogenetics. *BMC Evol. Biol.* 7, 241 (2007).
40. Wernersson, R. & Pedersen, A. G. RevTrans: Multiple alignment of coding DNA from aligned amino acid sequences. *Nucleic Acids Res.* 31, 3537–3539 (2003).
41. Keren, H., Eli, L. K., Laurent, G., Mayrose, I. A codon model for associating phenotypic traits with altered selective patterns of sequence evolution. *bioRxiv* (2020).
42. Gascuel, O. BIONJ: An improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol.* 14, 685–695 (1997).
43. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830 (2011).
44. James, G., Witten, D., Hastie, T. & Tibshirani, R. An Introduction to Statistical Learning. *Springer Texts*. 102, (2013).
45. Azouri, D., Abadi, S., Mansour, Y., Mayrose, I., Pupko, T. Harnessing machine learning to boost heuristic strategies for phylogenetic-tree search. OSF. https://doi.org/10.17605/OSF.IO/B8AQJ. (2020).
46. Desper, R. & Gascuel, O. The minimum evolution distance-based approach to phylogenetic inference. *Math. Evol. Phylogeny* 1–32 (2005).

Table

Table 1. Features used in the machine-learning framework

# Feature	Feature name	Tree	The represented action	Details
1	Total branch lengths	Initial tree (a in Fig. 1)	Shared for pruning and regrafting	The sum of branches in the starting tree
2	Longest branch			The length of the longest branch in the starting tree
3-4	Branch length		Both pruning and regrafting	The length of the branch that was being pruned or regrafted
5	Topology distance from the pruned node		Regrafting only	The number of branches in the path between the regrafting and the pruning branches, not including these branches
6	Branch length distance from the pruned node			The sum of branches in the path between the regrafting and the pruning branches, not including these branches
7	New branch length	Resulting tree (c_1 in Fig. 1)	Regrafting only	The approximated length of the newly formed branch formed due to regrafting[46].
8	New total branch lengths			The estimated total branch lengths of the resulting tree[46].
9-12	Number of species	Each of the four subtrees (b, c, c_1, c_2 in Fig. 1)	Both pruning and regrafting	The number of leaves in the subtree
13-16	Total branch lengths			The sum of branches in the subtree
17-20	Longest branch			The length of the longest branch in the subtree

The table lists the 20 features on which the machine-learning algorithm is based, extracted for each data point. The features are grouped according to the subtree considered. Some features are independent of the move, e.g., features 1 and 2.

Figures

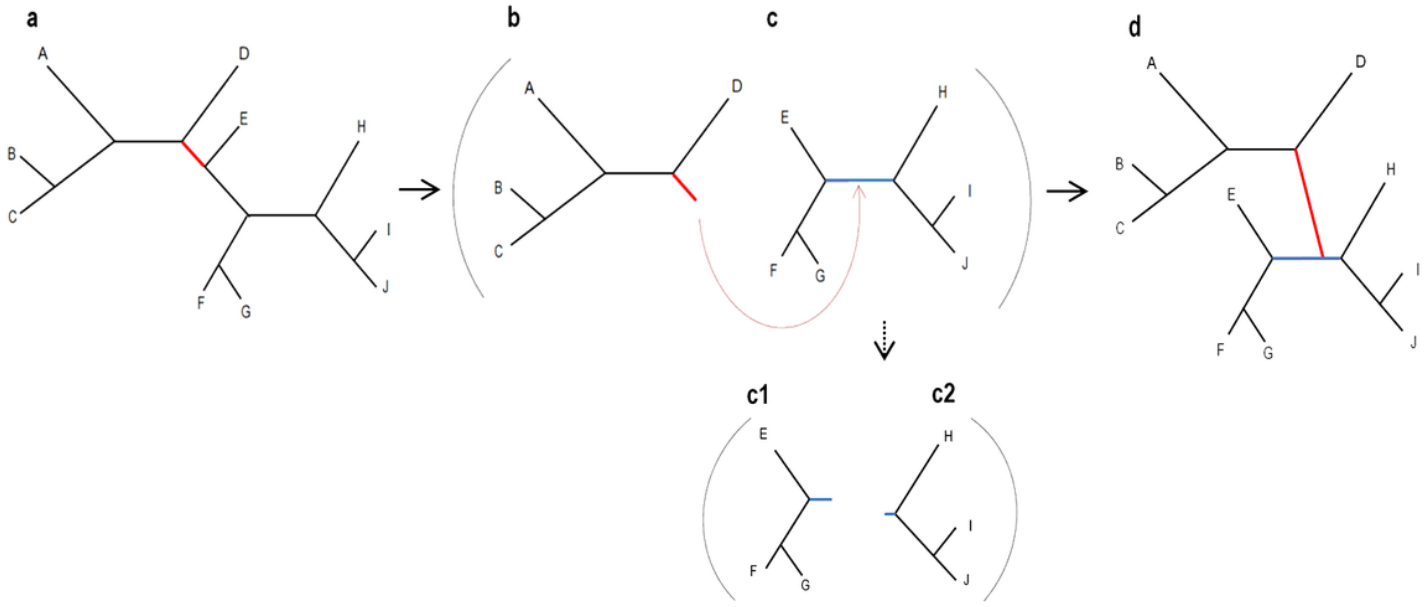

Figure 1

The trees defined by an SPR move. For each data sample two trees and four subtrees were considered: (a) an example of a starting tree; (b-c) the two subtrees induced by the pruned branch of tree a (in red), where (b) is the pruned subtree and (c) the remaining subtree; (c1-c2) the regrafted branch (in blue) also induces two subtrees, from both sides of the regrafted branch of subtree c; (d) the resulting tree following the SPR move.

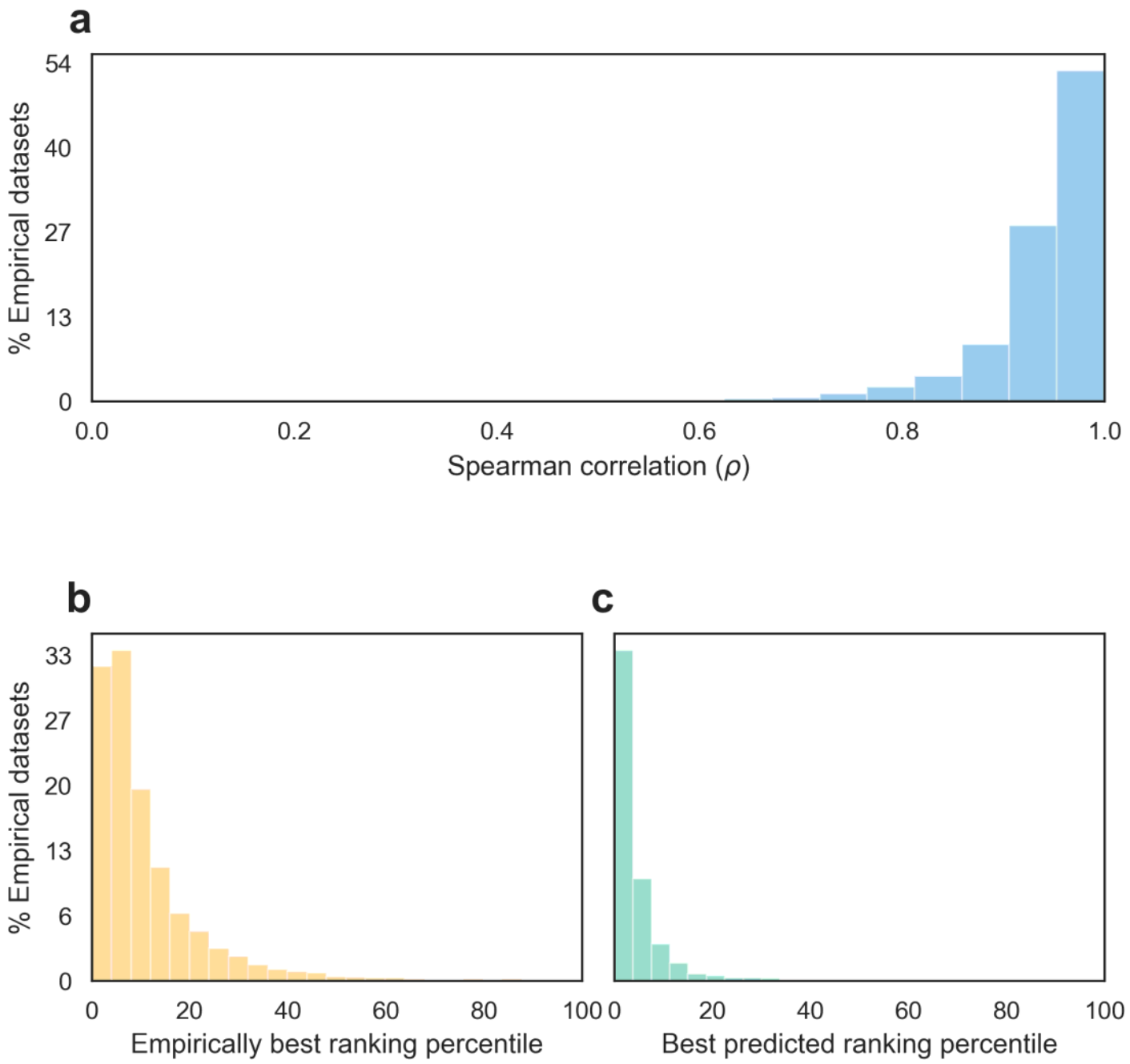

Figure 2

Performance evaluation scores on the empirical datasets. A histogram of the three performance scores of learning algorithm evaluated on 4,300 starting trees: (a) the Spearman correlation coefficient between the values of the experimentally and the predicted target values; (b) the predicted ranking percentile of the empirically best neighbor; (c) the empirical ranking percentile of the neighbor that was predicted to be the best. On the y axis are the proportions of the empirical datasets (in percentages).

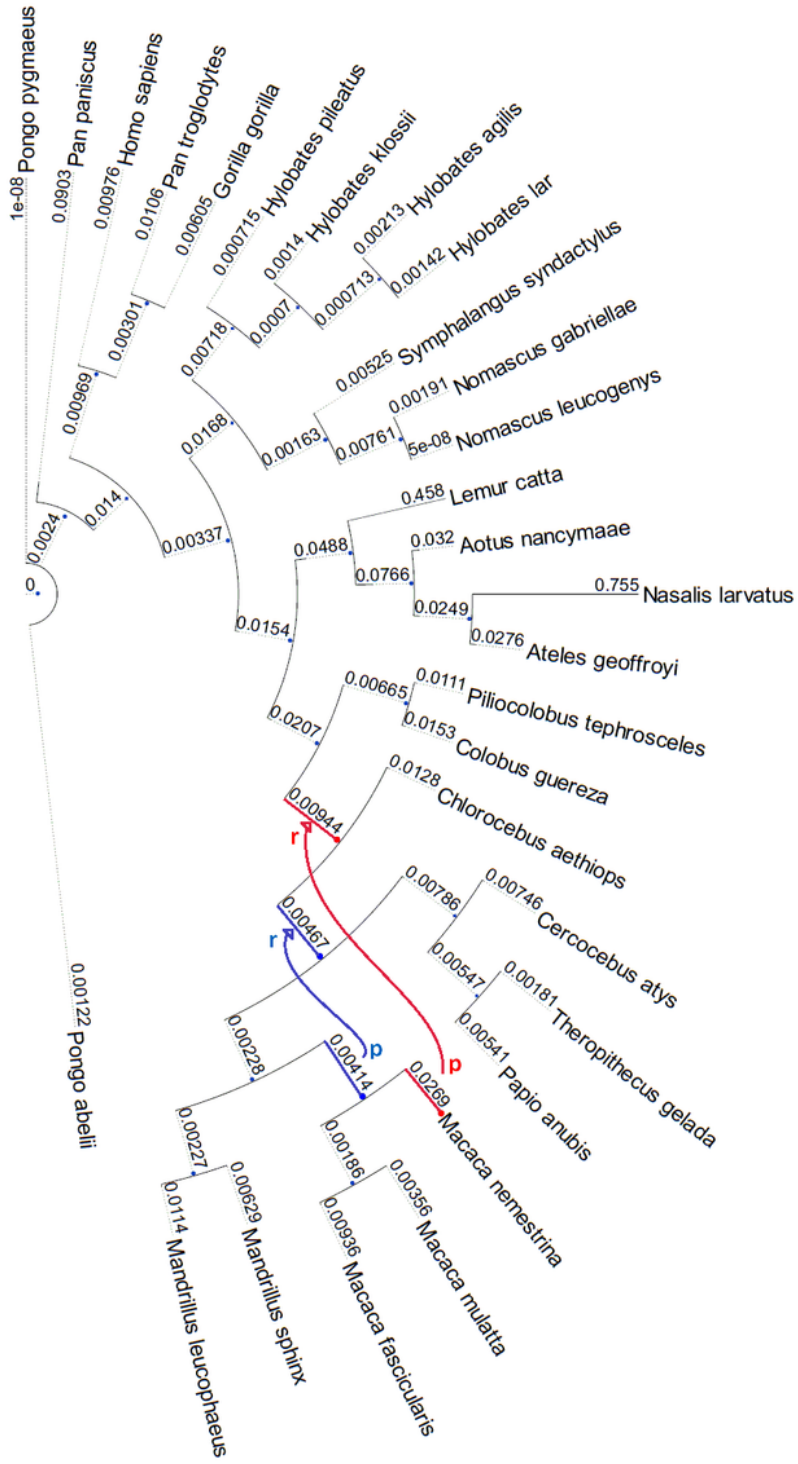

Figure 3

Performance evaluation when testing the trained model on an example of primate multiple sequence alignment. In red and blue are the best SPR and the predicted best SPR tree modifications, respectively. Marked with ‘p’ and ‘r’ are the specific pruning and regrafting locations, respectively.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryInformationAppendix.docx
- RS.pdf
- NCOMMS2029678codelink.pdf
- SupplementaryData1.xlsx