Molecular Biology and Evolution, 2025, **42**, 1–14
https://doi.org/10.1093/molbev/msaf051
Advance access publication 11 March 2025
Methods

MBE OXFORD

Phyloformer: Fast, Accurate, and Versatile Phylogenetic Reconstruction with Deep Neural Networks

Luca Nesterenko [1,†], **Luc Blassel** [1,†], **Philippe Veber** [1], **Bastien Boussau** [1,‡], **Laurent Jacob** [2,*,‡]

[1]Laboratoire de Biométrie et Biologie Évolutive, Université Lyon 1, Villeurbanne, France
[2]Laboratory of Computational and Quantitative Biology, Sorbonne Université, Paris, France

[†]These authors contributed equally to this work.
[‡]These authors contributed equally to this work.
***Corresponding author**: E-mail: laurent.jacob@cnrs.fr.
Associate editor: Tal Pupko

Abstract

Phylogenetic inference aims at reconstructing the tree describing the evolution of a set of sequences descending from a common ancestor. The high computational cost of state-of-the-art maximum likelihood and Bayesian inference methods limits their usability under realistic evolutionary models. Harnessing recent advances in likelihood-free inference and geometric deep learning, we introduce Phyloformer, a fast and accurate method for evolutionary distance estimation and phylogenetic reconstruction. Sampling many trees and sequences under an evolutionary model, we train the network to learn a function that enables predicting a tree from a multiple sequence alignment. On simulated data, we compare Phyloformer to FastME—a distance method—and two maximum likelihood methods: FastTree and IQTree. Under a commonly used model of protein sequence evolution and exploiting graphics processing unit (GPU) acceleration, Phyloformer outpaces all other approaches and exceeds their accuracy in the Kuhner–Felsenstein metric that accounts for both the topology and branch lengths. In terms of topological accuracy alone, Phyloformer outperforms FastME, but falls behind maximum likelihood approaches, especially as the number of sequences increases. When a model of sequence evolution that includes dependencies between sites is used, Phyloformer outperforms all other methods across all metrics on alignments with fewer than 80 sequences. On 3,801 empirical gene alignments from five different datasets, Phyloformer matches the topological accuracy of the two maximum likelihood implementations. Our results pave the way for the adoption of sophisticated realistic models for phylogenetic inference.

Keywords: phylogenetic reconstruction, neural network, attention, machine learning, regression

Introduction

Molecular phylogenies provide essential insights into evolutionary processes. They are employed in epidemiology to track viral spread (Hadfield et al. 2018), in virology to identify events of recombination (Nelson et al. 2008), in biochemistry to evaluate functional constraints operating on sequences (Harms and Thornton 2013), and in ecology to characterize biodiversity (Perez-Lamarque et al. 2022). Most of the time, molecular phylogenies are estimated from aligned nucleotide or amino acid sequences using probabilistic models in the maximum likelihood (ML) or Bayesian frameworks. Parameters of these models include rates of substitution, the topology of the phylogeny, and its branch lengths—representing the expected number of substitutions per site occurring along that branch. Typically, the objective of phylogenetic reconstruction is thus to infer both the topology of the tree and its branch lengths. In the ML framework, parameter inference is achieved by heuristics that attempt to maximize the likelihood. In the Bayesian framework, it is often achieved by Markov chain Monte Carlo algorithms that sample the posterior distribution. Both approaches are computationally expensive for two reasons. First, they need to explore the space of tree topologies, which grows super-exponentially in the number of leaves (Felsenstein 2004). Second, this exploration involves numerous computations of the likelihood, each obtained with a costly sum-product algorithm (Felsenstein's pruning algorithm; Felsenstein 1981). This computational cost has kept researchers from using more realistic models of sequence evolution, which would for instance take into account interactions between sites of a protein (as in e.g. Kleinman et al. 2010). Such simplifications are well-known to be problematic, as several reconstruction artifacts directly associated to model violations were discovered early in the history of model-based phylogenetic reconstruction (Weisburg et al. 1989; Yang 1996; Telford et al. 2005). Much faster methods exist, but they are generally less accurate (Guindon and Gascuel 2003). In particular, distance methods (e.g. neighbor joining [NJ], Saitou and Nei 1987; BioNJ, Gascuel 1997; FastME, Lefort et al. 2015) build a hierarchical clustering of sequences based on some estimate of their evolutionary pairwise distances, i.e. the sum of the branch lengths along the path between pairs of sequences on the true unobserved phylogenetic tree. While even a simple $\mathcal{O}(n^2)$ algorithm (Waterman et al. 1977) is guaranteed to reconstruct the true tree topology if applied to the true distances, making the problem of estimating the tree and the set of distances equivalent, algorithms such as NJ further provide the same guarantee even if

Received: July 23, 2024. **Revised:** January 16, 2025. **Accepted:** January 27, 2025

the estimated distances are at most half the shortest edge length in the tree away from their true value (Atteson 1999). This ensures the statistical consistency of such distance methods given unbiased estimates of the true evolutionary distances. In practice, distances are typically estimated under the same probabilistic models as ML and Bayesian methods but considering each pair separately—whereas the latter consider all sequences at once—which greatly simplifies computations but discards part of the global information contained in the full set of homologous sequences.

Here, we present Phyloformer (PF), a phylogenetic inference method exploiting all sequences at once with the speed of distance methods. Importantly, PF can handle complex models of sequence evolution for which likelihood computations would not be feasible. We build on recent advances in deep learning for multiple sequence alignments (MSAs, Rao et al. 2021) and in the likelihood-free inference paradigm (Fig. 1). Sometimes referred to as simulation-based inference (Lueckmann et al. 2021), this paradigm exploits the fact that simulating data under probabilistic models of sequence evolution is computationally affordable, even in cases where computing likelihoods under these models is expensive. Through simulation we sample a large number of phylogenetic trees and MSAs evolved along these trees, given a probabilistic model under which we want to perform phylogenetic inference. We then learn a function that takes an MSA as input and outputs the evolutionary distances between all pairs of sequences on the tree. This function provides a point inference of the *full set of pairwise distances* under the chosen probabilistic model, conditional to the observed MSA. Learning the function is

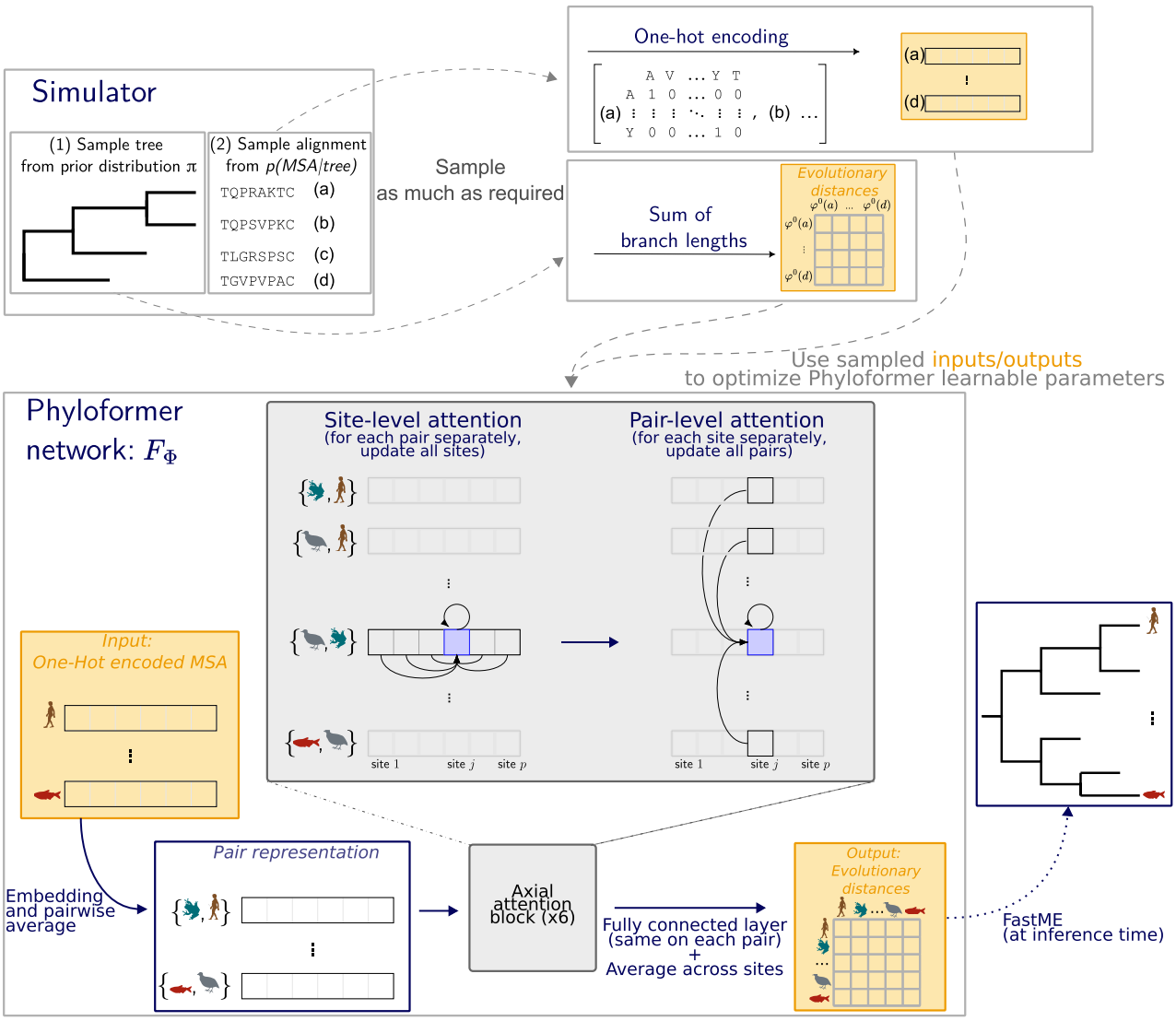

Fig. 1. Learning a function that reconstructs a phylogenetic tree from an MSA. We simulate phylogenetic trees and evolve MSAs along these trees under a given probabilistic model (Simulator panel). Once encoded, we use the examples of MSAs and corresponding trees to optimize the prediction function, described in the PF network panel. Each square denotes a vector of dimension d representing one site in one sequence or pair in the MSA, where the value of d can be different at each step. PF starts (bottom left) from a one-hot encoded MSA, and builds a representation for the pairs. These pairs then go through several layers of *axial attention* (central panel). Each of these layers shares information across sites within each pair and across pairs within each site, progressively building a new representation for each pair that accounts for the entire MSA. For every site of every pair, we finally apply the same fully connected network to the $\mathbb{R}^d$ embedding in the resulting representation. Finally, we average the embeddings across sites to predict the evolutionary distance between each pair (bottom right). At training time, we compare these distances against real ones to optimize the network parameters Φ. At inference time, we feed them to FastME to reconstruct a phylogeny.

computationally intensive, but once done, PF can be used in combination with a distance method to reconstruct a tree from an MSA very rapidly, regardless of the complexity of the model of sequence evolution. We show that under the common LG+GC model (Le and Gascuel 2008), PF leads to phylogenies as accurate as state of the art ML methods, as measured per the Kuhner–Felsenstein (KF) metric that accounts for both evolutionary distances and topologies, but runs two orders of magnitude faster than these methods on a graphics processing unit (GPU). When looking at the Robinson–Foulds (RF) metric that only measures topological accuracy, PF outperforms FastME, but falls behind ML methods, with performances degrading as the number of leaves increases. Under more realistic models, e.g. accounting for pairwise dependencies between sites, PF widens the gap with all other inference methods in the KF metric, and is on a par with ML methods in topological accuracy for small trees.

Related work (Mo et al. 2024) offer a recent review on deep learning for phylogenetics. Suvorov et al. (2019) and Zou et al. (2020) proposed likelihood-free methods for phylogenetic inference, by casting the problem as a classification across possible topologies. Given the super-exponential growth of the number of possible unrooted tree topologies in the number of sequences, they restricted themselves to trees with four leaves (quartet trees), that could then be combined to obtain larger trees (Strimmer and Von Haeseler 1996). Both methods relied on convolutional neural networks and were, therefore, sensitive to the order of the sequences in the alignment and restricted to a fixed sequence length—smaller sequences being accommodated with padding. More recently, while still only considering quartet trees, Tang et al. (2024) proposed a network that was independent of sequence order, and reported accuracies similar to Zou et al. (2020) using fewer training samples. Zaharias et al. (2022) showed that the accuracy of the network introduced in Zou et al. (2020) was lower than that of ML or distance methods when evaluated on difficult problems involving long branches and shorter sequences (200 sites), for both quartet trees and trees with 20 leaves. Smith and Hahn (2023) proposed a generative adversarial network for phylogenetic inference. While also likelihood-free, this approach required a new training for each inference, and did not scale beyond fifteen species. Jiang et al. (2022) introduced a distance-based learning method called Deep-learning Enabled Phylogenetic Placement (DEPP) for the related problem of adding new tips into an existing tree. This method relies on a convolutional network, trained on a very large reference empirical data set containing a sequence alignment and the corresponding reconstructed phylogeny. The network learns an embedding of the sequences that captures the pairwise distances on the tree, and can then be used to predict the distances between a query sequence and the sequences in the alignment. Based on these distances, the query sequence can be placed inside the tree using a dedicated distance-based placement algorithm (Balaban et al. 2020). DEPP thus uses deep learning for distance-based phylogenetics, like PF, but relies on a large empirical data set for its training, which needs to be redone if the reference phylogeny or alignment is changed. Another class of approaches has rather focused on optimizing the tree space exploration, with Azouri et al. (2021) using first a traditional machine learning model, namely a random forest regressor, to predict optimal, likelihood-maximizing, SPR moves. Second, the authors then resorted to reinforcement learning and a fully connected neural network architecture (Azouri et al. 2024), and showed that, allowing for suboptimal moves during tree space exploration, the method could outperform state of the art techniques. Our work is also related to the recent corpus of methods predicting contact between pairs of residues from MSAs, a crucial step in protein structure prediction (Jumper et al. 2021; Rao et al. 2021). These methods infer distances between sites (columns in the MSA), whereas we infer distances between sequences (rows in the MSA). Our network is trained end-to-end to predict distances, whereas the Rao et al. (2021) network is pretrained on a masked language modeling task to learn a data representation that is then used as input for residue contact prediction learning.

Results

Likelihood-Free Phylogenetic Inference with Phyloformer

PF is a learnable function for reconstructing a phylogenetic tree from an MSA representing a set of homologous sequences (Fig. 1). It produces an estimate, under a chosen probabilistic model, of the distances between all pairs of sequences, which is then fed to a fast distance-based method to infer a phylogenetic tree. The key feature of PF is its ability to produce pairwise distance estimates that account for all sequences in the alignment—providing more accuracy than the fast approaches that consider each pair of sequences independently—without computing likelihoods—leading to much faster inference than full ML or Bayesian approaches.

For a given model of sequence evolution $p(\text{MSA} \mid \tau, \theta)$ describing how an observed MSA evolves conditionally to a phylogeny τ and evolutionary parameters θ—substitution rates, equilibrium frequencies—and priors $\pi(\theta)$ and $\pi(\tau)$, we generate a large number of samples $\{(\text{MSA}, \tau, \theta)\}$ under the unnormalized posterior $p(\text{MSA}, \tau, \theta) = p(\text{MSA} \mid \tau, \theta)\pi(\tau)\pi(\theta)$ (Fig. 1, Simulator panel). Because there is a bijection between trees and evolutionary distances, these are also samples from $p(\text{MSA}, d, \theta) = p(\text{MSA} \mid d, \theta)\pi(d)\pi(\theta)$ where d is the set of evolutionary distances between pairs of leaves in τ and $\pi(d)$ is the distribution over distances induced by $\pi(\tau)$. We then use these samples to build a function estimating the tree τ, by optimizing a parameterized function $F_\Phi(\text{MSA})$ that takes the MSA as input and outputs point estimates of distances d. Parameters of $F_\Phi(\text{MSA})$ are estimated by minimizing the average absolute error between these point estimates and the real distances, which amounts to estimating the median of the posterior distribution $p(d \mid \text{MSA}, \theta)$, see supplementary methods 1.4, Supplementary Material online. At inference time, these distances can then be used by a distance-based method like FastME to reconstruct an estimate of τ. Assuming that the family of functions described by F_Φ is expressive enough and that enough samples are used, this approach offers posterior inference under the model (π, p), effectively replacing likelihood evaluations by samplings of $p(\text{MSA} \mid \tau, \theta)$.

Our F_Φ relies on self-attention to build a vector representation for each pair of sequences that contains all the information from the MSA required to determine the corresponding distance. During each self-attention block, the representation of each pair is updated using information extracted from itself and from all others. Self-attention is a general mechanism popularized by the Transformer architecture (Vaswani et al. 2017) that acts on an unordered set of objects $\{o_1, \ldots, o_N\}$—in our case, the set of pairs of sequences at a single site. A self-attention layer replaces each object o_i in the set by a weighted average

$\sum_{j=1}^{N} w_{ij} v(o_j)$ of all objects in the set. It relies on functions that computes the w_{ij}—how much object j weighs in the update of object i—and another function v that determines what information it contributes. Each of these functions depends on weights that are optimized during the learning process to make the predicted distances as close as possible to the real ones. For these reasons, we expect them to adaptively extract an MSA-aware representation for each pair that captures the relevant information from the whole alignment, as opposed to the initial representation where each pair is blind to the rest of the MSA. We provide more detail on our self-attention mechanism in the Scalable self-attention section. Because we maintain a separate vector representation for each site within each pair, every pair is itself a list of elements—the amino acids at a given site for the pair of sequences—and thereby also amenable to self-attention. We follow the axial attention strategy (Ho et al. 2019; Rao et al. 2021) and alternate between a separate update for each site—whereby information is shared among pairs as we just described—and a similar separate update for each pair whereby information flows among the sites. Following the axial attention blocks, we use a fully connected neural network on the enriched representation of each pair of sequences to predict the corresponding distance on the phylogenetic tree.

Performance Under a Standard Model of Evolution

We first assessed the performances of PF on data generated under the LG+GC model of sequence evolution, which combines the LG matrix of amino acid substitution (Le and Gascuel 2008) with rate heterogeneity across sites (Yang 1994). The LG model is widely used, implemented in many phylogenetic tools (Huelsenbeck and Ronquist 2001; Yang 2007; Höhna et al. 2016; Rambaut 2017) and amenable to likelihood computation, making it a good model to compare against state of the art ML inference methods. Following Szöllősi et al. (2022), we sampled trees under a birth–death process, subsequently rescaling the branches to simulate variations of the rate of sequence evolution (see supplementary methods 1.1, Supplementary Material online). We chose simulation parameters to match empirical data in the HOGENOM (Penel et al. 2009) and RaxMLGrove (Höhler et al. 2022) databases (see Online Methods and supplementary methods 1.1.1, Supplementary Material online). We then evolved MSAs of 50 sequences and 500 sites under LG+GC along these trees, and used the resulting data to train PF (see supplementary methods 1.2, Supplementary Material online). We compared PF followed by FastME to reconstruct the tree from estimated distances against two ML methods, IQTree and FastTree, and one distance method, FastME using LG pairwise distances. Figure 2a shows the average KF distance (Kuhner and Felsenstein 1994) between the true and reconstructed phylogeny for each of these methods over 500 samples from the same model for increasing numbers of leaves. The KF distance is widely used to compare phylogenies and captures both topological and branch length reconstruction errors. Under this metric, PF achieved a performance similar to ML methods. It is noteworthy that this performance was stable across numbers of leaves, even though our network was trained on 50-leaf phylogenies only. The performance was also stable when doing inference over a range of sequence lengths, even though PF was trained only on alignments with 500 positions (supplementary figures 13 and 14, Supplementary Material online). FastME with FastME distances was much less accurate. Interestingly, the high accuracy of PF—using a GPU—was achieved with the lowest runtime among all benchmarked methods (Fig. 3). In particular, it was up to 135 times faster than the ML method IQTree, for a similar

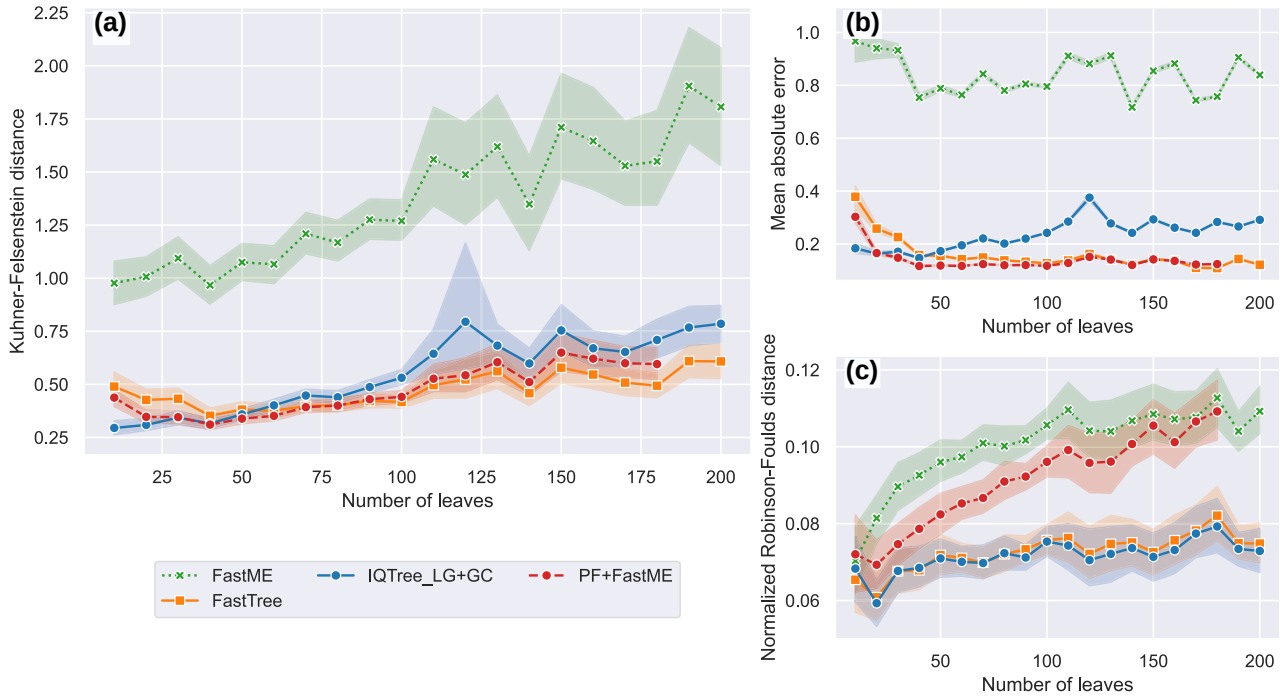

Fig. 2. Performance measures for different tree reconstruction method. a) KF distance, which takes into account both topology and branch lengths of the compared trees; b) MAE on pairwise distances, which ignores topology; and c) normalized RF distance, which only takes into account tree topology. The alignments for which trees are inferred were simulated under the LG+GC sequence model and are all 500 amino acids long. For each measure, we show 95% confidence intervals estimated with 1,000 bootstrap samples.

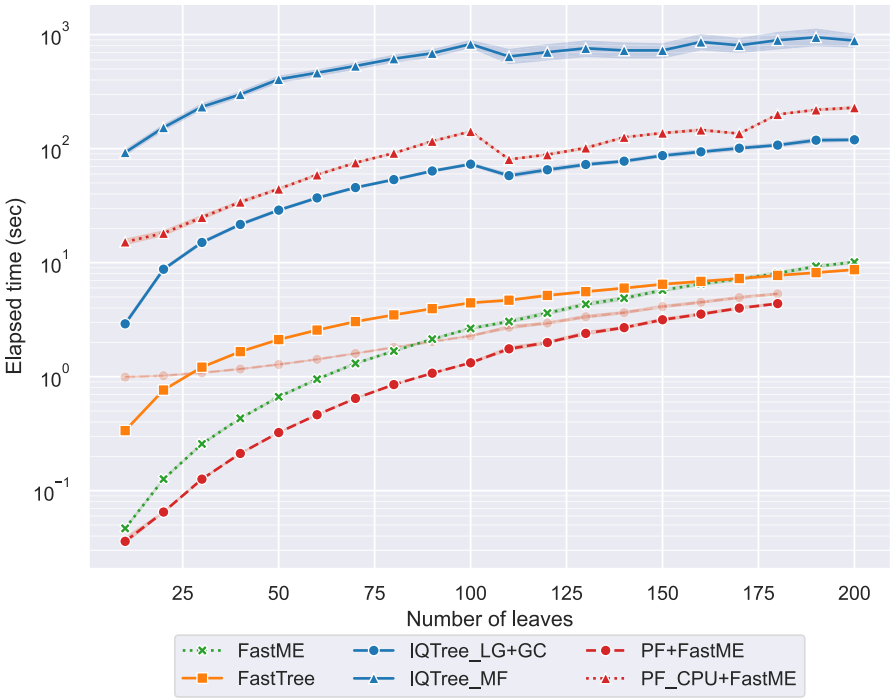

Fig. 3. Execution time for different tree reconstruction methods on the LG+GC test set with alignments of length 500. For IQTree ModelFinder (MF) times were measured on the Cherry testing set (see section Performance Under More Realistic Models). For all methods except PF, total wall time was measured. For PF, the elapsed time is the sum of the time it takes to infer the distances and the time FastME takes to infer the tree from these distances. It is important to note that the distance prediction time does not include the time it takes to load the PF weights to the GPU as we did that once before inferring distances for all the testing alignments. The faded red dashed line shows the PF execution times, added to the average central processing unit (CPU) to GPU model-loading time measured over 100 replicates. The slight discrepancy noticeable at 110 leaves is due to a change of CPUs in our high performance computing (HPC) environment (see end of Baselines).

accuracy. FastTree—a faster and supposedly less accurate heuristic for ML—also had similar accuracy on this dataset, but remained one order of magnitude slower than PF. PF was even twice as fast as FastME combined with LG distances. As PF itself runs FastME to reconstruct a tree from its distance estimates, this difference indicates that inferring distances that exploit the full MSA with a trained PF on a GPU is actually faster than computing the ML distances independently for each pair. Conversely, PF was the most memory intensive method, using up to 24.2 GB of GPU RAM for alignments of 180 sequences (supplementary figure 16, Supplementary Material online), which makes it impossible to handle larger trees on 32 Gb GPUs. Of note, memory usage can be halved in some cases by using automatic mixed precision. Furthermore, running PF inference on central processing units (CPUs) is possible with the same memory usage as on GPU. However, it is slower than IQTree and therefore not advantageous for models amenable to likelihood computation.

Figure 2b and c stratify the reconstruction error in terms of their topology (panel c, using the normalized RF metric Robinson and Foulds 1981) and pairwise distances (panel b, using the mean absolute error [MAE] between true and estimated distances). PF was more accurate than FastME on both criteria. On the other hand, PF had a larger topological error than ML methods as measured by the normalized RF distance, and increasingly so for larger numbers of leaves. On the contrary, it was better at estimating evolutionary distances. A possible explanation for this discrepancy is that since we control the tree diameter in our simulation, larger trees have shorter branches on average. As branch lengths decrease, the number of mispredicted branches increases leading to larger topological errors (see supplementary results 2.3, Supplementary Material online for an in depth explanation).

Finally, we investigated the ability of PF to handle gaps contained in empirical MSAs because of insertion–deletion (indel) events that have occurred during sequence evolution. Standard models of sequence evolution consider gaps as wildcard "X" characters, and thus cannot benefit from the information they provide. Models that account for insertion–deletion processes are more complicated to implement and more costly to run (Redelings and Suchard 2007), but can easily be included using our paradigm. We fine-tuned the PF network previously trained on ungapped LG+GC data on a smaller dataset that includes indels, inserted through a model of insertion/deletion events in Alisim (Ly-Trong et al. 2022), choosing parameters as in Trost et al. (2024). Figure 4 shows that the accuracy of all methods dropped on alignments that include gaps compared to alignments that do not (Fig. 2), probably because gaps remove information from the alignments. However, the difference between PF and ML methods shrinked, with PF outperforming ML methods according to the RF metric for 10 to 30-leaf trees. This is likely due to PF's ability to extract information from gaps, which are encoded as a separate character and not as a wildcard character. The drop in performance for larger trees is most likely caused by the same phenomenon as the one in Fig. 2 (see above and supplementary results 2.3, Supplementary Material online).

Performance Under More Realistic Models

Because ML and Bayesian inference approaches must compute the likelihood, in practice they can only be used under simple

models such as LG+GC for which these likelihood calculations are affordable. PF on the other hand can reconstruct phylogenies under arbitrarily complex models of sequence evolution, as long as we can efficiently sample training data from these models. We now illustrate this feature by considering inference tasks under two substitution models that relax common simplifying assumptions: independence between sites, and the homogeneity of selective constraints across sites. The first model we used (Cherry, supplementary methods 1.2, Supplementary Material online) is derived from a model of sequence evolution that includes pairwise amino acid interactions (Prillo et al. 2023). ML inference under such a model would be very costly for two reasons: the substitution matrix has size 400×400, and would need to be applied to pairs of interacting sites, which would need to be identified with additional computations. The second model (SelReg, supplementary methods 1.2, Supplementary Material online) draws different selective regimes for each site of the alignment: a site can evolve under neutral evolution, negative selection, or persistent positive selection. ML inference under such a model is achievable with a mixture model (e.g. Si Quang et al. 2008), but costly, because the SelReg mixture includes 263 distinct amino acid profiles, plus a profile for neutral evolution, and a different matrix for positively selected sites. We fine-tuned the PF network previously trained under the LG+GC model on alignments sampled under the Cherry or the SelReg model. We compared its performances against the same methods as before, but allowing IQTree to search for the best evolution model available (with the Model Finder option). Figure 5 shows that under both the Cherry and SelReg models all methods performed worse than under LG+GC, presumably because both models decrease the information provided by a given number of sites, by including pairwise correlations (Cherry), or positively selected sites that might saturate (SelReg). However, PF outperformed all other methods by a substantial margin under the KF metric, with distances around 1 whereas others range between 2 and up to 10 for IQTree under SelReg. Of note, the Model Finder option was costly, further increasing the computational edge of PF (Fig. 3). Not using this option markedly decreased the accuracy of IQTree on the Cherry alignments (supplementary figures 11 and 12, Supplementary Material online). As we observed under LG+GC, PF was better at estimating distances than topologies (Fig. 5), with the latter becoming more challenging for larger numbers of leaves. We observed a similar trend as for the LG – GC experiments, where the RF distances increased for larger trees (albeit at a much slower rate), making PF progressively lose its edge against misspecified ML methods when only considering topological accuracy. The cause of this degradation is likely the same as the one we proposed in section Performance Under a Standard Model of Evolution (details are provided in supplementary results 2.3, Supplementary Material online).

Phyloformer Is Likelihood-Free but Not Model-Free

Since it is trained on data simulated under a specific evolutionary model, PF is not a model-free method. As such, it is not immune to model misspecification, much like all likelihood-based tree reconstruction methods. In order to investigate the effects of this misspecification we simulated additional testing data using three evolutionary models for which the equilibrium frequencies as well as the exchange rates were as "far" as possible from LG (Minh et al. 2021; Norn et al. 2021), namely: HIVw, JTT, and mtRev. For each of these models, we simulated new 500 amino acid-long alignments, using the trees from the test set shown in section Performance Under a Standard Model of Evolution and with rate heterogeneity across sites (Yang 1994).

As in section Performance Under a Standard Model of Evolution, we inferred trees using PF followed by FastME, FastTree using the LG model, IQTree using the LG+GC model and FastME using distances computed by FastME under the LG model. To quantify the effect of misspecification we also

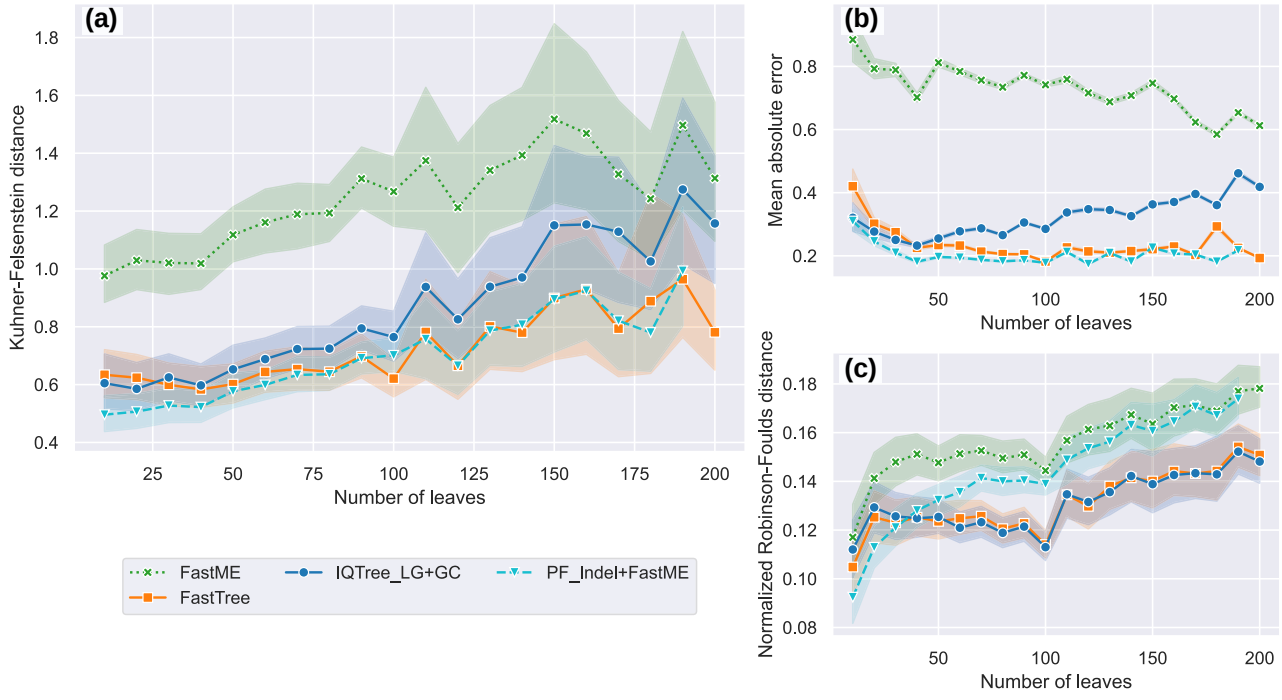

Fig. 4. Tree comparison metrics for different tree reconstruction methods on the LG+GC+indels test set (alignment length = 500). Legend as in Fig. 2, with PF fine-tuned on alignments with gaps named PF_{Indel} + FastME and in cyan.

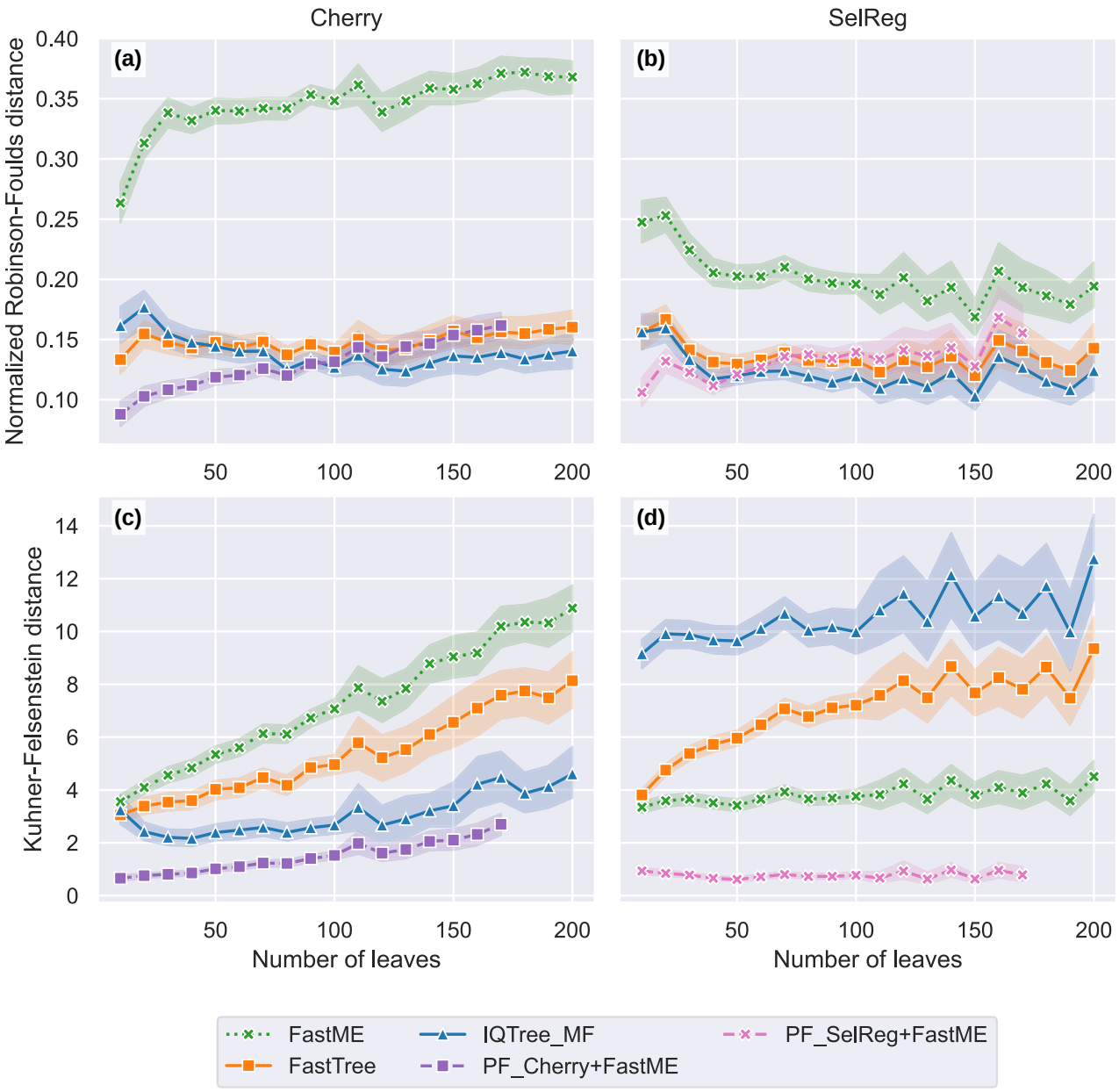

Fig. 5. Normalized RF distance (above) and KF distance (below) for different tree reconstruction methods on the Cherry (left) and SelReg (right) test sets (alignment length = 500).

inferred trees with IQTree using the correct substitution matrix (i.e. HIVw+GC, JTT+GC, and mtREV+GC).

When tested on these new datasets, and as measured by the KF distance, all LG tree-inference methods do worse than IQTree with the correct model (Fig. 6), except for FastTree when inferring larger trees from JTT alignments. However, it seems like all methods are similarly impacted, and the relationships between these methods is very similar to the one shown in Fig. 2. When looking at the normalized RF distance (see supplementary figure 15, Supplementary Material online), the same dynamic is present: misspecification degrades the accuracy of all methods but preserves their relative performances. Therefore, it seems likely that PF behaves quite similarly to other tree inference methods misspecification-wise, consistently with what was reported in Thompson et al. (2024), and is not oversensitive to this phenomenon. Of course, inferring a tree under the wrong evolutionary model for a given alignment will not yield the optimal tree, but that is the case for all other model-based tree inference methods.

Phyloformer Performs on Par with ML Methods on Empirical Data

We compared the performance of PF and other methods on 346 orthologous gene alignments from 36 Cyanobacteria (Szöllosi et al. 2013), reasoning that good reconstruction methods should more often infer trees that match the tree obtained on the concatenated gene alignments. We compared the LG+GC-with-indel version of PF to the same three methods assessed in section Performance Under a Standard Model of Evolution. Figure 7a shows that under the RF metric, PF performed as well as the other standard methods on empirical data, and did so faster—on a GPU.

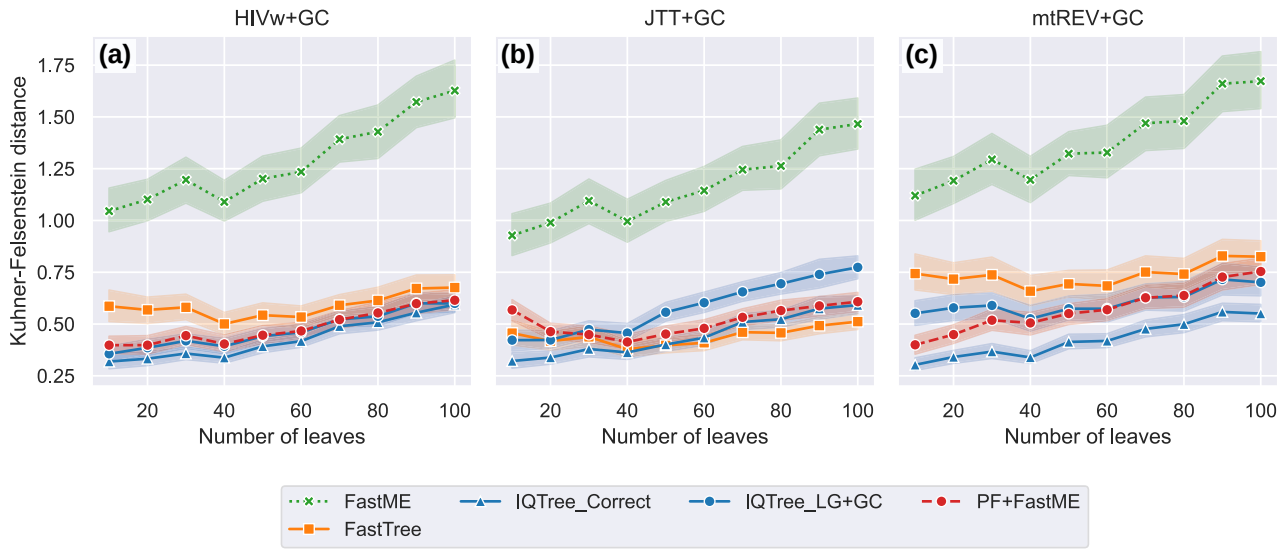

Fig. 6. KF distance, for different tree reconstruction methods applied to alignments simulated under the (a) HIVw, (b) JTT, and (c) mtREV evolutionary models. All methods except IQTree_Correct infer trees using the LG evolutionary model. IQTree_Correct inferes trees using the appropriate model for each testing dataset.

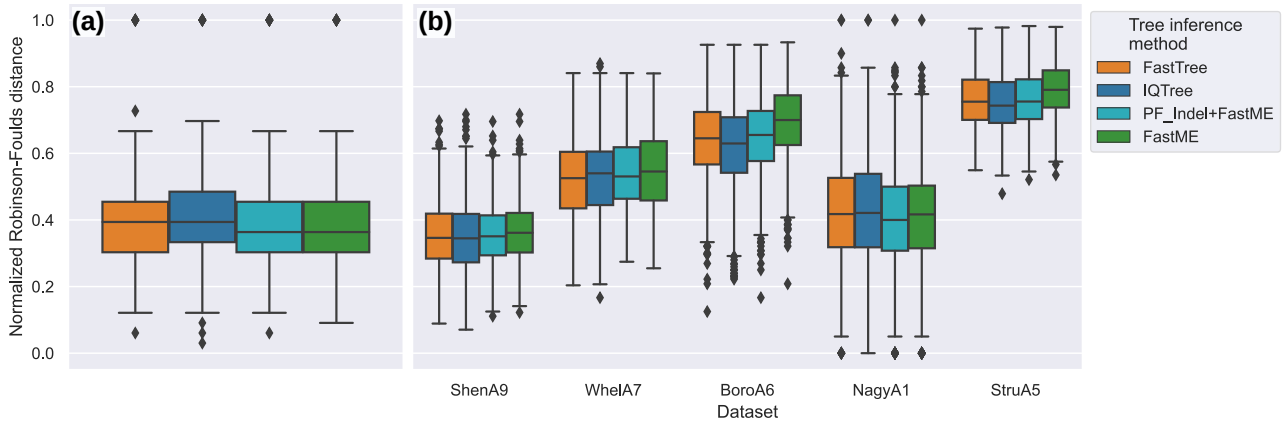

Fig. 7. Comparison of topology reconstruction accuracy between PF and other methods on empirical data. In both panels, we show the normalized RF distance between reconstructed gene trees and the corresponding concatenate tree. In a) inferred gene trees on alignments from Szöllosi et al. (2013) using the same pipeline as in section Performance Under a Standard Model of Evolution and with the gap-aware version of PF shown in Fig. 4. In b) gene alignments, species trees and some gene trees were obtained from Zhou et al. (2018). We inferred gene-trees using the gap-aware version of Phyloformer and FastME as in (a). The IQTree predictions were made in Zhou et al. (2018) under the evolutionary model found by IQTree ModelFinder, then 10 predictions were done and only the one with the best likelihood was kept. The datasets shown here have ≥ 80% of alignments detected as LG by IQTree.

We conducted a similar analysis on gene-trees over many different clades obtained from Zhou et al. (2018). In this study, the authors collected a large number of sequence datasets and inferred gene-trees using IQTree and FastTree under the evolutionary model found by IQTree's ModelFinder for each alignment. For IQTree they inferred 10 trees and only kept the one with the best likelihood. The authors also reconstructed species trees from concatenated alignments for each dataset. We reconstructed trees on the five ensembles of alignments where at least 80% of alignments were classified as LG by IQTree using the LG+GC-with-indel version of PF with FastME (see supplementary methods 1.5, Supplementary Material online).

We then compared our gene trees as well as the ones from Zhou et al. (2018) to the concatenate trees. Here again, Fig. 7b shows that in most cases PF performed as well as the best of 10 trees estimated with ML methods. Here the computational speed of PF shines as we were able to infer about 12, 000 trees in under 2 h with one GPU. In Zhou et al. (2018), the authors measured execution times of only 10% of tree inference tasks, for which the total runtimes of IQTree and FastTree were ∼ 10.5 days and 4 h, respectively. On the same subset of trees, we measured the total runtime of Phyloformer+FastME and standalone FastME at ∼ 11.5 and 15 min, respectively. Furthermore, PF consistently produced trees with a higher likelihood than FastME trees though still lower than pure ML methods (supplementary figure 7, Supplementary Material online).

Discussion

Drawing on recent breakthroughs in likelihood-free inference and geometric deep learning, we have demonstrated that PF achieves rapid and precise phylogenetic inference. The likelihood-free paradigm only requires samples from the probabilistic model of sequence evolution, which allows inference

under much more complex models than ML or Bayesian inference. Furthermore we exploited an amortized form of this paradigm, requiring a single training of a neural network that takes an MSA as input and outputs evolutionary distances between pairs of sequences—as opposed to approaches like approximate bayesian computation (Csilléry et al. 2010) that require a new sampling step at each inference. We based our neural network on axial self-attention, an expressive mechanism that accounts for the symmetries of the MSA and seamlessly handles arbitrary numbers of sequences of any length.

PF was faster and as accurate as ML inference methods on data sampled under the standard LG+GC model, as measured by the KF distance that captures both the topological and branch length reconstruction accuracy. Computing likelihoods under LG+GC is expensive but possible, making ML inference the gold standard: reaching the same accuracy faster was the best outcome one could hope for. However, PF performed worse than ML methods when focusing on topological accuracy only. Under more complex models accounting for local dependencies (Cherry) or heterogeneous selective pressures (SelReg), computing likelihoods is too costly, forcing ML methods to work under misspecified models, whereas PF can still perform inference under the correct model, without any effect on its speed. As a result, PF yields the most accurate inference by a substantial margin under the KF metric while retaining its computational edge. Nonetheless, both of the corresponding networks were trained on gapless data and are, therefore, only useful as a proof of concept: we do not recommend using them on empirical alignments—unless these alignments are gapless themselves.

More generally, we stress that likelihood-free inference using neural networks has a model-based nature identical to that of ML or Bayesian methods. It formally estimates the posterior distribution defined by the prior and probabilistic model used to simulate training data, accessing this model through sampling instead of likelihood evaluations. As such, it is not immune to model misspecification: for example, we observed that PF trained on LG+GC underperformed on data simulated under Cherry or SelReg and vice versa (supplementary figures 11, 12, and 17, Supplementary Material online), and that PF was as sensitive to misspecification of the matrix of amino acid substitution rates as ML methods (Fig. 6). Rather than replacing model choice, we believe that the crucial contribution of a likelihood-free method like PF is to offer a way to work under more realistic models of sequence evolution that were so far not amenable to inference.

It is noteworthy that the inference speed that we report for PF was recorded on a GPU, a less widespread hardware than the CPU used for other methods, which may limit its interest for analyzing a single gene alignment under models amenable to ML. On a CPU, PF has a runtime larger than that of IQTree. It is, therefore, not an interesting alternative for reconstructing a single tree on a CPU under standard models such as LG – GC, where ML methods are at least as accurate.

In its current form, PF also becomes less useful as the trees get larger. First, our tests revealed that PF's topological accuracy decreases as the number of leaves in the phylogeny increases, whereas ML approaches are more stable. This is likely due to our experimental setting, where larger trees have shorter branches on average. It may also be caused by PF's reliance on a distance matrix, which reduces the amount of information available for phylogenetic reconstruction compared to likelihood methods that estimate probability distributions for all ancestral states. In addition, PF's memory usage scales quadratically with the number of sequences (supplementary figure 16, Supplementary Material online), because it is mostly driven by applying self-attention to pairs of sequences. This prevents analyzing larger data sets beyond 180 sequences of length 500 on a GPU with 32 Gb of RAM. A better scaling version could be obtained by working at the sequence level—attempts to do so have scaled beyond 2,000 sequences but led to lower accuracies so far.

At the present time, the most useful version of PF is likely to be PF_{Indel}, which has been trained under the LG model of sequence evolution, with indels. This version could have a significant impact in experiments where many reconstructions are necessary, e.g. for bootstrapping and reconstructing several gene trees of a few dozen leaves from whole genomes or transcriptomes, and where branch lengths and topological accuracy are equally important. In such a situation, loading the network into the GPU memory only needs to be done once, which makes the method very efficient. In the future, we expect that PF will have its largest impact on phylogenetic inference after versions are trained on more realistic models of sequence evolution which could include model parameter heterogeneities along the sequence or between branches and position-specific dependencies among sites (Boussau and Gouy 2006; Blanquart and Lartillot 2008; Kleinman et al. 2010). Our self-attention network could exploit these latter dependencies via the addition of positional encodings—a standard approach in the transformers literature. It is yet unclear if a single, complex enough model of sequence evolution will be enough to capture all cases of interest, or if better inference will be achieved by offering a collection of trained networks corresponding to different realistic models. We will study this question in future work, and in the latter case we will develop an additional neural network to help choose the most relevant model for a given MSA, as currently offered by the ModelFinder option of IQTree. Versions of PF could also be trained on coding or noncoding nucleotide sequences.

Another important extension of PF will be to train with a topological loss function, e.g. directly minimizing the RF metric rather than a distance metric. Such a version would address the gap that we observed between accuracies in distance and topological reconstruction, and could also lead to a more scalable method by working around the need for all pairwise distances—of quadratic size in the number of sequences, whereas the tree itself has linear numbers of nodes and edges. We also believe that extending PF to unaligned sequences will be of interest, both because multiple alignments are computationally intensive, and because they are error-prone. This could be addressed by including the alignment step in the network (Petti et al. 2022; Llinares-López et al. 2023). Alternatively, one could forego alignment altogether, e.g. by maintaining a length-independent representation of each sequence throughout the network.

Beyond phylogenetic reconstruction, our network can be trained to infer other parameters of the simulation model. This would provide an efficient and flexible way to study phylodynamics, phylogeography, and selective pressures operating on the sequences, for instance.

Online Methods

The Phyloformer Neural Network

PF is a parameterized function F_Φ that takes as input an MSA of n sequences of length L and outputs an estimate of the $N = \binom{n}{2}$ distances between all pairs of sequences. Φ denotes the set of

learnable parameters of F_Φ. We then input these distances to FastME (Lefort et al. 2015) to obtain a phylogenetic tree (Fig. 1).

The PF network starts with a one-hot encoding of the aligned sequences: every sequence x is represented as a matrix $\varphi^{(0)}(x) \in \{0, 1\}^{22\times L}$ in which column j contains a single non-zero element $\varphi^{(0)}_{ij}(x) = 1$, whose coordinate $i \in \{1, \ldots, 22\}$ denotes the amino acid or gap present in sequence x at position j. It then represents each pair (x, x') of sequences in the MSA by the average of their individual representations, i.e. with a slight abuse of notation, $\varphi^{(0)}(x, x') = \frac{1}{2}(\varphi^{(0)}(x) + \varphi^{(0)}(x'))$. Of note, $\varphi^{(0)}(x, x')$ does not depend on the order of sequences x and x'. At this stage, the network represents each site within each pair independently of all others, encoding information such as "at site 4, sequences x and x' contain a Leucine and an Isoleucine." The whole purpose of F_ϕ is to account for relevant information about the evolutionary distance between x and x' contained in other sequences from the alignment. To extract this information, F_ϕ uses $r = 6$ self-attention layers (Vaswani et al. 2017) that iteratively build updated $\varphi^{(l)}(x, x') \in d \times L$ representations of each pair using all others in the MSA. More precisely, we use axial attention (Rao et al. 2021, Fig. 1, central panel) and successively update each pair (resp. site) separately by sharing information across sites (resp. pairs). Along each axis, we rely on a modified linear attention (Katharopoulos et al. 2020, see Scalable self-attention), with $h = 4$ attention heads and embeddings of dimension 64 for the value matrix and only 1 for the query and key matrices. The r axial attention blocks of PF output for every pair of sequences a tensor $\varphi^{(r)}(x, x') \in \mathbb{R}^{d\times L}$ informed by all other pairs in the same MSA. We convert this representation into a single estimate of the evolutionary distance between x and x' by applying an $\mathbb{R}^d \to \mathbb{R}$ fully connected layer to each site of each pair, followed by an average over the sites. We provide more details on the F_Φ architecture in supplementary section 1.3, Supplementary Material online.

Accounting for Symmetries

It is now well understood that accounting for known symmetries is key to the success of deep learning, as formalized in geometric deep learning (Bronstein et al. 2021). Following this principle, we parameterize the function F_Φ by a neural network that exploits two symmetries of the estimation task: the estimated evolutionary distances should not depend on the order of the n sequences or L sites in the MSA. More precisely, we want F_Φ to be equivariant by permutations of the sequences: if it returns values d_{ab}, d_{ac}, d_{bc} when presented with sequences (a, b, c), it should return d_{ac}, d_{bc}, d_{ab} when given (c, a, b) as input. On the other hand, when working with a model of evolution such as LG+GC which assumes the process of evolution being independent and identically distributed (i.i.d.) across all sites, it is desirable for the F_Φ function to be capable to exploit an additional symmetry, namely being invariant to site permutations which, given the i.i.d. assumption, simply lead to another instance of the same evolution process. The self-attention updates act on the $\mathbb{R}^d$ representations of a site within a pair of sequences regardless of their order, yielding the desired equivariances. Enforcing these equivariances would be more difficult if the updates were general functions acting on entire MSAs represented by $\mathbb{R}^{d\times N\times L}$ tensors. The final average across sites within each pair makes F_Φ invariant rather than equivariant by permutation of these sites. In addition because none of the operations in F_Φ depend on the number of sites or pairs, we can use the same F_Φ seamlessly on MSAs with an arbitrary number of sequences of arbitrary length. Finally, it is worth noting that, despite F_Φ being invariant to site permutations, the network, through the attention mechanism across different sites, is capable to account for interactions among them. This is demonstrated when the i.i.d. assumption is relaxed in the simulations under the Cherry model, with the network, regardless of being order-agnostic, still being capable of identifying coevolving sites in order to provide more accurate predictions (supplementary material 2.1, Supplementary Material online).

Scalable Self-Attention

Naive implementations of self-attention over M elements scale quadratically in M—in our case, both the number of sites and pairs of sequences. Indeed, softmax attention as introduced by Vaswani et al. (2017) is parameterized by three matrices $Q, K, V \in \mathbb{R}^{M\times d}$ for some embedding dimension d, respectively called Queries, Keys, and Values, and every update for an element i computes attention weights $(s_{i,1}, \ldots, s_{i,M}) = \mathrm{softmax}(\frac{q_i^\top K}{\sqrt{d}})$. We resorted to the linear attention of Katharopoulos et al. (2020), who exploited the fact that $s_{ij} = \frac{\langle\phi(q_i), \phi(k_j)\rangle}{\sum_{h=1}^{M}\langle\phi(q_i), \phi(k_h)\rangle}$ for some nonlinear infinite-dimensional mapping $\phi : \mathbb{R}^d \to \mathcal{H}$ to a Hilbert space $\mathcal{H}$ (Schölkopf and Smola 2002) and proposed to replace ϕ by some other nonlinear, finite-dimensional mappings $\tilde{\phi} : \mathbb{R}^d \to \mathbb{R}^t$. We can then rewrite the self-attention updates $z'_i = \sum_{j=1}^{M} s_{i,j}v_j$ as

$$z'_i = \frac{\sum_{j=1}^{M}\tilde{\phi}(q_i)^\top\tilde{\phi}(k_j)v_j}{\sum_{h=1}^{M}\tilde{\phi}(q_i)^\top\tilde{\phi}(k_h)} = \frac{\tilde{\phi}(q_i)^\top\sum_{j=1}^{M}\tilde{\phi}(k_j)v_j}{\tilde{\phi}(q_i)^\top\sum_{h=1}^{M}\tilde{\phi}(k_h)}. \quad (1)$$

Because we can precompute each of the two sums and reuse it for every query, this simple factorization reduces both the number of operations and memory usage from $\mathcal{O}(M^2 \cdot L \cdot d)$ to $\mathcal{O}(M \cdot L \cdot d \cdot t)$. Following Katharopoulos et al. (2020), we used an ELU-based mapping (Clevert et al. 2016)

$$\tilde{\phi}(x) = \begin{cases} x + 1, & \text{if } x > 0 \\ \exp(x) & \text{if } x \le 0, \end{cases}$$

where the operation is applied entrywise, yielding $\tilde{\phi}(x) \in \mathbb{R}^d$ vectors for $x \in \mathbb{R}^d$. In our experiments, we used $d = 64$ for the Values matrix, but noticed that using $d = 1$ for Queries and Keys led to slightly lower training-loss values (supplementary figure 21a, Supplementary Material online), while substantially reducing the memory footprint of the self-attention layers (supplementary figure 21b, Supplementary Material online). This observation is consistent with recent research showing that Transformers and other neural networks learn through gradual rank increase (Abbe et al. 2023; Zhao et al. 2023). However, applying (1) with queries and keys of dimension 1 leads to identical updates z'_i for all elements. To work around this issue, we normalized each update by the average of queries and the sum of keys instead of the usual sum of attention weights, leading to

$$z'_i = \frac{\tilde{\phi}(q_i)}{M^{-1}\sum_{g=1}^{M}\tilde{\phi}(q_g)} \cdot \frac{\sum_{j=1}^{M}\tilde{\phi}(k_j)v_j}{\sum_{h=1}^{M}\tilde{\phi}(k_h)}. \quad (2)$$

Training Phyloformer

We trained F_Φ using six NVIDIA A100 80 GB GPUs on simulated examples through a loss function (see Metrics) comparing the estimated and true evolutionary distance (Fig. 1). We used the Adam optimizer (Kingma and Ba 2015), batches of size 4 and a maximum learning rate of 10^{-3} with 3,000 linear warmup steps followed by a linear decrease of 213,270 steps, corresponding to 30 epochs. We also implemented an early stopping criterion that stopped training when the validation loss did not decrease over five successive 3,000 step intervals.

We first trained an F_Φ^{pre} function that served as a starting point for all the functions used in our experiments, by optimizing Φ with respect to the MAE loss for 20 epochs ($\approx$ 79 h) over the 170,616 examples (see section Online Methods) simulated under LG+GC, saving a model every 3,000 steps, and eventually retaining the one with lowest RF error (see Metrics) over the validation dataset (17,016 examples). For the results in Fig. 2, we further optimized the parameters of F_Φ^{pre} for 4 epochs (20 h) with respect to the MRE loss leading to a slightly improved error over small distances (supplementary figure 18, Supplementary Material online) and on the overall RF metric (supplementary figure 10, Supplementary Material online). For the results in Figs. 4 and 5, we further optimized the parameters of F_Φ^{pre} for the MAE loss on gapped MSAs and MSAs generated under the Cherry or SelReg substitution models, respectively (see Datasets).

Baselines

IQTree LG+GC (Minh et al. 2020, v2.2.0) reconstructs phylogenies in the ML framework. It first estimates several parsimony trees along with one reconstructed through a distance method, then optimizes branch lengths and other parameters of the model of sequence evolution, while performing local topological rearrangements (nearest neighbor interchanges, NNIs) to maximize the likelihood. We ran it with the LG model of amino acid substitution (Le and Gascuel 2008) combined with a continuous gamma distribution to model rate heterogeneity across site (Yang 1993). In our experiments, we did five rounds of NNIs since we observed that optimizing for more rounds rarely improved the topology of the final tree while substantially adding to the running time. The software was run with `iqtree2 -T 1 -m LG+GC -n 5`.

IQTree MF uses the MF mode of IQTree (Kalyaanamoorthy et al. 2017), in which likelihoods of an initial tree are computed for a large set of substitution models and models of rate-heterogeneity across sites. The best fitting model is selected using BIC. The rest of the tree search is done as above but using the selected model for likelihood estimations. The software was run with `iqtree2 -T 1 -n 5`.

FastTree (Price et al. 2010, v2.1.11 SSE3) reconstructs a starting tree using an algorithm inspired from NJ (Saitou and Nei 1987) which is subsequently refined with topological rearrangements to optimize the minimum evolution criterion. The tree is then improved using ML with NNIs. It was run under the LG+G4 model of sequence evolution. The software was run with `fasttree -lg -gamma`.

FastME (Lefort et al. 2015, v2.1.6.4) computes a distance matrix using ML, then reconstructs a tree topology using BioNJ (Gascuel 1997) and further refines it via topological rearrangements which seek to optimize the Balanced Minimum Evolution score. In virtually all performed experiments, we observed that the FastME tree search algorithm led to slightly better performances than the NJ algorithm (Saitou and Nei 1987). This is consistent with the existing literature showing that these BME-decreasing topological moves also decrease the RF error (Desper and Gascuel 2004; Sy Vinh and von Haeseler 2005). We did not resort to the --gamma option as in our experiments we observed that this lead to worse performances. Using FastME as our baseline distance method makes the comparison with PF insightful, as the only difference between the two methods is the distance matrix used as input. The software was run with fastme --nni --spr --protein=LG to reconstruct trees using the inbuilt evolutionary distance estimation and simply with fastme --nni --spr when PF's predicted distance matrix was provided.

All methods were run on a single CPU thread (Intel Xeon E5-2660 2.20 GHz for trees of size 10 to 100 and Intel Xeon E5-2650 v3 for trees of size 110 to 200) except for PF distance prediction which was run on a single GPU (NVIDIA V100 32 GB). The experiments run in (Zhou et al. 2018) show that RAxML-NG and IQTree2 often have very similar outputs, in many cases the tree topologies are identical. This phenomenon is likely to be also present on simulated data. Therefore, in an effort to reduce the computational footprint of this study we chose to run only one of these two methods, and chose IQTree2 since we also use it to simulate MSAs.

Datasets

We generated ultrametric phylogenies under a birth–death process. We used 50-leaf trees for training, and 10-leaf to 200-leaf trees for testing. We rescaled branch lengths as in Szöllősi et al. (2022) to yield nonultrametric trees. Finally, we rescaled each tree to resemble trees found in public databases of empirical trees (see also supplementary methods 1.1.1, Supplementary Material online, the effect of such a choice for the distribution of diameters along with its possible drawbacks is further discussed in supplementary results 2.3, Supplementary Material online). We used each rescaled phylogeny to simulate one MSA with AliSim (Ly-Trong et al. 2022) for the LG+GC model, or in-house code for Cherry, or Pastek (Duchemin et al. 2023) for SelReg. For LG+GC, we sampled the parameter of the gamma distribution to match values estimated on empirical data. We provide more details in supplementary methods 1, Supplementary Material online. While it would be possible to train PF models on trees of different number of leaves and/or MSAs of different lengths, for the sake of implementation simplicity and GPU memory efficiency we chose to only train on a single tree size and alignment length. This ensures that we fill GPU memory with useful data during training and not just padding tokens. MSA length seems to have little impact at inference time (see supplementary figures 13 and 14, Supplementary Material online), which probably remains true at training time.

Metrics

We now describe the metrics used throughout this article to compare phylogenies or optimize our network.

Let d_i be the ith of N true evolutionary distances in a phylogeny, and $\hat{d}_i$ the corresponding estimate output by a given tree inference method. Then the MAE and MRE are defined as

$$\ell_{\text{MAE}} = \frac{1}{N}\sum_{i=1}^{N} |d_i - \hat{d}_i| \quad \text{and} \quad \ell_{\text{MRE}} = \frac{1}{N}\sum_{i=1}^{N} \frac{|d_i - \hat{d}_i|}{d_i}.$$

When used to compute the loss during PF training, $\hat{d}_i$ values correspond to distance estimates directly output by F_Φ. When used as a metric (e.g. in Fig. 2) we use $\hat{d}_i$ values extracted from the reconstructed tree, by summing all branch lengths on the paths between each pair of leaves—even for PF—in order to fairly compare different methods.

In phylogenetic trees, each branch describes a bipartition of the set of leaves, paired with a weight (i.e. the branch length). Let A and B be the sets of leaf-bipartitions describing trees T_A and T_B, and $w_{e,T}$ the weight of a bipartition e in tree T. Then, the Normalized RF distances and the KF distance between T_A and T_B can be written

$$\mathrm{RF}_{\mathrm{norm}}(T_A, T_B) = (|A| + |B|)^{-1}(|A \cup B| - |A \cap B|)$$

$$\text{and} \quad \mathrm{KF}(T_A, T_B)^2 = \sum_{e \in A \cap B} (w_{e,T_A} - w_{e,T_B})^2 + \sum_{e \in A \setminus B} w_{e,T_A}^2 + \sum_{e \in B \setminus A} w_{e,T_B}^2.$$

Supplementary Material

Supplementary material is available at *Molecular Biology and Evolution* online.

Acknowledgments

The authors thank Dexiong Chen, Flora Jay, Martin Ruffel, and Johanna Trost for insightful discussions.

Funding

This work was funded by the Agence Nationale de la Recherche (ANR-20-CE45-0017). It was granted access to the high performance computing (HPC)/AI resources of Institut du développement et des ressources en informatique scientifique (IDRIS) under the allocation AD011011137R1 made by the Grand Équipement National de Calcul Intensif (GENCI). We estimate that our computations on GPUs to experiment different architectures, and to train and test the networks have generated about 520 kg eCO2. This includes 15 kg eCO2 for training the final PF network on LG − GC. Part of this work was performed using the computing facilities of the CC LBBE/Pôle Rhône-Alpes de bioinformatique (PRABI). The taxon silhouettes in Fig. 1 are modified from public domain images in the PhyloPic database.

Conflict of Interests

None declared.

Data Availability

The code for Phyloformer, the pretrained models, and all the datasets analyzed in this work can be found at https://github.com/lucanest/Phyloformer.

References

Abbe E, Bengio S, Boix-Adserà E, Littwin E, Susskind JM. Transformers learn through gradual rank increase. 2023. https://openreview.net/forum?id=qieeNlO3C7.

Atteson K. The performance of neighbor-joining methods of phylogenetic reconstruction. *Algorithmica*. 1999:25(2-3):251–278. https://doi.org/10.1007/PL00008277.

Azouri D, Abadi S, Mansour Y, Mayrose I, Pupko T. Harnessing machine learning to guide phylogenetic-tree search algorithms. *Nat Commun*. 2021:12(1):1983. https://doi.org/10.1038/s41467-021-22073-8.

Azouri D, Granit O, Alburquerque M, Mansour Y, Pupko T, Mayrose I, Rzhetsky A. The tree reconstruction game: phylogenetic reconstruction using reinforcement learning. *Mol Biol Evol*. 2024:41(6):msae105. https://doi.org/10.1093/molbev/msae105.

Balaban M, Sarmashghi S, Mirarab S. APPLES: scalable distance-based phylogenetic placement with or without alignments. *Syst Biol*. 2020:69(3):566–578. https://doi.org/10.1093/sysbio/syz063.

Blanquart S, Lartillot N. A site- and time-heterogeneous model of amino acid replacement. *Mol Biol Evol*. 2008:25(5):842–858. https://doi.org/10.1093/molbev/msn018.

Boussau B, Gouy M. Efficient likelihood computations with nonreversible models of evolution. *Syst Biol*. 2006:55(5):756–768. https://doi.org/10.1080/10635150600975218.

Bronstein MM, Bruna J, Cohen T, Velickovic P. 2021. Geometric deep learning: grids, groups, graphs, geodesics, and gauges, CoRR arXiv, arXiv:2104.13478, preprint: not peer reviewed. https://doi.org/10.48550/arXiv.2104.13478.

Clevert D, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs). In: Bengio Y, LeCun Y, editors. 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings 2016. http://arxiv.org/abs/1511.07289.

Csilléry K, Blum MG, Gaggiotti OE, François O. Approximate Bayesian computation (ABC) in practice. *Trends Ecol Evol*. 2010:25(7):410–418. https://www.sciencedirect.com/science/article/pii/S0169534710000662. https://doi.org/10.1016/j.tree.2010.04.001.

Desper R, Gascuel O. Theoretical foundation of the balanced minimum evolution method of phylogenetic inference and its relationship to weighted least-squares tree fitting. *Mol Biol Evol*. 2004:21(3):587–598. https://doi.org/10.1093/molbev/msh049.

Duchemin L, Lanore V, Veber P, Boussau B. Evaluation of methods to detect shifts in directional selection at the genome scale. *Mol Biol Evol*. 2023:40(2):msac247. https://doi.org/10.1093/molbev/msac247.

Felsenstein J. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol*. 1981:17(6):368–376. https://doi.org/10.1007/BF01734359.

Felsenstein J. *Inferring phylogenies*. Vol. 2. Sunderland, MA: Sinauer Associates; 2004.

Gascuel O. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol Biol Evol*. 1997:14(7):685–695. https://doi.org/10.1093/oxfordjournals.molbev.a025808.

Guindon S, Gascuel O. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol*. 2003:52(5):696–704. https://doi.org/10.1080/10635150390235520.

Hadfield J, Megill C, Bell SM, Huddleston J, Potter B, Callender C, Sagulenko P, Bedford T, Neher RA, Kelso J. Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics*. 2018:34(23):4121–4123. https://doi.org/10.1093/bioinformatics/bty407.

Harms MJ, Thornton JW. Evolutionary biochemistry: revealing the historical and physical causes of protein properties. *Nat Rev Genet*. 2013:14(8):559–571. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4418793/. https://doi.org/10.1038/nrg3540.

Ho J, Kalchbrenner N, Weissenborn D, Salimans T. 2019. Axial attention in multidimensional transformers. CoRR arXiv, arXiv:1912.12180, preprint: not peer reviewed. https://doi.org/10.48550/arXiv.1912.12180.

Höhler D, Pfeiffer W, Ioannidis V, Stockinger H, Stamatakis A. RaxML Grove: an empirical phylogenetic tree database. *Bioinformatics*. 2022:38(6):1741–1742. https://doi.org/10.1093/bioinformatics/btab863.

Höhna S, Landis MJ, Heath TA, Boussau B, Lartillot N, Moore BR, Huelsenbeck JP, Ronquist F. RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Syst Biol*. 2016:65(4):726–736. http://academic.oup.com/sysbio/article/65/4/726/1753608. Publisher: Oxford Academic. https://doi.org/10.1093/sysbio/syw021.

Huelsenbeck JP, Ronquist F. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*. 2001:17(8):754–755. https://doi.org/10.1093/bioinformatics/17.8.754.

Jiang Y, Balaban M, Zhu Q, Mirarab S. DEPP: deep learning enables extending species trees using single genes. *Syst Biol*. 2022:72(1):17–34. https://doi.org/10.1093/sysbio/syac031.

Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, *et al*. Highly accurate protein structure prediction with AlphaFold. *Nature*. 2021:596(7873):583–589. https://doi.org/10.1038/s41586-021-03819-2.

Kalyaanamoorthy S, Minh BQ, Wong TK, Von Haeseler A, Jermiin LS. ModelFinder: fast model selection for accurate phylogenetic estimates. *Nat Methods*. 2017:14(6):587–589. https://doi.org/10.1038/nmeth.4285.

Katharopoulos A, Vyas A, Pappas N, Fleuret F. Proceedings of the 37th international conference on machine learning. *PMLR*. 2020:119:5156–5165.

Kingma DP, Ba J. Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations; 2015 May 7–9; San Diego, CA, USA.

Kleinman CL, Rodrigue N, Lartillot N, Philippe H. Statistical potentials for improved structurally constrained evolutionary models. *Mol Biol Evol*. 2010:27(7):1546–1560. https://doi.org/10.1093/molbev/msq047.

Kuhner MK, Felsenstein J. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol Biol Evol*. 1994:11:459–468. https://doi.org/10.1093/oxfordjournals.molbev.a040126.

Le SQ, Gascuel O. An improved general amino acid replacement matrix. *Mol Biol Evol*. 2008:25(7):1307–1320. https://doi.org/10.1093/molbev/msn067.

Lefort V, Desper R, Gascuel O. FastME 2.0: a comprehensive, accurate, and fast distance-based phylogeny inference program. *Mol Biol Evol*. 2015:32(10):2798–2800. https://doi.org/10.1093/molbev/msv150.

Llinares-López F, Berthet Q, Blondel M, Teboul O, Vert J-P. Deep embedding and alignment of protein sequences. *Nat Methods*. 2023:20(1):104–111. https://doi.org/10.1038/s41592-022-01700-2.

Lueckmann J-M, Boelts J, Greenberg D, Goncalves P, Macke J. Benchmarking simulation-based inference. In: Banerjee A, Fukumizu K, editors. Proceedings of the 24th International Conference on Artificial Intelligence and Statistics of Proceedings of Machine Learning Research. Vol. 130. PMLR; 2021. p. 343–351. https://proceedings.mlr.press/v130/lueckmann21a.html.

Ly-Trong N, Naser-Khdour S, Lanfear R, Minh BQ. AliSim: a fast and versatile phylogenetic sequence simulator for the genomic era. *Mol Biol Evol*. 2022:39(5):msac092. https://doi.org/10.1093/molbev/msac092.

Minh BQ, Dang CC, Vinh LS, Lanfear R. QMaker: fast and accurate method to estimate empirical models of protein evolution. *Syst Biol*. 2021:70(5):1046–1060. https://doi.org/10.1093/sysbio/syab010.

Minh BQ, Schmidt HA, Chernomor O, Schrempf D, Woodhams MD, von Haeseler A, Lanfear R, Teeling E. IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era. *Mol Biol Evol*. 2020:37(5):1530–1534. https://doi.org/10.1093/molbev/msaa015.

Mo YK, Hahn MW, Smith ML. Applications of machine learning in phylogenetics. *Mol Phylogenet Evol*. 2024:196(11):108066. https://www.sciencedirect.com/science/article/pii/S1055790324000587. https://doi.org/10.1016/j.ympev.2024.108066.

Nelson MI, Viboud C, Simonsen L, Bennett RT, Griesemer SB, St. George K, Taylor J, Spiro DJ, Sengamalay NA, Ghedin E, *et al*. Multiple reassortment events in the evolutionary history of H1N1 influenza a virus since 1918. *PLoS Pathog*. 2008:4(2):e1000012. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2262849/. https://doi.org/10.1371/journal.ppat.1000012.

Norn C, André I, Theobald DL. A thermodynamic model of protein structure evolution explains empirical amino acid substitution matrices. *Protein Sci*. 2021:30(10):2057–2068. https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.4155. https://doi.org/10.1002/pro.v30.10.

Penel S, Arigon A-M, Dufayard J-F, Sertier A-S, Daubin V, Duret L, Gouy M, Perrière G. Databases of homologous gene families for comparative genomics. *BMC Bioinformatics*. 2009:10(S6):S3. https://doi.org/10.1186/1471-2105-10-S6-S3.

Perez-Lamarque B, Öpik M, Maliet O, Afonso Silva AC, Selosse MA, Martos F, Morlon H. Analysing diversification dynamics using barcoding data: the case of an obligate mycorrhizal symbiont. *Mol Ecol*. 2022:31(12):3496–3512. https://doi.org/10.1111/mec.v31.12.

Petti S, Bhattacharya N, Rao R, Dauparas J, Thomas N, Zhou J, RushAM, Koo P, Ovchinnikov S, Borgwardt K. End-to-end learning of multiple sequence alignments with differentiable Smith–Waterman. *Bioinformatics*. 2022:39(1):btac724. https://doi.org/10.1093/bioinformatics/btac724.

Price MN, Dehal PS, Arkin AP. FastTree 2–approximately maximum-likelihood trees for large alignments. *PLoS One*. 2010:5(3):e9490. https://doi.org/10.1371/journal.pone.0009490.

Prillo S, Deng Y, Boyeau P, Li X, Chen P-Y, Song YS. CherryML: scalable maximum likelihood estimation of phylogenetic models. *Nat Methods*. 2023:20(8):1232–1236. https://doi.org/10.1038/s41592-023-01917-9.

Rambaut A. Seq-Gen. 2017. http://tree.bio.ed.ac.uk/software/seqgen/.

Rao RM, Liu J, Verkuil R, Meier J, Canny JF, Abbeel P, Sercu T, Rives A. MSA transformer. In: Meila M, Zhang T, editors. Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research. Vol. 139. PMLR; 2021. p. 8844–8856. https://proceedings.mlr.press/v139/rao21a.html.

Redelings BD, Suchard MA. Incorporating indel information into phylogeny estimation for rapidly emerging pathogens. *BMC Evol Biol*. 2007:7(1):40. https://doi.org/10.1186/1471-2148-7-40.

Robinson DF, Foulds LR. Comparison of phylogenetic trees. *Math Biosci*. 1981:53(1-2):131–147. https://doi.org/10.1016/0025-5564(81)90043-2.

Saitou N, Nei M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*. 1987:4:406–425. https://doi.org/10.1093/oxfordjournals.molbev.a040454.

Schölkopf B, Smola AJ. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press; 2002.

Si Quang L, Gascuel O, Lartillot N. Empirical profile mixture models for phylogenetic reconstruction. *Bioinformatics*. 2008:24(20):2317–2323. https://doi.org/10.1093/bioinformatics/btn445.

Smith ML, Hahn MW. Phylogenetic inference using generative adversarial networks. *Bioinformatics*. 2023:39(9):btad543. https://doi.org/10.1093/bioinformatics/btad543.

Strimmer K, Von Haeseler A. Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. *Mol Biol Evol*. 1996:13(7):964–969. https://doi.org/10.1093/oxfordjournals.molbev.a025664.

Suvorov A, Hochuli J, Schrider DR. Accurate inference of tree topologies from multiple sequence alignments using deep learning. *Syst Biol*. 2019:69(2):221–233. https://doi.org/10.1093/sysbio/syz060.

Sy Vinh L, von Haeseler A. Shortest triplet clustering: reconstructing large phylogenies using representative sets. *BMC Bioinformatics*. 2005:6(1):92. https://doi.org/10.1186/1471-2105-6-92.

Szöllősi GJ, Höhna S, Williams TA, Schrempf D, Daubin V, Boussau B. Relative time constraints improve molecular dating. *Syst Biol*. 2022:71:797–809. https://doi.org/10.1093/sysbio/syab084.

Szöllosi GJ, Rosikiewicz W, Boussau B, Tannier E, Daubin V. Data from: efficient exploration of the space of reconciled gene trees. 2013. https://datadryad.org/stash/dataset/doi:10.5061/dryad.pv6df. Artwork Size: 45688314 bytes Pages: 45688314 bytes.

Tang X, Zepeda-Nuñez L, Yang S, Zhao Z, Solís-Lemus C. Novel symmetry-preserving neural network model for phylogenetic inference. *Bioinform Adv*. 2024:4(1):vbae022. https://doi.org/10.1093/bioadv/vbae022.

Telford MJ, Wise MJ, Gowri-Shankar V. Consideration of RNA secondary structure significantly improves likelihood-based estimates

of phylogeny: examples from the bilateria. *Mol Biol Evol*. 2005:22(4):1129–1136. https://doi.org/10.1093/molbev/msi099.

Thompson A, Liebeskind BJ, Scully EJ, Landis MJ. Deep learning and likelihood approaches for viral phylogeography converge on the same answers whether the inference model is right or wrong. *Syst Biol*. 2024:73(1):183–206. https://doi.org/10.1093/sysbio/syad074.

Trost J, Haag J, Höhler D, Jacob L, Stamatakis A, Boussau B, Crandall K. Simulations of sequence evolution: how (un) realistic they are and why. *Mol Biol Evol*. 2024:41(1):msad277. https://doi.org/10.1093/molbev/msad277.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. *Adv Neural Inf Process Syst*. 2017:30. https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Waterman MS, Smith TF, Singh M, Beyer WA. Additive evolutionary trees. *J Theor Biol*. 1977:64(2):199–213. https://doi.org/10.1016/0022-5193(77)90351-4.

Weisburg WG, Giovannoni SJ, Woese CR. The Deinococcus-Thermus phylum and the effect of rRNA composition on phylogenetic tree construction. *Syst Appl Microbiol*. 1989:11(2):128–134. http://www.ncbi.nlm.nih.gov/sites/entrez?Db=pubmed&Cmd=Retrieve&list_uids=11542160&dopt=abstractplus. https://doi.org/10.1016/S0723-2020(89)80051-7.

Yang Z. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Mol Biol Evol*. 1993:10:1396–1401. https://academic.oup.com/mbe/article-abstract/10/6/1396/988090. https://doi.org/10.1093/oxfordjournals.molbev.a040082.

Yang Z. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J Mol Evol*. 1994:39(3):306–314. https://doi.org/10.1007/BF00160154.

Yang Z. Among-site rate variation and its impact on phylogenetic analyses. *Trends Ecol Evol*. 1996:11(9):367–372. https://www.sciencedirect.com/science/article/pii/0169534796100410. https://doi.org/10.1016/0169-5347(96)10041-0.

Yang Z. PAML 4: phylogenetic analysis by maximum likelihood. *Mol Biol Evol*. 2007:24(8):1586–1591. https://doi.org/10.1093/molbev/msm088.

Zaharias P, Grosshauser M, Warnow T. Re-evaluating deep neural networks for phylogeny estimation: the issue of taxon sampling. *J Comput Biol*. 2022:29(1):74–89. https://doi.org/10.1089/cmb.2021.0383.

Zhao J, Zhang Y, Chen B, Schaefer FT, Anandkumar A. Incremental low-rank learning. 2023. https://openreview.net/forum?id=Xm9AvjEfdE.

Zhou X, Shen X-X, Hittinger CT, Rokas A. Evaluating fast maximum likelihood-based phylogenetic programs using empirical phylogenomic data sets. *Mol Biol Evol*. 2018:35(2):486–503. https://doi.org/10.1093/molbev/msx302.

Zou Z, Zhang H, Guan Y, Zhang J. Deep residual neural networks resolve quartet molecular phylogenies. *Mol Biol Evol*. 2020:37(5):1495–1507. https://doi.org/10.1093/molbev/msz307.