DEPP: Deep Learning Enables Extending Species Trees using Single Genes

Yueyu Jiang,[1] Metin Balaban,[2] Qiyun Zhu,[3] and Siavash Mirarab[1*]

[1] *Department of Electrical and Computer Engineering, UC San Diego, CA 92093, USA*
[2] *Bioinformatics and Systems Biology Graduate Program, UC San Diego, CA 92093, USA*
[3] *Center for Fundamental and Applied Microbiomics, Arizona State University, Tempe, AZ 85281, USA*
[*] Corresponding author: smirarab@ucsd.edu

Abstract

Identifying samples in an evolutionary context is a fundamental step in the study of microbiome, and more broadly, biodiversity. Extending a *reference* phylogeny by placing new *query* sequences onto it has been increasingly used for sample identification and other applications. Existing phylogenetic placement methods have assumed that the query sequence is homologous to the data used to infer the reference phylogeny. Thus, they are designed to place data from a single gene onto a gene tree (e.g., they can place 16S sequences onto a 16S gene tree). While this assumption is reasonable, ultimately, sample identification is a question of identifying the species not individual genes. The placement of single gene data on a gene tree is therefore used as a proxy for a more ambitious goal: extending a *species tree* given sequence data from one or more gene. This goal poses difficult algorithmic questions. Nevertheless, a sufficiently accurate solution would not only improve sample identification using marker genes, it would also help achieving the long-standing goal of combining 16S and metagenomic data. We approach this problem using deep neural networks (DNN) and introduce a method called DEPP. Given a reference species tree and sequence data from one (or a handful of) genes, DEPP learns how to extend the species tree to include new species. DEPP does not rely on pre-specified models of sequence evolution or gene tree discordance; instead, it uses highly parameterized DNNs to learn both aspects from the data. We test DEPP both in simulations and on real microbial data and show high accuracy.

Keywords: Phylogenetic placement, Neural networks, Deep learning, Gene tree discordance, Microbiome analyses

1 Introduction

In recent years, phylogenetic inference has found widespread application in identifying organisms that make up a biological sample [1–3]. Microbiome analyses often rely on phylogenetic analyses to identify species present in an environment sampled using the 16S marker gene or genome-wide (metagenomic) data [4–10]. By using the phylogenetic context, we can identify species even when exact matches to the reference datasets are not present. Similarly, outside microbiome analyses, identifying new and known species using (meta)barcoding and genome skimming data both rely on phylogenetic analyses [11–14].

In these high-throughput applications, phylogenetic placement [15–19] – adding a new sequence onto an existing reference tree – is sufficient, and the more challenging *de novo* reconstruction is neither necessary nor always desirable [20]. Phylogenetic placement has a long history [21–23], and maximum likelihood placement methods [15–17] are widely-used in microbiome analyses. However, the focus of these algorithms is to add sequences from a single gene onto a tree showing its evolutionary history (i.e., the gene tree). Phylogenetic relationships change across the genome [24, 25], and accounting for such discordance is a subject of much recent method development [26]. Given data from individual genes, it must be assumed that they have evolved on a gene tree, not the species tree. Thus, existing methods place on gene trees but use the gene tree as a proxy for the species tree, hoping that their differences is not consequential. For example, species identification using marker genes such as 16S or COI [27,28]

implicitly assumes that the gene tree is close enough to the species tree to allow accurate identification of the *species* by placement on the gene tree. More recently, inserting new genomes onto a species tree by minimizing quartet distance has become possible [18], but this approach requires genome-wide data and complicated analyses pipelines.

Users of phylogenetic placement often face a question: given query sequence data from a single gene (or a handful of genes), is it possible to place the query onto the species tree (a goal we name *discordant placement*). The correct position of a query on the species tree is not always determinable from a single gene. Nevertheless, the gene tree is related to the species tree, and gene data have *some* information about the correct placement on the species tree, giving us hope that discordant placement can be achieved with sufficient if imperfect accuracy.

The ability to extend a species tree using single-gene data would be tremendously useful in studying ecology. Microbiome analyses are split between metagenomic and 16S data, which remain mostly disconnected. Accurate discordant placement methods would enable researchers to add 16S samples onto species trees and treat them as if they were metagenomic samples, albeit with less signal and more uncertainty. Thus, it would become possible to combine their 16S and metagenomic data (Fig. 1) in downstream analyses such as sample differentiation [29,30] and taxonomic profiling. Similarly, the ability to place eukaryotic species sampled only through their barcode genes (e.g., COI) on species trees will hugely benefit research in ecology.

How can we approach the discordant placement problem? One solution is to misuse the existing methods and simply assume the species tree and the gene tree are the same. However, it is unclear how these methods are expected to behave when the tree and sequence data are incompatible. The standard phylogenetics methods are model-based: they assume data has evolved on a tree according to a model of sequence evolution (e.g., GTR [31]) and infer the tree using maximum likelihood, Bayesian sampling, or model-corrected distances [26]. While this paradigm has enjoyed much success, its reliance on explicit models can make it less accurate in the face of deviations from assumptions of the model [32–36]. Using existing methods for the discordant placement problem would lead to a mis-specified model, which may result in low accuracy. We suggest that instead of relying on the dominant paradigm, discordant placement can be solved using machine learning.

As early as the 1990s, there were attempts to approach phylogenetic inference using general purpose models from machine learning models such as neural networks. Dopazo *et al.* [37] formulated the problem as unsupervised learning, and designed a neural network that reflected the tree shape. More recently, the success of deep neural networks (DNN) in solving other challenging problems has motivated new efforts to adopt DNNs in phylogenetics. Zou *et al.* [38] and Suvorov *et al.* [39] have used supervised learning and a two-step approach: for every four species (quartet), classify the input data to one of the three possible quartet topologies, then combine these quartet trees. The supervised learning formulation raises a question: where are we to find labeled training data in the high volume needed by DNNs? Both papers turn to simulations for an answer: use complex models to simulate data on known trees, from which we then train the model. The point of these methods is to use complex generative models that can be sampled but do not avail themselves to scalable inference. However, learning from simulated data runs the risk of missing relevant parts of the huge parameter space and model misspecification. As Zaharias *et al.* [40] recently showed, these methods have lower accuracy than standard methods in careful benchmarking.

Phylogenetic placement offers a way to use general purpose models without simulations. Given a reference tree, however computed, and sequence data which are a function of the tree, we can use the reference data to train a machine learning model (such as a DNN) that can place a query sequence onto the reference tree. The reference tree may be a species tree inferred using large numbers of genes, using complex models, and perhaps after spending much computational resources. Such reference data are increasingly available. For example, several comprehensive trees were published in the past year on tens of thousands of microbial species [10,41,42] using

120 to 400 genes, with analyses that took $> 200,000$ hours of CPU and GPU time in one case. These available trees are excellent candidates for providing the training data.

Why would we need to turn to black-box machine learning models for discordant placement? Note that discordant placement only poses a weak connection between sequence data and the tree: the sequence data should be a function of the tree (but not only the tree). In contrast, traditional models require that sequence data have evolved on the tree according to a specific Markov model, which is a much stronger assumption. A compelling reason to use machine learning is that it provides ways to learn general functions (here from sequences to positions on trees). Thus, they have the potential to enable extending species trees using single genes. By avoiding explicit mechanistic models, machine learning has the potential to build general models that map sequences onto trees even when the tree and the sequences are not fully compatible. Such a model would, in effect, simultaneously place sequences onto a gene tree and *reconcile* [43] the differences with the species tree.

In this paper, we introduce the Deep-learning Enabled Phylogenetic Placement (DEPP) framework (Fig. 1). Given a reference tree, inferred in any way, and some sequence data labeled by leaves of the tree but potentially evolved on a tree incongruent with the reference tree, DEPP learns a neural network to embed sequences in a high dimensional Euclidean space, such that pairwise distances in the new space correspond to the square root of tree distances. Given such a model, the placement of new sequences can proceed by computing the embedding, computing distances, and using distance-based phylogenetic placement [19].

2 Methods

Let T denote a rooted phylogenetic tree on a set of n taxa $\{t_i\}_{i=1}^n$ represented as leaves and each branch labelled with its length. The tree T defines a distance matrix where each entry $d_{ij} \in \mathbb{R}^+$ corresponds to the path length between leaves i and j. A distance matrices may or may not be equal to that of some tree, but when it does match a tree, it matches a *unique* tree and is called additive [44]. Assume we are given a set of sequences S, and each $s_i \in S$ corresponds to a leaf t_i. Computing distance between sequences produces a sequence distance matrix. Assuming specific Markov models of sequence evolution, we can define phylogenetic distances that converge to additivity as sequence lengths increase [45]. For example, under the Jukes Cantor (JC) model [46], $\frac{3}{4}\ln(1-\frac{4}{3}h)$ asymptotically converges to additivity where h is the hamming distance between sequences.

The vanilla phylogenetic placement problem takes as input (i) a reference tree T and its associated sequences, and (ii) a query sequence q; it outputs a placement of q, which consists of a specific position of a particular edge of T and the length of a new terminal branch. While past methods were based on likelihood [15–17], recently, a distance-based least-squares phylogenetic placement (LSPP) was formulated (Fig. 1) and solved in a tool called APPLES [19]. It finds the placement with $\text{argmin}_T \sum_{i=1}^n w_i(D_i - d_i(T))^2$ where $d_i(T)$ represents tree-based distances to query, and D_i represents sequence-based distances between the query and the taxon i, and w_i is a weight, set by default to D_i^{-2} according to the FM method [47].

The vanilla placement problem assumes that the sequence data used for inferring the reference tree T and inserting the query are identical. Here, we break this assumption. We assume we are given a *reference* tree T (typically a species tree) inferred from *any* source of data and sequence data S (typically from a single gene) that were not necessarily used to infer T but are influenced by T. Thus, traditional methods for obtaining distances from S would not match d_{ij}. The discordant phylogenetic placement seeks to find the best placement of the query on T using S *despite* disagreements between S and T. When the discordance between data and the reference tree is because T is a species tree whereas the sequence data come from a single gene, the ideal placement is defined by the true placement of the species on the species tree.

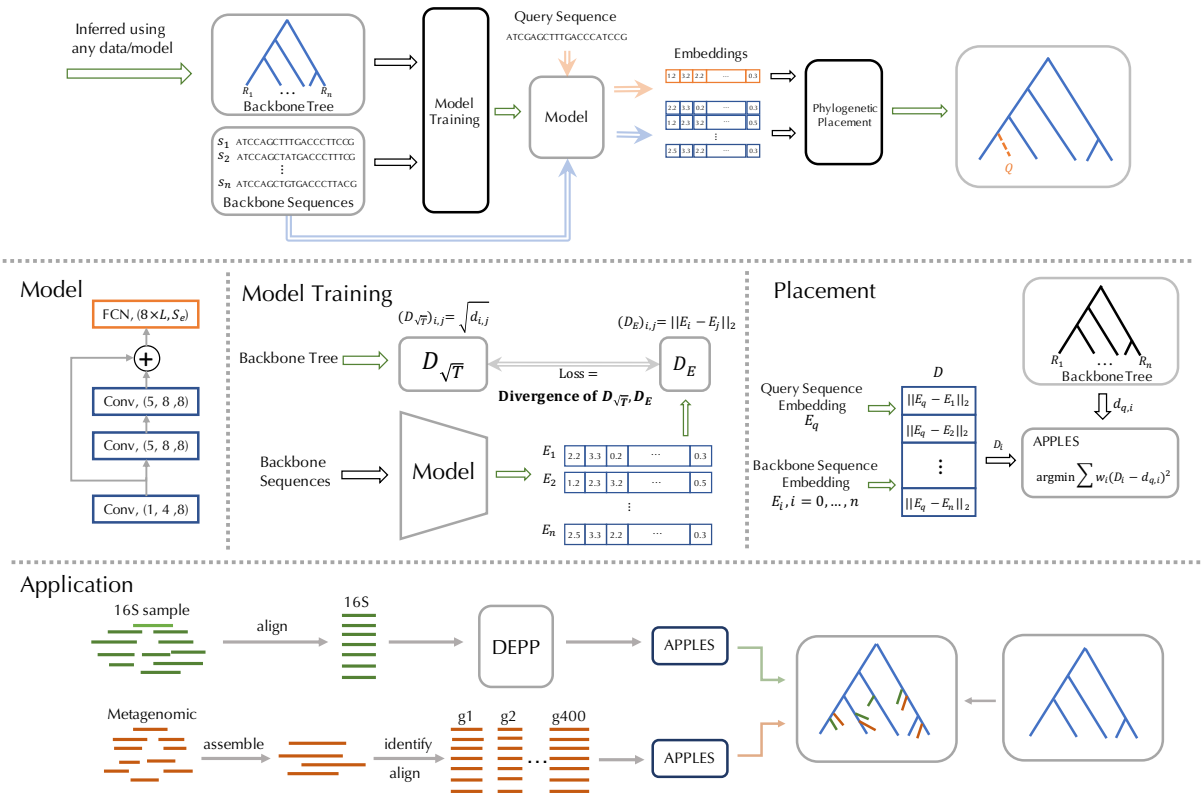


Figure 1: **Overview of the method.** Top panel shows the overall process. A neural network model is trained from a given reference tree, however inferred, and reference sequences labeled by the leaves of the tree. The model maps sequences to embeddings in $\mathbb{R}^k$ dimensions. Once sequences are embedded, phylogenetic placement can be performed using distances between embeddings of the query and reference species. Middle panel shows the structure of the CNN model and both training and placement processes. The training process minimizes the difference between Euclidean distances in the $\mathbb{R}^k$ space and the square root of tree distances across all pairs of species. The placement process computes Euclidean distances between embeddings of the query and those of reference sequences, and the resulting distance vector is used within a (weighted) least square error minimization framework (e.g., APPLES) for placement. Bottom: Using DEPP, we can combine various types of samples by adding them onto the same tree.

2.1 DEPP Algorithm

We formulate phylogenetic inference as a supervised learning problem. The training data are T and S, using which, parameters of some model are learned. This model is then used to add query sequences to the tree (Fig. 1). The distance-based LSPP problem provides a clean mathematical formulation. In the LSPP framework, the training dataset can be expressed as $\{(s_i, s_j), d_{ij}\}$ for all pairs of i and j; the training process should learn a mapping from data to pairwise distances. Once the model is trained, we can map a given query sequence q to a vector of distances $d_{q1} \dots d_{qn}$ and can solve LSPP using APPLES to add q to T.

Sequence embedding. To map sequences to distances, we build a neural network model that embeds sequences in the $\mathbb{R}^k$ space and use distances between embeddings. This idea enjoys a strong theoretical justification. As de Vienne et al. showed [48] and Layer and Rhodes elaborated [49], for any tree T, there *exists* a collection of points $P = \{\Phi(t_i)\}_{i=1}^n$ in the $\mathbb{R}^{n-1}$ Euclidean space such that the distance between the points $\Phi(t_i)$ and $\Phi(t_j)$ equals to $\sqrt{d_{ij}}$. Thus, if sequences are a function of the tree, there must exist an embedding that corresponds to the tree. The challenge is finding that embedding, a problem amenable to deep learning. Thus, our learning objective is to build a model that given a sequence s embeds it as $\Phi(s)$.

We treat T and S as training data and seek a model that maximizes the match of Euclidean

distances between embeddings and the square root of phylogenetic distances in the reference tree (i.e., $\sqrt{d_{ij}}$). To make this precise, we need to define a measure of matrix similarity. While any metric can be used (and measures such as log-determinant divergence have shown promise [50]), here, we simply use mean squared error, seeking:

$$\arg\min_{\Phi} \sum_{i,j} (\|\Phi(s_i) - \Phi(s_j)\|_2 - \sqrt{d_{ij}})^2$$

where $\Phi : \{A, C, G, T\}^L \to \mathbb{R}^k$ is an embedding of sequences in the Euclidean space, and d_{ij} give pairwise path distances in the reference tree T. While we focus on nucleotides here, a similar formulation can be used for amino acid sequences or any type of character data. Note that the estimated distance of i and j is $(\|\Phi(s_i) - \Phi(s_j)\|_2)^2$.

Motivated by strong evidence in distance-based phylogenetics that weighting down long distances improves accuracy [19, 22, 51], we define a weighted version of the objective function:

$$\arg\min_{\Phi} \sum_{i,j} \frac{1}{d_{ij}^2} (\|\Phi(s_i) - \Phi(s_j)\|_2 - \sqrt{d_{ij}})^2 \ . \tag{1}$$

Note that other weightings schemes [22, 47, 52] can be used and will be tested.

The result of Layer and Rhodes (LR) requires $n - 1$ dimensions, which introduces some challenges. According to the theory, the number of dimensions needs to increase by one after inserting the query. Our supervised learning formulation does not allow that (the embedding size is fixed after training). Thus, there is no guarantee that the embeddings remain correct after addition, even if they are before addition. However, we note that, in LR embeddings, adding a leaf would require simply dividing one of the $n - 1$ dimensions into two dimensions, leaving the rest of the embeddings intact. Thus, one can hope that having one less dimension has a minimal practical impact. More broadly, for large n, training models with n-dimensional embedding is impractical. Thus, we often set $k < n - 1$, and the gap can be more than an order of magnitude for some of our tests described below.

CNN. We use a convolutional neural network (Fig. 1). Nucleotide sequences are encoded using 4-bit one-hot binary vectors (gaps are encoded as all zeros); we refer to each bit as a channel. To accommodate reference species that have multiple copies of a gene, we change the encoding, so that instead of a binary vector, it includes the frequencies of the nucleotide characters among the gene copies. These encodings provide the input "features" that are processed through three linear convolutional layers, each followed by a nonlinear layer and a fully-connected linear layer.

As detailed in Appendix A, a convolutional layer applies a set of parametrized kernels by convolving them across its input (i.e., using the dot product of the kernel entries and the input). Convolutional layers are usually used as feature extractors, and multiple layers are used to detect high-level abstraction from the input. Here, we use them to enable the model to go beyond the traditional i.i.d models of sequence evolution and capture k-mer signatures. The first convolutional layer takes as input L features, each encoded as four channels, and outputs an $8 \times L$ matrix by applying a kernel size of 1 (but applied to all four input channels). The next two convolutional layers each have a kernel size of 5 (i.e., operating on 5-mers). Input of each layer is padded with zeros on both sides so that the output has the same length as the input. The input to the second convolutional layer is added to the output of the third convolutional layer, forming a residual block, which is a cornerstone of deep learning [53]. Using residual blocks can help solve the problem of gradient vanish, which is why they are commonly used, including for sequence data analyses [54, 55]. To enable the model to capture nonlinear relations, after each convolutional layer, we use a nonlinear layer built using the continuously differentiable exponential linear unit, which has performed well in other contexts [56]. The last layer is fully connected, taking features from outputs of convolutional layers and producing the final

embeddings. This layer aggregates the signal from convolutional layers. Each activation in the output is connected to all the inputs, and the output is a weighted sum of all the inputs.

Training process. We trained the model with Eq. 1 as the loss function using the stochastic gradient descent algorithm RMSProp, which divides the gradient by a running average of its recent magnitude to speed up training [57]. The batch size is fixed to 32. We check the training loss each 50 epochs and stop the training when the value of the loss function fails to decrease in two consecutive checks. The model with the optimal objective function value is chosen.

Distance correction. Our results below showed a bias towards underestimating distances in the testing set but not the training set (we will discus this later). We can attempt to *correct* the bias in the distances in a *post-hoc* manner if the reference tree is ultrametric. Note that making any references tree ultrametric with or without fossils is possible using model-free methods such as LSD [58] and wLogDate [59]. Given an ultrametric reference tree, unless the query species falls as an outgroup to all the species in the reference tree, we know exactly the maximum distance of the query to all reference species. We assume the scenario where the query falls outside the tree does not happen; that is, we assume the reference tree is comprehensive (e.g., includes at least one archaea and one bacteria). We denote the diameter (twice the height) of the ultrametric tree as d_M. At the test time, for each query q, we first use DEPP with uncorrected distances to infer to which side of the root the query belongs. Then, we compute the median distance from q to all leaves in the *other side* and call this distance d_Q. If our initial placement is on the correct side of the root and the query distances are consistent with height of the ultrametric tree, then d_Q should match d_M. To the extent that d_Q diverges from d_M, it is a bias that we can *correct* by a simple scaling. Thus, after obtaining distances $d_{q1} \dots d_{qn}$ of q to the reference set, we correct each distance to $d_{qi}\frac{d_M}{d_Q}$. With this correction, the median of the tree diameter based on computed distances would match the diameter of the backbone tree.

2.2 Experimental setup

We used three datasets to evaluate DEPP.

Simulated dataset. We use an existing simulated dataset [60], with 200 ingroup species and gene trees that are discordant with the species tree due to incomplete lineage sorting (ILS). The dataset contains model conditions corresponding to high, medium, and low ILS, each with 50 replicates. We arbitrarily selected the first 1, 2, 4, ..., 32 genes for each replicate. Two replicates of high discordance case where all gene sequences are identical are excluded.

Tree-of-life (TOL) marker gene. Zhu *et al.* [61] built a species tree of 10,575 prokaryotic genomes using ASTRAL from 381 marker genes. We dated this TOL tree using wLogDate [59] with a calibration point at the root of Cyanobacteria at 2.55 billion years ago. We categorized the marker genes into three equal-sized groups based on the rank of quartet-distance [62] between their gene tree and the species tree. For each discordance category, we selected 10 most commonly present genes among all species (Table S3). The mean gene tree discordance with the species tree, measured using the quartet distance, is 0.18, 0.34, 0.50 in low, medium, and high discordant groups. Gene alignments are available from Zhu *et al.* [61].

TOL amplicon. Due to their wide usage in amplicon sequencing, we also tested 16S and 5S genes from the same set of 10,575 microbial genomes. Genes were predicted using RNAmmer [63] and aligned using UPP [64]. In genomes with multiple copies of 16/5S, we train and test DEPP on all copies, but the backbone tree for APPLES+JC has branch lengths estimated from an arbitrary copy. For 16S, we also perform analyses with three regions commonly used in amplicon sequencing: V3+V4, V4 (100bp), and V4 (150bp). When using subsets, sometimes two species that are not sisters have identical sequences. Branch length re-estimation under the JC model cannot handle such cases; thus, we remove such species (between 367 and 934 in different cases) to allow comparison but also show DEPP results with those included. In addition, we removed from 16S datasets any sequence that was shorter than half of the average sequence length (removing less than 1% of species; see Table S5).

On these datasets, we evaluate the performance of four methods. Because no method other than DEPP designed for discordant placement using a single gene exist, we compare DEPP to other vanilla placement methods.

EPA-ng. This maximum likelihood method [17] is widely used but is *not* designed for discordant placement. Nevertheless, we test it for placing gene data on the species tree, with branch lengths re-estimated using RAxML, under the GTR $+\Gamma$ model.

INSTRAL. This method updates a species tree given input gene trees (already updated to include the query) and accounts for discordance by maximizing the quartet score [18]. Here, input gene tree are inferred using FastTree-II [65]. While INSTRAL does account for discordance, it requires *at least two* genes and is designed for cases with many genes.

APPLES+JC. We use APPLES in its default settings where it computes distances using Jukes Cantor (JC) model [46], chosen because Balaban *et al.* found no evidence that more complex models improve accuracy [19]. The branch lengths of reference trees are re-estimated based on the JC model using FastME [66] or FastTree-II [65] (Table S4). APPLES also includes two options d_m and b_m to set $w_{qi} = 0$ (i.e., ignore distances) for at most $n - b_m$ reference taxa per query when $\delta_{qi} > d_m$. We use $d_m = 0.2, b_m = 25$ for all datasets.

DEPP. We implemented DEPP using PyTorch and treeswift python packages [67] and trained the models on 2080Ti NVIDIA GPUs. The embedding size is set to 128 for simulated data and 512 for the real TOL dataset. Other hyperparameters are fixed to their defaults unless otherwise specified (see Table S2). DEPP is trained on the reference tree and is used to compute distances that are then fed to APPLES. Reference trees in our experiments are all ultrametric, allowing us to use the distance correction. Given more than one gene, we simply concatenate them. APPLES is used identically to JC, except that we adjust d_m to match branch length units: 2×10^5 generations for simulated data (where tree heights are $5 \times 10^5, 2 \times 10^6$, and 10^7 for high, medium, and low discordance) and 0.25 (billion years) for the TOL dataset (height: 4.2 billion years).

Evaluation procedure. To ensure query sets (i.e., testing data) are separate from the training data, we removed 5% of species (10 for simulations and ≈ 445 for real data) from the species tree to obtain reference trees. These left-out species are used as the query. The reference tree is the true species tree for the simulated dataset and the ASTRAL tree from the TOL dataset [61]. For testing, each query taxon is placed independently, and the result is compared against the full reference tree before pruning the query (i.e., the true tree for simulations and the ASTRAL tree for TOL). The error metric we report is the number of edges between the position on the reference tree and the inferred placement, computed using the Dendropy package [68]. In total, we have 8,941 test cases for the simulated and 14,239 test cases for the TOL dataset. INSTRAL fails in 39/8941 tests; we exclude these cases for INSTRAL (but not others).

3 Results

3.1 Simulated dataset

DEPP parameters and running time. As the training epochs advance, the loss function (1) drops rapidly and stabilizes after around 500 epochs in a typical case (Fig. S1). Here, training, which is a one-time process for each reference tree, finishes in around 20 minutes on a machine with one 2080Ti NVIDIA GPU and 8 CPU cores; placement of 10 queries takes 20 seconds. DEPP is not very sensitive to its hyperparameter settings. It becomes significantly less accurate if the distance correction is turned off (p-value: 0.036; Table S2). However, it is mostly robust to weighing schema, numbers of layers in the network, and the embedding size (Table S2). To see statistically significant reductions in accuracy on a single-gene case, the embedding size has to be dropped to 16. These results motivated the choice of default settings.

Comparison to other methods. On the simulated data, DEPP generally outperforms alternative methods when given a single gene (Fig. 2a). However, the absolute error and the

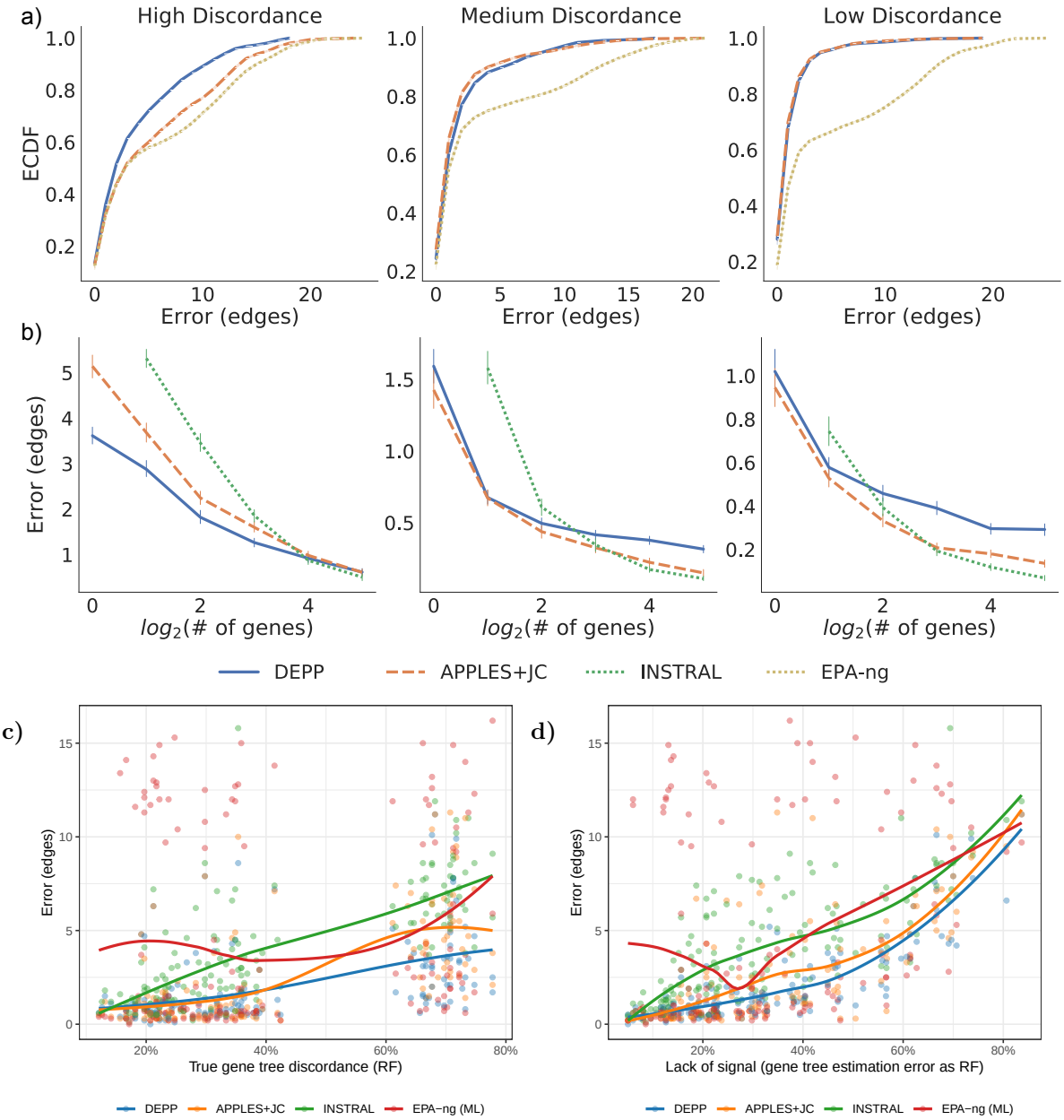


Figure 2: **Results on simulated datasets.** a) Empirical cumulative distribution function (ECDF) of the placement error on a single gene for high, medium, and low discordance (69% 34%, and 21% mean RF). INSTRAL needs at least 2 genes. b) Mean and standard error of placement error versus the number of genes. EPA-ng, which was much worse than other methods on average, is shown in Figure S2. (c,d) Sensitivity to gene properties. The impact of (c) true gene tree discordance (RF distance between true gene trees and the species tree) and (d) gene signal (RF distance between true gene tree and the estimated gene tree) on error on the single-gene data, combining all discordance levels. Dots represent the mean of 10 queries for each of 148 replicate datasets. See Fig. S4 for sensitivity to query properties.

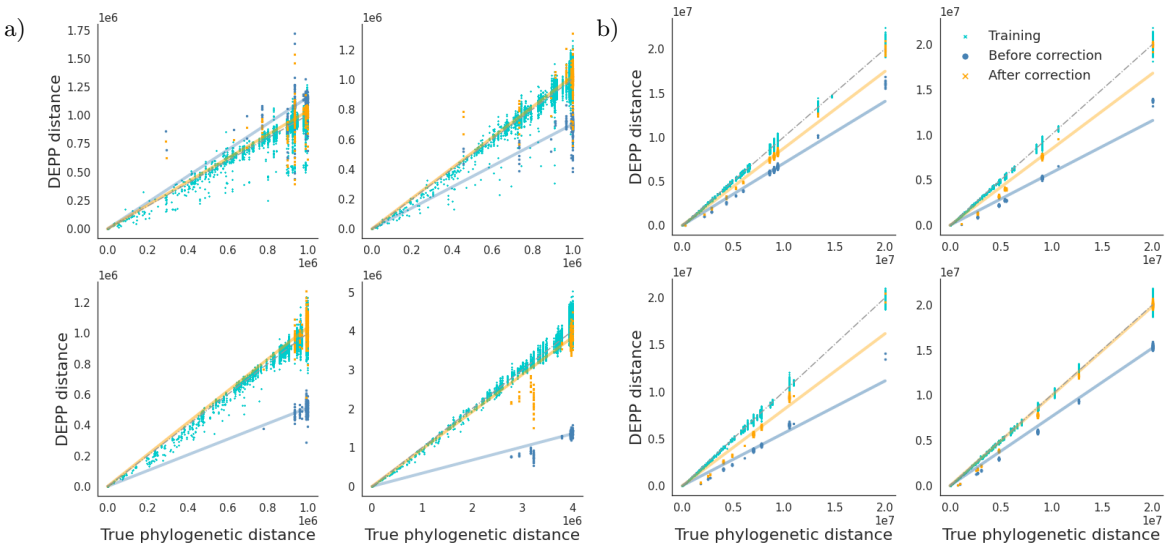


Figure 3: Examples of phylogenetic distance calculation using DEPP. Phylogenetic distance on the true species tree (in units of numbers of generations) versus the distance calculated by DEPP for (a) examples with high error (11, 8, 4, and 7, clockwise) and (b) examples with zero error. Results are shown for training data and testing data before and after the correction, with lines fitted to testing data (with non-negative intercept). Dashed gray line: identity line. High errors correspond to long terminal and short internal branches; see Fig. S3 for trees.

relative ranking of methods both depend on the level of gene tree discordance. When discordance is high, DEPP has the best accuracy among the methods, with the mean error around 3.6 edges. For context, random placements on these species trees would give a placement error of 14 edges on average (empirically computed). While DEPP and APPLES+JC have low errors (e.g., up to two edge error) with about the same frequency, DEPP has fewer cases with very high error. When the gene tree discordance level is medium (or low), DEPP and JC models have similar performance, and the error of both methods is as low as 1.5 edges (or 1 edge) on average. EPA-NG, which is not designed for discordant placement, has unusually high error on *some* trees and reasonable error elsewhere (Figs. 2a and S2); thus, we exclude it from other analyses.

As the number of genes increases, unsurprisingly, the mean errors of all methods reduce (Fig. 2b). For example, all methods converge to ≈0.5 edges error with 32 genes and high discordance. Both DEPP and APPLES+JC are more accurate than INSTRAL for low numbers of genes (≤ 4) but not for larger numbers of genes. In fact, as the number of genes increases to 32, INSTRAL starts to have the best accuracy, a result consistent with the theory as INSTRAL is statistically consistent under conditions simulated here [18].

Drivers of the error. To examine the causes behind performance patterns, we study the impact of two factors on error: true gene tree discordance and the strength of the phylogenetic signal. Since these factors change from one gene to another, we focus on results using a single gene. The amount of gene tree discordance, measured as the Robinson Foulds (RF) distance [69] between true gene trees and species trees, affects all methods, especially when discordance is very high; however, DEPP is impacted less than other methods (Fig. 2c). The signal is a function of many factors, including sequence length, tree height, and the rate of evolution. When a gene has weak signal, it is hard to place it, even on the gene tree. To quantify lack of signal, here, we use the RF distance between true gene trees (which we know in simulations) and those estimated using FastTree-II [65]. All methods experience a sharp rise in the error when phylogenetic signal weakens, a pattern most pronounced at the lowest levels of signal (Fig. 2d). According to an ANOVA test (Table S1), both gene tree discordance and signal have a significant impact on the

placement error ($p-$value$< 10^{-3}$). However, for DEPP, the signal explains 33.2% of the total variance in error, and gene tree discordance explains only 0.7% of the remaining variance. For the JC model, the signal explains 24.3% of the variance, and discordance explains 5.9% of the rest. Thus, compared to the JC model, DEPP is more robust to gene tree discordance but not to a lack of signal.

We further examine the example cases where DEPP performs well or poorly. We compare true phylogenetic distances versus distances computed by DEPP for low and high error cases (Fig. 3). Both high error and low error cases seem to result in unbiased distances computed in the training set; however, high error examples have much higher variance. The high variance can be attributed to a lack of signal: two identical sequences in the gene may belong to different parts of the species tree, a problem that the model cannot overcome. At the time of testing (e.g., for a query), distances before correction are systematically underestimated in most cases (see Discussions). Our distance correction for ultrametricity goes a long way in reducing that bias but does not eliminate it. In cases with high error, DEPP distances can be far from the true distances, and a large range of values are estimated for pairs with equal true distances. In some cases, DEPP assigns small distances to some reference species that have high distances to the query (Fig. 3; bottom left).

Examining example trees shows that these cases of high error correspond to "novel" query taxa that are on long branches on sparsely sampled clades (Fig. S3). Comparing the error of DEPP to the length of the terminal branch of the query, we observe that the longest branches lead to high error (Fig. S4); however, it appears that shortest of branches also have high error, perhaps because distinguishing very short branches requires strong signal. Similarly, queries that have a large clade as sister tend to be more difficult for DEPP when gene tree discordance is high (Fig. S4). Trees that lead to high error tend to have long terminal branches and short branches close to the root, a condition that corresponds to rapid radiations; in contrast, easy cases are those with shorter terminal branches and long branches closer to the root (Fig. S3).

3.2 The real TOL dataset

DEPP obtains substantial improvements compared to APPLES+JC on the real ToL dataset (Fig. 4). Since EPA+ng did not substantially outperform APPLES+JC in the simulations (and running EPA-ng on backbone trees with 10,575 species is challenging, if not impossible), here, we only compare to APPLES+JC.

Observed queries. A scenario where the query sequences is identical to a species present in the reference tree is expected to happen reasonably often for large backbone trees such as ToL. Under such conditions, DEPP has close to perfect accuracy for all genes except 16S (Figs. 4a and S5). For 16S, which can have multiple copies per species, the error slightly increases; DEPP find the correct placement in 89% of cases but in average has and error of 0.56 edges. Such errors are inevitable if the multiple copies of 16S are very different. Patterns of error do not change whether we train DEPP using weighting scheme described before or by simply selecting an arbitrary 16S copy (Fig. S6). While APPLES+JC also had low error for many genes, for eight genes, it had considerable levels of error on the training datasets (Fig. S5).

Novel queries. In the most interesting case when the query sequences are novel (i.e., are not in the training set), DEPP greatly outperforms APPLES+JC (Figs. 4a). On average, the placement error of DEPP (2.71 edges) is about half of APPLES+JC (5.52 edges). Moreover, while JC and DEPP find the correct placement about the same number of times, DEPP is less often far away from the correct placement compared to JC (Fig. S7). For example, in 89.2% of cases classified as low discordance, placement error of DEPP is 4 edges or lower while only 74.1% of JC69 placements are within 4 edges; to get 89% of the cases covered, we need to increase the error tolerance to 17 edges for JC. Given the ambitious goal of updating a species tree using a

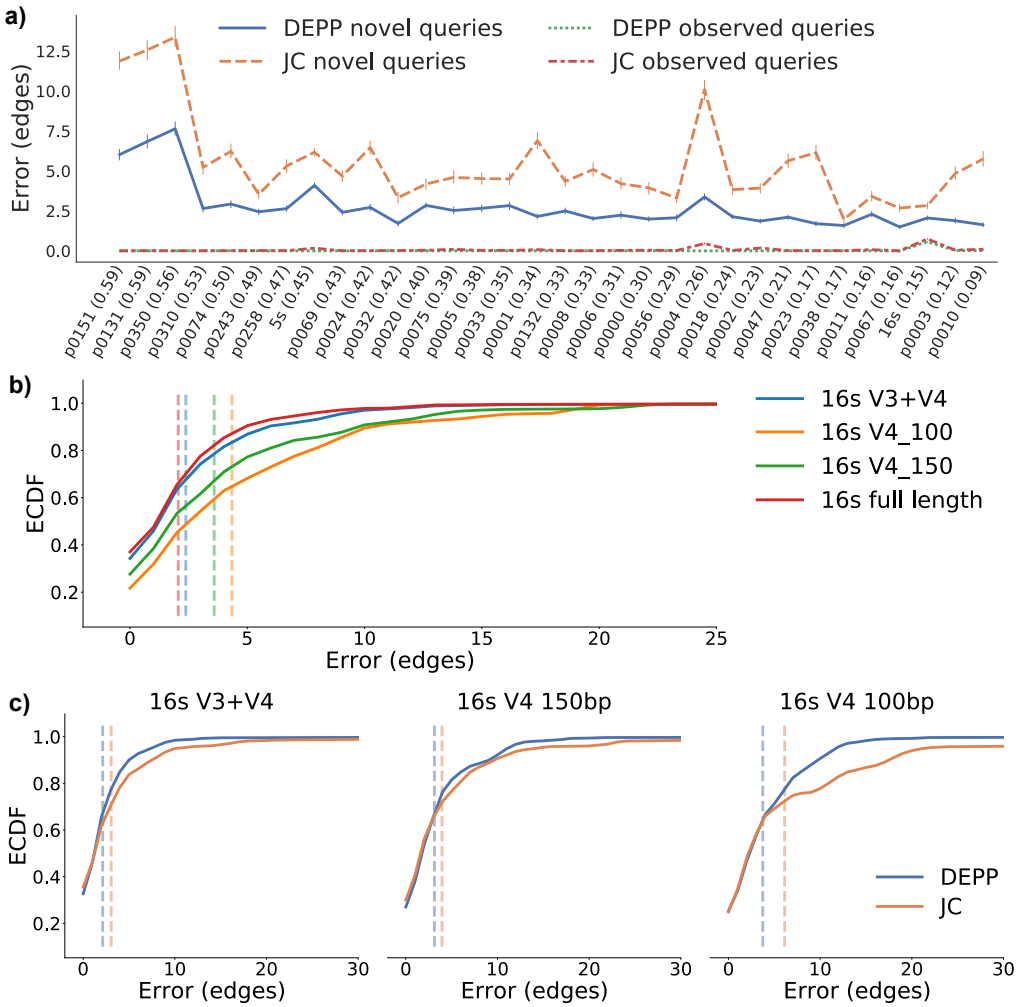


Figure 4: **Results on the TOL dataset.** a) Mean and standard error of placement error for APPLES+JC and DEPP, applied to novel queries (testset) and known queries (trainset), on the species tree from TOL dataset, which is treated as the ground truth. x-axis shows genes, ordered by their quartet distance to the reference species tree (shown parenthetically). For novel queries, the placement is performed in a leave-out fashion. For Empirical Cumulative Distribution (ECDF) of the placement error, see Fig. S7. b) ECDF of placement error on the 16S gene with full length or individual regions of 16S used in amplicon sequencing. V3+V4 regions is $\approx$ 400bp. Dashed lines show mean error. c) APPLES+JC versus DEPP on 16S regions. We cut x-axis at 25, but see Fig. S11 for the full range.

single gene, the accuracy of DEPP is reasonably high for practical applications. Across all 32 genes, in 78% of tests, DEPP finds a placement within 3 edges of the correct placement; for the full-length 16S gene, this value is 82%. A random placement on this trees with 10,575 leaves is, on average, 26 edges away from the correct placement. Our results are also consistent with using 16S as a marker gene, which among the 32 genes had among the lowest mean error rates (2.06 edges). Finally, note that similar to simulated datasets, accuracy of DEPP is a function of the accuracy of its distances. Calculated distances have very little bias and high variance where DEPP works well (Fig. S8) but high variance and bias when it works poorly (Fig. S9).

Fragments. When 16S analysis is restricted to fragments typically used in microbiome analyses, DEPP is able to retain high accuracy as long as the region is long enough (Figs. 4b). For V3+V4 region, which is $\approx$ 400bp long, DEPP has only a slight (0.3 edge on average) increase in error, showing the entire 16S is not needed for accurate placement. However, the more restricted sub-regions of V4 with 150bp and especially 100bp do show a reduction in accuracy.

APPLES+JC has higher error than DEPP when given fragments, especially when given 100bp reads from V4 (Figs. 4c). Recall that to run APPLES+JC, we had to exclude challenging cases where identical fragments formed to non-monophyletic groups in the reference species tree. DEPP can handle such challenging cases, with only a slight increase in error (Fig. S10).

Running time. On these large datasets, neither training nor placement were prohibitively costly. On the 16S gene with 6497 species, training takes around two hour 20 minutes and around 5GB of memory using a 2080Ti NVIDIA GPU. Placement of 200 queries on the backbone uses only 2 minutes and 30 seconds.

4 Discussion and conclusions

We introduced a deep learning approach for extending an existing phylogenetic tree without a need for pre-specified models of sequence evolution or gene tree discordance. Our approach learns how to add new taxa by capturing patterns in an existing tree/data pair. Thus, it uses the backbone alignment and tree to learn a model that maps sequences onto the tree. This automatic learning of the model eliminates the need for assuming rigid models. Beyond accounting for discordance, the DEPP framework leads to other benefits. For example, when the reference tree is dated, DEPP outputs the divergence of the query to the reference species in the unit of time, unlike traditional placement methods that use the unit of expected numbers of substitutions.

Given a correct model of sequence evolution and no discordance, we see no reason machine learning should be more accurate than traditional phylogenetics. The power of DEPP is in its ability to learn from data without knowing the model. The model misspecification in our analyses came from gene tree discordance, but other forms of model misspecification exist and should be explored in future. Moreover, other reasons for discordance (e.g., the reference tree may be the taxonomic tree or built from morphological data) can also be imagined and provide potential use cases of DEPP. Finally, beyond accuracy, saving computational effort can provide a compelling reason to use black-box methods. If an expensive model is used to infer a tree, perhaps a black-box model can learn its essential features so that the tree can be extended further without repeating the effort. Such an approach would be presumably cheaper than applying the expensive model to the entire dataset. We leave the exploration of such applications to the future work.

Here, we trained our model on backbone trees that ranged in size from 200 to 10,000 species. While deep learning is believed to require extremely large labeled datasets for training, we were able to train DEPP, which has a moderate number of layers, with only 200 species because we use pairwise information. Thus, with 200 species, we have $\binom{200}{2} = 19900$ observed pairwise distances for training. Nevertheless, it is reasonable to expect that as reference trees become more densely sampled, the accuracy of DEPP would increase. Moreover, recall that our embedding is much smaller than size of the tree. Our results indicate that while theory suggests we need $n-1$ dimensions for Euclidean embedding of trees, far fewer dimensions suffice in practice; we had to reduce k to 16 to observe substantial drops in accuracy on simulated data (Table S2). Note that LR embedding states that $n-1$ dimensions are sufficient, but it does not state that $n-1$ dimensions are necessary.

Our results on the prokaryotic real dataset indicates that DEPP *can* overcome horizontal transfer to find the right species tree placement, *as long as* similar patterns of HGT are observed in the training data. Note that HGT is the main cause of gene tree discordance in prokaryotes, even for the marker genes such as 16S [70, 71]. However, despite relatively low levels of error overall, a small tail of placements far away from the optimal species placement with errors >20 edges is observed for many genes (Fig. S7). This long tail may be a signature of horizontally transferred genes pointing to a very different position on the species tree than the genome-wide position. When HGT events are observed in the training data, DEPP has a chance to learn

them and account for them. However, when an HGT event is novel (not seen among training data), DEPP has no way of recovering the correct position given just that one gene. These novel HGT events are a likely source of those infrequent cases of large error.

Similar to unobserved HGT events, the accuracy of DEPP can be negatively impacted by unobserved sequence patterns. Distances on the testing data were systematically underestimated before correction. We attribute this bias to the inability of the model to easily take advantage of novel data (e.g., changes in sites that are invariable in training data). The trained neural networks likely down-weight unobserved letters in specific sites, leading to distance underestimation. We hope to remedy this limitation of our model in future work by more explicit modelling of unseen data, by data augmentation, or by changing the loss function.

The most immediate use of DEPP is going to be in connecting 16S and metagenomics. DEPP often places 16S within three branches of the correct edge in the species tree; thus, while some errors remain, DEPP results should enable combined 16S and metagenomic analyses that were not possible before. Given a set of 16S and a set of metagenomic samples, DEPP can add *all* samples to the same underlying tree. This unified view of multiple data types will then enable downstream statistical analyses (e.g., Unifrac or taxonomic identification) to analyze both sets of samples jointly. Clearly, 16S has limited data compared to metagenomics, and DEPP cannot overcome that lack of signal. For example, given two identical 16S sequences that belong to different parts of the species tree, it has no way of distinguishing their placement. But results on fragmentary 16S data showed that even with as low as 400bp sequences, reasonably accurate models can be trained.

Availability.

DEPP is publicly available at https://github.com/yueyujiang/DEPP.
Data from this paper are available at https://github.com/yueyujiang/DEPP_data.

Funding.

This work was supported by the National Science Foundation (NSF) grant IIS-1845967 to Y.J. and S.M. and NSF NSF-1815485 to M.B. and S.M.; Computations were performed on the San Diego Supercomputer Center (SDSC) through XSEDE allocations, which is supported by the NSF [ACI- 1053575].

References

[1] P. D. N. Hebert, A. Cywinska, S. L. Ball, and J. R. DeWaard. Biological identifications through DNA barcodes. *Proceedings of the Royal Society B: Biological Sciences*, 270(1512):313–321, 2003.

[2] K. A. Seifert, R. A. Samson, J. R. deWaard, J. Houbraken, C. A. Levesque, J.-M. Moncalvo, G. Louis-Seize, and P. D. N. Hebert. Prospects for fungus identification using CO1 DNA barcodes, with Penicillium as a test case. *Proceedings of the National Academy of Sciences*, 104(10):3901–3906, 2007.

[3] Kasper Munch, Wouter Boomsma, Eske Willerslev, and Rasmus Nielsen. Fast phylogenetic DNA barcoding. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1512):3997–4002, 12 2008.

[4] Jo Handelsman. Metagenomics: application of genomics to uncultured microorganisms. *Microbiology and molecular biology reviews : MMBR*, 68(4):669–85, 12 2004.

[5] Mgi Langille, J Zaneveld, J G Caporaso, Daniel McDonald, D Knights, Ja Reyes, Jc Clemente, De Burkepile, Rl Vega Thurber, Rob Knight, Rg Beiko, and C Huttenhower. Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. *Nature biotechnology*, 31(9):814–21, 2013.

[6] S Sunagawa, D R Mende, G Zeller, F Izquierdo-Carrasco, S a Berger, J R Kultima, L P Coelho, M Arumugam, J Tap, H B Nielsen, S Rasmussen, S Brunak, O Pedersen, F Guarner, W M de Vos, J Wang, J Li, J Dore, S. Dusko Ehrlich, a Stamatakis, and P Bork. Metagenomic species profiling using universal phylogenetic marker genes. *Nature methods*, 10(12):1196–1199, 2013.

[7] Frederick A Matsen. Phylogenetics and the Human Microbiome. *Systematic Biology*, 64(1):e26–e41, 1 2015.

[8] Nam-phuong Nguyen, Siavash Mirarab, Bo Liu, Mihai Pop, and Tandy Warnow. TIPP: taxonomic identification and phylogenetic profiling. *Bioinformatics*, 30(24):3548–3555, 12 2014.

[9] Duy Tin Truong, Eric A Franzosa, Timothy L Tickle, Matthias Scholz, George Weingart, Edoardo Pasolli, Adrian Tett, Curtis Huttenhower, and Nicola Segata. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature Methods*, 12(10):902–903, 10 2015.

[10] Francesco Asnicar, Andrew Maltez Thomas, Francesco Beghini, Claudia Mengoni, Serena Manara, Paolo Manghi, Qiyun Zhu, Mattia Bolzan, Fabio Cumbo, Uyen May, Jon G Sanders, Moreno Zolfo, Evguenia Kopylova, Edoardo Pasolli, Rob Knight, Siavash Mirarab, Curtis Huttenhower, and Nicola Segata. Precise phylogenetic analysis of microbial isolates and genomes from metagenomes using PhyloPhlAn 3.0. *Nature Communications*, 11(1):2500, 12 2020.

[11] W. John Kress, David L. Erickson, F. Andrew Jones, Nathan G. Swenson, Rolando Perez, Oris Sanjur, and Eldredge Bermingham. Plant DNA barcodes and a community phylogeny of a tropical forest dynamics plot in Panama. *Proceedings of the National Academy of Sciences*, 106(44):18621–18626, 11 2009.

[12] Donald L J Quicke, M. Alex Smith, Daniel H. Janzen, Winnie Hallwachs, Jose Fernandez-Triana, Nina M. Laurenne, Alejandro Zald??var-River??n, Mark R. Shaw, Gavin R. Broad, Seraina Klopfstein, Scott R. Shaw, Jan Hrcek, Paul D N Hebert, Scott E. Miller, Josephine J. Rodriguez, James B. Whitfield, Michael J. Sharkey, Barbara J. Sharanowski,

Reijo Jussila, Ian D. Gauld, Douglas Chesters, and Alfried P. Vogler. Utility of the DNA barcoding gene fragment for parasitic wasp phylogeny (Hymenoptera: Ichneumonoidea): Data release and new measure of taxonomic congruence. *Molecular Ecology Resources*, 12(4):676–685, 7 2012.

[13] Jesús A. Ballesteros and Gustavo Hormiga. Species delimitation of the North American orchard-spider Leucauge venusta (Walckenaer, 1841) (Araneae, Tetragnathidae). *Molecular Phylogenetics and Evolution*, 121(December 2017):183–197, 4 2018.

[14] Kristine Bohmann, Siavash Mirarab, Vineet Bafna, and M. Thomas P. Gilbert. Beyond DNA barcoding: The unrealized potential of genome skim data in sample identification. *Molecular Ecology*, 29(14):2521–2534, 7 2020.

[15] Frederick A Matsen, Robin B Kodner, and E Virginia Armbrust. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC bioinformatics*, 11(1):538, 10 2010.

[16] Simon A. Berger, Denis Krompass, and Alexandros Stamatakis. Performance, accuracy, and Web server for evolutionary placement of short sequence reads under maximum likelihood. *Systematic biology*, 60(3):291–302, 5 2011.

[17] Pierre Barbera, Alexey M Kozlov, Lucas Czech, Benoit Morel, Diego Darriba, Tomáš Flouri, and Alexandros Stamatakis. EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences. *Systematic Biology*, 68(2):365–369, 3 2019.

[18] Maryam Rabiee and Siavash Mirarab. INSTRAL: Discordance-Aware Phylogenetic Placement Using Quartet Scores. *Systematic Biology*, 69(2):384–391, 8 2020.

[19] Metin Balaban, Shahab Sarmashghi, and Siavash Mirarab. APPLES: Scalable Distance-Based Phylogenetic Placement with or without Alignments. *Systematic Biology*, 69(3):566–578, 5 2020.

[20] Stefan Janssen, Daniel McDonald, Antonio Gonzalez, Jose A. Navas-Molina, Lingjing Jiang, Zhenjiang Zech Xu, Kevin Winker, Deborah M. Kado, Eric Orwoll, Mark Manary, Siavash Mirarab, and Rob Knight. Phylogenetic Placement of Exact Amplicon Sequences Improves Associations with Clinical Information. *mSystems*, 3(3):00021–18, 4 2018.

[21] Joseph Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 11 1981.

[22] Richard Desper and Olivier Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *Journal of computational biology*, 9(5):687–705, 10 2002.

[23] Siavash Mirarab, Nam Nguyen, and Tandy Warnow. SEPP: SATé-Enabled Phylogenetic Placement. In *Pacific Symposium On Biocomputing*, pages 247–58. WORLD SCIENTIFIC, 12 2012.

[24] Wayne P. Maddison. Gene Trees in Species Trees. *Systematic Biology*, 46(3):523–536, 9 1997.

[25] James H. Degnan and Laura A Salter. Gene tree distributions under the coalescent process. *Evolution*, 59(1):24–37, 1 2005.

[26] Tandy Warnow. *Computational phylogenetics: An introduction to designing methods for phylogeny estimation*. Cambridge University Press, 2017.

[27] Konstantinos T. Konstantinidis and James M. Tiedje. Genomic insights that advance the species definition for prokaryotes. *Proceedings of the National Academy of Sciences*, 102(7):2567–2572, 2 2005.

[28] Jesse R. Zaneveld, Catherine Lozupone, Jeffrey I. Gordon, and Rob Knight. Ribosomal RNA diversity predicts genome diversity in gut bacteria and their relatives. *Nucleic Acids Research*, 38(12):3869–3879, 7 2010.

[29] Catherine Lozupone and Rob Knight. UniFrac : a New Phylogenetic Method for Comparing Microbial Communities UniFrac : a New Phylogenetic Method for Comparing Microbial Communities. *Applied and environmental microbiology*, 71(12):8228–8235, 2005.

[30] Frederick A Matsen and Steven N Evans. Edge Principal Components and Squash Clustering: Using the Special Structure of Phylogenetic Placement Data for Sample Comparison. *PLoS ONE*, 8(3), 2013.

[31] Simon Tavaré. Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences. *Lectures on Mathematics in the Life Sciences*, 17:57–86, 1986.

[32] Lars S Jermiin, Renee A Catullo, and Barbara R Holland. A new phylogenetic protocol: dealing with model misspecification and confirmation bias in molecular phylogenetics. *NAR Genomics and Bioinformatics*, 2(2), 6 2020.

[33] Suha Naser-Khdour, Bui Quang Minh, Wenqi Zhang, Eric A. Stone, and Robert Lanfear. The Prevalence and Impact of Model Violations in Phylogenetic Analysis. *Genome Biology and Evolution*, 11(12):3341–3352, 12 2019.

[34] Nicolas Lartillot, Henner Brinkmann, and Hervé Philippe. Suppression of long-branch attraction artefacts in the animal phylogeny using a site-heterogeneous model. *BMC Evolutionary Biology*, 7(Suppl 1):S4, 2007.

[35] Jack Sullivan and David L. Swofford. Should We Use Model-Based Methods for Phylogenetic Inference When We Know That Assumptions About Among-Site Rate Variation and Nucleotide Substitution Pattern Are Violated? *Systematic Biology*, 50(5):723–729, 9 2001.

[36] Aaron L. Halpern and William J. Bruno. Evolutionary distances for protein-coding sequences: modeling site- specific residue frequencies. *Molecular Biology and Evolution*, 15(7):910–917, 7 1998.

[37] Joaquín Dopazo and José María Carazo. Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. *Journal of Molecular Evolution*, 44(2):226–233, 1997.

[38] Zhengting Zou, Hongjiu Zhang, Yuanfang Guan, and Jianzhi Zhang. Deep Residual Neural Networks Resolve Quartet Molecular Phylogenies. *Molecular Biology and Evolution*, 37(5):1495–1507, 5 2020.

[39] Anton Suvorov, Joshua Hochuli, and Daniel R. Schrider. Accurate Inference of Tree Topologies from Multiple Sequence Alignments Using Deep Learning. *Systematic Biology*, 69(2):221–233, 9 2020.

[40] Paul Zaharias, Martin Grosshauser, and Tandy Warnow. Re-evaluating Deep Neural Networks for Phylogeny Estimation: The issue of taxon sampling. In *RECOMB (To appear)*, 2021.

[41] Qiyun Zhu, Uyen Mai, Wayne Pfeiffer, Stefan Janssen, Francesco Asnicar, Jon G. Sanders, Pedro Belda-Ferre, Gabriel A. Al-Ghalith, Evguenia Kopylova, Daniel McDonald, Tomasz Kosciolek, John B Yin, Shi Huang, Nimaichand Salam, Jian-yu Jiao, Zijun Wu, Zhenjiang Z. Xu, Kalen Cantrell, Yimeng Yang, Erfan Sayyari, Maryam Rabiee, James T Morton, Sheila Podell, Dan Knights, Wen-jun Li, Curtis Huttenhower, Nicola Segata, Larry Smarr, Siavash Mirarab, and Rob Knight. Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea. *Nature Communications*, 10(1):5477, 12 2019.

[42] Donovan H. Parks, Maria Chuvochina, Pierre-Alain Chaumeil, Christian Rinke, Aaron J. Mussig, and Philip Hugenholtz. A complete domain-to-species taxonomy for Bacteria and Archaea. *Nature Biotechnology*, 38(9):1079–1086, 9 2020.

[43] J.-P. JP Doyon, Vincent Ranwez, Vincent Daubin, and V. Berry. Models, algorithms and programs for phylogeny reconciliation. *Briefings in Bioinformatics*, 12(5):392–400, 9 2011.

[44] Peter Buneman. A note on the metric properties of trees. *Journal of Combinatorial Theory, Series B*, 17(1):48–50, 1974.

[45] Peter J Lockhart, Michael A Steel, Michael D Hendy, and David Penny. Recovering evolutionary trees under a more realistic model of sequence evolution. *Molecular Biology and Evolution*, 11(4):605–612, 7 1994.

[46] T H Jukes and C R Cantor. Evolution of protein molecules. In *Mammalian protein metabolism, Vol. III (1969), pp. 21-132*, volume III, pages 21–132. 1969.

[47] W. M. Fitch and E. Margoliash. Construction of Phylogenetic Trees. *Science*, 155(3760):279–284, jan 1967.

[48] D. M. de Vienne, Sébastien Ollier, and Gabriela Aguileta. Phylo-MCOA: A Fast and Efficient Method to Detect Outlier Genes and Species in Phylogenomics Using Multiple Co-inertia Analysis. *Molecular Biology and Evolution*, 29(6):1587–1598, 6 2012.

[49] Mark Layer and John A. Rhodes. Phylogenetic trees and Euclidean embeddings. *Journal of Mathematical Biology*, 74(1-2):99–111, 2017.

[50] Pengtao Xie, Wei Wu, Yichen Zhu, and Eric P. Xing. Orthogonality-promoting distance metric learning: Convex relaxation and theoretical analysis, 2018.

[51] Olivier Gascuel. On the Optimization Principle in Phylogenetic Analysis and the Minimum-Evolution Criterion. *Molecular Biology and Evolution*, 17(3):401–405, mar 2000.

[52] William A. Beyer, Myron L. Stein, Temple F. Smith, and Stanislaw M. Ulam. A molecular sequence metric and evolutionary trees. *Mathematical Biosciences*, 1974.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[54] Nathan Killoran, Leo J Lee, Andrew Delong, David Duvenaud, and Brendan J Frey. Generating and designing dna with deep generative models. *arXiv preprint arXiv:1712.06148*, 2017.

[55] Zhengting Zou, Hongjiu Zhang, Yuanfang Guan, and Jianzhi Zhang. Deep residual neural networks resolve quartet molecular phylogenies. *Molecular Biology and Evolution*, 37(5):1495–1507, 2020.

[56] Jonathan T Barron. Continuously differentiable exponential linear units. *arXiv*, pages arXiv–1704, 2017.

[57] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[58] Thu-Hien To, Matthieu Jung, Samantha Lycett, and Olivier Gascuel. Fast Dating Using Least-Squares Criteria and Algorithms. *Systematic Biology*, 65(1):82–97, 1 2016.

[59] Uyen Mai and Siavash Mirarab. Log Transformation Improves Dating of Phylogenies. *Molecular Biology and Evolution*, page msaa222, 9 2020.

[60] Siavash Mirarab and Tandy Warnow. ASTRAL-II: Coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. *Bioinformatics*, 31(12):i44–i52, 6 2015.

[61] Qiyun Zhu, Uyen Mai, Wayne Pfeiffer, Stefan Janssen, Francesco Asnicar, Jon G Sanders, Pedro Belda-Ferre, Gabriel A. Al-Ghalith, Evguenia Kopylova, Daniel McDonald, Tomasz Kosciolek, John B Yin, Shi Huang, Nimaichand Salam, Jian-Yu Jiao, Zijun Wu, Zhenjiang Z Xu, Kalen Cantrell, Yimeng Yang, Erfan Sayyari, Maryam Rabiee, James T. Morton, Sheila Podell, Dan Knights, Wen-Jun Li, Curtis Huttenhower, Nicola Segata, Larry Smarr, Siavash Mirarab, and Rob Knight. Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea. *Nature Communications*, 10(1):5477, dec 2019.

[62] Andreas Sand, Morten Holt, Jens Johansen, Rolf Fagerberg, Gerth Brodal, Christian Pedersen, and Thomas Mailund. Algorithms for Computing the Triplet and Quartet Distances for Binary and General Trees. *Biology*, 2(4):1189–1209, 9 2013.

[63] Karin Lagesen, Peter Hallin, Einar Andreas Rødland, Hans-Henrik Stærfeldt, Torbjørn Rognes, and David W. Ussery. RNAmmer: consistent and rapid annotation of ribosomal RNA genes. *Nucleic Acids Research*, 35(9):3100–3108, may 2007.

[64] Nam-phuong D. Nguyen, Siavash Mirarab, Keerthana Kumar, and Tandy Warnow. Ultra-large alignments using phylogeny-aware profiles. *Genome Biology*, 16(1):124, 12 2015.

[65] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. FastTree-2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE*, 5(3):e9490, 3 2010.

[66] Vincent Lefort, Richard Desper, and Olivier Gascuel. FastME 2.0: A comprehensive, accurate, and fast distance-based phylogeny inference program. *Molecular Biology and Evolution*, 32(10):2798–2800, 2015.

[67] N. Moshiri. TreeSwift: A massively scalable Python tree package. *SoftwareX*, 11:100436, 1 2020.

[68] Jeet Sukumaran and Mark T Holder. DendroPy: a Python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–1571, 2010.

[69] DF Robinson and LR Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131–147, 1981.

[70] J. Peter Gogarten, W. Ford Doolittle, and Jeffrey G. Lawrence. Prokaryotic Evolution in Light of Gene Transfer. *Molecular Biology and Evolution*, 19(12):2226–2238, 12 2002.

[71] Ramunas Stepanauskas, Julia M Brown, Uyen Mai, Oliver Bezuidt, Maria Pachiadaki, Joe Brown, Steven Biller, Paul M Berube, Nicholas R Record, and Siavash Mirarab. Gene exchange networks define species-like units in marine prokaryotes. *bioRxiv*, page 2020.09.10.291518, 1 2020.

Supplementary Material

A Background on CNNs

Convolutional layers. Denote an input feature map of the intermediate layer of a model as $X \in \mathbb{R}^{L \times c_{in}}$, where L is the length of the feature map (here, sequence length), and c_{in} is the number of input channels (here, 4 for the initial layer). Also given is a filter $F \in \mathbb{R}^{k \times c_{in} \times c_{out}}$ where k is the kernel size and c_{out} is the number of output channels. The convolutional operation is defined as:

$$Y(l,t) = \sum_{i=0}^{k} \sum_{c=0}^{c_{in}} X(l+i,c) \times F(i,c,t)$$

One-dimensional convolutional layer use sliding windows to process the input sequence. Each site in the output is the weighted sum of the fragment in the corresponding windows plus the bias.

In a neural network, convolutional layers are usually used as feature extractors. Multiple convolutional layers are able to detect high-level abstraction from the input. One-dimensional convolutional layer use sliding windows to process the input sequence. Each site in the output is the weighted sum of the fragment in the corresponding windows plus the bias. The weights are determined by a learnable kernel.

Fully Connected Layer. In a neural network, fully connected layers are usually used as the last few layers to aggregate information and get the final output. In the fully connected layer, each activation in the output has connections with all the input. Specifically, each output dimension is a weighted sum of all the input, and the weights are optimized during training.

Nonlinear layer. Convolutional layers and fully connected layers consist of only linear operations. Introducing nonlinear layers can give the model the ability to capture nonlinear relations. While many nonlinear kernels exist, in this work, we use Continuously Differentiable Exponential Linear Units (CELU) as the nonlinear layer applied at the end of each layer. CELU has the following form:

$$\mathrm{CELU}(x,\alpha) = \begin{cases} x, & x \leq 0, \\ \alpha(e^{\frac{x}{\alpha}} - 1), & \text{otherwise} \end{cases} \tag{S1}$$

B Supplementary Tables

Table S1: ANOVA statistical tests of the impact of gene tree discordance (RF between true gene trees and the species tree) and lack of gene signal (RF between true gene trees and estimated gene trees), both log transformed for better linearization, on the placement error of DEPP and APPLES+JC.

Method	Variable	Df	Sum Sq	Mean Sq	F value	p-value	Explained %
DEPP	log(1 - gt. err)	1	5431.62	5431.62	740.64	$< 10^{-5}$	33.2
	log(1 - discord)	1	113.08	113.08	15.42	0.00009	0.7
	Residuals	1477	10831.87	7.33			66.1
JC	log(1 - gt. err)	1	6255.57	6255.57	515.12	0	24.3
	log(1 - discord)	1	1514.27	1514.27	124.69	0	5.9
	Residuals	1477	17936.73	12.14			69.8

Table S2: Impact of hyperparameters on the error of DEPP. We test on the single-gene high discordance simulated dataset. Bold: the default setup, used for the results reported in the paper. k: embedding size. To change the architecture, we change the number of resblocks. Various weighting schemes are incorporated in (1). We test with and without distance correction for ultrametricity. Error: mean placement error (edges). Statistical tests of significance compare each condition to the default condition using a two-sided paired Student's t test (p-values).

k	16	32	64	**128**	128	128	128	128	128	128	256	128
resblocks	1	1	1	**1**	3	5	1	1	1	1	1	1
weighting	$\frac{1}{\delta^2}$	$\frac{1}{\delta^2}$	$\frac{1}{\delta^2}$	$\frac{1}{\delta^2}$	$\frac{1}{\delta^2}$	$\frac{1}{\delta^2}$	1	$\frac{1}{\sqrt{\delta}}$	$\frac{1}{\delta}$	$\frac{1}{\delta^4}$	$\frac{1}{\delta^2}$	$\frac{1}{\delta^2}$
correction	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
error	3.87	3.67	3.65	**3.61**	3.62	3.61	3.75	3.65	3.76	4.73	3.61	3.77
p-values	0.054	0.564	0.701	N/A	0.909	0.977	0.063	0.572	0.061	0.000	1.000	0.036

Observations:

- Reducing embedding size (k) down to 32 leads to non-significant increases in error and increasing to 256 does not further decrease the error. Reducing k to 16 starts to show signs of statistically significant increases in error.
- Increasing the number of resblocks has no effect on accuracy.
- The default weighting is better than no weighting and differences are close to statistical significance.
- Weighting by $\frac{1}{\delta^2}$ and $\frac{1}{\sqrt{\delta}}$ lead to small differences in accuracy; while the default is the best, the difference to others is mostly non-significant. Weighting by $\frac{1}{\delta}$ *may* slightly reduce accuracy. Further decreasing the impact of long branches ($\frac{1}{\delta^4}$) reduces accuracy dramatically and statistically significantly.
- Distance corrections leads to a statistically significant reduction in the error.

Table S3: Properties of selected genomes from the TOL dataset [41]. Length indicates the length of the gene's protein alignment. Occupancy shows the number of species that has the each gene in their genome. QD: Quartet distance between the published gene tree and the species tree. Finally, group shows the assignment of genes to the three groups of low, medium, and high discordance based on the QD.

marker	uniprot id	gene	length	occupancy	QD	group
p0010	C9RQN5	rpoB	1429	9911	0.09324	low
p0003	A0A0T5XBS3	rpoC	1652	10249	0.11948	low
p0067	P62663	rpsC	239	10295	0.16015	low
p0011	Q1IJG7	alaS	1098	10253	0.16170	low
p0038	K9TTL6	rplB	287	10208	0.16852	low
p0023	D3R080	infB	1155	10197	0.17352	low
p0002	B1AI94	ftsH	721	10243	0.22639	low
p0047	D2PUL3	Kfla (0407)	1113	10201	0.20781	low
p0018	B5I9V0	pgk	409	9969	0.23642	low
p0004	P19486	tuf	424	10054	0.25889	low
p0056	B9CN99	rimI	816	10047	0.28959	mid
p0000	B3PLT3	oppF-valS	1631	9985	0.29736	mid
p0006	Q7US70	cysS	537	9977	0.31252	mid
p0008	D5WTU4	serS	428	9944	0.32622	mid
p0132	A0A0P0KSS3	metAP1b	333	9745	0.33467	mid
p0001	A0A173SLF7	fusA1	880	10237	0.34126	mid
p0033	D3LW72	HMPREF0889 (1213)	550	10315	0.35079	mid
p0005	E1GXM3	ileS	1220	10187	0.37819	mid
p0075	E8X3N9	AciX9 (1167)	848	10135	0.38996	mid
p0020	D8PAC1	thrS	651	10063	0.39864	mid
p0032	Q9X3X7	topA	1212	10101	0.41792	high
p0024	A5F9G1	epd	341	9337	0.42137	high
p0069	B1GZ24	folD	296	9389	0.43362	high
p0258	E1WYY2	tyrS	423	9238	0.46740	high
p0243	D5E5H2	hisRS	478	9769	0.48789	high
p0074	A0A173ZQ45	rhlE	747	8977	0.50476	high
p0310	P75510	trpS	346	9368	0.53021	high
p0350	B8J115	metN	348	9536	0.56286	high
p0131	A0A1C6GRF3	potA1	525	9110	0.58579	high
p0151	I5AQ83	EubceDRAFT1 (0090)	1320	10070	0.59077	high

Table S4: Software used for branch length re-estimation. We used FastME, except for conditions where we had some sequences with identical sequences, where FastME would fail to retain the backbone topology.

Dataset	Software	# of genes
Simulated data	FastTree	1
	FastME	2, 4, 8, 16, 32
Real TOL data	FastME	1

Commands used: :

- fastme -i $seq -u $tree -o $output –dna=JC69
- FastTreeMP -nosupport -nt -nome -noml -log tree.log -intree $tree <$seq>$output

Table S5: Number of species left after filtering out sequences that is shorten than roughly half the average length.

Dataset	Unfiltered	full length	V3+V4	V4 150bp	V4 100bp
# of species	7813	7797	7730	7731	7740
Threshold	N/A	800	200	75	50

Supplementary Figures

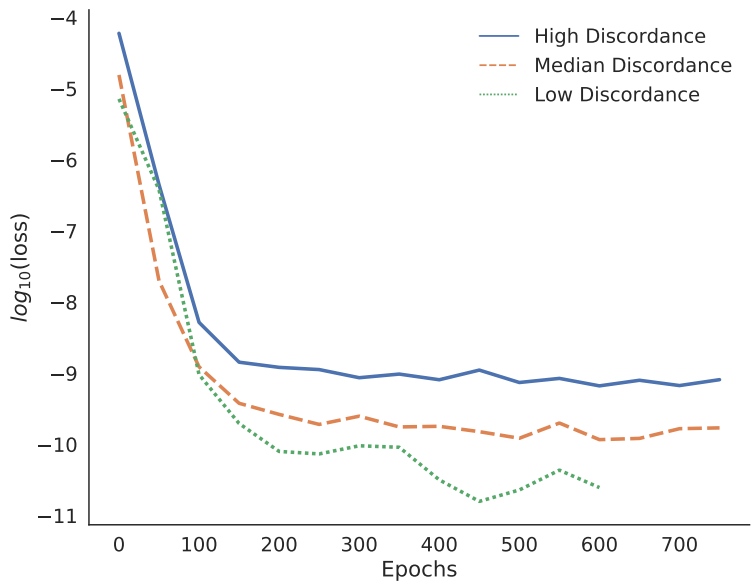


Figure S1: Train loss

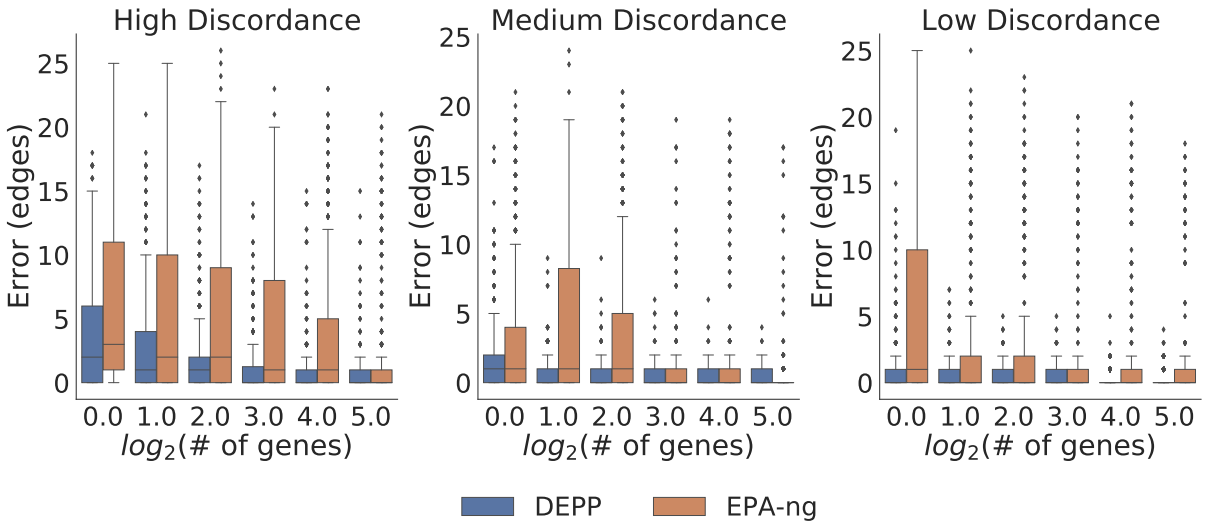


Figure S2: Error for DEPP and EPA-ng on all conditions of the simulated data.

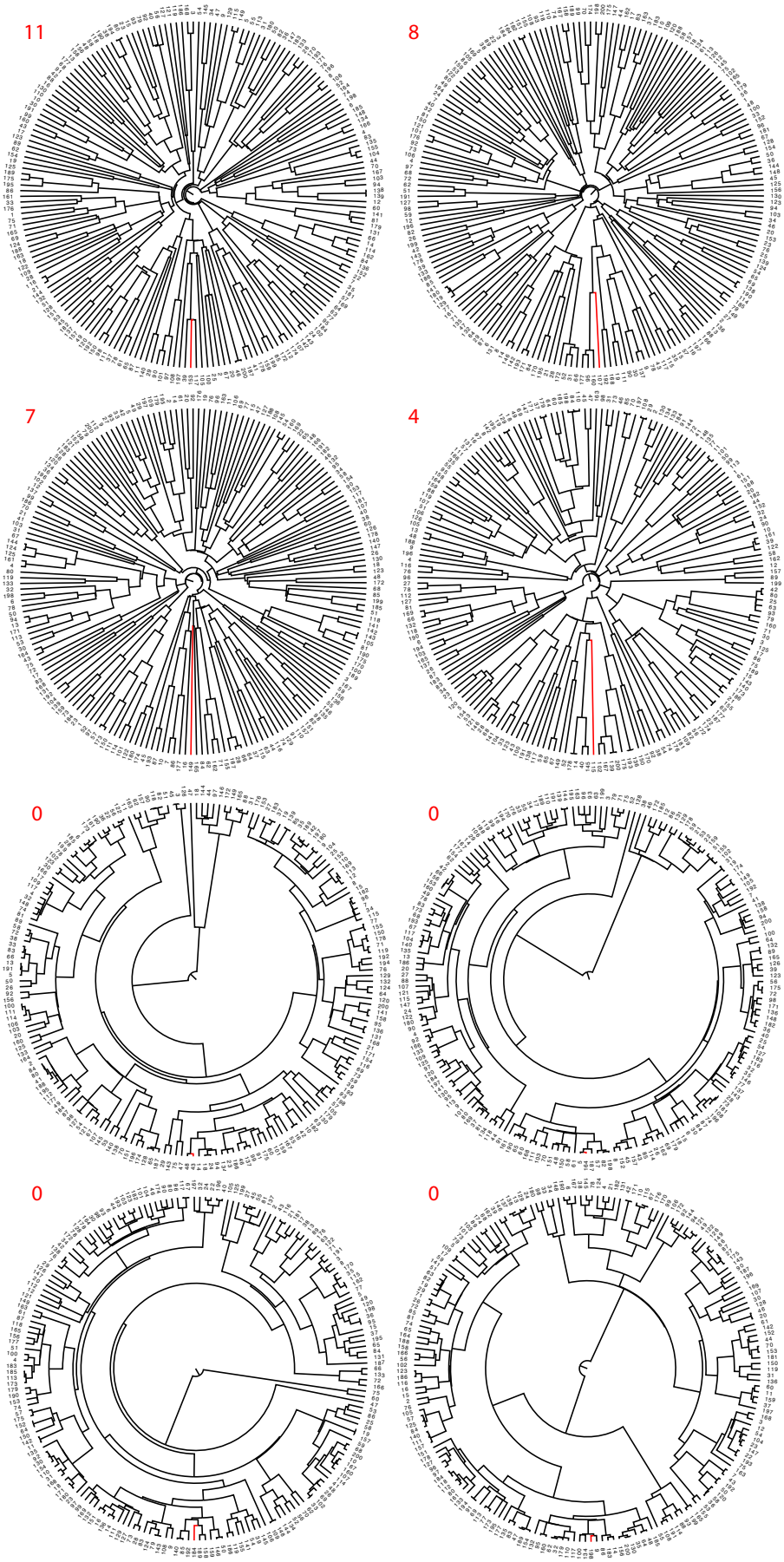
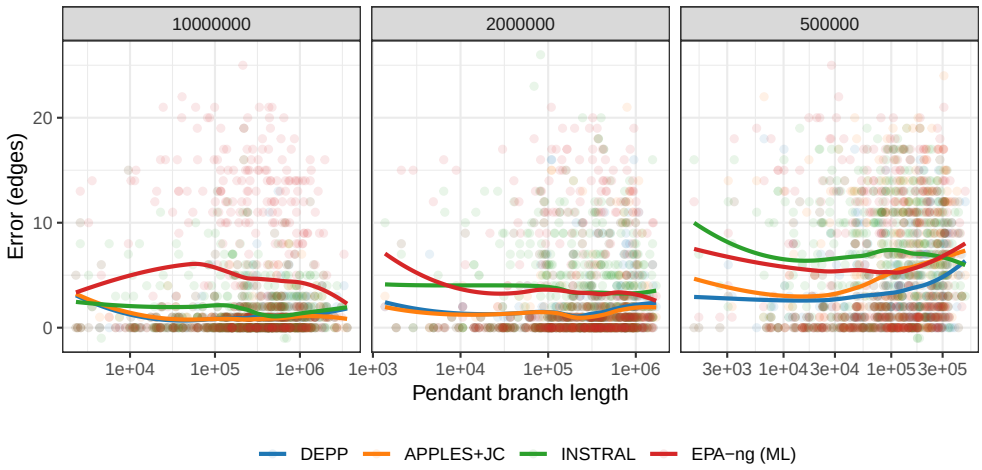
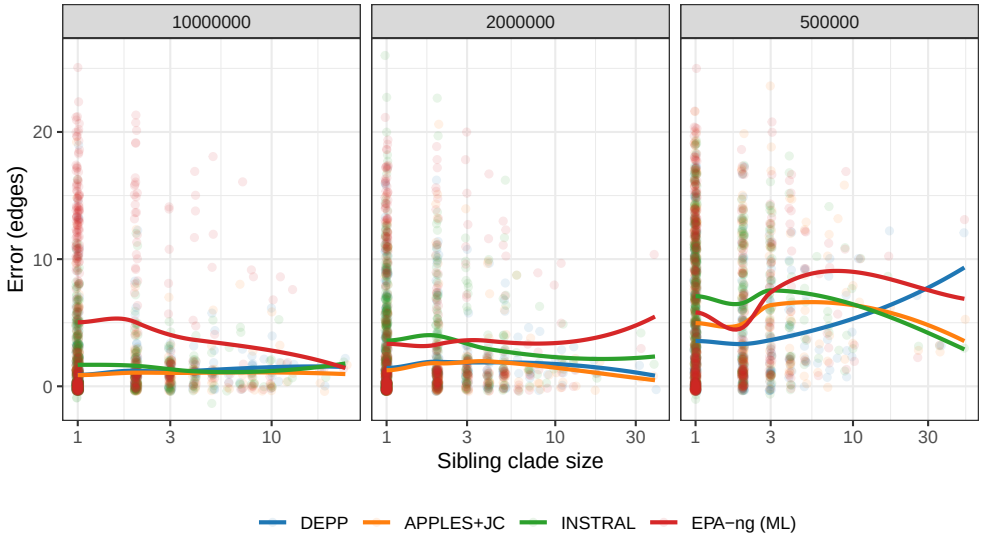


Figure S3: Examples of phylogenetic trees with high (top) and low (bottom) placement error by DEPP. For each tree, we show the (correctly placed) query in red and show error of DEPP on top in red.



(a)



(b)

Figure S4: **Impact of query properties on error.** For each mode condition, we show the impact of the (a) length of the terminal branch of the query and (b) the number of leaves in the sibling clade of the query on the accuracy of all models when run with a single gene. Each dot represents one of the 10 queries for each of 150 replicate datasets and lines show a fitted loess curve. The boxes correspond to levels of discordance, showing low, medium, and high from left to right.

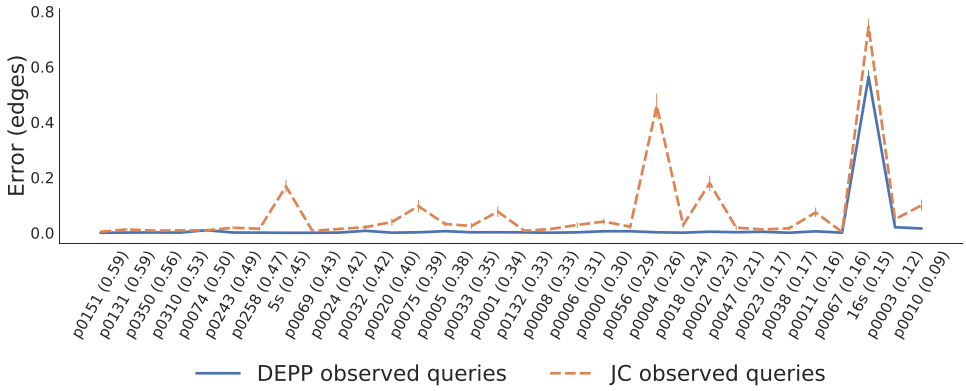


Figure S5: The mean and standard error of errors of APPLES+JC versus DEPP when tested on known sequences (i.e., those in the reference set). Note that, unlike other genes, here, 16S has multiple copies, causing the spike in the error.

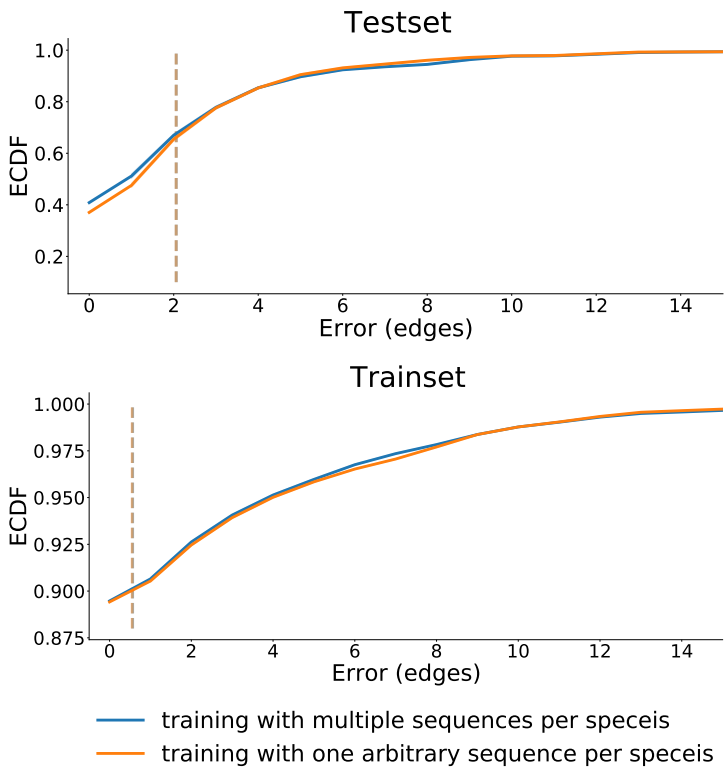


Figure S6: **Impact of weighted encoding on WoL 16S data.** Figure shows the comparison between training the model using all the available copies (using weighted encoding described in the text) versus selecting one arbitrary copy to represent each species. With multiple copies, sequenced are encoded by the frequency of the nucleotide character at each site. The results shows that using more sequences per species has a very small impact on the accuracy. Nevertheless, we use all the available sequences for training in all other experiments under the assumption that using all the sequences can reduce the variance of the DNN model.

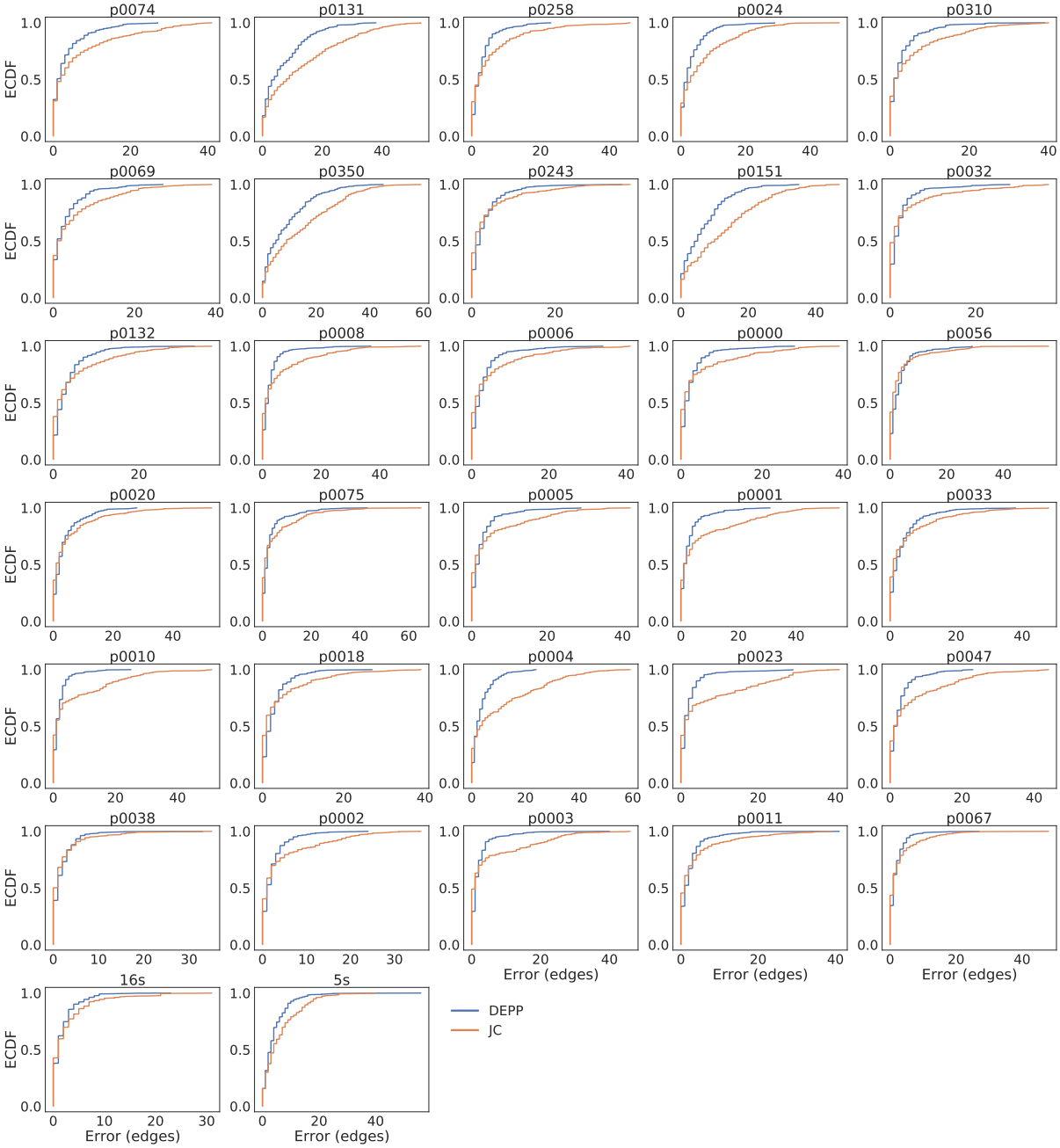


Figure S7: **Placement error on real microbial data in leave-out-experiments** We show the Empirical Cumulative Distribution (EDCF) of the placement error for JC and DEPP per gene case. The top two rows correspond to genes with high discordance, the next two rows medium discordance, and the next two rows low discordance.

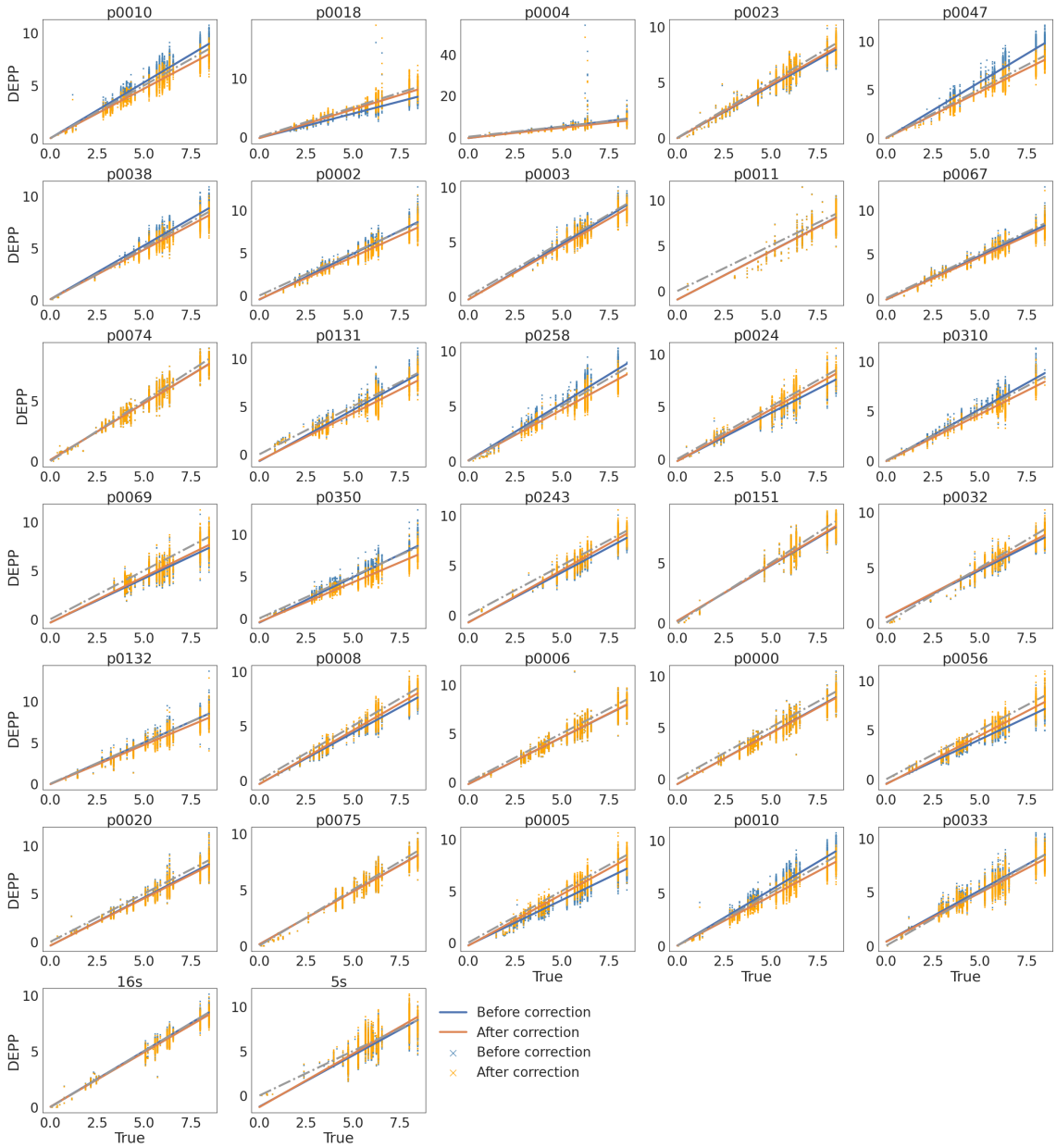


Figure S8: **Examples of distance calculation using DEPP with error equal to zero** x-axis: the true phylogenetic distance between the query; y-axis: distance calculated by DEPP. Each panel is a query arbitrarily chosen from the experiment corresponding to the gene indicated by the panel's tile. A line is fitted to points before and after correction. See Table S3 for gene properties.

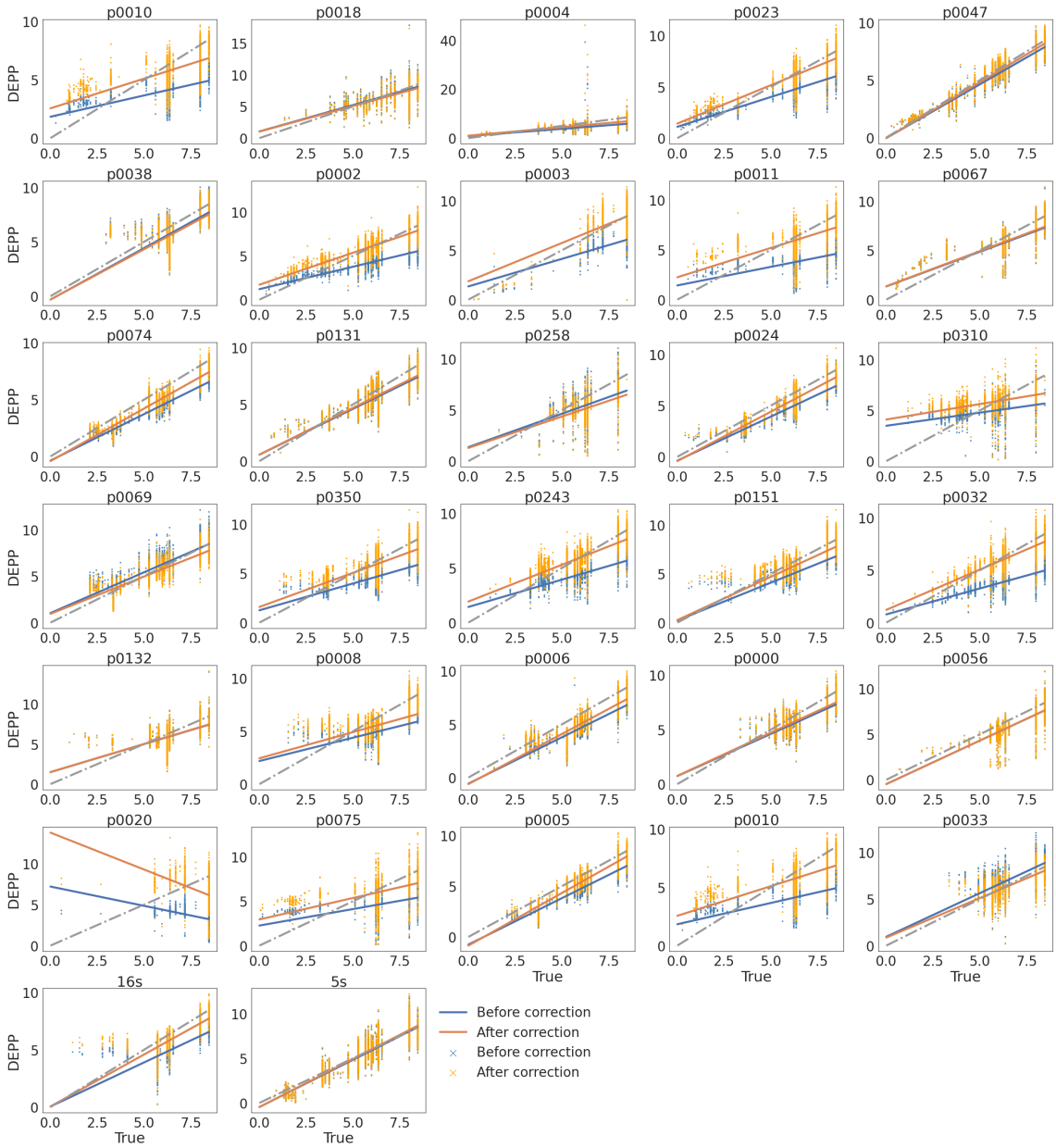


Figure S9: **Examples of distance calculation using DEPP with error larger than 15** x-axis: the true phylogenetic distance between the query; y-axis: distance calculated by DEPP. Each panel is a query arbitrarily chosen from the experiment corresponding to the gene indicated by the panel's tile. A line is fitted to points before and after correction. See Table S3 for gene properties.

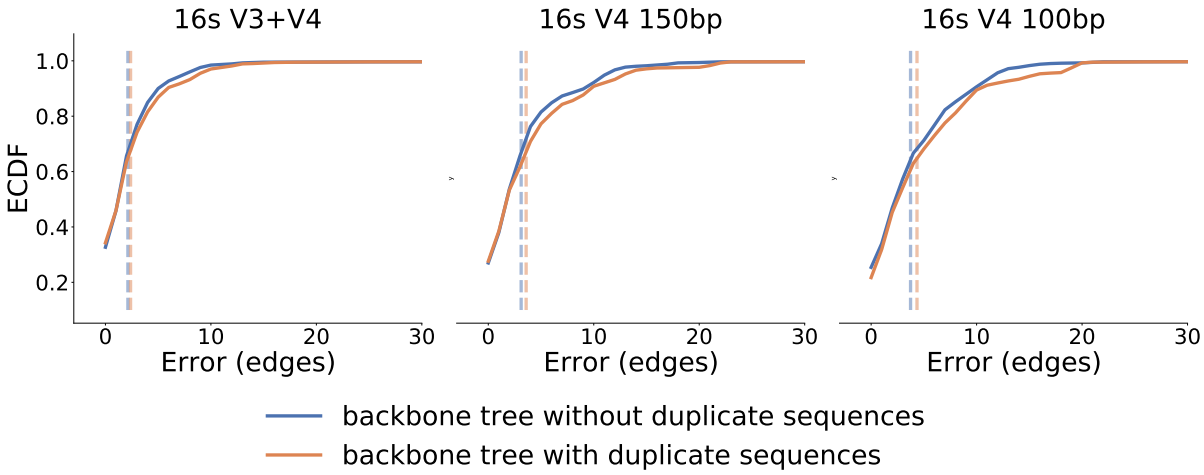


Figure S10: **Impact of duplicate sequences.** In the experiment of placement using individual regions of 16S, we run DEPP with or without duplicate sequences in the backbone tree that are not monophyletic (presence of these prevents branch length re-estimation for APPLES+JC but does not prevent DEPP from running). In the main paper, we show results without those species (to enable comparisons with APPLES+JC). Here, we show the results of DEPP both with and without those non-monophyletic duplicate sequences. The figure indicates that non-monophyletic duplicate sequences in the backbone tree have a limited negative influence on the performance.

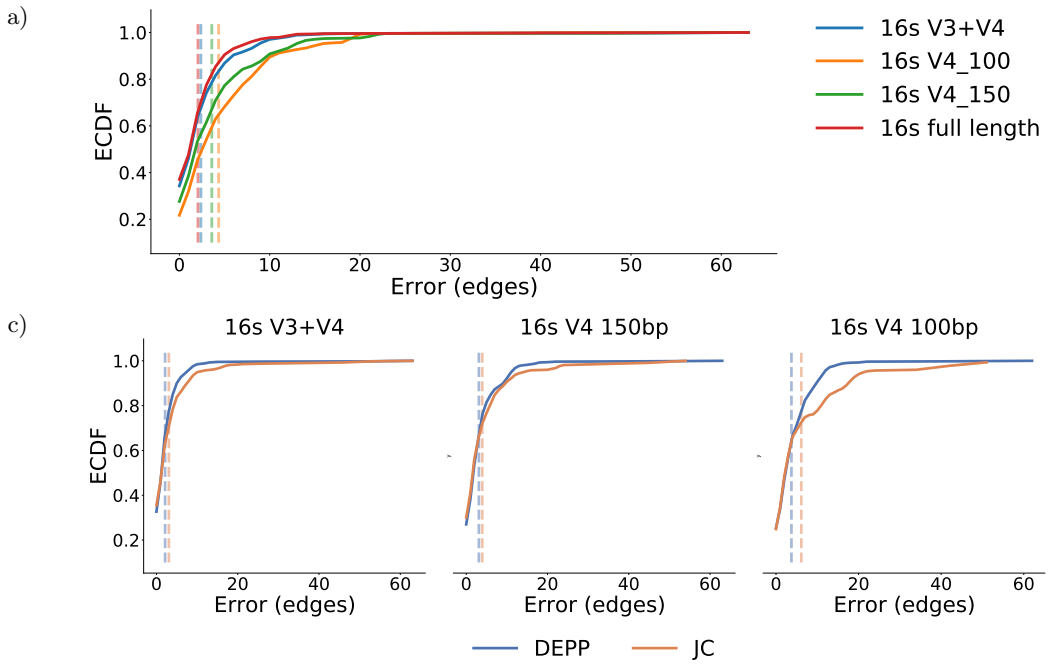


Figure S11: **Complete Empirical CDF on the TOL dataset.** In this figure we show the complete ECDF figure on WoL data, which is a supplement of Fig. S7)