Syst. Biol. 0(0):1–18, 2022

https://doi.org/10.1093/sysbio/syac031
Advance Access Publication 29 April 2022

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, 9500 Gilman Drive, La Jolla, CA 92093, USA;* [2]*Bioinformatics and Systems Biology Graduate Program, UC San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA and* [3]*Center for Fundamental and Applied Microbiomics, Arizona State University, 1151 S Forest Ave, Tempe, AZ 85281, USA*

[*]*Correspondence to be sent to: Department of Electrical and Computer Engineering, UC San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA; E-mail: y5jiang@ucsd.edu.*

Received 18 May 2021; reviews returned 13 April 2022; accepted 22 April 2022
Associate Editor: Claudia Solis-Lemus

Abstract.—Placing new sequences onto reference phylogenies is increasingly used for analyzing environmental samples, especially microbiomes. Existing placement methods assume that query sequences have evolved under specific models directly on the reference phylogeny. For example, they assume single-gene data (e.g., 16S rRNA amplicons) have evolved under the GTR model on a gene tree. Placement, however, often has a more ambitious goal: extending a (genome-wide) species tree given data from individual genes without knowing the evolutionary model. Addressing this challenging problem requires new directions. Here, we introduce Deep-learning Enabled Phylogenetic Placement (DEPP), an algorithm that learns to extend species trees using single genes without prespecified models. In simulations and on real data, we show that DEPP can match the accuracy of model-based methods without any prior knowledge of the model. We also show that DEPP can update the multilocus microbial tree-of-life with single genes with high accuracy. We further demonstrate that DEPP can combine 16S and metagenomic data onto a single tree, enabling community structure analyses that take advantage of both sources of data. [Deep learning; gene tree discordance; metagenomics; microbiome analyses; neural networks; phylogenetic placement.]

In recent years, phylogenetic inference has found widespread application in identifying organisms that make up a biological sample (Hebert et al. 2003; Seifert et al. 2007; Munch et al. 2008). Microbiome analyses often rely on phylogenetic analyses to identify species present in an environment sampled using the 16S rRNA gene amplicon sequencing or whole metagenome shotgun sequencing data (Handelsman 2004; Langille et al. 2013; Sunagawa et al. 2013; Matsen 2015; Nguyen et al. 2014; Truong et al. 2015; Asnicar et al. 2020). Using the phylogenetic context, we can identify species even when exact matches to the reference data sets are not present. Similarly, outside microbiome analyses, identifying new and known species using (meta)barcoding and genome skimming data rely on phylogenetic analyses (Kress et al. 2009; QUICKE et al. 2012; Ballesteros and Hormiga 2018; Bohmann et al. 2020).

In these high-throughput applications, phylogenetic placement—adding a new sequence onto an existing reference tree—is sufficient, and the more challenging *de novo* reconstruction is neither necessary nor always more accurate (Janssen et al. 2018). Phylogenetic placement has a long history of method development (Felsenstein 1981; Desper and Gascuel 2002; Mirarab et al. 2012; Matsen et al. 2010; Berger et al. 2011; Barbera et al. 2019; Balaban et al. 2020). However, these algorithms are designed to add sequences from a single gene family onto a tree showing its evolutionary history (i.e., the gene tree). Phylogenetic relationships change across the genome (Maddison 1997; Degnan and Salter 2005) due to processes such as horizontal gene transfer (HGT) (Ochman et al. 2000), and accounting for such discordance is a subject of much recent method development (Warnow 2017). 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 are not consequential. For example, species identification using marker genes such as 16S or COI (e.g., Konstantinidis and Tiedje 2005; Zaneveld et al. 2010) 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, Rabiee and Mirarab (2020) and Mai and Mirarab (2022) enabled inserting a new taxon onto a species tree by minimizing quartet distance, but this approach requires genome-wide data.

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 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 16S and metagenomic data (Fig. 1) in downstream analyses such as sample differentiation (Matsen and Evans 2013) using methods such as UniFrac (Lozupone and Knight 2005) and taxonomic

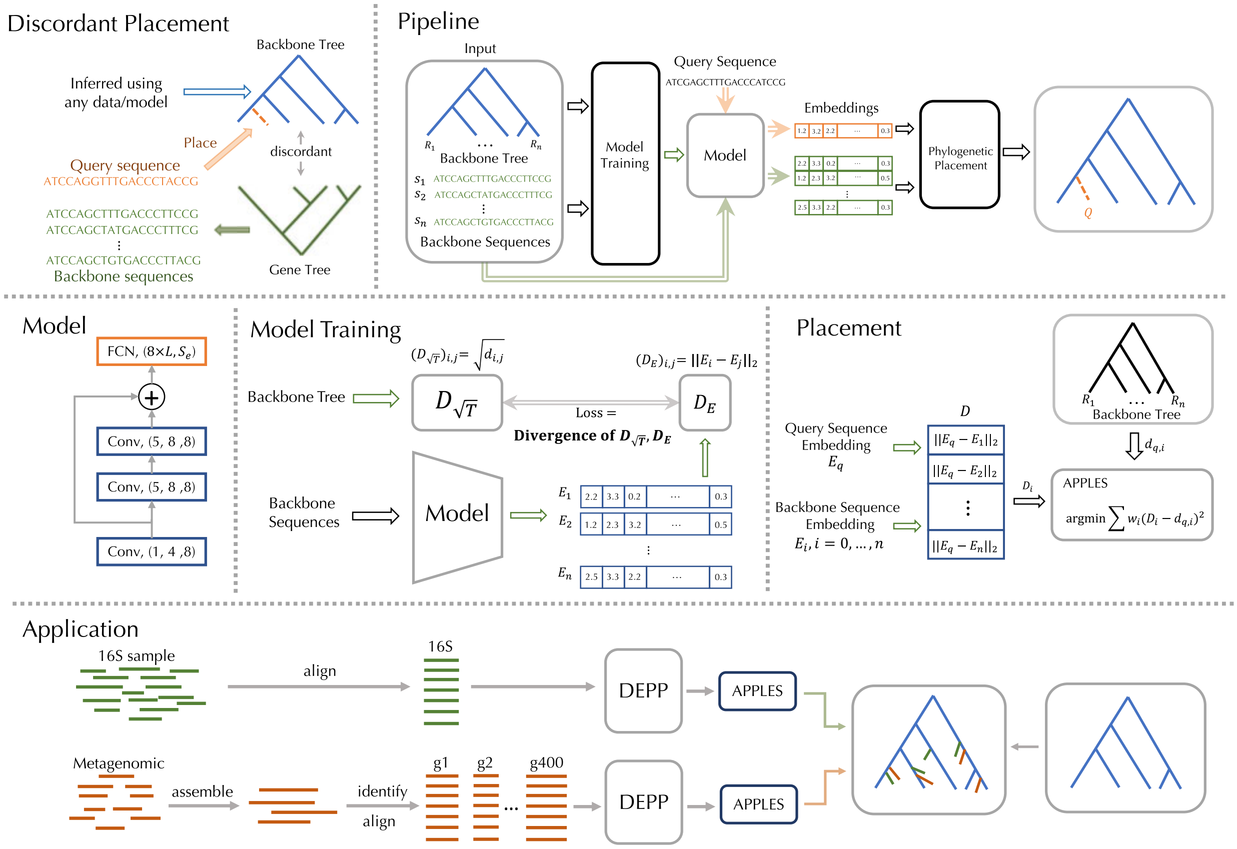

FIGURE 1. Overview of the method. Top: an illustration of discordant placement and the overall process. A neural network model is trained from a given reference tree, however inferred, and reference sequences labeled by the leaves of the tree. The model maps sequences to embeddings in k dimensions. Once sequences are embedded, phylogenetic placement is performed using distances between embeddings of the query and reference species. Middle: 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 references. The resulting distance vector is placed using weighted least square error minimization (APPLES). Bottom: The process for combining 16S and metagenomic samples using DEPP.

profiling. In particular, the ability to add *all* types of data, including 16S, metagenome-assembled genomes (MAGs), and marker genes on the same backbone tree using the same algorithm will be helpful in ensuring consistency for downstream applications. For example, we often have a large number of 16S samples and a smaller number of metagenomic samples (presumably with better differentiation ability) available to study the impact of microbiome on a phenotype. Inserting all samples onto the same tree will enable probing the associations between composition of samples and phenotypes using a unified analysis. Similarly, the ability to place eukaryotic species sampled only through their barcode genes (e.g., COI) on species trees will greatly benefit ecology research.

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 sequence data have not directly evolved on the tree. The standard phylogenetics methods are model-based: they assume data have evolved on a tree according to a model of sequence evolution (e.g., GTR Tavaré 1986) and infer the tree using maximum likelihood, Bayesian sampling, or model-corrected distances (Warnow 2017). While this paradigm has enjoyed much success, the performance of those methods is less predictable in the face of deviations from assumptions of the model (Halpern and Bruno 1998; Sullivan and Swofford 2001; Lartillot et al. 2007; Naser-Khdour et al. 2019; Jermiin et al. 2020). In particular, the impacts of gene tree discordance are poorly understood. We suggest that instead of relying on the dominant paradigm, discordant placement can be approached using machine learning.

As early as the 1990s, researchers attempted phylogenetic inference using general purpose machine learning models, such as neural networks. Dopazo and Carazo (1997) formulated the problem as unsupervised learning and designed a neural network that reflected the tree shape. More recently, the success of deep neural networks (DNNs) in solving other challenging problems has motivated efforts to adopt DNNs in phylogenetics. Zou et al. (2020) and Suvorov et al. (2020) have used a semisupervised approach with two steps: for every four species (a quartet), classify the input data to one of the three possible quartet topologies, then combine these quartet trees. This 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. (2022) recently showed, these methods can 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 recently with tens of thousands of microbial species (Zhu et al. 2019; Asnicar et al. 2020; Parks et al. 2020) using 120–400 genes, with analyses that took >200,000 h of CPU and GPU time in one case. These available trees are excellent candidates for providing the training data.

Why do we turn to black-box machine learning models for discordant placement? Discordant placement weakens the connection between sequence data and the reference tree. While the sequence data are still assumed to be a function of the reference tree, we avoid the assumption that sequence data have evolved *directly* on the tree according to a specific Markov model. A compelling reason to use machine learning is that it provides ways to learn general functions. 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, enabling placement onto species trees using single genes. Such a model would, in effect, simultaneously place sequences onto a gene tree and *reconcile* (Doyon et al. 2011) the differences with the species tree.

In this article, 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.

Materials and Methods

Discordant Phylogenetic Placement

The standard phylogenetic placement takes as input a *reference* tree T, its associated sequences S, and a query sequence q. The output is the best placement of q on T, which consists of a specific position of a particular edge of T and the length of a new terminal branch. In this article, we assume the relationship between the reference tree and sequences is indirect. Thus, S (typically sequences from a single gene) does not directly evolve on T (typically a species tree) but is influenced by T. Also, T can be inferred from any source of data (not just S). We define the discordant phylogenetic placement as the problem of finding the best placement of the query on T using S *despite* disagreements between S and T. The meaning of “best placement” depends on the context. When the discordance between data and the reference tree is because T is a species tree but the sequence data come from a single gene, we define the ideal placement as the true placement of the species on the species tree. This definition is meant to enable the application laid out before; namely, updating the species tree and identifying samples using a single gene. If this problem could be solved completely accurately, we could build species trees using single genes, and we could fully identify samples using their marker genes. Alternative definition of “best” could be imagined. For example, we could seek the place of queries that would minimize the distance between the updated tree and the true gene tree; such definitions may be appropriate for other applications.

Background on Distance-Based Estimation

Let T denote a rooted phylogenetic tree on a set of n taxa $\{t_i\}_{i=1}^{n}$ represented as leaves and each branch labeled with its length. The tree T defines a distance matrix where each entry $d_{ij} \in \mathbb{R}^+$ is the path length between leaves i and j. A distance matrix 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 (Buneman 1974). 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. These distances can converge to additivity if computed under the correct statistical model. For example, under the Jukes and Cantor (1969) (JC) model, $\frac{3}{4}\ln(1-\frac{4}{3}h)$ asymptotically converges to additivity where h is the hamming distance between sequences. Note that in discordant phylogenetic placement, traditional methods for obtaining distances from S would not match d_{ij} as the sequences have not evolved on the reference tree directly.

Placement as Supervised Learning

We approach discordant placement using supervised learning (Fig. 1). The training data are T and S, and our model is a convolutional neural network (CNN). We use a distance-based approach and express the training data as $\{((s_i,s_j),d_{ij})\}$, where $s_i,s_j \in S$ are pairs of reference sequences and d_{ij} is the distance between taxa i and j on the tree T. Due to the discordance, which leads to model mis-specification, using JC or similar distances may not be accurate. Instead, we use machine learning to compute distances. In addition, since real data sets almost always include missing data, we use machine learning to also reconstruct the missing parts of a sequence to obtain more accurate distances.

Learning objective.—To compute distances, we build a CNN that embeds sequences in the $\mathbb{R}^k$ space; we then use distances between embeddings as estimates of phylogenetic distances. The use of embeddings enjoys a theoretical justification. As de Vienne et al. (2012) showed and Layer and Rhodes (2017) elaborated, 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, and we use machine learning to find an embedding that minimizes the divergence between embedding distances and (the square root of) distances on the given reference tree.

More precisely, we treat the reference tree T and sequences 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., $\sqrt{d_{ij}}$). The square root is to match the theory by de Vienne et al. (2012) and Layer and Rhodes (2017). To make our goal 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, e.g., Xie et al. 2018), here, we simply use mean squared error, seeking:

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

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 (Fitch and Margoliash 1967; Beyer et al. 1974; Gascuel 2000; Desper and Gascuel 2002; Balaban et al. 2020), we define a weighted version of the objective function:

$$\underset{\Phi}{\arg\min} \sum_{i,j} \frac{1}{d_{ij}} \left(\left\| \Phi(s_i) - \Phi(s_j) \right\|_2 - \sqrt{d_{ij}} \right)^2 \qquad (1)$$

Embedding size. The Layer and Rhodes (2017) (LR) formulation 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. In practice, we use a rule of thumb to select the default k (which the user can change), setting $k=10\sqrt{n}$, rounded to the nearest power of 2 from below (i.e., $2^{\lfloor \frac{1}{2}\log_2(100n) \rfloor}$).

Model structure and training

CNN. We use a convolutional neural network (Fig. 1). Nucleotide sequences are encoded using 4-bit one-hot binary vectors; we refer to each bit as a channel. Gaps can be encoded as all zeros (DEPP version $\leq$ 0.1.13) or by setting all four channels to $\frac{1}{4}$ (0.1.13 $<$ v). Moreover, as detailed below, we can use a separate model to guess the best values to represent a gap (v $\geq$0.2.0). 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 C of the Supplementary material available on Dryad at https://doi.org/10.6076/D14G68, a convolutional layer applies a set of parameterized 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).

The 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 (He et al. 2016). Using residual blocks can help solve the vanishing gradient problem, which is why they are commonly used, including for sequence data analyses (Killoran et al. 2017; Zou et al. 2020). 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 (Barron 2017). The last layers are fully connected, taking features from outputs of convolutional layers and producing the final embeddings or the probability vectors. 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.

Handling missing data. Multiple sequence alignments almost always include gaps, which may represent missing data or indels. While indels may represent real signal, just like traditional maximum likelihood phylogenetic models, we can treat gaps as missing data. We can do so by leaving the one-hot encoding ambiguous (e.g., setting all four channels to $\frac{1}{4}$). An alternative is to amputate the missing data. Motivated by the BERT (Devlin et al. 2018) model used extensively in NLP, DEPP includes a reconstruction neural network to guess the best encoding for missing data (implemented since v0.2.2). The input of the model is a sequence with missing data (gaps) encoded as a one-hot vector, and the output is the reconstruction of the sequence where the sites with gaps are probability vectors inferred by the model. During training, we randomly select sites and label them as gaps to provide supervised signal for learning the reconstruction objective. This learning objective is the Kullback–Leibler divergence between the one-hot encoding of the letter and the output probability vector:

$$\underset{\Theta}{\arg\min} \sum_{i \in D_{\text{mask}}} D_{\text{KL}}(I_i \| O_i) \tag{2}$$

where D_{mask} are randomly chosen masked sites, and I_i and O_i are the one-hot encoding of the letter and the output probability vector for the site i. The reconstruction model, which consists of three convolutional layers and one fully connected layer, is trained separately from the DEPP encoder. At the time of testing/placement, a query sequence is first run through the reconstruction model to fill in the gaps, and the output of the reconstruction network is fed into the DEPP encoder to generate embeddings.

Training. We trained the model with Eq. 1 as the loss function for DEPP encoder and Eq. 2 as the loss function for reconstruction network using the stochastic gradient descent algorithm RMSProp, which divides the gradient by a running average of its recent magnitude to speed up training (Tieleman and Hinton 2012). The batch size is fixed to 32. We check the training loss every 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.

Placement.—Once the CNN model is trained, we use it to map a given query sequence q to a vector of distances $D_1 \ldots D_n$. For data sets with missing data (gaps), we compute two sets of distances, $\{D_i\}_{i=1}^n$ and $\{D_i^R\}_{i=1}^n$, using the models with and without gap reconstruction, respectively. The final distances is set to the weighted sum of the distances, that is, $(1-\alpha)D_i + \alpha D_i^R$, where α is the proportion of the sites with gaps in the query sequences. The weighted sum is used to reduce the impact of reconstructed bases (which are guessed, as opposed to being observed) on the final distance and will be empirically tested. Given these distances, we then place q onto T using distance-based placement (Balaban et al. 2020), which uses dynamic programming to find the placement with the minimum $\sum_{i=1}^n D_i^{-2}(D_i - d_{qi}(T))^2$, where $d_{qi}(T)$ represents the tree-based distance between the query and each taxon i (Fig. 1).

Uncertainty calculation.—In principle, bootstrapping, the dominant method used in phylogenetics, can be used to estimate uncertainty around distances and thus placements. However, bootstrapping assumes i.i.d models, and convolutional networks like DEPP do not treat input as i.i.d. Moreover, bootstrapping would require retraining our model on each replicate bootstrap, which we do not afford. Instead, we measure the uncertainty using a subsampling procedure recently proposed by Rachtman et al. (2021) based on solid grounds from the nonparametric support estimation literature. For each query, we randomly select m sites and mask them as gaps. The masked sequences are input to the pretrained DEPP model, which produces a distance vector corresponding to the distances from the query to backbone species. We repeat this step r times and get r distance vectors $\hat{D}_1 \ldots \hat{D}_r$. A correction is then applied to the r distance vectors as

$$\tilde{D}_i = \sqrt{\frac{m}{n}}(\hat{D}_i - \bar{D}) + D,$$

where $\bar{D}$ is the average over $\hat{D}_i$, D is the distance vector corresponds to the sequence with no site masked, and n is the length of the sequence. We use $\tilde{D}_i$ for placement and get r placements for each query, which are then used to calculate support of the placements by counting the number of times each edge is chosen. In our experiment, we choose $r = 200$ and set $m = n/\log^{0.1}(n)$.

The $\sqrt{\frac{m}{n}}$ term adjusts for the increased variance of estimates obtained from fewer data points. For a statistically consistent estimator $\hat{\theta}_n$ of a parameter θ based on n data points, if there is some rate of convergence τ_n such that $\tau_n(\hat{\theta}_n - \theta)$ weakly converges to *some* distribution as $n \to \infty$ (Assumption 2.2.1 of Politis et al. 1999), then, under forgiving conditions, the distribution of $\tau_b(\hat{\theta}_b - \theta_n)$ converges to that of $\tau_n(\hat{\theta}_n - \theta)$ as $n \to \infty$ as long as $b \to \infty$ and $b/n \to 0$ (see Theorem 2.2.1 in Politis et al. 1999). While the choice of τ_n is not obvious in general, by central theorem limit, for any estimator that can be described as the sum of independent random variables, $\tau_n = \sqrt{n}$ is the correct choice. Motivated by this observation, we set $\tau_n = \sqrt{n}$ which gives us the $\sqrt{\frac{m}{n}}$ correction term, noting that we have no proof that the rate of convergence of our estimator is proportional to $\sqrt{n}$ (or for that matter, that our estimator is consistent). We will evaluate the support values empirically.

DEPP implementation details.—We implemented DEPP using PyTorch and treeswift python packages (Moshiri 2020) and trained the models on 2080Ti NVIDIA GPUs. The embedding size k is set to 128 for data sets with 200 taxon and 512 for larger data sets (including the real web-of-life [WoL] data set). Other hyperparameters are fixed to their defaults (Table S2 of the Supplementary material available on Dryad) unless otherwise specified. DEPP is trained on the reference tree and is used to compute distances that are then fed to APPLES-II (Balaban et al. 2022), used identically to APPLES-II+JC (see below). Branch lengths of the backbone tree provided to DEPP are re-estimated using RAxML-8 (Stamatakis 2014) under the GTR+CAT model. Given more than one gene, DEPP has two options: concatenating genes or computing a summary of distances. For each query, we can compute the distance between a query and backbone species j according to each of the N genes, obtaining $D_j^1 \dots D_j^N$ (ignoring missing genes). We summarize all D_j^i values by setting D_j to the average of all D_j^i values that fall between 25 and 75 percentiles of all D_j^i values (to remove the impact of outlier genes).

Methods Compared

We compared DEPP v.0.2.2 (unless otherwise specified) to three methods.

EPA-ng. (Barbera et al. 2019) This maximum likelihood method 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-ng, under the GTR+Γ model.

INSTRAL. (Rabiee and Mirarab 2020) 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. Here, input gene trees are inferred using FastTree-II (Price et al. 2010) or RAxML. Results with FastTree or RAxML give similar results (Fig. S16 of the Supplementary material available on Dryad), so in the main paper, we use the results with input tree built by FastTree-II. 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 (Jukes and Cantor 1969) chosen because Balaban et al. (2020) found no evidence that more complex models improve accuracy. The branch lengths of reference trees are re-estimated based on the JC model using RAxML-ng (Kozlov et al. 2019). 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 $d_m = 0, b_m = 5$ for all data sets.

Data Sets

Simulated data set. For studying the gene tree and species tree discordance due to incomplete lineage sorting (ILS), we use a published simulated data set (Mirarab and Warnow 2015), with 200 ingroup species and gene trees that are discordant with the species tree due to ILS. The data set contains model conditions corresponding to high, medium, and low ILS, each with 50 replicates. We arbitrarily selected the first $2^0, \dots 2^5$ genes for each replicate.

We also simulated a second data set that consists of gene trees and species trees discordant due to HGT in addition to low levels of ILS. We used Simphy 1.0 (Mallo et al. 2016) to simulate 10 replicates, each with 10,000 ingroup species and 500 genes. Species trees are simulated using 10^8-generation birth-death process (Kendall 1948) with birth and death rates set to 5×10^{-7} and 4.167×10^{-7}, respectively. This would lead to low levels of ILS—average normalized RF distance between species and gene trees due to ILS is 0.03. In addition to ILS, each gene goes through HGT at a rate drawn from lognormal distribution with $\mu \sim \mathcal{N}(-18, 0.4), \sigma^2 = 0.75$. Simphy uses a HGT model that reduces the probability of transfer proportionally to the distance between source and recipient. The combined effect of ILS and HGT leads to 0.44 (normalized RF) median gene tree discordance. Given true gene trees simulated using Simphy, we simulate alignments with length ranging between 231 and 2054 using Indelible (Fletcher and Yang 2009). Since training a model on 10,000 species takes considerable computational resources, we then pick five genes per each replicate. To do so, we ordered the genes by the number of species that are horizontally transferred and pick 50th, 150th, 250th, 350th, and 450th genes in that the ordered gene list to ensure our test cases include genes with different level of HGT.

WoL marker genes. Zhu et al. (2019) built a species tree of 10,575 prokaryotic genomes using ASTRAL-MP (Yin et al. 2019) from 381 marker genes and computed mutation unit branch lengths using 100 sites randomly selected from each of the 381 marker genes. We categorized the marker genes into three equal-sized groups based on the rank of quartet-distance (Sand et al. 2013) between their gene tree and the species tree. For each discordance category, we selected the 10 most commonly present genes among all species (Table S3 of the Supplementary material available on Dryad). The mean gene tree discordance with the species tree, measured using the quartet distance, is 0.18, 0.34, and 0.50 in low, medium, and high discordant groups. Gene alignments are available from Zhu et al. (2019).

WoL rRNA genes. 16S and 5S rRNA genes were predicted using RNAmmer (Lagesen et al. 2007) and aligned using UPP (Nguyen et al. 2015). In genomes with multiple copies of 16/5S, we train and test DEPP on all copies and re-estimate backbone tree branch lengths for APPLES+JC using an arbitrary copy. Due to their wide usage in microbiome analyses, we also perform analyses with three regions of 16S commonly used in amplicon sequencing: V3+V4 (≈ 400 bp), V4 (100 bp), and V4 (150 bp). We removed from 16S data sets any predicted sequence output by RNAmmer that was shorter than half of the average sequence length (removing less than 1% of species; see Table S4 of the Supplementary material available on Dryad).

Traveler's diarrhea microbiomes. Quality-controlled 16S rRNA gene amplicons and manually curated MAGs were derived from a study by Zhu et al. (2018), which identified novel pathogenic profiles from the fecal samples of 22 Traveler's diarrhea (TD) patients as compared with seven healthy traveler (HT) controls. The 16S rRNA amplicon sequence variants (ASVs) were generated using Deblur from QIIME 2 and are 250 bp long. The 381 marker genes were identified using PhyloPhlAn on the translated protein sequences inferred by Prodigal from the contigs included in each MAG. This protocol is identical to that used in the WoL study.

Evaluation Procedure and Leave-Out Experiments

To ensure query sets (i.e., testing data) are separate from the training data, we removed 5% of species (10 and 500 for simulations with 200 and 10,000 backbone taxon and ≈445 for real data) from the species tree to obtain reference trees. We did not re-estimate species trees after removing queries. These left-out species are used as the query. The reference tree is the true species tree for the simulated data set and the ASTRAL tree for the WoL data set (Zhu et al. 2019). For the simulated data set, branch lengths of the backbone species tree are estimated using sites randomly selected from the genes we used in the experiments (32 genes for ILS data and 5 genes for HGT data) with each gene providing 500 sites. For WoL data set, branch lengths of the species tree are available from the original study (estimated under GTR+Γ from 100 randomly selected sites from 381 marker genes). Training is done using DEPP v0.2.2. 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 WoL). The error metric we report is the number of edges between the position on the reference tree and the inferred placement. In total, we have 8934 and 25,000 test cases for the ILS simulated and HGT simulated data respectively and 14,616 test cases for the WoL data set. INSTRAL fails in 66/9000 tests; we exclude these cases for all the methods.

In addition, we categorize test cases by their level of ILS, level of HGT, and phylogenetic signal. We compute the level of ILS by measuring the Robinson and Foulds (1981) (RF) distance between true gene trees and the species tree. The phylogenetic signal is a function of many factors, including sequence length, tree height, and the rate of evolution. Here, to quantify the lack of signal, we use the RF distance between true gene trees and those estimated using FastTree-II (Price et al. 2010). These two measurements are per backbone tree. In contrast, we measure HGT levels on a per query basis by inspecting species close to the query species in the true gene tree and their placement in the species tree. Specifically, for the five nearest species in the gene tree to the query q (denote them by N_5), we compute the sum of their path length (number of branches) to q in the species tree. Note that this sum can never be less than 17, which is the value obtained if N_5 are the five closest leaves to q in the species tree and the topology is identical and balanced. We measure HGT as the average path length of N_5 above 17; that is, $(-17+\sum_{i\in N_5} e_{q,i})/5$ where $e_{q,i}$ is the number of branches from the query q to species i in the species tree. Thus, 0 means the query is placed in a similar context in the gene tree and there is no HGT *close* to that leaf, while a high value indicates that the species close to the query in the gene tree are far away from it in the true species tree, indicating recent HGT.

Case Study on TD

For ASV placement, a single model is trained using DEPP v.0.2.2 with the reference tree set to the WoL species tree (backbone tree) and backbone sequences coming from V4 region of 16S (≈250 bp). The trained model is applied on the ASV in the TD data set to calculate the distance matrix between the sequences in the studied data set and the backbone sequences. We removed one gene (p0150), where all backbone sequences were gapped for at least half of the sites. Finally, we train a model using DEPP v.0.2.2 on the remaining sequences, giving us 380 models in total. We release and maintain these reference DEPP models for public use (see Data and Code Availability); v1.0.0 of the database is used in these analyses.

For placing MAGs, first, we used UPP to extend the alignments of all 380 marker genes to include the markers identified in the TD data set. Given these alignments, we used DEPP and the 380 trained models to compute distances between query marker genes and backbone WoL species, resulting in 380 distance matrices. We then use the distance-summary option of DEPP (i.e., mean distance in the interquartile range) to summarize all distances and use APPLES-II for placement.

To compare pairs of samples, we use weighted UniFrac (Lozupone and Knight 2005; McDonald et al. 2018). A feature table containing the frequency of ASVs or the number of reads matching a MAG in each sample is available. We use the feature table and the placement tree to calculate weighted UniFrac between each pair of the samples using QIIME 2 (Bolyen et al. 2019). We then use the PERMANOVA (Anderson 2001), as implemented in QIIME 2, to compare the HT group and the TD group (the number of permutation is set to be 10^6-1). To visualize the correlation between samples, we apply principal coordinates analysis (PCoA) on the weighted UniFrac distance matrix using QIIME-2 (Halko et al. 2011; Legendre and Legendre 2012), picking the top 3 coordinates for visualization.

We calculated a *MAG coverage* metric for each sample to represent the proportion of sequencing data covered by a MAG. It equals $(\sum_{i\in M} L_i C_i)/(\sum_{i\in N} L_i C_i)$, where M is the set of contigs constituting MAGs, N is the set of all contigs in a metagenome assembly, L is the length (bp) of a contig, C is the coverage of a contig as determined by the average number of times each nucleotide of the contig is included in any sequencing read recruited to the contig.

Results

Evaluation on Simulated Data Sets

DEPP training and parameter sensitivity.—We start by evaluating DEPP on simulated data sets, testing the ability to train the CNN model in reasonable times. As the training epochs advance, the loss function (1) drops rapidly and stabilizes after around 500 epochs in a typical case (Fig. S1 of the Supplementary material available on Dryad). Here, training, which is a one-time process for each reference tree, finished in around 20 min for the 200-taxon data set and 260 minutes for 10,000-taxon data set, on a machine with one 2080Ti NVIDIA GPU and 8 CPU cores. Placement of 1000 queries took 4 seconds for the 200-taxon and 30 s for the 10,000-taxon data sets using a single CPU core. On the small 200-taxon data set, EPA-ng has an advantage in terms of running time. However, in the larger HGT data set (10,000-taxon), DEPP placements are faster than the alternatives with half the running time of EPA-ng. In terms of the memory usage, APPLES+JC has the lowest memory consumption, while the memory usage of DEPP is 9-fold lower than EPA-ng on the larger data set.

DEPP is mostly robust to weighing schema, with all four schemes tested resulting in statistically indistinguishable performance (Table S2 of the Supplementary material available on Dryad). Models with more parameters, that is, deeper network or larger embeddding size, tend to have better performance but also longer training time. For example, for the 200-taxon tree, the time for training a model with one residual block is around 15 min while this number goes up to 20 min when the model has five residual blocks. Reducing the number of residual blocks from five to one or reducing the embedding size to 32 reduce the accuracy significantly (Table S2 of the Supplementary material available on Dryad). In our final model, we use five residual blocks for backbone tree with 200 taxon and one residual block for the rest of data set to trade-off performance and training time. Note that while the preliminary results motivated the choice of default settings used in the rest of analyses, we did not select the optimal settings for this data set and have not tested various settings on other data sets (thus, hyperparameters are not overfit to the data). The use of the gap reconstruction model dramatically improves accuracy when the query has 40% or more gaps, and the use of the weighted approach results in further reductions in error (Fig. S17 of the Supplementary material available on Dryad).

Comparison to other methods.—We now compare accuracy of DEPP to distance-based APPLES-II (Balaban et al. 2022) used with the standard JC model, maximum likelihood method EPA-ng (Barbera et al. 2019), and the quartet-based discordant-aware method INSTRAL (Rabiee and Mirarab 2020). Note that APPLES-JC and EPA-ng are not designed for discordant placement using a single gene, and INSTRAL is designed only for data sets with many genes (at least two but ideally many more). However, since no existing method is designed for discordant placement, we had to compare it to these existing methods.

ILS discordance. On the 200-taxon data set, DEPP is comparable to EPA-ng and outperforms APPLES+JC and INSTRAL when given a single gene (Fig. 2a,c). While DEPP and EPA-ng have similar average error rates overall, DEPP has fewer cases with error above 10 edges (Fig. 2a). When the gene tree discordance level is medium (or low), DEPP, EPA-ng, and APPLES+JC have similar average error, which is as low as 1.5 edges (or 0.9 edge) on average but DEPP has a shorter error tail (Fig 2a). DEPP has the lowest mean error among the methods when discordance is high (3.38 edges for DEPP versus 3.50 for EPA-ng and 4.31 for APPLES+JC). For context, random placements on these species trees would give a placement error of 14 edges on average (empirically computed). Furthermore, DEPP outperforms other methods in difficult cases when the phylogenetic signal is weak (Fig. 2b).

All methods experience a sharp rise in the error when the phylogenetic signal weakens (Fig. S5a of the Supplementary material available on Dryad) or discordance increases (Fig. S5b,c of the Supplementary material available on Dryad). According to an ANOVA test (Table S1 of the Supplementary material available on Dryad), both gene tree discordance and signal have a significant impact on the placement error (P-value$<10^{-20}$). However, these two factors combined explain only around 15% of the variance in error for DEPP and EPA-ng.

As the number of concatenated genes increases, unsurprisingly, the mean errors of all methods reduce (Fig. 2c).

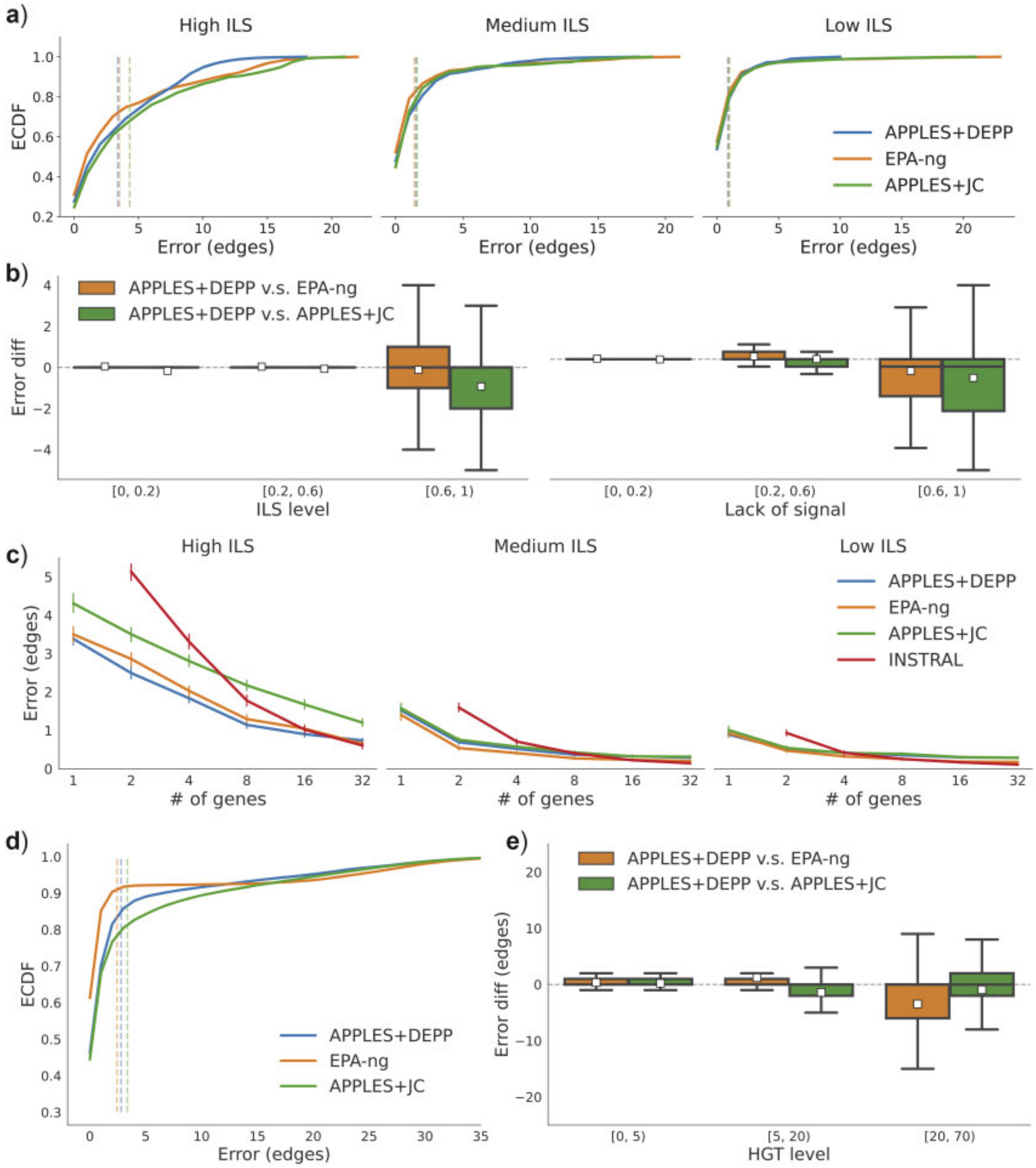

FIGURE 2. Results on simulated data sets. a) Empirical cumulative distribution function (ECDF) of the placement error on a single gene for high, medium, and low discordance (69% 34%, and 21% mean RF). INSTRAL needs at least two genes. b) Sensitivity to gene properties on ILS data. Error comparison between DEPP and other method using a single gene on (left) different level of true gene tree discordance (RF distance between true gene trees and the species tree) and (right) different level of gene signal missing (RF distance between true gene trees and estimated gene trees) combining all discordance levels. y-axis: the error of DEPP minus error an alternative method. Hallow squares: mean error difference. c) Mean and standard error of placement error versus the number of genes on ILS data. d) ECDF of the placement error on HGT data. e) Error comparison between DEPP and other method on different level of HGT. HGT is measured by $\sum_{i\in N_5} e_{q,i}$, where N_5 is the five closest species on the gene tree and $e_{q,n}$ is the number of branches between queries and species i on species tree.

Computing per-gene distances and summarizing them instead of concatenating them increases accuracy under some conditions but reduces accuracy under others (Fig. S2 of the Supplementary material available on Dryad). DEPP, EPA-ng as well as APPLES+JC are more accurate than INSTRAL for low numbers of genes (≤4) but not for more 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 ILS.

We further examine the example cases where DEPP performs well or poorly. We compare the ML phylogenetic distances computed using RAxML on the true species tree versus distances computed by DEPP for low and high error cases (Fig. 3). Both high- and low-error cases seem to result in unbiased distances computed

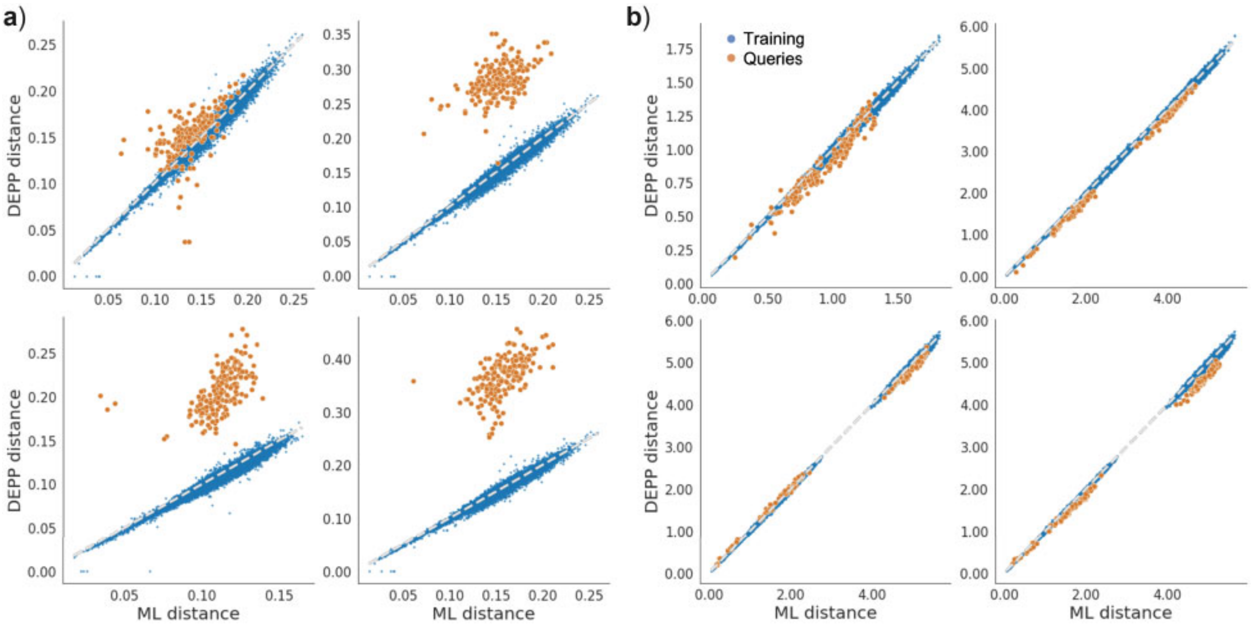

FIGURE 3. Examples of phylogenetic distance calculation using DEPP. Phylogenetic distance on the true species tree (in units of substitution per site) versus the distance calculated by DEPP for a) examples with high error (11 edges of error) and b) examples with zero error. Dashed line: identity line. High errors correspond to long terminal and short internal branches; see Figure S3 of the Supplementary material available on Dryad for trees.

in the training set; however, high error examples have 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 are systematically overestimated in cases with high error, and a large range of values are estimated for pairs with equal ML 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 tend to correspond to novel query taxa; that is, those on long branches on sparsely sampled clades (Fig. S3 of the Supplementary material available on Dryad). Queries with higher terminal branch lengths lead to higher error (Fig. S4 of the Supplementary material available on Dryad); however, it appears that the shortest of branches also have a higher error, perhaps because distinguishing very similar taxa requires strong signal. Similarly, queries with a large clade as the sister tend to be more difficult for DEPP (Fig. S4 of the Supplementary material available on Dryad). 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 of the Supplementary material available on Dryad).

HGT discordance. On the 10,000-taxon data set, which includes HGT, EPA-ng has the best overall accuracy, with DEPP coming as a close second (mean error: 2.43 and 2.80, respectively). However, these average performances mask larger differences as HGT levels change. Breaking down the data set by the HGT level per query taxon, we observe that DEPP has slightly worse accuracy than EPA-ng for the queries with low or medium HGT level but better accuracy with high levels of HGT. DEPP has the largest advantage with the most challenging cases when gene sequences moved far away; e.g., error for DEPP is 3.5 edges better than EPA-ng and 0.9 edges better than APPLES+JC on average with the highest level of HGT.

The Real WoL Data Set

We then tested DEPP on the real WoL data set using 30 marker genes, preselected to represent the range of discordance among all 381 genes, in addition to 16S and 5S. We tested DEPP, EPA-ng, and APPLES+JC using both novel queries (left-out) and observed queries (training data). Despite the size of the data sets, neither DEPP training nor placement was prohibitively slow. For example, on the 16S gene with 7407 species, training takes 240 min and uses 5 GB of memory using a 2080Ti NVIDIA GPU. Placement of 800 queries on the backbone takes less than 40 s using a single core. At the placement time, for the 30 marker genes leave-out experiments, DEPP is faster than both EPA-ng and APPLES+JC (Table S5 of the Supplementary material available on Dryad). Moreover, DEPP, once trained (a one-time process) uses ten times less memory than EPA-ng.

Given large backbone trees such as WoL, a query sequence will have a considerable chance of matching

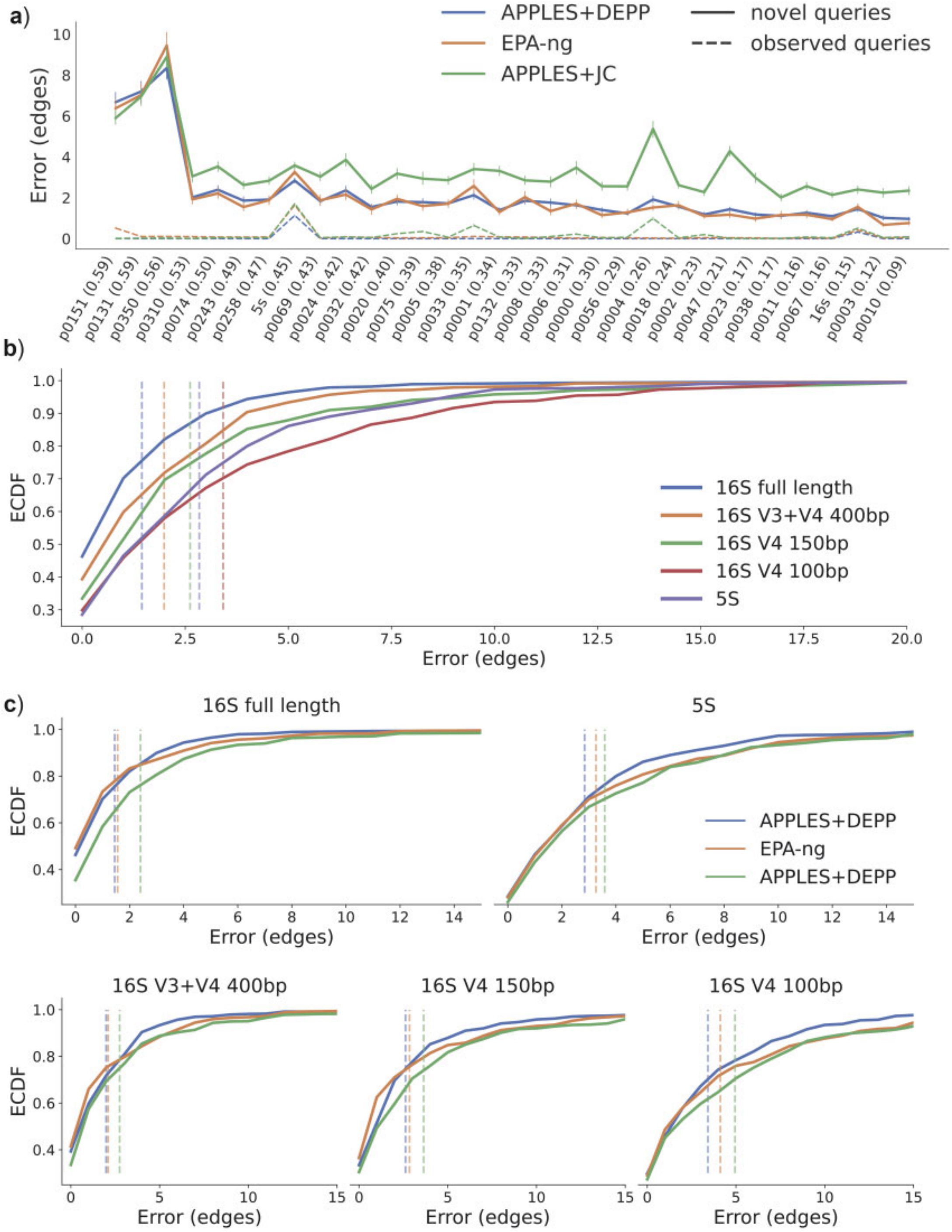

FIGURE 4. Results on the WoL data set. a) Mean and standard error of placement error for APPLES+JC, EPA-ng and DEPP, applied to novel queries and known queries, on the species tree from WoL data set, which is treated as the ground truth. The x-axis shows genes, ordered by their quartet distance to the reference species tree (shown parenthetically). For novel queries, the placement is performed in a leave-out fashion. For ECDF of the placement error, see Figure S8 of the Supplementary material available on Dryad. b) ECDF of placement error on the 16S gene with full length or individual regions of 16S used in amplicon sequencing. V3+V4 region is ≈ 400 bp. Dashed lines show the mean error. c) Methods comparison on rRNA genes. We cut the x-axis at 15 for better visualization; see Figure S11 of the Supplementary material available on Dryad for the full range.

a species present in the reference tree. Under such conditions, DEPP has better performance over all the genes compared with EPA-ng and APPLES+JC (Fig. S6 of the Supplementary material available on Dryad). DEPP has close to perfect accuracy for all genes except the multicopy genes 16S and 5S (Fig. 4a and Fig. S6 of the Supplementary material available on Dryad). For these genes, the error slightly increases; DEPP finds the optimal placement in 91% (16S) and 73% (5S) of cases but, on average, has an error of 0.32 (16S) and 1.13 (5S) edges. Patterns of error do not change whether we train DEPP using the weighting scheme described before or by simply selecting an arbitrary copy (Fig. S7 of the Supplementary material available on Dryad). While APPLES+JC and EPA-ng also had low error levels for many genes, they had considerable error levels on the training data sets for eight and three genes, respectively (Fig. S6 of the Supplementary material available on Dryad).

In the most interesting case, when the query sequences are novel (i.e., are not in the training set), both DEPP and EPA-ng greatly outperform APPLES+JC (Fig. 4a). On average, the placement error of DEPP (2.17 edges) and EPA-ng (2.15 edges) is much lower than APPLES+JC (3.34 edges). Moreover, EPA-ng and DEPP have low error about the same number of times (respectively, 89% and 88% of DEPP and EPA-ng placements have four edges or less error). However, DEPP is less often far away from the optimal placement. For example, on average, the maximum error of DEPP for each gene is seven edges lower than EPA-ng; or, the placement error of EPA-ng is larger than 15 edges in 3.1% of cases compared to 2.4% for DEPP. Thus, just like the simulated data set, DEPP has fewer cases of high error.

Across all 32 genes, in 85% of tests, DEPP and EPA-ng find a placement within 3 edges of the optimal placement; for the full-length 16S gene, this value is 91% for DEPP and 87% for EPA-ng. A random placement on a tree with 10,575 leaves is, on average, 26 edges away from the optimal placement. Our results are also consistent with using 16S as a marker gene, which among the 32 genes had one of the lowest mean error rates (1.43 edges). Finally, note that similar to simulated data sets, 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. S9 of the Supplementary material available on Dryad) but high variance and bias when it works poorly (Fig. S10 of the Supplementary material available on Dryad).

Going beyond a single gene, given 50 randomly selected genes, DEPP used with distance summary strategy was able to place within three branches of the optimal placement in 94% of cases, with a mean error of only 0.98 edges (Fig. S12 of the Supplementary material available on Dryad). Accuracy slightly degrades if we concatenate genes instead of summarizing distances among them (e.g., mean error increases to 1.6 edges). Thus, DEPP can not only place using single genes, but it can also place with high accuracy given data from multiple genes.

Testing performance on rRNA genes 16S and 5S, average error for DEPP is lower than EPA-ng or APPLES+JC for both genes (Fig. 4c). Mean error for DEPP over all rRNA data is 2.48 edges compared to 2.81 for EPA-ng and 3.47 for APPLES+JC. For full-length 16S, DEPP and EPA-ng find the optimal placements about the same number of times; however, DEPP is more often *close* to the optimal placements (e.g., 91% of queries are within three edges for DEPP compared to 87% for EPA-ng and 80% for APPLES+JC). When given short amplicon-length sequences, the advantage of DEPP over alternatives becomes more substantial. For example, given 100 bp amplicons from the V4 region, DEPP has an average error of 3.4 edges while the average error of EPA-ng and APPLES+JC are 4.12 and 4.94 edges, respectively. While 3.4 edges of error may sound high, note that insertion of a 100 bp read into a *species tree* is clearly a difficult task. Comparing performance on different rRNA genes, for 16S data, longer sequences give better performance for all the methods (Fig. 4). Interestingly, 5S sequences (which are ≈100 bp) have better accuracy than 16S V4 region sequences with the similar length, possibly indicating that 5S carries more phylogenetic signal or less discordance with the species tree.

Support accuracy. We test our method of measuring support on 30 marker genes of WoL data versus the support values generated by EPA-ng. Support values are vastly different between EPA-ng and DEPP. While EPA-ng tends to produce 100% support for a single placement for most queries, DEPP generate far more placements, most with low support (Fig. 5a). Across all 12,727 queries, EPA-ng generates only 14,355 placements versus 163,110 produced by DEPP. DEPP estimates full support for a single placement for only 9.8% of queries whereas EPA-NG produces full support for 85% of queries. Because of its high confidence in its unique placement, no threshold of EPA-NG support can produce low levels of false positive detection (Fig. 5b), in contrast to DEPP, which can produce FPRs close to zero. Comparing FPR and recall, DEPP can achieve the same recall level as EPA-NG with far lower levels of FPR (Fig. 5b).

Both EPA-NG and DEPP support values tend to be higher in distribution for correct placements than incorrect placements (Fig. 5a,c). However, DEPP support values show a much larger gap between correct and incorrect branches and are more predictive of accuracy compared to EPA-ng (Fig 5a,c). Both methods clearly overestimate support so that even branches with 100% support are often incorrect; for example, only 66%, 71%, and 76% of the DEPP placements with support ≥0.9, ≥0.95, and ≥0.99 are correct. However, the overestimation problem is worse for EPA-ng, which gives high support in the vast majority of cases (Fig. 5c,d). Overall, 62% of wrong placements with EPA-ng have 100% support, compared with only 0.2% for DEPP.

Case Study on Combined 16S rRNA and Shotgun Metagenomic Data

We next studied how adding the MAGs and 16S rRNA ASVs onto the same tree enables new analyses. We used the data set by Zhu et al. (2018) with gut microbiomes from seven healthy controls (HT) and 22 patients with TD, all of whom were sampled using both 16S amplicon sequencing and metagenomics with available MAGs and ASVs. We added the ASVs and MAGs onto the same WoL backbone tree using DEPP (see Materials and

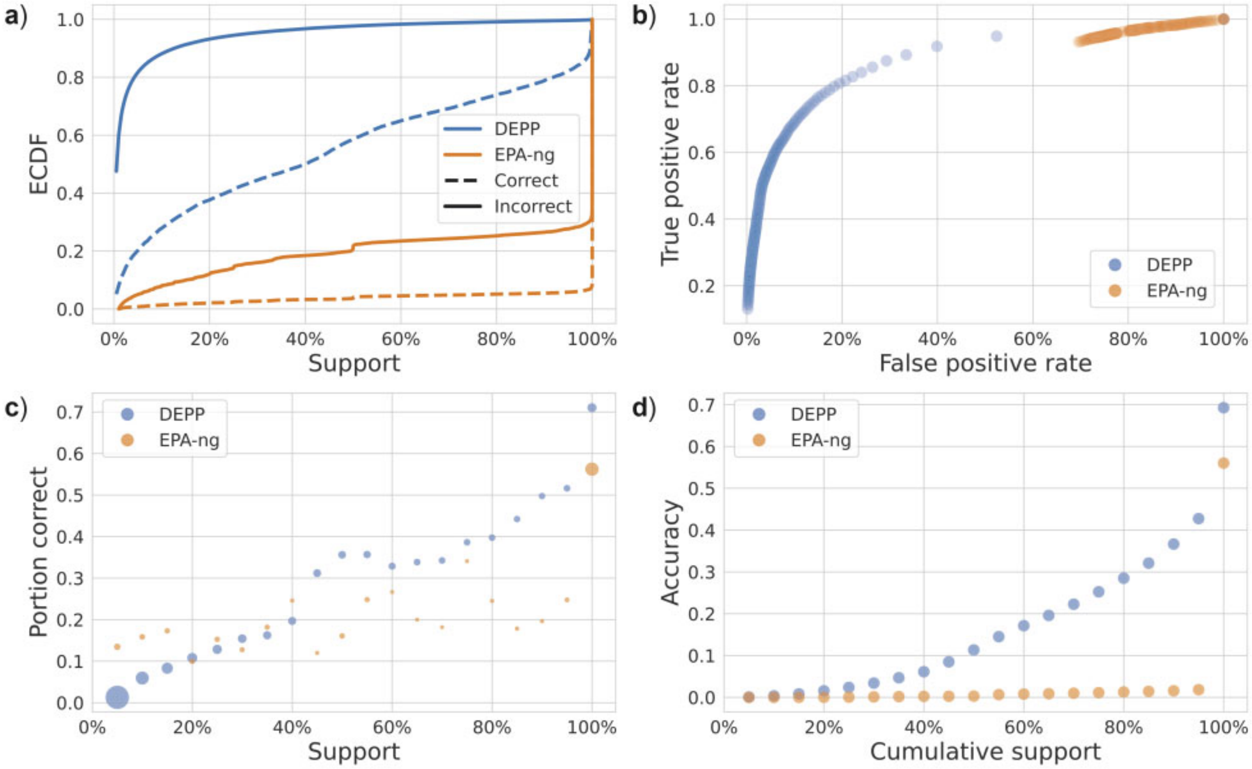

FIGURE 5. Support of the placements. a) Empirical cumulative distribution function (ECDF) of the supports. b) Receiver operating characteristic (ROC) curve. TP, TN, FP, FN is defined by correct placement with support above the threshold, incorrect placement with support below the threshold, incorrect placement with support above the threshold, correct placement with support below the threshold respectively (support threshold step size: 0.5%). c) Correlation between support and the correctness of the placements (area of each dot is proportional to the number of placements it contains). d) Correlation between support and placement accuracy. x-axis: cumulative support of the top placements (placements with highest supports); y-axis: proportion of the cases when the correct placements are among the top placements. These figures are using 163,110 placements for DEPP and 14,355 placements for EPA-ng in total across 12,727 queries on 30 marker genes of the WoL data set.

Methods section) obtaining two placement profiles for each subject. We compare pairs of profiles using the weighted UniFrac (Lozupone and Knight 2005) distance.

The UniFrac distances of the MAG profile of a sample to the ASV profiles of other samples were higher on average than its distance to the ASV profile of the same sample in all except one case (Fig. 6a). In nine samples, the MAG profile had a lower distance to its own ASV than *any* other sample. The intrasample distances between the ASV and MAG profiles substantially reduced as the MAGs represented a larger proportion of the sequencing data of each sample (Fig. 6a; P-value $=0.001$), whereas distances across samples slightly increase (P-value $=$ 0.02). As a result, the gap between inter- and intrasample distances grew substantially with higher MAG coverage (Fig. 6a).

Placements on a single tree enabled us to visualize all ASV- and MAG-informed community structures using a unified Principal Coordinates Analysis (PCoA) (Fig. 6b). This results show that for some samples, ASV and MAG placements indicate extremely similar community structures while for others, there is a substantial disagreement between the two. The intrasample distances in the PCoA plots tend to be shorter for high coverage MAGs (Fig. S13a of the Supplementary material available on Dryad).

MAGs and ASVs can both distinguish healthy and diseased samples (P-values: 0.019 and 0.024 using the standard PERMANOVA method), but MAGs provide a higher statistical power that implies a larger effect size (Fig. 6c). While the median ASV distances between pairs of TD samples are similar to distances between TD and HD samples, both intragroup median MAG distances are lower than intergroup MAG median. Furthermore, combining two type of data, 16S and MAG, which is enabled by DEPP, provides a large separation with much larger F statistics and increased statistical significant (P-values: 0.002) (Fig. 6c).

Despite the substantial agreement between MAG and ASV placements, there are also differences. For three samples (78, 10, 80,152), the ASV/MAG agreement is low compared to the background distance levels. Zhu et al. (2018) characterized these samples as suffering from the co-infection of multiple Enterobacteriaceae organisms (*Escherichia*, *Enterobacter*, *Klebsiella*, and *Citrobacter*), which increased the challenge of accurately binning contigs from these closely related microbes—a possible explanation for the relatively low congruence. In addition, there are several groups that are found by MAGs or ASVs but not the other (Fig. 6e and Fig. S15 of the Supplementary material available on Dryad). The MAG placements include a clade representing

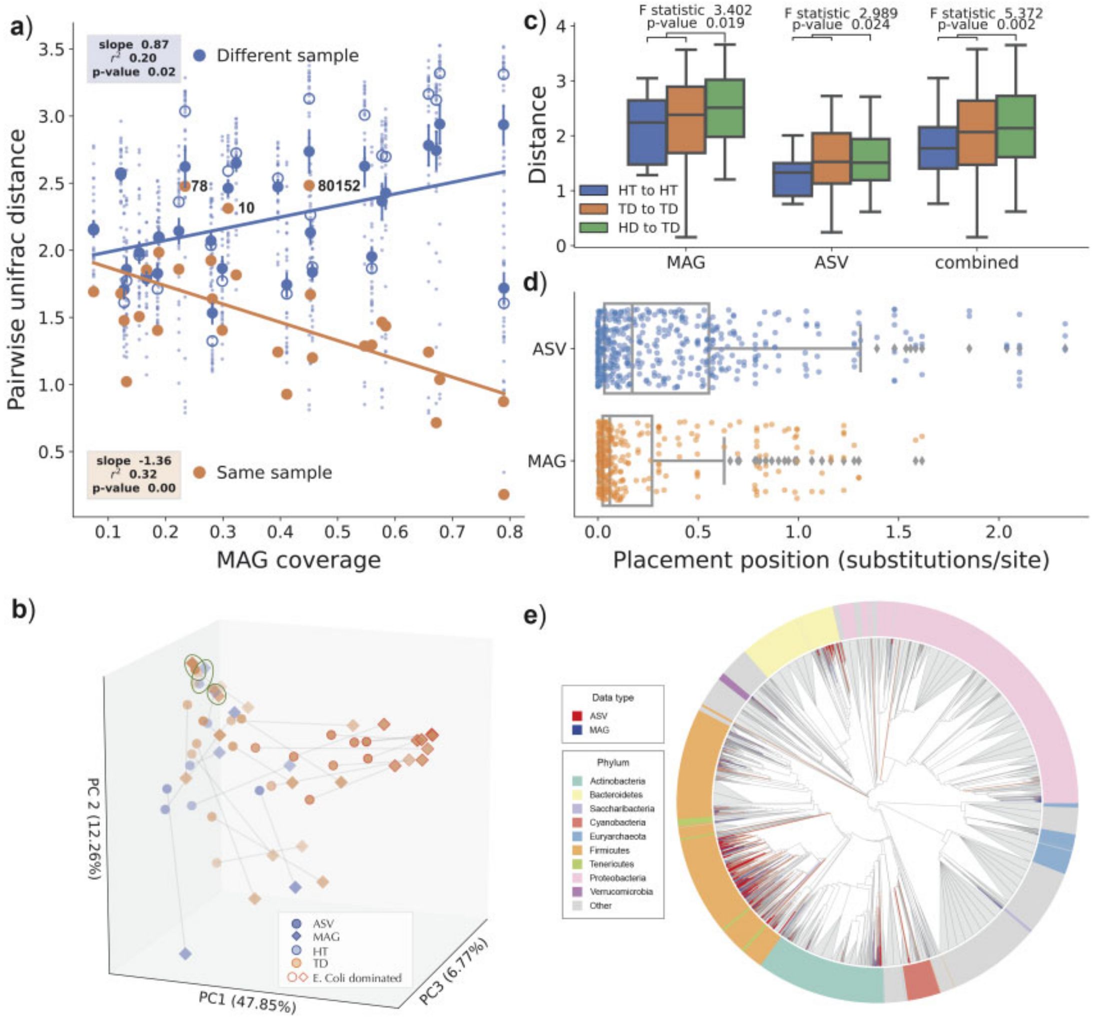

FIGURE 6. Combined MAG and ASV results on the TD data set. a) For every sample, the UniFrac distance (y-axis) is shown between its MAG placements and ASV placements of the same sample and ASV placements of other samples versus the MAG coverage (proportion of sequencing data covered by the MAGs). For intersample comparisons, we show all pairs of comparisons (small dots), the mean and standard error (large sold dots and bars), and the median (empty circles). b) PCoA analyses of combined ASV and MAG placements based on weighted UniFrac Distances. Each sample is represented with two connected dots, one for MAG and one for ASV. Examples of samples where ASV and MAG have very low distances are highlighted in green. Nine samples dominated by *E. coli* are highlighted with a solid border. c) Distribution of UniFrac distances among pairs of samples within HT or TD group and across the groups, using MAGs, ASVs or the two data combined. F-statistic and P-values are calculated using the PERMANOVA test. d) Distance of the placement position (geometric mean of tMRCA of the species in the closest sibling clade to the query) using ASV and MAG data; each dot represents one bin or ASV. e) The ASV and MAG placements on the WoL phylogenetic tree (the lightest shade of grey: backbone taxa).

Saccharibacteria which is not represented in the ASV placements. This group of bacteria is classified under the candidate phyla radiation, which has distinct physiological and genetic characteristics from all remaining bacteria. Commonly used 16S rRNA primers have reduced sensitivity in capturing the Saccharibacteria group (Castelle and Banfield 2018) that, according to Zhu et al. (2018), may be responsible for the disease status. On the other hand, the ASV placements include several clades under Cyanobacteria not found by MAGs. These may represent the commonly seen contamination from chloroplasts of dietary plants in 16S analyses (Di Rienzi et al. 2013), a problem that does not afflict metagenomic assembly.

Beyond the detected groups, branch lengths also reveal interesting patterns. While MAGs tend to be placed close to the tips, a majority of ASV placements are deep in the tree (Fig. 6d). These more basal placements of ASVs are consistent with the lower phylogenetic signal included in short sequences, which can lead to less specific characterizations. Shorter terminal branches highlight the advantage of MAGs versus 16S rRNA amplicons in understanding microbiome compositions.

We observed substantial correlation (r^2: 0.57; P-value $< 10^{-5}$) between the Faith's phylogenetic (alpha) diversity computed using ASV and MAG placements (Fig. S14 of the Supplementary material available on Dryad). These strong but imperfect correlations once again show the two sources of data capture similar but subtly different patterns. Overall, due to the more basal placements discussed earlier, alpha diversity measured using ASV tends to be higher. Several low alpha diversity cases are related to nine TD samples reported by Zhu et al. (2018) to be dominated by *Escherichia coli*. The first axis in the combined PCoA analysis clearly separates these samples from others using MAGs, but the separation is less strong using ASVs, a pattern observed if PCoA analyses are performed separately for ASV and MAGs (Fig. S13b,c of the Supplementary material available on Dryad).

Discussion

We introduced a deep-learning approach for extending an existing phylogenetic tree without a need for prespecified models of sequence evolution or gene tree discordance. Our approach learns how to add new taxa by capturing patterns in an existing reference set. 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. Given a correct sequence evolution model and no discordance, we see no reason machine learning should be more accurate than traditional phylogenetics. The novelty of DEPP is in its ability to learn from data without knowing the model. For example, in our simulations where the GTR+Gamma model of sequence evolution and MSC or HGT models of gene tree discordance were used, DEPP was able to match the accuracy of maximum likelihood placement (and surpassed it when discordance was high) without any direct knowledge of the underlying model. Note that once its model is trained (a one time process per reference tree), DEPP provides substantial running time and memory advantages compared to EPA-ng, even when their accuracy is similar (Table S5 of the Supplementary material available on Dryad).

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 data set. We leave the exploration of such applications to future work.

The specific formulation that we chose, embedding sequences in high-dimensional spaces, allowed us to define a loss function that can be easily optimized using back propagation. One can argue that the ideal loss function for placement would be one that evaluates the accuracy of the final placement, not the distances. Designing such loss functions would be easy enough. However, optimizing a loss function with discrete components (e.g., the placement branch) will loose the differentiability of the loss function, which is necessary for backpropagation using standard methods. Finding ways to perform backpropagation in partially differentiable spaces like phylogenetic placement is an interesting topic for 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 data sets 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 $\binom{200}{2}=19,900$ 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 from 128 to 32 to observe substantial drops in accuracy on simulated data (Table S2 of the Supplementary material available on Dryad). Note that LR embedding states that $n-1$ dimensions are sufficient, but it does not state that $n-1$ dimensions are necessary. Future work should explore recent advances in hyperbolic neural networks (Ganea et al. 2018) and hyperbolic distances (Tabaghi and Dokmanić 2020) to overcome limitations of Euclidean distances.

Our choices of hyperparameters such as k and the training parameters such as the stopping criterion were based on preselected values that were not fine-tuned on any data set. In extensive simulations, this simple procedure was necessary due to computational reasons. When applying in practice on real data, it is possible to fine-tune all the hyperparameters using a validation set. We can first randomly select a subset of the reference species as the validation set, use these as testing data to tune the parameters for the given data set, and then train one last time with all the data with the fine-tuned parameters. Note that such a procedure will require repeated training and can be slow, and our preliminary results (Table S2 of the Supplementary material available on Dryad) show that the gain in accuracy obtained can be small.

Our results on the real prokaryotic data set shed light on the ability of DEPP to overcome horizontal transfer in some but not all cases. HGT is the main cause of gene tree discordance in prokaryotes, even for the marker genes such as 16S (Gogarten et al. 2002). 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. S8 of the Supplementary material available on Dryad). 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, unobserved sequence patterns can negatively impact the accuracy of DEPP. For the cases with high errors, we observe distances being overestimated, especially when the ML distances are low. 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). We hope to remedy this limitation of our model in future work by more explicit modeling of unseen data, data augmentation, or changing the loss function.

The most immediate use of DEPP is in connecting 16S and metagenomics, as we demonstrated in our case study. DEPP often places 16S within three branches of the correct edge in the species tree; thus, while some errors remain, DEPP results enable combined 16S and metagenomic analyses with high accuracy. The current practice to combine results from 16S and metagenomic data is to use each data type to perform *taxonomic* identification and use the resulting classification in downstream analyses. Taxonomic classification clearly has less resolution than phylogenetic placement. DEPP (and more broadly, discordant placement) allows a phylogenetic, instead of taxonomic, approach for combining data. Once sequences from all the data types are added to the same tree, many downstream measurements, such as UniFrac distances and beta diversity, can be measured on the combined data. Our case study demonstrated that DEPP is capable of resolving real microbial community structures using either 16S amplicon or metagenomic data. Moreover, on this data set, we observed improved ability to distinguish healthy and diseased samples by combining 16S and MAG data. Thus, combining both sources of data can reveal patterns that are relevant to the pathogenic profiles of the samples and the clinical status of the subjects. We saw remarkable levels of agreement between MAGs and ASV data in our case study, but also disagreements. While our data tended to provide evidence supporting the advantages of metagenomics over 16S amplicons, some limitations of MAGs were also revealed. Thus, due to pros and cons of each data type, the two sources of information are likely to remain complementary. Given 16S and metagenomic data, DEPP can add *all* samples to the same underlying tree, and this unified view of multiple data types enables downstream statistical analyses (e.g., Unifrac) to analyze both sets of samples jointly.

Supplementary material

Data available from the Dryad Digital Repository: https://doi.org/10.6076/D14G68.

Data and Code Availability

DEPP is publicly available at https://github.com/yueyujiang/DEPP.

Data from this article are available at https://tera-trees.com/data/depp/.

Funding

This work was supported by the National Institute of Health (NIH) grant 1R35GM142725 to Y.J. and S.M.; National Science Foundation (NSF) [IIS-1845967 to Y.J. and S.M.]; NSF [NSF-1815485 to M.B. and S.M.]; and an Arizona State University start-up grant to Q.Z.; Computations were performed on the San Diego Supercomputer Center (SDSC) through XSEDE allocations, which is supported by the NSF [ACI- 1053575].

References

Anderson M.J. 2001. A new method for non-parametric multivariate analysis of variance. Austral Ecol. 26(1):32–46.

Asnicar F., Thomas A.M., Beghini F., Mengoni C., Manara S., Manghi P., Zhu Q., Bolzan M., Cumbo F., May U., Sanders J.G., Zolfo M., Kopylova E., Pasolli E., Knight R., Mirarab S., Huttenhower C., Segata N. 2020. Precise phylogenetic analysis of microbial isolates and genomes from metagenomes using PhyloPhlAn 3.0. Nat. Commun. 11(1):2500.

Balaban M., Jiang Y., Roush D., Zhu Q., Mirarab S. 2022. Fast and accurate distance-based phylogenetic placement using divide and conquer. Mol. Ecol. Res. 22(3):1213–1227.

Balaban M., Sarmashghi S., Mirarab S. 2020. APPLES: scalable distance-based phylogenetic placement with or without Alignments. Syst. Biol. 69(3):566–578.

Ballesteros J.A., Hormiga G. 2018. Species delimitation of the North American orchard-spider Leucauge venusta (Walckenaer, 1841) (Araneae, Tetragnathidae). Mol. Phylogenet. Evol. 121:183–197.

Barbera P., Kozlov A.M., Czech L., Morel B., Darriba D., Flouri T., Stamatakis A. 2019. EPA-ng: massively parallel evolutionary placement of genetic sequences. Syst. Biol. 68(2):365–369.

Barron J.T. 2017. Continuously differentiable exponential linear units. *arXiv*, pages arXiv–1704.

Berger S.A., Krompass D., Stamatakis A. 2011. Performance, accuracy, and Web server for evolutionary placement of short sequence reads under maximum likelihood. Syst. Biol. 60(3):291–302.

Beyer W.A., Stein M.L., Smith T.F., Ulam S.M. 1974. A molecular sequence metric and evolutionary trees. Math. Biosci. 19(1-2):9–25.

Bohmann K., Mirarab S., Bafna V., Gilbert M.T.P. 2020. Beyond DNA barcoding: the unrealized potential of genome skim data in sample identification. Mol. Ecol. 29(14):2521–2534.

Bolyen E., Rideout J.R., Dillon M.R., Bokulich N.A., Abnet C.C., Al-Ghalith G.A., Alexander H., Alm E.J., Arumugam M., Asnicar F., Bai Y., Bisanz J.E., Bittinger K., Brejnrod A., Brislawn C.J., Brown C.T., Callahan B.J., Caraballo-Rodríguez A.M., Chase J., Cope E.K., Da Silva R., Diener C., Dorrestein P.C., Douglas G.M., Durall D.M., Duvallet C., Edwardson C.F., Ernst M., Estaki M., Fouquier J., Gauglitz J.M., Gibbons S.M., Gibson D.L., Gonzalez A., Gorlick K., Guo J., Hillmann B., Holmes S., Holste H., Huttenhower C., Huttley G.A., Janssen S., Jarmusch A.K., Jiang L., Kaehler B.D., Kang K.B., Keefe C.R., Keim P., Kelley S.T., Knights D., Koester I., Kosciolek T., Kreps J., Langille M.G.I., Lee J., Ley R., Liu Y.-X., Loftfield E., Lozupone C., Maher M., Marotz C., Martin B.D., McDonald D., McIver L.J., Melnik A.V., Metcalf J.L., Morgan S.C., Morton J.T., Naimey A.T., Navas-Molina J.A., Nothias L.F., Orchanian S.B., Pearson T., Peoples S.L., Petras D., Preuss M.L., Pruesse E., Rasmussen L.B., Rivers A., Robeson M.S., Rosenthal P., Segata N., Shaffer M., Shiffer A., Sinha R., Song S.J., Spear J.R., Swafford A.D., Thompson L.R., Torres P.J., Trinh P., Tripathi A., Turnbaugh P.J., Ul-Hasan S., van der Hooft J.J.J., Vargas F., Vázquez-Baeza Y., Vogtmann E., von Hippel M., Walters W., Wan Y., Wang M., Warren J., Weber K.C., Williamson C.H.D., Willis A.D., Xu Z.Z., Zaneveld J.R., Zhang Y., Zhu Q., Knight R., Caporaso J.G. 2019. Reproducible, interactive, scalable and extensible microbiome data science using qiime 2. Nat. Biotechnol. 37(8):852–857.

Buneman P. 1974. A note on the metric properties of trees. J. Combin. Theory Ser. B 17(1):48–50.

Castelle C.J., Banfield J.F. 2018. Major new microbial groups expand diversity and alter our understanding of the tree of life. Cell 172(6):1181–1197.

de Vienne D.M., Ollier S., Aguileta G. 2012. Phylo-MCOA: a fast and efficient method to detect outlier genes and species in phylogenomics using multiple co-inertia analysis. Mol. Biol. Evol. 29(6):1587–1598.

Degnan J.H., Salter L.A. 2005. Gene tree distributions under the coalescent process. Evolution 59(1):24–37.

Desper R., Gascuel O. 2002. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. J. Comput. Biol. 9(5):687–705.

Devlin J., Chang M.-W., Lee K., Toutanova K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Di Rienzi S.C., Sharon I., Wrighton K.C., Koren O., Hug L.A., Thomas B.C., Goodrich J.K., Bell J.T., Spector T.D., Banfield J.F., Ley R.E. 2013. The human gut and groundwater harbor non-photosynthetic bacteria belonging to a new candidate phylum sibling to Cyanobacteria. eLife 2:e01102.

Dopazo J., Carazo J.M. 1997. Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. J. Mol. Evol. 44(2):226–233.

Doyon J.-P.J., Ranwez V., Daubin V., Berry V. 2011. Models, algorithms and programs for phylogeny reconciliation. Brief. Bioinformatics 12(5):392-400.

Felsenstein J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17(6):368–376.

Fitch W.M., Margoliash E. 1967. Construction of phylogenetic trees. Science 155(3760):279–284.

Fletcher W., Yang Z. 2009. INDELible: a flexible simulator of biological sequence evolution. Mol. Biol. Evol. 26(8):1879–1888.

Ganea O.E., Bécigneul G., Hofmann T. 2018. Hyperbolic neural networks. In: Bengio S., Wallach H., Larochelle H., Grauman K., Cesa-Bianchi N., Garnett R., editors. Advances in Neural Information Processing Systems, vol. 2018. Red Hook (NY): Curran Associates Inc. p. 5345–5355.

Gascuel O. 2000. On the optimization principle in phylogenetic analysis and the minimum-evolution criterion. Mol. Biol. Evol. 17(3):401–405.

Gogarten J.P., Doolittle W.F., Lawrence J.G. 2002. Prokaryotic evolution in light of gene transfer. Mol. Biol. Evol. 19(12):2226–2238.

Halko N., Martinsson P.-G., Shkolnisky Y., Tygert M. 2011. An algorithm for the principal component analysis of large data sets. SIAM J. Sci. Comput. 33(5):2580–2594.

Halpern A.L., Bruno W.J. 1998. Evolutionary distances for protein-coding sequences: modeling site-specific residue frequencies. Mol. Biol. Evol. 15(7):910–917.

Handelsman J. 2004. Metagenomics: application of genomics to uncultured microorganisms. Microbiol. Mol. Biol. Rev. 68(4):669–685.

He K., Zhang X., Ren S., Sun J. 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, p. 770–778.

Hebert P.D.N., Cywinska A., Ball S.L., DeWaard J.R. 2003. Biological identifications through DNA barcodes. Proc. R. Soc. Lond. Ser. B 270(1512):313–321.

Janssen S., McDonald D., Gonzalez A., Navas-Molina J.A., Jiang L., Xu Z.Z., Winker K., Kado D.M., Orwoll E., Manary M., Mirarab S., Knight R. 2018. Phylogenetic placement of exact amplicon sequences improves associations with clinical information. mSystems 3(3):00021–18.

Jermiin L.S., Catullo R.A., Holland B.R. 2020. A new phylogenetic protocol: dealing with model misspecification and confirmation bias in molecular phylogenetics. NAR Genomics Bioinformatics 2(2):lqaa041.

Jukes T.H., Cantor C.R. 1969. Evolution of protein molecules. In: Mammalian protein metabolism, vol. III (1969). p. 21–132.

Kendall D.G. 1948. On the generalized "birth-and-death" process. Ann. Math. Stat. 19(1):1–15.

Killoran N., Lee L.J., Delong A., Duvenaud D., Frey B.J. 2017. Generating and designing DNA with deep generative models. arXiv preprint arXiv:1712.06148.

Konstantinidis K.T., Tiedje J.M. 2005. Genomic insights that advance the species definition for prokaryotes. Proc. Natl. Acad. Sci. USA 102(7):2567–2572.

Kozlov A.M., Darriba D., Flouri T., Morel B., Stamatakis A. 2019. Raxml-ng: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. Bioinformatics 35(21):4453–4455.

Kress W.J., Erickson D.L., Jones F.A., Swenson N.G., Perez R., Sanjur O., Bermingham E. 2009. Plant DNA barcodes and a community phylogeny of a tropical forest dynamics plot in Panama. Proc. Natl. Acad. Sci. USA 106(44):18621–18626.

Lagesen K., Hallin P., Rødland E.A., Stærfeldt H.-H., Rognes T., Ussery D.W. 2007. RNAmmer: consistent and rapid annotation of ribosomal RNA genes. Nucleic Acids Res. 35(9):3100–3108.

Langille M., Zaneveld J., Caporaso J.G., McDonald D., Knights D., Reyes J., Clemente J., Burkepile D., Vega Thurber R., Knight R., Beiko R., Huttenhower C. 2013. Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. Nat. Biotechnol. 31(9):814–821.

Lartillot N., Brinkmann H., Philippe H. 2007. Suppression of long-branch attraction artefacts in the animal phylogeny using a site-heterogeneous model. BMC Evol. Biol. 7(Suppl 1):S4.

Layer M., Rhodes J.A. 2017. Phylogenetic trees and Euclidean embeddings. J. Math. Biol. 74(1-2):99–111.

Legendre P., Legendre L. 2012. Numerical ecology. Amsterdam: Elsevier.

Lozupone C., Knight R. 2005. UniFrac: a new phylogenetic method for comparing microbial communities. Appl. Environ. Microbiol. 71(12):8228–8235.

Maddison W.P. 1997. Gene trees in species trees. Syst. Biol. 46(3):523–536.

Mai U., Mirarab S. 2022. Completing gene trees without species trees in sub-quadratic time. Bioinformatics 38(6):1532–1541.

Mallo D., De Oliveira Martins L., Posada D. 2016. SimPhy: phylogenomic simulation of gene, locus, and species trees. Syst. Biol. 65(2):334–344.

Matsen F.A. 2015. Phylogenetics and the human microbiome. Syst. Biol. 64(1):e26–e41.

Matsen F.A., Evans S.N. 2013. Edge principal components and squash clustering: using the special structure of phylogenetic placement data for sample comparison. PLoS One 8(3):e56859.

Matsen F.A., Kodner R.B., Armbrust E.V. 2010. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. BMC Bioinformatics 11(1):538.

McDonald D., Vázquez-Baeza Y., Koslicki D., McClelland J., Reeve N., Xu Z., Gonzalez A., Knight R. 2018. Striped unifrac: enabling microbiome analysis at unprecedented scale. Nat. Methods 15:847–848.

Mirarab S., Nguyen N., Warnow T. 2012. SEPP: SATé-Enabled Phylogenetic Placement. In: Pacific Symposium on Biocomputing. World Scientific, p. 247–258.

Mirarab S., Warnow T. 2015. ASTRAL-II: coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. Bioinformatics 31(12):i44–i52.

Moshiri N. 2020. TreeSwift: a massively scalable Python tree package. SoftwareX 11:100436.

Munch K., Boomsma W., Willerslev E., Nielsen R. 2008. Fast phylogenetic DNA barcoding. Philos. Trans. R. Soc. B 363(1512): 3997–4002.

Naser-Khdour S., Minh B.Q., Zhang W., Stone E.A., Lanfear R. 2019. The prevalence and impact of model violations in phylogenetic analysis. Genome Biol. Evol. 11(12):3341–3352.

Nguyen N.-P., Mirarab S., Kumar K., Warnow T. 2015. Ultra-large alignments using phylogeny-aware profiles. Genome Biol. 16(1):124.

Nguyen N.-P., Mirarab S., Liu B., Pop M., Warnow T. 2014. TIPP: taxonomic identification and phylogenetic profiling. Bioinformatics 30(24):3548–3555.

Ochman H., Lawrence J.G., Groisman E.A. 2000. Lateral gene transfer and the nature of bacterial innovation. Nature 405(6784): 299–304.

Parks D.H., Chuvochina M., Chaumeil P.-A., Rinke C., Mussig A.J., Hugenholtz P. 2020. A complete domain-to-species taxonomy for Bacteria and Archaea. Nat. Biotechnol. 38(9):1079–1086.

Politis D.N., Romano J.P., Wolf M. 1999. Subsampling. Berlin, Germany: Springer Science & Business Media.

Price M.N., Dehal P.S., Arkin A.P. 2010. FastTree-2 – approximately maximum-likelihood trees for large alignments. PLoS One 5(3):e9490.

Quicke D.L.J., Alex Smith M., Janzen D.H., Hallwachs W., Fernandez-Triana J., Laurenne N.M., Zaldívar-Riverón A., Shaw M.R., Broad G.R., Klopfstein S., Shaw S.R., Hrcek J., Hebert P.D.N., Miller S.E., Rodriguez J.J., Whitfield J.B., Sharkey M.J., Sharanowski B.J., Jussila R., Gauld[deceased] I.D., Chesters D., Vogler A.P. 2012. Utility of the DNA barcoding gene fragment for parasitic wasp phylogeny (Hymenoptera: Ichneumonoidea): data release and new measure of taxonomic congruence. Mol. Ecol. Resour. 12(4):676–685.

Rabiee M., Mirarab S. 2020. INSTRAL: discordance-aware phylogenetic placement using quartet scores. Syst. Biol. 69(2):384–391.

Rachtman E., Sarmashghi S., Bafna V., Mirarab S. Forthcoming. 2021. Uncertainty quantification using subsampling for assembly-free estimates of genomic distance and phylogenetic relationships. Cell Syst.

Robinson D., Foulds L. 1981. Comparison of phylogenetic trees. Math. Biosci. 53(1-2):131–147.

Sand A., Holt M., Johansen J., Fagerberg R., Brodal G., Pedersen C., Mailund T. 2013. Algorithms for computing the triplet and quartet distances for binary and general trees. Biology 2(4):1189–1209.

Seifert K.A., Samson R.A., DeWaard J.R., Houbraken J., Levesque C.A., Moncalvo J.-M., Louis-Seize G., Hebert P.D.N. 2007. Prospects for fungus identification using CO1 DNA barcodes, with Penicillium as a test case. Proc. Natl. Acad. Sci. USA 104(10):3901–3906.

Stamatakis A. 2014. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics 30(9):1312–1313.

Sullivan J., Swofford D.L. 2001. 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? Syst. Biol. 50(5):723–729.

Sunagawa S., Mende D.R., Zeller G., Izquierdo-Carrasco F., Berger S.A., Kultima J.R., Coelho L.P., Arumugam M., Tap J., Nielsen H.B., Rasmussen S., Brunak S., Pedersen O., Guarner F., de Vos W.M., Wang J., Li J., Doré J., Ehrlich S.D., Stamatakis A., Bork P. 2013. Metagenomic species profiling using universal phylogenetic marker genes. Nat. Methods 10(12):1196–1199.

Suvorov A., Hochuli J., Schrider D.R. 2020. Accurate inference of tree topologies from multiple sequence alignments using deep learning. Syst. Biol. 69(2):221–233.

Tabaghi P., Dokmanić, I. 2020. Hyperbolic distance matrices. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1728–1738.

Tavaré S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. Lectures on Mathematics in the Life Sciences, vol. 17. p. 57–86.

Tieleman T., Hinton G. 2012. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA: Neural Netw. Mach. Learn. 4(2):26–31.

Truong D.T., Franzosa E.A., Tickle T.L., Scholz M., Weingart G., Pasolli E., Tett A., Huttenhower C., Segata, N. 2015. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. Nat. Methods 12(10):902–903.

Warnow T. 2017. Computational phylogenetics: an introduction to designing methods for phylogeny estimation. Cambridge, United Kingdom: Cambridge University Press.

Xie P., Wu W., Zhu Y., Xing E. 2018. Orthogonality-promoting distance metric learning: convex relaxation and theoretical analysis. In: International Conference on Machine Learning, PMLR. p. 5403–5412.

Yin J., Zhang C., Mirarab S. 2019. ASTRAL-MP: scaling ASTRAL to very large datasets using randomization and parallelization. Bioinformatics 35(20):3961–3969.

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

Zaneveld J.R., Lozupone C., Gordon J.I., Knight R. 2010. Ribosomal RNA diversity predicts genome diversity in gut bacteria and their relatives. Nucleic Acids Res. 38(12): 3869–3879.

Zhu Q., Dupont C.L., Jones M.B., Pham K.M., Jiang Z.-D., DuPont H.L., Highlander S.K. 2018. Visualization-assisted binning of metagenome assemblies reveals potential new pathogenic profiles in idiopathic travelers' diarrhea. Microbiome 6(1):201.

Zhu Q., Mai U., Pfeiffer W., Janssen S., Asnicar F., Sanders J.G., Belda-Ferre P., Al-Ghalith G.A., Kopylova E., McDonald D., Kosciolek T., Yin J.B., Huang S., Salam N., Jiao J.-Y., Wu Z., Xu Z.Z., Cantrell K., Yang Y., Sayyari E., Rabiee M., Morton J.T., Podell S., Knights D., Li W.-J., Huttenhower C., Segata N., Smarr L., Mirarab S., Knight R. 2019. Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea. Nat. Commun. 10(1):5477.

Zou Z., Zhang H., Guan Y., Zhang J. 2020. Deep residual neural networks resolve quartet molecular phylogenies. Mol. Biol. Evol. 37(5):1495–1507.