APPLES-2: Faster and More Accurate Distance-based Phylogenetic Placement using Divide and Conquer

Metin Balaban,[1] Daniel Roush,[2] Qiyun Zhu,[2] and Siavash Mirarab[3*]

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

Abstract

Phylogenetic placement of query samples on an existing phylogeny is increasingly used in microbiome analyses and other biological studies. As the size of available reference trees used in microbiome analyses continues to grow, there is a growing need for methods that place sequences on ultra-large trees with high accuracy. In this paper, we build on our previous work and introduce APPLES-2, a distance-based phylogenetic placement method. In extensive studies, we show that APPLES-2 is more accurate and much more scalable than APPLES and even some of the leading maximum likelihood methods. This scalability is owed to a divide-and-conquer technique that limits distance calculation and phylogenetic placement to parts of the tree most relevant to each query. The increased scalability and accuracy enables us to study the effectiveness of APPLES-2 for placing microbial genomes on a dataset of 10,575 microbial species using subsets of 381 marker genes. APPLES-2 has very high accuracy in this setting, placing 97% of query genomes within three branches of the optimal position in the species tree using 50 marker genes. As a proof of concept, we also tested APPLES-2 for placing scaffolds produced by metagenomic assembly. It was able to place 3318 scaffolds on the backbone tree with 10575 taxa in less than 3 hours. The accuracy of these placement was high as long as a scaffold included tens of marker genes.

Keywords: Phylogenetic placement, Distance-based methods, Metagenomics, Microbiome

1 Introduction

Phylogenetic placement of query samples on an existing phylogeny is increasingly used in diverse downstream applications such as microbiome profiling [1–7] and epidemic tracking [8, 9]. A main attraction of placing new sequences onto an existing phylogeny is computational expediency: the running time of phylogenetic placement is a fraction of the time needed for *de novo* reconstruction and can grow linearly with the number query samples assuming they are placed independently. To take advantage of this potential, many methods have been developed using a wide range of algorithmic techniques [e.g., 10–18].

A major attraction of phylogenetic placement is that it enables placement of sequences on very large trees. In applications of placement for microbiome analyses, sequences obtained from amplicon sequencing or metagenomic samples are placed into a reference phylogeny composed of known organisms. Depending on the datatype and pipeline, we may decide to place reads directly [4, 12] or may place marker genes obtained from metagenome-assembled genomes (MAGs) [6]. Large 16S databases have existed for more than a decade [19, 20] and genome-wide references trees with ten thousand species and more have been developed recently [e.g., 21, 22]. Moreover, close to a million microbial genomes are available in the RefSeq and GenBank databases. Although there is much redundancy among assembled genomes, we can expect that even larger and more diverse reference trees will be available in the near future. Development of bigger reference sets has a strong motivation: the density of reference set has been known to play a crucial role in the accuracy of downstream analyses [23–25]. Thus, if downstream methods can handle them, we should ideally use these dense reference datasets.

Despite their promise, two types of challenges emerge when reference datasets increase in size: scalability and accuracy. The issue of scalability is well-understood: placement methods may not be able to place on ultra-large reference trees with reasonable running time, and equally important, with reasonable amounts of memory. Moreover, handling ultra large reference trees can be subject to numerical issues. Less appreciated is the observation that as the dataset size increases, the accuracy of the algorithms may reduce and updated strategies may be needed. Thus, for placement methods to reach their full potential and take advantage of the available ultra-large reference trees, both scalability and accuracy of existing methods need to improve.

One recent advance in phylogenetic placement on ultra-large reference trees is the introduction of distance-based placement. Balaban et al. [26] introduced a Least Squares Phylogenetic Placement (LSPP) framework and a method called APPLES for distance-based placement. The input to APPLES is a reference (*a.k.a* backbone) phylogenetic tree T with n leaves and a vector of distances δ_{qi} between the query taxon q and every taxon i on T. Although machine-learning based methods shows substantial promise [17], typically, input distances are obtained by calculating sequence distances between query and backbone taxa followed by a phylogenetic correction using a statistically consistent method under a model such as Jukes-Cantor(JC69) [27]. APPLES introduced a dynamic programming algorithm to find a placement of q that minimizes weighted least squares error $\sum_{i=1}^{n} w_{qi}(\delta_{qi} - d_{qi}(T))^2$ where $d_{qi}(T)$ represents the path distance from q to backbone taxon i on T. The weights w_{qi} reduce the impact of large distances, which have larger variance, and by default w_{qi} is set to δ_{qi}^{-2} following Fitch and Margoliash [28] (FM) weighting. In extensive simulation studies, APPLES was found to come very close in accuracy to a leading maximum likelihood method (pplacer [29]), but unlike ML methods, was able to scale to trees with up to 200,000 taxa.

Despite the relatively high accuracy and scalability of APPLES, it has room for improvement. Its memory usage and speed both grow linearly with the size of the dataset, which can start to become slow for references with many hundreds of thousands of species. A bigger challenge is that computing distances across very diverse species found in ultra-large trees can become problematic, an issue that APPLES only tried to address using FM weighting. A more direct algorithm that accounts for very diverse sequences in the backbone has the potential to further improve accuracy. Moreover, APPLES lacked several features that help usability (including handling of amino acids and building precomputed reference packages). Finally, APPLES has not been previously tested in the context of microbiome analyses where it has the most potential. In this paper, we introduce APPLES-2, a new version of APPLES that improves both accuracy and scalability. We test APPLES2 on new simulated and empirical datasets representative of microbiome analyses. We show that it can place scaffolds from a metagenomic sample onto a large reference tree of more than 10,000 species both given individual marker genes or concatenations of several markers.

2 Results

2.1 APPLES-2 Framework

To improve the accuracy, scalability, and usability of APPLES, we designed a new version called APPLES-2, which includes a major algorithmic change and several new features.

The most consequential change in APPLES-2 is that it adopts a divide-and-conquer approach to improve both accuracy and scalability using two inter-related techniques. There is strong evidence in distance-based phylogenetics literature that using short distances instead of long distances can result in dramatic improvements in accuracy [30–32]. For example, the DCM family of methods that result in fast converging methods [33–35] mostly rely on dividing taxa into smaller subsets with lower distances. To take advantage of this insight, we enable APPLES-2 to use distances that are either smaller than a threshold d_f or among the lowest b distances (Material and Methods). Ignoring distances larger than the d_f threshold also gives us an opportunity to avoid computing all n distances so that the running time could grow sublin-

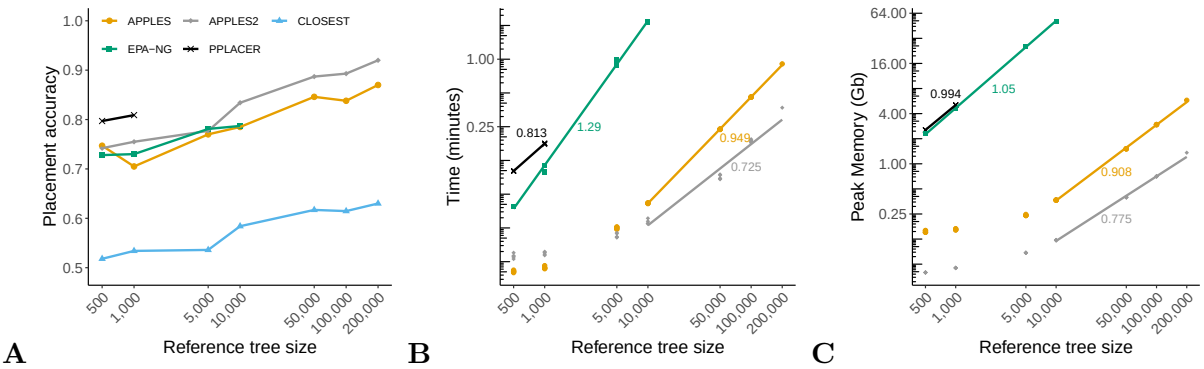

Figure 1: **Results on RNASim-VS**. Placement accuracy (a), running time (b), and peak memory usage (c) for a single placement with taxon sampling ranging from 500 to 200,000. (b,c) Lines are fitted in the log–log scale and their slope (indicated on the figure) empirically estimates the polynomial degree of the asymptotic growth. APPLES and APPLES-2 lines are fitted to ≥ 10000 points because the earlier values are small and irrelevant to asymptotic behavior. All calculations are on 8-core, 2.6GHz Intel Xeon CPUs (Sandy Bridge) with 64GB of memory, with each query placed independently and given 1 CPU core and the entire memory. Missing results indicate that the tool fails to run or complete in 48h.

early with the size of the reference tree. To do so, we divide the backbone tree T into subsets that are somewhat larger than d_f in diameter (maximum pairwise path distance between any two leaves), choose one representative from each subset, and compute distances of the query only to these representatives. Then, we compute all distances in the cluster with the least distance to our query taxon. See Material and Methods for details.

Besides divide-and-conquer, APPLES-2 adds several new features detailed in Material and Methods: *i*) It enables protein sequences as input using the Scoredist algorithm [36]. *ii*) It enables balanced minimum evolution (BME) weighting $w_{qi} = 2^{-(1+p_{qi})}$ [37]. *iii*) It adds the capability to precompute and load an APPLES database to save much of the preprocessing time. *iv*) It enables automatic re-estimation of backbone branch lengths using FastTree-2 [38].

2.2 Leave-out Experiments

We performed a series of leave-out experiments on several datasets where we removed one or more taxa from a backbone tree and placed them back using several methods, including APPLES, APPLES-2, EPA-NG, and pplacer (see Material and Methods). We then measure the number of branches between the obtained placement and the true placement (for simulated datasets) or the optimal placement based on the existing tree (for empirical datasets).

2.2.1 Single-gene Placement

To test single-gene accuracy, we used a simulated RNASim dataset (see Material and Methods) where all methods face model misspecification due to the non *i.i.d.* simulation procedure [39]. Despite the model misspecification, APPLES-2 is able to find the best placement of query sequences with up to 92% accuracy (placement on the correct branch) when the backbone size is $n = 200,000$ (Fig.1a, Table 1). APPLES-2 has lower mean delta error (-0.09 edges on average) and higher accuracy (+3.2% on average) compared to APPLES for all cases except for $n = 500$, where both methods have similar performance. The least accurate method is the closest match approach (placing the query as a sister taxon to the best matching backbone taxa), with accuracy that ranges between 52% to 63%.

As reported before [26], the ML method pplacer fails to run on backbone trees with 5000 sequences or more, and only distance based methods APPLES and APPLES-2 can handle trees larger than $10,000$ taxa. Among smaller trees, pplacer is the most accurate method for size 500 and 1000 and has 6% and 5% higher accuracy than APPLES-2. However, APPLES-2 either

	n = 500		n = 1,000		n = 5,000		n = 10000		n = 100,000		n = 200,000	
	%	Δe	%	Δe	%	Δe	%	Δe	%	Δe	%	Δe
APPLES-2	74	0.33	76	0.31	78	0.35	83	0.25	89	0.13	92	0.11
APPLES	75	0.32	71	0.43	77	0.37	79	0.33	84	0.25	87	0.25
CLOSEST	52	1.16	53	1.18	54	1.15	59	0.90	61	0.69	63	0.70
EPA-ng	73	0.33	73	0.31	78	0.24	79	0.22	n.p	n.p	n.p	n.p
pplacer	80	0.23	81	0.20	fail (800)	n.p	n.p	n.p	n.p	n.p	n.p	n.p

Table 1: Percentage of correct placements (shown as %) and the average placement error (Δe) on the RNASim-VS with various backbone size (n). % and Δe is over 1000 placements (except $n = 200,000$, which is over 200 placements). Running pplacer and EPA-ng was not possible (n.p) for trees with more than 10,000 leaves and failed in some cases (number of fails shown) for 5,000 and 10,000 leaves.

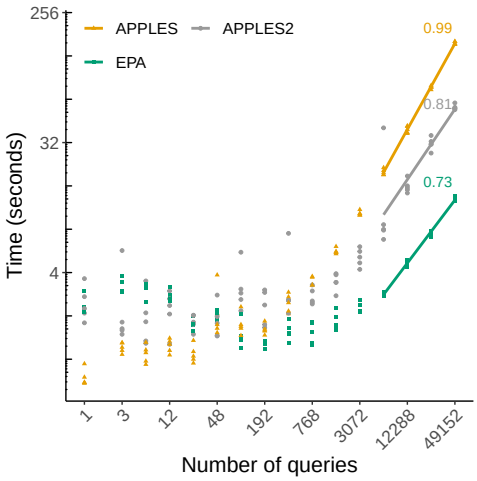

Figure 2: **Scalability with respect to the number of queries.** We show wall-clock running time with respect to increased numbers of queries placing on a tree with 500 taxa in one execution of the tool given 28 CPU cores and 28 threads on an Intel Xeon E5 CPU with 64 GB of memory. Lines are fitted to $x \geq 6144$ points because the earlier values are small and irrelevant to asymptotic behavior.

matches or improves on the accuracy of the ML method EPA-NG (for up to 4%) on instances where EPA-NG manages to complete ($n \leq 10,000$). Thus, APPLES-2 is more accurate than some ML methods (EPA-ng) and less accurate than others (pplacer).

Over all dataset sizes, we see an increased accuracy with the increased taxon sampling. Thus, the denser reference sets improve the accuracy for all methods, including APPLES-2. However, only distance-based methods can handle very large backbone tree. Thus, overall, the best accuracy over any reference set size is obtained by APPLES-2 which finds the correct placement in 92% of cases on a backbone tree of 200,000 species.

Running time and memory. Our benchmarking indicates that running time of APPLES-2 grows empirically as $O(n^{0.73})$ (Fig. 1b); this sublinear running time growth with the backbone size is consistent with our theoretical expectations (see Material and Methods). APPLES-2 is the fastest method on backbones with 5,000 or more taxa, offering up to 2.5× speed-up in average compared to APPLES on a 200,000-taxon tree. EPA-NG is faster than pplacer but slower than both versions of APPLES (with running time that grows super-linearly). On the 10,000 taxon dataset, EPA-NG takes 60× longer than APPLES-2 on average. APPLES-2 consistently uses less memory than all other tools (Fig. 1c) and is the only tool with sublinear memory complexity (empirically close to $O(n^{0.78})$). On the largest backbone tree with 200,000 taxa, APPLES-2 requires only 1.4GB of memory compared to 5.8GB needed by APPLES.

We also evaluated the impact of the number of queries on the running time (Fig. 2), comparing APPLES, APPLES-2, and EPA-NG, all run in the parallel mode. All three methods finish

placement of up to 1,536 queries in less than 4 seconds given 28 CPU cores with no clear trend in running time. Methods start to display an asymptotic behavior after placing 6144 queries or more. While APPLES scales linearly with respect to the number of queries, APPLES-2 and EPA-NG grow in sublinear rate. EPA is able to place 49152 queries in 13 seconds on average, 4.3 times faster than the second best method APPLES-2, which takes 56 seconds and is 2.8 times faster than APPLES.

2.2.2 Multi-gene Web of Life (WoL) Dataset

We then tested the utility of distance-based phylogenetic placement on the WoL biological dataset [40] with 381 marker genes and 10575 microbial taxa with 1,000 to 9,000 sequences used as backbone and the rest as query (see Material and Methods). Note that on this real dataset, the optimal placement is the one matching published species tree as the true tree is not known. We compare APPLES-2 only to APPLES because the large backbone size (up to 9000) combined with high alignment length (71,178 sites for nucleotide alignment) does not permit testing ML methods due to their high computational demand.

We start by simply concatenating the best 50 marker genes (those with minimum distance to the species tree; see Material and Methods). On these data, APPLES-2 achieves outstanding accuracy, placing query sequences with 74% accuracy and 0.54 edges of error on average on backbones with 1000 taxa (Fig. 3a). A striking 97% of the queries are placed within three or fewer branches away from the optimal branch. Note that here we are using 50/381 marker genes and a much simpler methodology compared to the original study. In comparison, APPLES achieves 60% accuracy with 1.11 average error on the same dataset. As the size of the reference increases from 1000 to 3000 and 9000, APPLES-2 is able to maintain high accuracy, placing within three branches of the optimal placement in 97% and 96% of cases, respectively. Increasing the backbone size also amplifies the difference between APPLES and APPLES-2 (going from a difference of 0.57 edges of error on average for $n = 1000$ to +0.75 and +1.04 for $n = 3000$ and $n = 9000$).

Data type. We next test the impact of data type used for placement. Removing the third codon position from nucleotide alignments improves placement accuracy substantially (Fig. 3b) for both versions of APPLES. Interestingly, APPLES-2 seems to be more robust to data type as the difference in the average error is 0.59 and 1.84 for APPLES-2 and APPLES, respectively. The third codon position often poses a stronger violation on stationarity assumption than the first and second codon positions [41, 42] and saturates faster, especially among very divergent taxa. Recall that APPLES-2 ignores distances among very divergent sequences, which is consistent with its higher robustness to the third codon position. Note that the original study [40] that built our gold-standard in these analyses inferred gene trees using amino acid data. We do not observe a significant error difference between using nucleotide (first two codon positions only) and protein sequences for APPLES-2 on the smallest backbone with $n = 1000$ (Fig. 3a). As the number of taxa in the backbone increases the accuracy gap between using nucleotide and protein sequences widens in favor of nucleotide sequences, reaching 4% for $n = 9000$. We remind the reviewer that amino acid distances are computed under the Scoredist algorithm, which is different from the models used in the original study to infer the reference tree [40].

Marker selection. Next, we investigate the impact of varying the number of marker genes used in a leave-k-out experiment ($k = 500$) where genes are either randomly chosen or among the best ones according to distance of their gene tree to the species tree (Material and Methods).

Interestingly, using all the marker genes is not the most accurate strategy (Fig. 3c). When the 50 *best* genes (i.e., those with lowest quartet distance the species tree) are selected, APPLES-2 places 472 out of 500 queries (94% rate) within two branches away from the optimal branch;

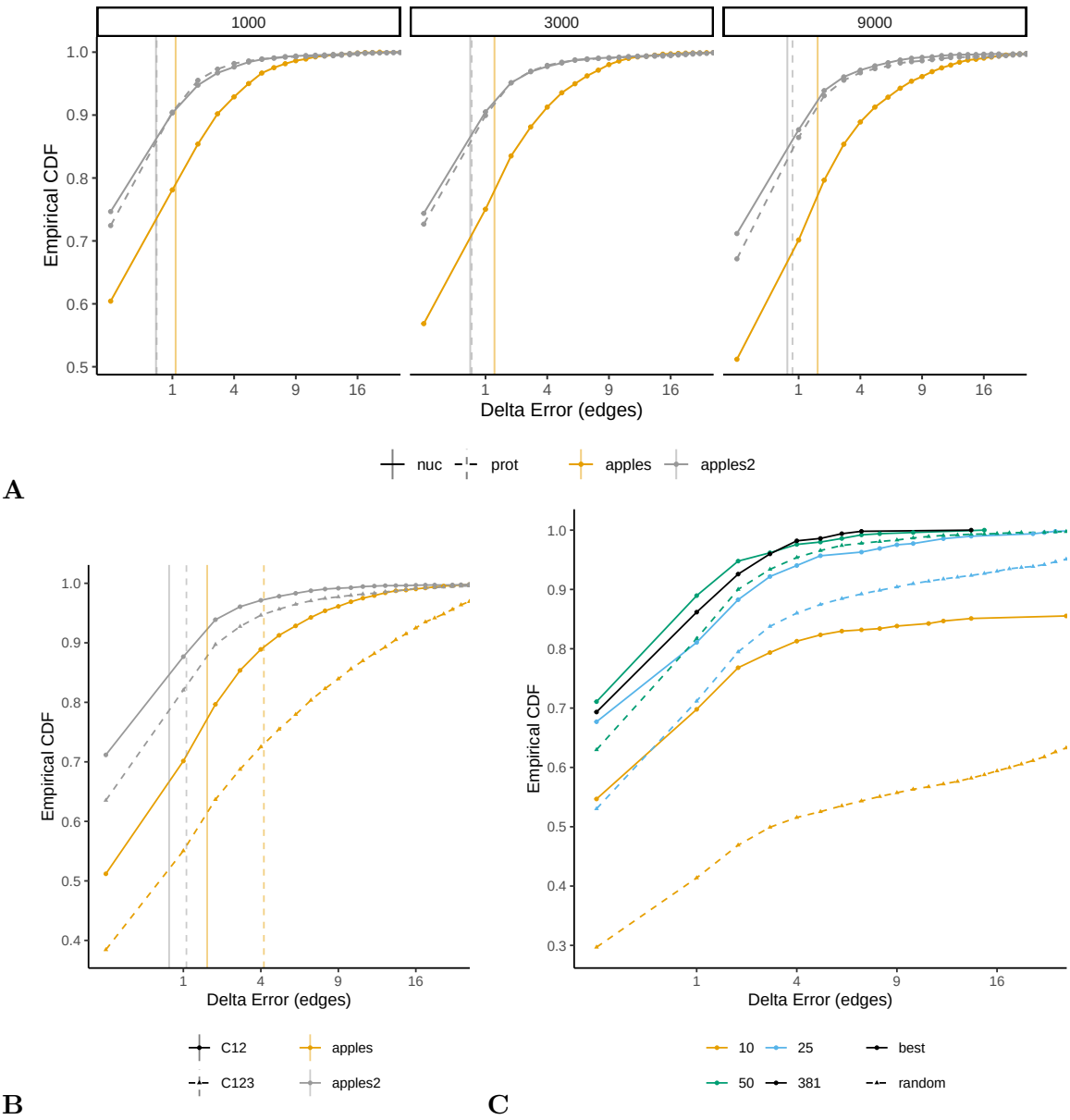

Figure 3: (a) Empirical cumulative distribution function (CDF) of placement error on backbones ranging from 1000 to 9000 taxa. Results are based on 10000 query placement for each backbone size. We show placement using nucleotide (first and second codon position) and protein sequences. (b) Placement accuracy is impacted by filtering third codon position from the alignment. C12: only the first two codon positions are retained in the alignment; C123: all three positions used. (c) Impact of marker gene selection on placement accuracy. We control for number of genes selected and gene selection strategy: choosing randomly versus genes with lowest discordance with species tree (best). Vertical lines show mean error. X-axis is displayed in square-root scale and is cut at 20 edges.

in contrast, using all genes, 462 queries are within two branches (92% rate). Using the best 50 genes results in 0.59 average delta error, which is 0.02 less than using all genes in the data set. However, reducing the number of best genes to 25 and 10 increases error to 1.09 and 5.99 edges, respectively. Thus, 50 best genes is the sweet spot in terms of accuracy among levels we tested.

There is a dramatic difference between selecting genes randomly or using best genes (Fig. 3c). A random selection of 10 genes results in very low accuracy (within 3 edges from the optimal branch only 41% of the time and and average 16.9 edges error) whereas 10 best genes result in 5.99 edges of error on average. Nevertheless, as few as 25 randomly selected genes provide acceptable placement accuracy where 84% of queries are placed within 3 edges from the optimal

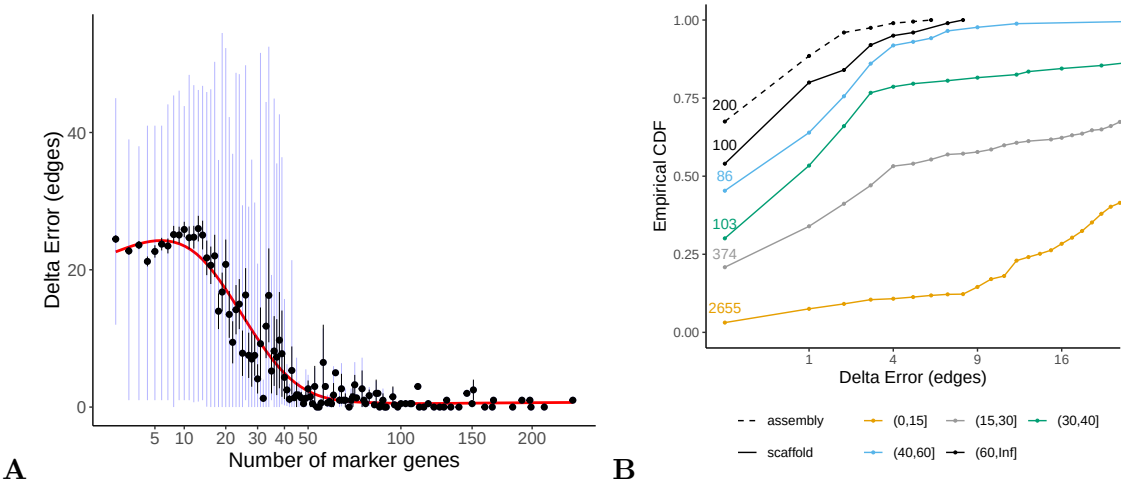

Figure 4: **Results on WoL-metagenomic data set**. a) The relationship between number of marker genes in a scaffold and the error. Dots show mean, black error bars show standard error, and light blue error bars show the central 90% range. Red curve is fitted to the data using the LOESS algorithm. X-axis is shown in square-root scale. b) Placement error CDF for simulated metagenomic assemblies and scaffolds. Each bin indicates the number of genes in the scaffold or assembly. We show the number of queries in each bin next to its curve.

location (error: 3.08 edges on average); yet, 25 best genes continue to be better (error: 1.09 edges on average). With 50 genes, the error is 0.39 less when using best genes compared to randomly selected genes. Unlike best genes, the placement error using random genes monotonically decreases as the number of genes increases, culminating in 0.61 edges when all genes are selected (Fig. S5). Overall, the difference between random and best genes is wider when few genes are available and diminishes as more genes are added. Note that *best* and *random* strategies are equivalent when all 381 genes in the data set are used,

2.3 Placement of assemblies and scaffolds

While our previous analyses showed that APPLES-2 has outstanding accuracy using best or random subsets of marker genes sampled across microbial genomes, we often do not have entire genomes. Instead, we have MAGs and scaffolds from which MAGs are generated. We next test APPLES-2 in a simulation that generated scaffolds and assemblies, similar to MAGs, by assembling reads simulated from a subset of 200 genomes in the WoL dataset (see Material and Methods). We ask whether high accuracy can be maintained and whether scaffolds contain a sufficient number of marker genes to allow accurate phylogenetic placement.

Our simulated assemblies included 105 to 365 marker genes (Fig. S6a). With these number of markers, APPLES-2 achieves 68% accuracy and places 195 of 200 simulated assemblies with error no larger than three (Fig. 4b). The error is never more than 6 edges. The placement error has a weak but statistically significant anti-correlation with the number of marker genes available in the assembly ($p = 0.005$ according to Pearson's correlation; $\rho = -0.196$; see Fig. S7).

Our assembly procedure produces 3318 unique scaffolds of $\geq$ 10kbp (Fig. S6b), among which 663 has more than fifteen marker genes and 289 has more than 30 marker genes. The placement error is clearly a function of the number of genes in each scaffold (Fig. 4a). Scaffolds with less than 15 genes have very high error on average (23.61 edges), but also have high variance (with 10% of such scaffolds leading to error up to three edges). Once scaffolds start to have more than approximately 15 genes, the error starts to fall and becomes consistently low after around 30 genes (Fig. 4a). While placement accuracy for scaffolds that contains 30 to 40 genes is 30%, 77% can be placed with error no more than 3 edges (Fig. 4b). As the number of genes in the scaffold increases, the accuracy also increases; on average, placement error for scaffolds with 60

or more genes is only one edge (1.03), 92% are within three edges of the optimal placement, and the maximum error observed is eight edges.

Both multiple sequence alignment using UPP and the phylogenetic placement step using APPLES-2 used in the scaffold placement workflow are fast. Running UPP to align all 3318 scaffolds for each gene to the backbone alignment takes 99 seconds in average (lowest 22 and highest 467 seconds) using 6 CPU cores. APPLES-2 takes 2.87 seconds per query scaffold on the backbone tree of size 10575 using 12 CPU cores.

3 Discussion

We presented APPLES-2: an improved distance based phlogenetic placement tool for inserting new taxa on large phylogenetic trees. Inspired by DCM-like methods [35], our divide-and-conquer approach improved placement accuracy beyond its predecessor APPLES and made it comparable to or better than ML-based tool EPA-NG on single gene data sets. Furthermore, we showed that APPLES-2 is even more scalable than APPLES, reducing running time and memory consumption, and can achieve high accuracy on diverse multi-gene datasets.

Some of the new features of APPLES-2 increase usability and completeness of the tool but have limited impact on accuracy and scalability. For example, we implemented BME weighting. However, despite the previous literature suggesting BME weighting is preferable to alternatives [37], we observed that BME is slightly less accurate than the default FM weighting scheme for all dataset sizes (Fig. S4; difference between FM and BME mean error is 0.26 edges on average). Based on these results, we continued to use FM as the default weighting method everywhere but provide BME as a new option to the users. Similarly, using amino acid sequences did not show any improvements, but we enable it for cases when only amino acid data is available. Despite declining opportunities, future changes could seek to further improve the accuracy. For example, at the expense of higher computational cost, one can select centroid sequences for partitions of the backbone MSA via ancestral state reconstruction instead of consensus — a technique used by Balaban et al. [43] (also see ancestral k-mers [18]).

While accuracy is typically high, on a minority of queries results of APPLES-2 are far from the correct placement. A reasonable question is whether these highly inaccurate instances can be identified by APPLES-2. While we leave a more elaborate exploration to future work, we have identified several interesting patterns (Fig. 5). First, we observe a correlation between APPLES-2's objective function value, the Minimum Least Square Error (MLSE; denoted by Q), and accuracy (Fig. 5a). We see a dramatic increase in error for $Q \geq 1$. Note that high MLSE (e.g $Q \geq 100$) does not indicate that APPLES-2 fails to optimize its objective function — APPLES-2 solves the objective problem *exactly* (i.e., is not heuristic). High MLSE can result from sequence data and tree distances being very incompatible. This incompatibility may be due to several reasons such as lack of signal, model violation, and horizontal gene transfer (HGT). Interestingly, despite its reduced accuracy, APPLES-2 can still find a good placement for many queries with $Q \geq 100$ ($\approx 50\%$ of such queries have error below 3 eges). Secondly, when a query is placed with zero pendant edge length, the placement error is significantly higher than otherwise ($p < 10^{-10}$, two-sample Wilcoxon test). The average error is respectively 16.2 and 27.9 when only pendant or both pendant and distal edge length is zero whereas it is only 0.8 on average when pendant edge length is larger than zero (Fig. 5b). We have also noticed that erroneous placements with zero pendant edge lengths are more prevalent in the query sequences with fewer genes (Fig. S8). Thus, users of APPLES-2 should be skeptical of the placements with zero pendant and distal branch length and/or high MLSE error (which APPLES-2 outputs). In future work, these features can be used to develop a predictive value indicating possible errors in placement.

Our studies on microbial data showed that APPLES-2 can phylogenetically place and hence identify genome-wide shotgun data with promising accuracy after they are assembled. Using a simulated dataset, we showed that assembled genomes can be placed on the species tree with

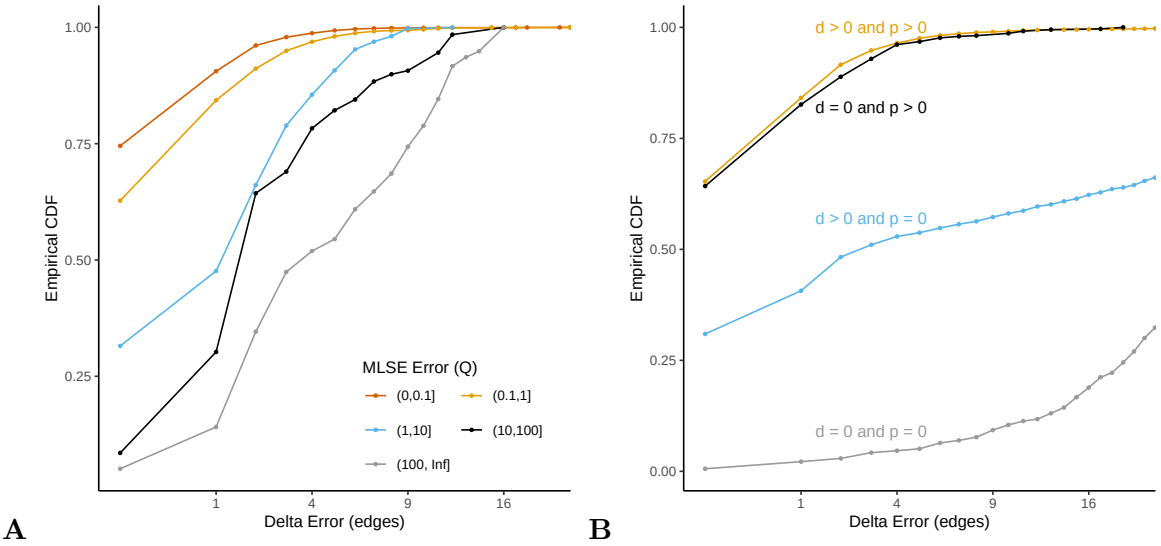

Figure 5: **Detecting erroneous placements**. Empirical CDF of placement error controlled for a) optimized MLSE objective function (Q), and b) having distal and pendant edge equal to zero. Results are based 27844 queries with 1 to 150 random marker genes placed on Web of Life dataset. Queries with zero objective function (i.e. queries that have an exact match to a backbone species) are omitted. Queries with less than 40 marker genes are omitted in panel a.

great accuracy. Patterns are more intricate for scaffolds: the accuracy tends to be low on small scaffolds with few gens but improve for scaffolds that have moderately large number of marker genes. Besides their reduced numbers of genes, scaffolds present several challenges that may contribute to their lower accuracy: (i)) In comparison to assemblies, scaffolds in a metagenomic sample are more prone to assembly errors and chimeras. (ii)) Genes located on the same syntenic block have similar gene trees, which can introduce a bias in the placement. Thus, factors such HGT and may have a bigger impact on scaffolds. (iii)) Even when a scaffold has many genes, it may not include the best marker genes; i.e., those genes with maximum signal and concordance to the species tree.

Our results clearly showed that the choice of genes matters. While a random selection of 25 marker genes were adequate for placing queries in most cases, a targeted gene selection strategy outperformed random selection (e.g., $p = 6.9 \times e^{-11}$ for 25 marker genes). The results indicate that certain marker genes serve better at predicting location of a query species in the backbone tree. This observation leads to two related questions. Given fully assembled genomes, should we use all or a subsample of available genes? Our data supports the idea that using a subset of genes is preferable. However, a more fruitful approach may be weighting genes (or even sites within genes) differently to further improve accuracy. Such a goal seems amenable to machine learning approaches that can learn optimal weights.

The second question is how to handle scaffolds from metagenomic assemblies, which include only a handful of genes. There are always more scaffolds with few genes than those with many genes. Thus, requiring a large number of genes would reduce the number of scaffolds placed, which has the potential to reduce the accuracy of downstream analyses. Our results indicate scaffolds with a modest number of genes (e.g., with 30 or more) are enough to place them phylogenetically. But the vast majority of scaffolds have fewer than 15 marker genes, and *some* of these *can* be placed accurately. We leave it to future work to design a more principled framework for deciding which scaffolds can be placed accurately and which cannot. We also leave to the future work to answer a more challenging question: for downstream applications, is it better to place a few scaffolds that have many genes (or perhaps binned contigs) with high confidence or is it better to place all or most scaffolds with lower confidence hoping that noise will be overcome by the large number of placements? Answering these questions requires careful

experimental procedures that are outside the scope of the present study.

Finally, in this paper, we focused on single query placement and observed that given multiple marker genes, APPLES-2 can insert a new genome into backbone tree with high accuracy. These results open up an exciting opportunity. By spending less computational budget than *de novo* phylogenetics, successive insertion of genomes can enable expanding the existing large microbial phylogenies [e.g., 40] to contain hundreds of thousands of sequences. Future work should explore the best pipelines for achieving this goal.

4 Material and Methods

4.1 Details of APPLES-2 Algorithm

Divide-and-conquer. Without loss of generality, we assume that for a certain query taxa q, $\delta_{q1} \leq \delta_{q2} \leq \delta_{q3} \leq \ldots \delta_{qn}$ holds. In APPLES-2, we introduce a threshold option $d_f \in \mathbb{R}_{\geq 0}$ which sets $w_{qi} = 0$ (i.e. ignore backbone taxon i) when $\delta_{qi} \geq d_f$ for a query taxon q. We introduce a second option $b \in \mathbb{N}_{\geq 0}$ which retains the standard weighting (i.e. $w_{qi} = \delta_{qi}^{-2}$) for backbone taxa $1 \leq i \leq b$. Consequently the new LSPP objective function becomes $\sum_{i=1}^{b} w_{qi}(\delta_{qi} - d_{qi}(T))^2 + \sum_{i=b+1}^{n} \mathbf{1}(d_f - \delta_{qi}) w_{qi}(\delta_{qi} - d_{qi}(T))^2$ where $\mathbf{1}(x)$ is the unit step function: $\mathbf{1}(x) = 0$ for $x < 0$ and $\mathbf{1}(x) = 1$ for $x \geq 0$. By default, APPLES-2 uses $d_f = 0.2$ and $b = 25$. These default values are chosen based on an empirical evaluation on an example dataset (Fig. S1) and are kept fixed across all of our other experiments.

To avoid computing all distances, during preprocessing of the reference set, we cluster the backbone alignment and tree T using the linear-time TreeCluster algorithm [43] (akin to OTU-clustering) to find the minimum number of clusters such that the maximum pairwise distance in each cluster is no more than $1.2 \times d_f$. The threshold 1.2 is adjustable, is chosen empirically, and APPLES-2 is robust to this choice (see Fig S3). Then, we select a representative sequence per partition by computing consensus sequence among all sequences belonging to the partition. Let $P_1, P_2, \ldots, P_k$ denote partitions of leaves of T and $C(j)$ denote centroid sequence of partition P_j. Without loss of generality, we assume that $\delta_{qC(1)} \leq \delta_{qC(3)} \leq \delta_{qC(3)} \leq \ldots \delta_{qC(k)}$. The distance between q and a backbone taxa $i \in P_j$ is computed only if either $\delta_{qC(j)} \leq d_f$ or $\sum_{i=1}^{j-1} |P_i| < b$ holds. The time complexity of distance calculation per query is in the order of $O(\max(b, m)L)$ where L is alignment length and m is number of backbone taxa whose distance to the query is less than or equal to d_f.

Since in APPLES2 a subset of distances are calculated, we have re-designed its dynamic programming algorithm so that it automatically works on the backbone tree induced to the taxa for which distances are computed. The updated dynamic programming algorithm scales with the number of edges in the induced tree, which can be as low as $O(\max(b, m))$ (if the chosen leaves are a connected subtree) and as high as $O(n)$ (when chosen leaves span all of a caterpillar tree).

New features. Four new features are added to APPLES.

1. Several tools [36, 44–46] offer distance calculation from protein sequences using analytical [27, 36, 47] and maximum likelihood (ML) [48, 49] models. In order to provide support for protein alignments, we implement Scoredist algorithm, which has achieved better accuracy than other analytical models in previous tests [36]. Scoredist computes pairwise distances according to the BLOSUM62 [50] matrix, normalizes the distances with respect to expected distance and minimum possible distance, applies a logarithmic correction, and scales distances using empirically derived coefficients. Like JC69 distances, in APPLES-2, Scoredist distance calculation is powered by Numpy [51] vectorized operations and is extremely fast.

2. APPLES implemented three weighting schemes FM[28], BE[52], and OLS[53]. Balaban et al. [26] demonstrated that FM weighing [28] given by $w_{qi} = \delta_{qi}^{-2}$ results in the best

placement accuracy among these methods. However, it did not implement balanced minimum evolution (BME) weighting [37], which has been among the most promising methods. APPLES-2 implements BME which corresponds to setting $w_{qi} = 2^{-(1+p_{qi})}$ where p_{qi} is the topological distance between q and a backbone taxa i. Note that BME weights are much more challenging to incorporate into the dynamic programming because a BME weight is not simply a function of calculated distances but is rather a function of the placement on the tree. Thus, unlike the previous weighting schemes, the BME weight changes as we examine different placements. Overcoming this hurdle required implementing a more complex dynamic programming.

3. We allow precomputation of a database (called APPLES database) that consists of a backbone alignment and tree, including centroid sequences and the leaf clustering, which can be stored and distributed. The database can be reused for the analysis of different query datasets.

4. When a backbone alignment is provided, APPLES-2 can re-estimate branch lengths of the input tree using FastTree-2 [38] under the JC69 model to match the model used for estimating distances.

4.2 Experimental Setup

4.2.1 Data sets

RNASim. We reuse the RNASim-VS simulated RNA data set from Balaban et al. [26] which creates subsets from a simulated RNASim data set [39]. It comprises five replicates of data sets with size (n): 500, 1000, 5000, 10,000, 50,000, and 100,000, and 1 replicate with size 200,000. Smaller datasets are subsets of species on the larger ones chosen randomly. Each replicate contains the alignment of a single gene and the true tree. We perform a leave-one-out test on this dataset with randomly chosen 200 query sequences for each replicate. We also adopted the RNASim-QS dataset from Balaban et al. [26] which is itself based on the RNASim data set [39]; this data set comprises five replicates with varying numbers of queries, ranging from 1 to 49,152 with backbones of size $n = 500$. In both RNASim data sets, backbone tree topology is estimated from true MSA using RAxML [54] for alignments with less than 5000 sequences and FastTree-2 [38] for larger alignments according to GTRGAMMA model. In all cases, branch length is re-estimated using FastTree-2 [38] to be consistent minimum evolution units.

Web of Life (WoL). Zhu et al. [40] built a species tree of 10,575 prokaryotic genomes from 381 marker genes using ASTRAL [55]. We first determine a set of marker genes according to several selection strategies which will be discussed later. We remove sites that contain gaps in 95% or more of the sequences in the protein MSA using TrimAl [56]. The triming is only to speed up analyses and has no positive impact on accuracy; in fact, it very slightly *decreases* accuracy (see Fig. S2). Three data sets of size (n) 1000, 3000, and 9000 with 10 replicates are created by randomly subsampling the protein MSA of the selected marker genes. For each replicate, a disjoint set of 1000 sequences from the same dataset are randomly selected as query sequences. Query (protein) sequences are re-aligned to the backbone MSA using UPP [57]. Finally, backbone and query MSA of selected genes are concatenated and backtranslated from protein alignment to nucleotide. Then, we create two concatenated alignments; one with all three codon positions and one with third codon position removed. Unless it is stated otherwise, we use the nucleotide MSA with first and second codon positions only.

In all cases, we use the ASTRAL tree available from the original publication with query taxa removed from it as the backbone tree. However, we recompute its branch lengths using FastTree-2 [38] in the minimum evolution branch length unit.

We determine a marker gene set by controlling for two parameters; the number of genes (k)

and a selection strategy. Two selection strategies are *random* (among all 381) and *best*, which means top k marker genes with the lowest quartet distance [58] to the species tree are selected. Unless stated otherwise, we choose $k = 50$ coupled with the *best* strategy (which results in lowest, median, and highest quartet-distance to be $0.058, 0.125, 0.17$ respectively). Concatenated MSA using the default marker gene set contains $71,798$ sites, compared to $482,334$ sites when all 381 marker genes are concatenated and trimmed.

Simulated genome assemblies and scaffolds. Our protocol for generating simulated genome sequencing data begins with randomly selecting 200 reference genomes from the WoL dataset (10 genomes are randomly selected from each of 20 genome bins of equal genome count with the bins determined by ascending terminal branch length). Next, we run InSilicoSeq [59] v1.5.1 (using NovaSeq settings) to simulate 3 M 150 bp paired-end reads per genome. For assembly, first we run PEAR [60] v0.9.11 to merged read pairs, then run SPAdes [61] v3.14.1 with a k-mer size cascade of 21,33,55,77,99 to assemble them into scaffolds. We then run Prodigal [62] v2.6.3 to identify open reading frames (ORFs) from the scaffolds, and finally run PhyloPhlAn [63] commit 2c0e61a to identify the same 381 marker genes.

All the genes were then independently aligned to the backbone marker genes using UPP [57], and markers from the same assembly or scaffold were concatenated. We then try to place the samples on the tree using either 1) the assembly (i.e., which can be fragmented and can include errors, compared with the genome from which it is simulated), or 2) individual scaffolds (small portions of the genome). We only include scaffolds that are $\geq$ 10 kbp in our analyses. Note that here, instead of testing on microbial communities, we use an *in-silico* approach and simply generate reads from individual microbial genomes and assemble them separately. We leave it to future work to simulate mixed metagenomic reads and evaluate accuracy under such scenarios.

4.2.2 Methods compared

For RNASim-VS dataset, in addition to APPLES, we compare APPLES-2 to two ML methods, pplacer [29] and EPA-ng [64]. We run pplacer (v1.1.alpha19-0-g807f6f3) and EPA-ng (v0.3.2) in their default mode using GTR+Γ model and use their best hit (ML placement). We also compare with a simple method referred to as CLOSEST that places the query as the sister to the species with the minimum distance to it. We compare to EPA-ng in analyses that concerned scalability (e.g., RNASim-QS). In multigene data sets (WoL), we compare APPLES-2 to APPLES only, since the size of the backbone tree and length of MSA does not permit ML based methods to run on the largest WoL data set.

4.2.3 Evaluation metric

In simulations, we use the known true tree as the gold standard whereas on the empirical data, we use the ASTRAL tree on the full set of species as the gold standard. We measure the accuracy of a placement using the number of edges between the position on the gold-standard tree and the inferred placement (i.e., node distance [65]). In the simulation datasets, because we place on the estimated tree and not the true tree, we need to slightly update the metric of the error: we use *delta error*, which measures the increase in the number of false-negative bipartitions after placement compared to before placement [12].

Availability

The APPLES-2 is publicly available in open-source from `github.com/balabanmetin/apples`.

Acknowledgments.

We thank Daniel McDonald for his valuable feedback. This work was supported by the National Science Foundation (NSF) grant IIS-1565862 to S.M, NSF grant NSF-1815485 to M.B. and

S.M, 2020 UCSD Center for Microbiome Innovation Grand Challenge award to M.B., and an Arizona State University start-up grant awarded to Q.Z. Computations were performed on the San Diego Supercomputer Center (SDSC) through XSEDE allocations, which is supported by the NSF grant ACI-1053575.

References

[1] S. Janssen, D. McDonald, A. Gonzalez, J. A. Navas-Molina, L. Jiang, Z. Z. Xu, K. Winker, D. M. Kado, E. Orwoll, M. Manary, et al., Phylogenetic Placement of Exact Amplicon Sequences Improves Associations with Clinical Information, *mSystems* **3**, 18 (2018), ISSN 2379-5077.

[2] F. A. Matsen, Phylogenetics and the human microbiome, *Systematic Biology* **64**, e26 (2014), ISSN 1063-5157.

[3] F. A. Matsen and S. N. Evans, Edge Principal Components and Squash Clustering: Using the Special Structure of Phylogenetic Placement Data for Sample Comparison, *PLoS ONE* **8** (2013), ISSN 19326203.

[4] N. N.-p. Nguyen, S. Mirarab, B. B. Liu, M. Pop, and T. Warnow, TIPP: Taxonomic Identification and Phylogenetic Profiling, *Bioinformatics* **30**, 3548 (2014), ISSN 1367-4803.

[5] L. R. Thompson, J. G. Sanders, D. McDonald, A. Amir, J. Ladau, K. J. Locey, R. J. Prill, A. Tripathi, S. M. Gibbons, G. Ackermann, et al., A communal catalogue reveals Earth's multiscale microbial diversity, *Nature* **551**, 457–463 (2017), ISSN 0028-0836.

[6] F. Asnicar, A. M. Thomas, F. Beghini, C. Mengoni, S. Manara, P. Manghi, Q. Zhu, M. Bolzan, F. Cumbo, U. May, et al., Precise phylogenetic analysis of microbial isolates and genomes from metagenomes using PhyloPhlAn 3.0, *Nature Communications* **11**, 2500 (2020), ISSN 2041-1723.

[7] A. E. Darling, G. Jospin, E. Lowe, F. A. Matsen, H. M. Bik, and J. A. Eisen, PhyloSift: phylogenetic analysis of genomes and metagenomes, *PeerJ* **2**, e243 (2014), ISSN 2167-8359.

[8] P. Libin, E. V. Eynden, F. Incardona, A. Nowé, A. Bezenchek, A. Sönnerborg, A. M. Vandamme, K. Theys, and G. Baele, PhyloGeoTool: Interactively exploring large phylogenies in an epidemiological context, *Bioinformatics* (2017), ISSN 14602059.

[9] Y. Turakhia, B. Thornlow, A. S. Hinrichs, N. De Maio, L. Gozashti, R. Lanfear, D. Haussler, and R. Corbett-Detig, Ultrafast Sample Placement on Existing Trees (UShER) Empowers Real-Time Phylogenetics for the SARS-CoV-2 Pandemic, *bioRxiv* 2020.09.26.314971 (2020).

[10] P. Barbera, A. M. Kozlov, L. Czech, B. Morel, D. Darriba, T. Flouri, and A. Stamatakis, EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences, *Systematic Biology* **68**, 365 (2019), ISSN 1063-5157.

[11] F. A. Matsen, R. B. Kodner, and E. V. Armbrust, pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree., *BMC bioinformatics* **11**, 538 (2010), ISSN 1471-2105.

[12] S. Mirarab, N. Nguyen, and T. Warnow, SEPP: SATé-Enabled Phylogenetic Placement., *Pacific Symposium On Biocomputing* 247–258 (2012), ISSN 17935091.

[13] Q. Zheng, C. Bartow-McKenney, J. S. Meisel, and E. A. Grice, HmmUFOtu: An HMM and phylogenetic placement based ultra-fast taxonomic assignment and OTU picking tool for microbiome amplicon sequencing studies, *Genome Biology* **19**, 82 (2018), ISSN 1474-760X.

[14] M. Stark, S. A. Berger, A. Stamatakis, and C. von Mering, MLTreeMap–accurate Maximum Likelihood placement of environmental DNA sequences into taxonomic and functional reference phylogenies., *BMC genomics* **11**, 461 (2010), ISSN 1471-2164.

[15] M. Rabiee and S. Mirarab, INSTRAL: Discordance-aware Phylogenetic Placement using Quartet Scores, *bioRxiv* **432906** (2018).

[16] D. G. Brown and J. Truszkowski, LSHPlace: fast phylogenetic placement using locality-

sensitive hashing., *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing* 310–319 (2013), ISSN 2335-6936.

[17] Y. Jiang, M. Balaban, Q. Zhu, and S. Mirarab, Depp: Deep learning enables extending species trees using single genes, *bioRxiv* (2021), arXiv:https://www.biorxiv.org/content/early/2021/01/24/2021.01.22.427808.full.pdf.

[18] B. Linard, K. Swenson, and F. Pardi, Rapid alignment-free phylogenetic identification of metagenomic sequences, *Bioinformatics* **35**, 3303 (2019), ISSN 1367-4803.

[19] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen, Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB, *Appl. Environ. Microbiol.* **72**, 5069 (2006), ISSN 0099-2240.

[20] C. Quast, E. Pruesse, P. Yilmaz, J. Gerken, T. Schweer, P. Yarza, J. Peplies, and F. O. Glöckner, The SILVA ribosomal RNA gene database project: improved data processing and web-based tools, *Nucleic acids research* gks1219 (2012), ISSN 0305-1048.

[21] Q. Zhu, U. Mai, W. Pfeiffer, S. Janssen, F. Asnicar, J. G. Sanders, P. Belda-Ferre, G. A. Al-Ghalith, E. Kopylova, D. McDonald, et al., Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea, *Nature Communications* **10**, 5477 (2019), ISSN 2041-1723.

[22] D. H. Parks, M. Chuvochina, P.-A. Chaumeil, C. Rinke, A. J. Mussig, and P. Hugenholtz, A complete domain-to-species taxonomy for Bacteria and Archaea, *Nature Biotechnology* **38**, 1079 (2020), ISSN 1087-0156.

[23] D. McDonald, A. Birmingham, and R. Knight, Context and the human microbiome, *Microbiome* **3**, 52 (2015), ISSN 2049-2618.

[24] E. Pasolli, F. Asnicar, S. Manara, M. Zolfo, N. Karcher, F. Armanini, F. Beghini, P. Manghi, A. Tett, P. Ghensi, et al., Extensive Unexplored Human Microbiome Diversity Revealed by Over 150,000 Genomes from Metagenomes Spanning Age, Geography, and Lifestyle, *Cell* (2019), ISSN 10974172.

[25] S. Nayfach, Z. J. Shi, R. Seshadri, K. S. Pollard, and N. C. Kyrpides, New insights from uncultivated genomes of the global human gut microbiome, *Nature* **568**, 505 (2019), ISSN 14764687.

[26] M. Balaban, S. Sarmashghi, and S. Mirarab, APPLES: Scalable Distance-Based Phylogenetic Placement with or without Alignments, *Systematic Biology* **69**, 566 (2020), ISSN 1076836X.

[27] T. H. Jukes and C. R. Cantor, in *In Mammalian protein metabolism, Vol. III (1969), pp. 21-132* (1969), vol. III, 21–132.

[28] W. M. Fitch and E. Margoliash, Construction of Phylogenetic Trees, *Science* **155**, 279 (1967), ISSN 0036-8075.

[29] F. A. Matsen, R. B. Kodner, and E. V. Armbrust, pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree., *BMC bioinformatics* (2010), ISSN 14712105.

[30] J. Felsenstein, Inferring phytogenies, *Sunderland* (2003).

[31] R. Desper and O. Gascuel, Fast and accurate phylogeny minimum-evolution principle, *J Comput Biol* **9**, 687 (2002), ISSN 1066-5277.

[32] J. Whitfield, Mathematics of Evolution and Phylogeny. * Edited by Olivier Gascuel, *Briefings in Bioinformatics* (2008), ISSN 1467-5463.

[33] P. Erdos, M. Steel, L. Szekely, and T. Warnow, A few logs suffice to build (almost) all trees: Part II, *Theoretical Computer Science* **221**, 77 (1999), ISSN 03043975.

[34] D. H. Huson, S. M. Nettles, and T. J. Warnow, Disk-covering, a fast-converging method for phylogenetic tree reconstruction., *Journal of computational biology* **6**, 369 (1999), ISSN 1066-5277.

[35] D. H. Huson, L. Vawter, and T. J. Warnow, Solving large scale phylogenetic problems using DCM2., *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology* 118–129 (1999), ISSN 15530833.

[36] E. L. Sonnhammer and V. Hollich, Scoredist: A simple and robust protein sequence distance estimator, *BMC Bioinformatics* **6**, 1 (2005), ISSN 14712105.

[37] R. Desper and O. Gascuel, Theoretical Foundation of the Balanced Minimum Evolution Method of Phylogenetic Inference and Its Relationship to Weighted Least-Squares Tree Fitting, *Molecular Biology and Evolution* **21**, 587 (2004), ISSN 07374038.

[38] M. N. Price, P. S. Dehal, and A. P. Arkin, FastTree-2 – Approximately Maximum-Likelihood Trees for Large Alignments, *PLoS ONE* **5**, e9490 (2010), ISSN 1932-6203.

[39] S. Guo, L.-S. Wang, and J. Kim, Large-scale simulation of RNA macroevolution by an energy-dependent fitness model (2009).

[40] Q. Zhu, U. Mai, W. Pfeiffer, S. Janssen, F. Asnicar, J. G. Sanders, P. Belda-Ferre, G. A. Al-Ghalith, E. Kopylova, D. McDonald, et al., Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea, *Nature Communications* **10**, 5477 (2019), ISSN 2041-1723.

[41] M. J. Phillips, F. Delsuc, and D. Penny, Genome-scale phylogeny and the detection of systematic biases, *Molecular Biology and Evolution* (2004), ISSN 07374038.

[42] O. Jeffroy, H. Brinkmann, F. Delsuc, and H. Philippe, Phylogenomics: the beginning of incongruence?, *Trends in Genetics* **22**, 225 (2006), ISSN 01689525.

[43] M. Balaban, N. Moshiri, U. Mai, and S. Mirarab, TreeCluster: clustering biological sequences using phylogenetic trees, *bioRxiv* 591388 (2019).

[44] V. Lefort, R. Desper, and O. Gascuel, FastME 2.0: A comprehensive, accurate, and fast distance-based phylogeny inference program, *Molecular Biology and Evolution* **32**, 2798 (2015), ISSN 15371719.

[45] P. Rice, L. Longden, and A. Bleasby, EMBOSS: The European Molecular Biology Open Software Suite (2000).

[46] D. D. Womble, GCG: The Wisconsin Package of sequence analysis programs., *Methods in molecular biology (Clifton, N.J.)* (2000), ISSN 10643745.

[47] M. Kimura, *The Neutral Theory of Molecular Evolution* (Cambridge University Press, 1983), ISBN 9780521231091, `https://www.cambridge.org/core/product/identifier/9780511623486/type/book`.

[48] S. Whelan and N. Goldman, A General Empirical Model of Protein Evolution Derived from Multiple Protein Families Using a Maximum-Likelihood Approach, *Molecular Biology and Evolution* **18**, 691 (2001), ISSN 1537-1719.

[49] S. Q. Le and O. Gascuel, An improved general amino acid replacement matrix, *Molecular Biology and Evolution* (2008), ISSN 07374038.

[50] S. Henikoff and J. G. Henikoff, Amino acid substitution matrices from protein blocks, *Proceedings of the National Academy of Sciences of the United States of America* (1992), ISSN 00278424.

[51] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al., Array programming with NumPy (2020).

[52] W. A. Beyer, M. L. Stein, T. F. Smith, and S. M. Ulam, A molecular sequence metric and evolutionary trees, *Mathematical Biosciences* (1974), ISSN 00255564.

[53] L. L. Cavalli-Sforza and A. W. Edwards, Phylogenetic analysis. Models and estimation procedures., *American journal of human genetics* **19**, 233 (1967), ISSN 0002-9297.

[54] A. Stamatakis, RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies, *Bioinformatics* **30**, 1312 (2014), ISSN 1460-2059.

[55] C. Zhang, M. Rabiee, E. Sayyari, and S. Mirarab, ASTRAL-III: polynomial time species

tree reconstruction from partially resolved gene trees, *BMC Bioinformatics* **19**, 153 (2018), ISSN 1471-2105.

[56] S. Capella-Gutiérrez, J. M. Silla-Martínez, and T. Gabaldón, trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses, *Bioinformatics* **25**, 1972 (2009), ISSN 1367-4803.

[57] N.-P. N.-p. D. Nguyen, S. Mirarab, K. Kumar, and T. Warnow, Ultra-large alignments using phylogeny-aware profiles, *Genome Biology* **16**, 124 (2015), ISSN 1465-6906.

[58] A. Sand, M. Holt, J. Johansen, R. Fagerberg, G. Brodal, C. Pedersen, and T. Mailund, Algorithms for Computing the Triplet and Quartet Distances for Binary and General Trees, *Biology* **2**, 1189 (2013), ISSN 2079-7737.

[59] H. Gourlé, O. Karlsson-Lindsjö, J. Hayer, and E. Bongcam-Rudloff, Simulating Illumina metagenomic data with InSilicoSeq, *Bioinformatics* (2019), ISSN 14602059.

[60] J. Zhang, K. Kobert, T. Flouri, and A. Stamatakis, PEAR: A fast and accurate Illumina Paired-End reAd mergeR, *Bioinformatics* (2014), ISSN 13674803.

[61] A. Bankevich, S. Nurk, D. Antipov, A. A. Gurevich, M. Dvorkin, A. S. Kulikov, V. M. Lesin, S. I. Nikolenko, S. Pham, A. D. Prjibelski, et al., SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing, *Journal of Computational Biology* (2012), ISSN 10665277.

[62] D. Hyatt, G. L. Chen, P. F. LoCascio, M. L. Land, F. W. Larimer, and L. J. Hauser, Prodigal: Prokaryotic gene recognition and translation initiation site identification, *BMC Bioinformatics* (2010), ISSN 14712105.

[63] N. Segata, D. Börnigen, X. C. Morgan, and C. Huttenhower, PhyloPhlAn is a new method for improved phylogenetic and taxonomic placement of microbes, *Nature Communications* (2013), ISSN 20411723.

[64] P. Barbera, A. M. Kozlov, L. Czech, B. Morel, D. Darriba, T. Flouri, and A. Stamatakis, EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences, *Systematic Biology* **68**, 365 (2019), ISSN 1063-5157.

[65] B. Linard, N. Romashchenko, F. Pardi, and E. Rivals, PEWO: a collection of workflows to benchmark phylogenetic placement, *Bioinformatics* (2020), ISSN 1367-4803.

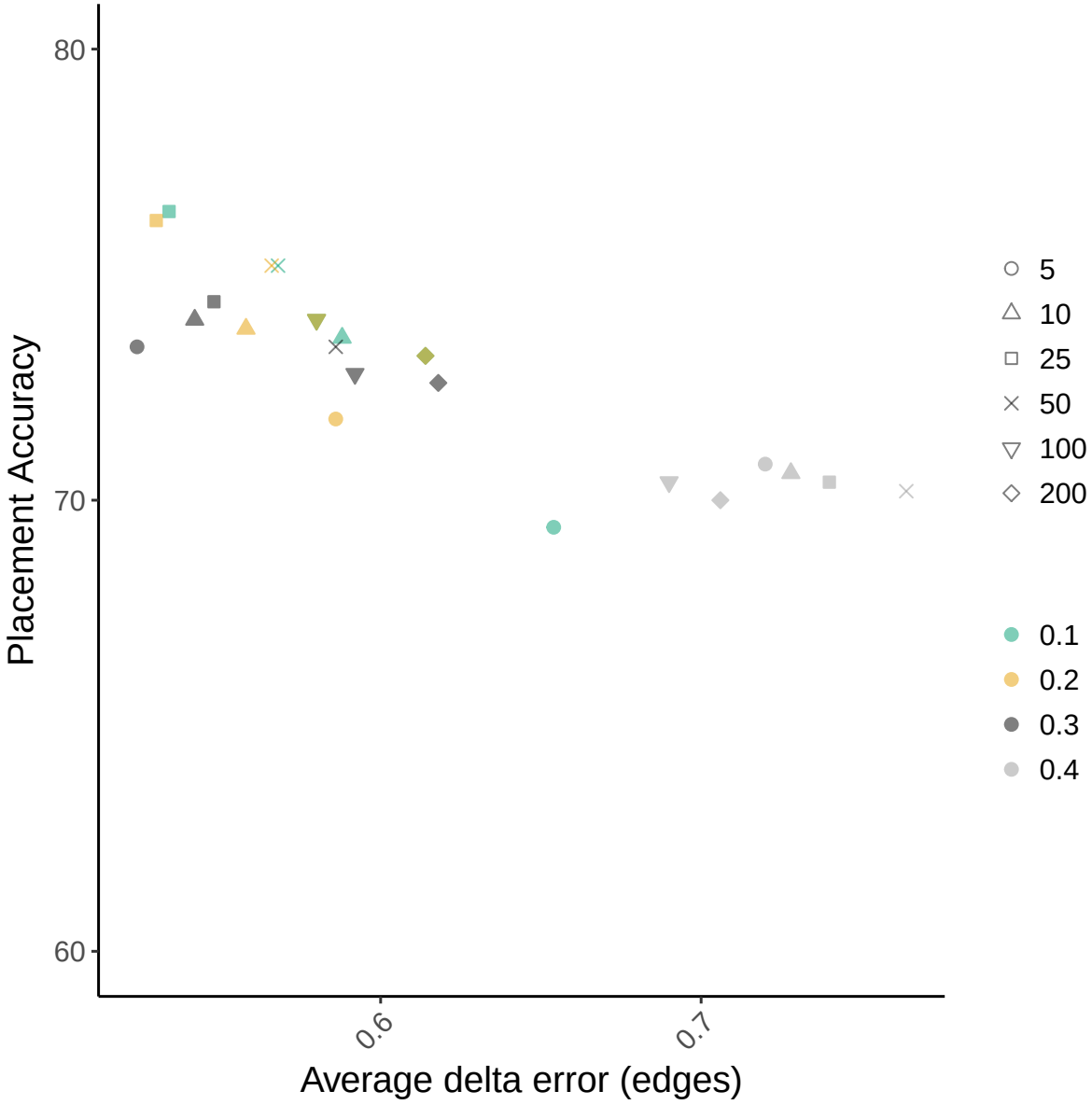

Figure S1: **Exploring the best choice of d_f and b parameters.** Results are based on 500 placements on WoL dataset using APPLES-2. Shape indicates d_f parameter and color indicates b parameter.

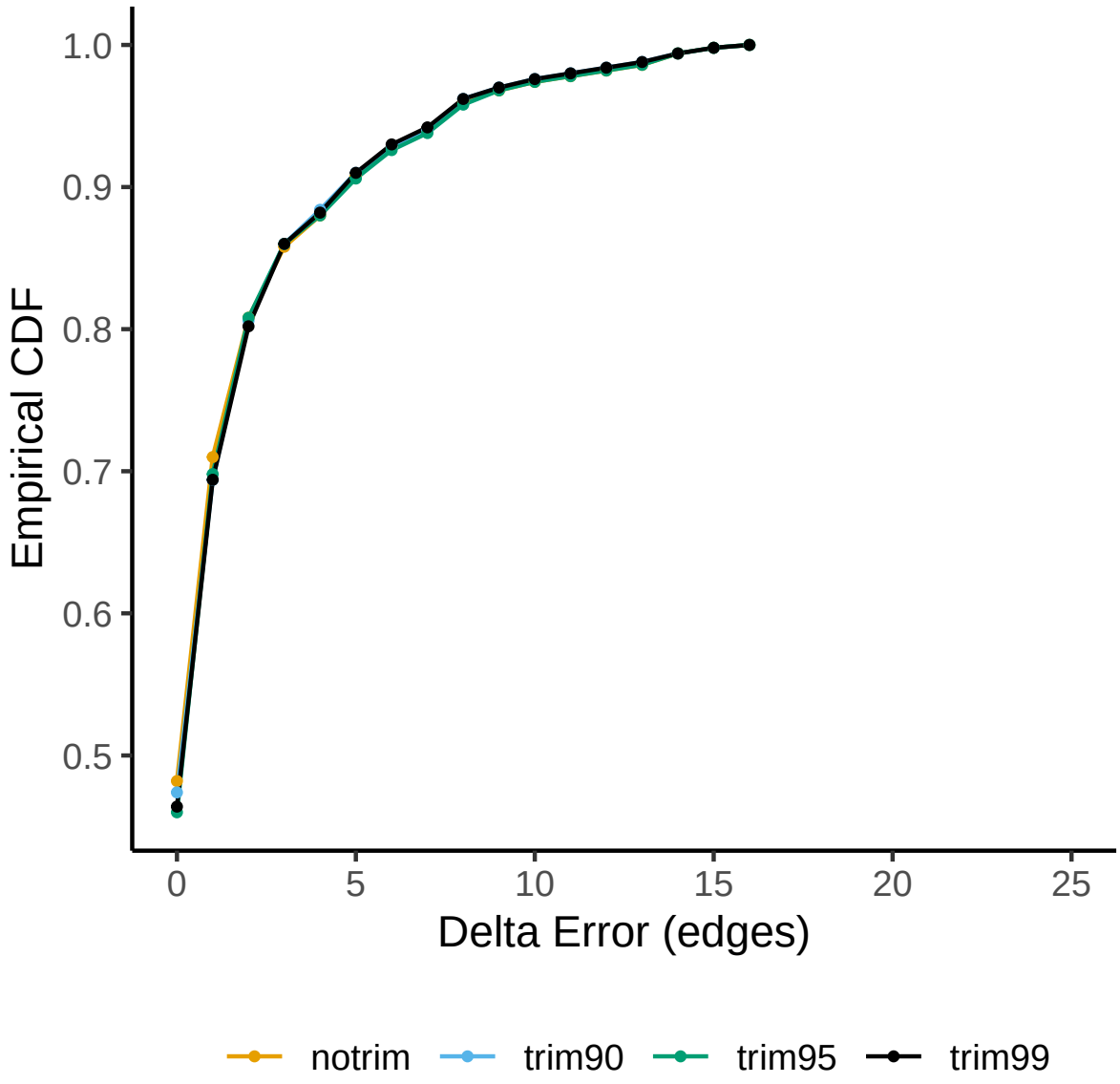

Figure S2: **Impact of alignment trimming.** Results are based on 500 placements on WoL backbone tree using APPLES. Trimming sites with 90, 95, or 99 percent gaps or more does not dramatically impact placement accuracy compared to using untrimmed alignment. Trimming reduces accuracy only slightly. For improved speed and memory usage, we therefore trim the sites with 95% or more gaps.

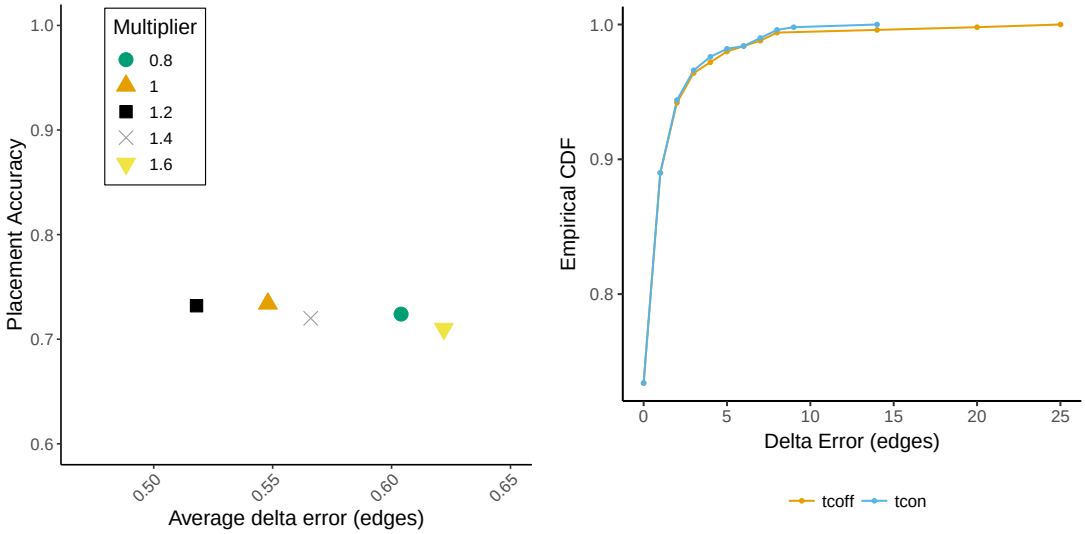

Figure S3: **Clustering the backbone using TreeCluster.** a) Exploring the multiplier for TreeCluster maximum diameter threshold. Results are based on 500 placements on WoL species tree using APPLES-2. Placement accuracy does not change as this threshold changes and the average error increases slightly if other values are used. b) Using TreeCluster to divide the backbone tree into smaller subsets (shown as *tcon* in the figure) not only improves the running time but also slightly improves the placement accuracy compared to using APPLES-2 on the full backbone tree (shown as *tcoff* in the figure) .

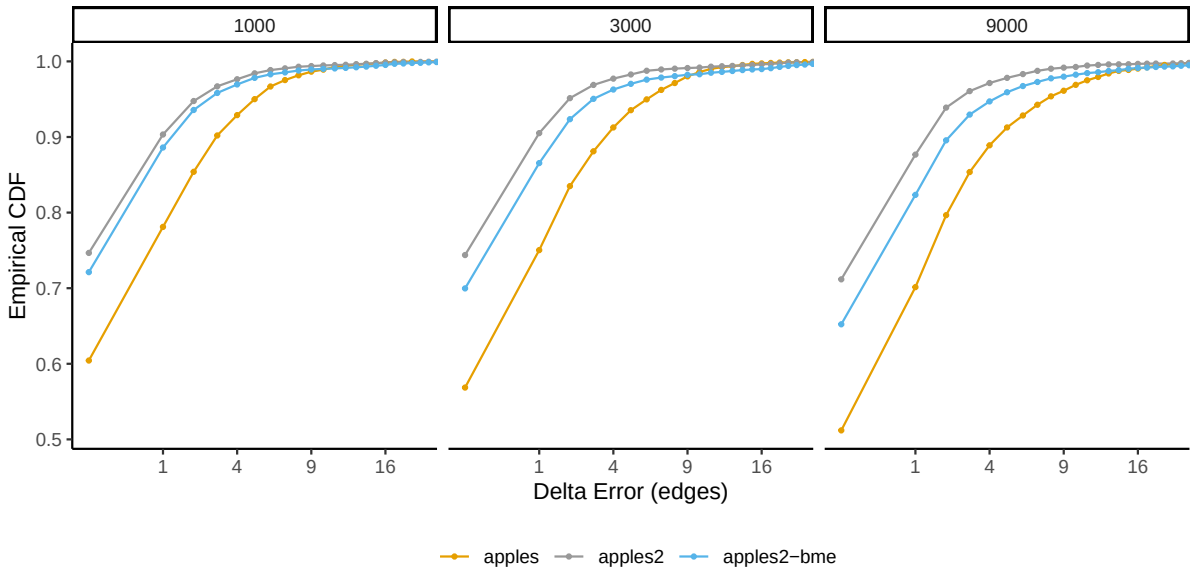

Figure S4: **Balanced minimum evolution weighting accuracy.** We compare BME weighting to FM, which is the default weighting scheme in APPLES and APPLES-2.

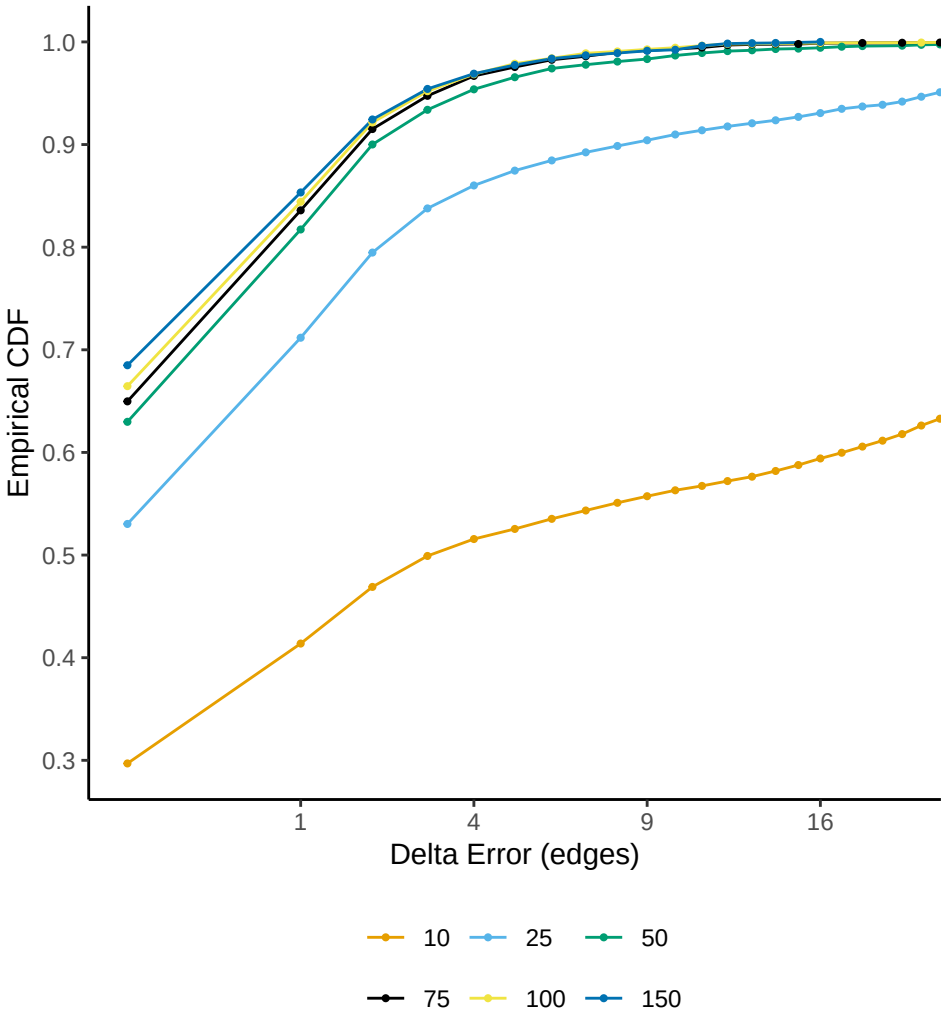

Figure S5: **Placement accuracy vs size of the marker gene set coupled with random selection strategy.** As the number of genes increases, accuracy increases consistently. However, after 50 genes, there are diminishing returns in accuracy as more genes are added.

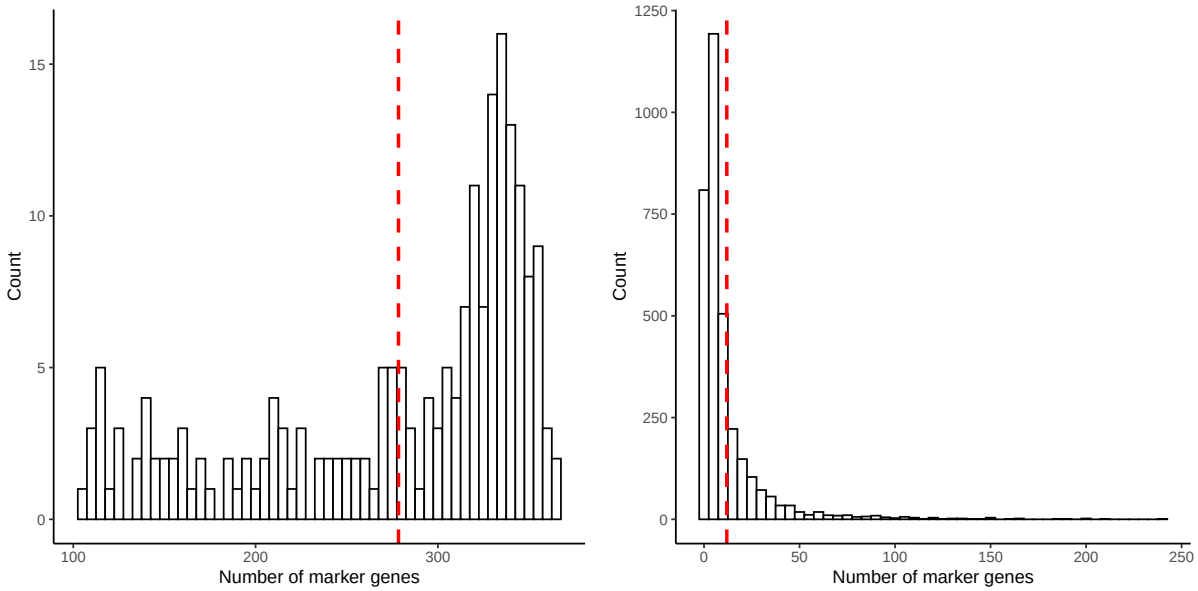

Figure S6: **Distribution of the number of marker genes in assemblies and scaffolds.** Red dashed line shows the mean. Left: assemblies. Right: scaffolds. Note that most scaffolds have relatively few genes.

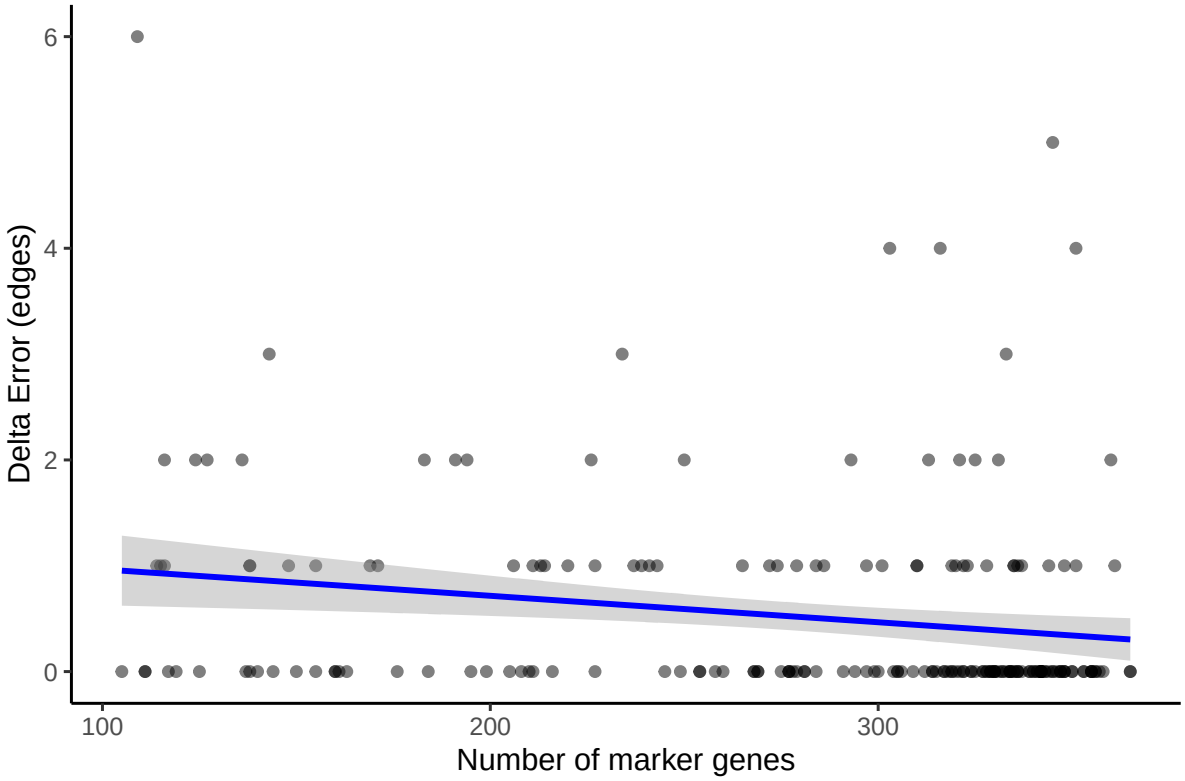

Figure S7: **Number of marker genes vs placement error for assemblies.** There is a weak but statistically significant correlation between the number of genes and error for placing assemblies.

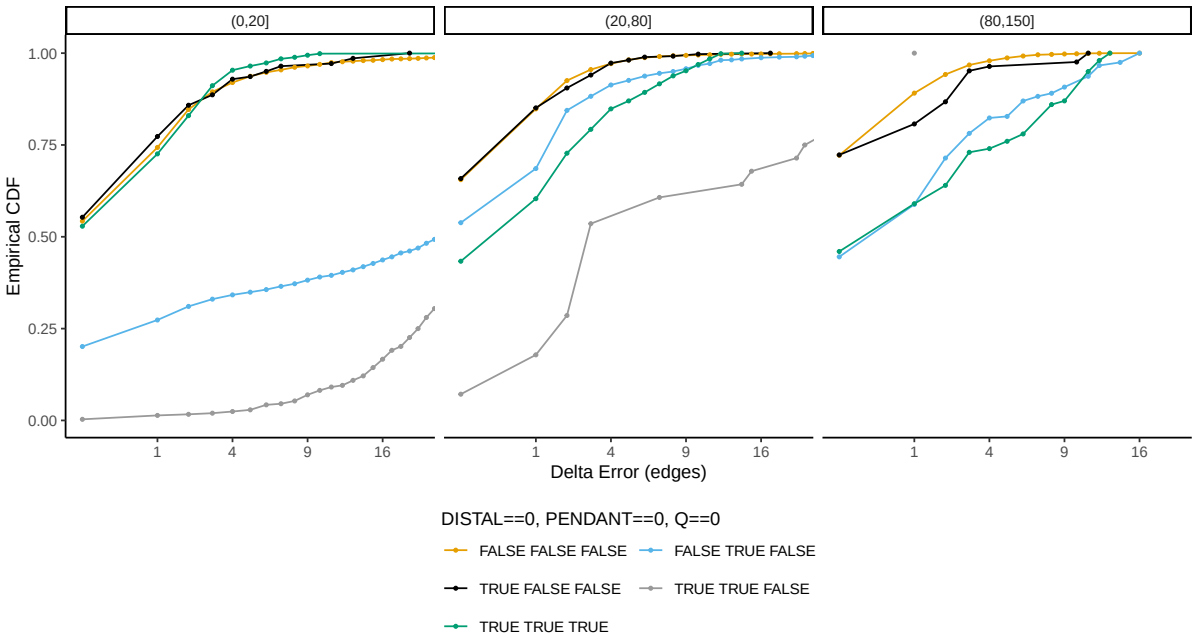

Figure S8: **Relationship between zero length pendant and distal placement edges and number of genes in the query.** Color shows all combinations of boolean values whether distal or pendant edge length is zero or MLSE error (Q) is zero. Panels show the various ranges of number of genes present in the query.