Published in final edited form as:
Nat Biotechnol. 2024 May ; 42(5): 768–777. doi:10.1038/s41587-023-01868-8.

Generation of accurate, expandable phylogenomic trees with uDANCE

Metin Balaban[1], **Yueyu Jiang**[2], **Qiyun Zhu**[3,4], **Daniel McDonald**[5], **Rob Knight**[5,6,7,8], **Siavash Mirarab**[2,6,8,*]

[1]Bioinformatics and Systems Biology Graduate Program, UC San Diego, CA, USA

[2]Department of Electrical and Computer Engineering, UC San Diego, CA, USA

[3]Biodesign Center for Fundamental and Applied Microbiomics, Arizona State University, Tempe, AZ, USA

[4]School of Life Sciences, Arizona State University, Tempe, AZ, USA

[5]Department of Pediatrics, University of California San Diego, La Jolla, CA, USA

[6]Department of Computer Science and Engineering, Jacobs School of Engineering, University of California San Diego, La Jolla, CA, USA

[7]Department of Bioengineering, University of California San Diego, La Jolla, CA, USA

[8]Center for Microbiome Innovation, Jacobs School of Engineering, University of California San Diego, La Jolla, CA, USA

Abstract

Phylogenetic trees provide a framework for organizing evolutionary histories across the tree of life and aid downstream comparative analyses such as metagenomic identification. Methods that rely on single marker genes such as 16S rRNA have produced trees of limited accuracy with hundreds of thousands of organisms, whereas methods that use genome-wide data are not scalable to large numbers of genomes. We introduce uDance, a method that enables updatable genome-wide inference using a divide-and-conquer strategy that refines different parts of the tree independently and can build off of existing trees, with high accuracy and scalability. With uDance, we infer a species tree of roughly 200,000 genomes using 387 marker genes, totaling 42.5 billion amino acid residues.

The set of available genomes is expanding quickly. Cataloging the constantly growing set of genomes is challenging, and phylogenetic trees provide a framework for organizing the

[*] smirarab@ucsd.edu .

Author Contributions Statement
S.M and M.B conceived and designed the uDance method. M.B, Y.J, and S.M performed simulation studies. All authors contributed to building and analyses of the real biological dataset. All authors contributed to the writing of the paper. All authors reviewed and edited the manuscript.

Code Availability
The code is publicly available at https://github.com/balabanmetin/uDance [25] under BSD 3-Clause license.

Competing Interests Statement
The authors declare no competing interests.

data. Besides providing a hypothesis on the evolutionary history, reference phylogenies are increasingly used in microbiome profiling [1, 2, 3]. Several widely-used databases, such as 16S rRNA [4, 5], based on single marker genes consist of millions of sequences. However, trees inferred from genome-wide data (e.g., The Web of Life [6] and GTDB [7]) reflect the evolutionary history of microbes more accurately than those marker-based methods. Some of these genome-wide analyses use discordance-aware methods [8] that are robust in the face of gene tree heterogeneity [9, 10, 11] due to prevasive horizontal gene transfer (HGT) [12, 13], incomplete lineage sorting (ILS), and other causes [14]. However, due to methodological limits, existing genome-wide trees have several orders of magnitude fewer taxa than single marker trees. Currently, there is a trade-off between two paradigms: a large number of genes and better accuracy on one side (genome-wide trees) and a large number of organisms represented on the other (single-marker trees). One solution, adopted by both GTDB [7] and PhyloPhlAn-3 [15], is to use many genes but select a tiny fraction of sites from each gene (Table 1); this approach clearly misses out on the full benefits of genome-wide data.

We aim to break the trade-off by creating genome-wide trees that are nearly as densely sampled as marker gene trees. The main obstacle is increasingly not a lack of data but a lack of scalable algorithms. For example, using current methods, it is not possible to extend the Web of Life tree [6] (inferred from ≈ 10,000 species and 381 genes) to include hundreds of thousands of leaves; it took 100,000 CPU hours to infer this tree using RAxML [16] and ASTRAL [17], and some tools used in this analyses do not scale beyond tens of thousands of genomes. Moreover, existing approaches focus on *de novo* inference and are not designed for updating existing trees; they often require repeating almost all of the inference without including the possible benefit that would come from previous analyses. Phylogenetic placement methods [18] do allow new sequences and genomes to be inserted [19] onto a tree in a scalable fashion [20, 21]. However, these do not infer relationships between query sequences and do not allow updates to the “backbone” tree on which new sequences are placed.

To achieve scalable genome-wide phylogenetic inference with the ability to be updated continuously, we turn to the divide-and-conquer paradigm. A rich algorithmic history (reviewed by Warnow [22]) has established that by inferring smaller subtrees that are then merged, we can extend the scalability of the existing methods *and also* improve the accuracy of analyses (e.g. [23, 24]). Here, we provide a framework, uDance (Updating trees using Divide ANd ConquEr) [25], that is suitable for both *de novo* reconstruction and incrementally updating existing phylogenies using genome-wide or single-gene data. Unlike phylogenetic placement methods, uDance obtains a fully resolved tree with updated resolutions for the backbone. It uses a divide-and-conquer algorithm to refine different parts of the tree independently (Fig 1), which enables it to analyze ultra-large datasets (e.g., hundreds of thousands of genomes and hundreds of genes) in reasonable times and incrementally grow and update trees.

Using uDance, we built a 199,330-genome microbial tree of life from 387 marker genes, resulting in one order of magnitude improvement over the tree from the Web of Life in terms of the number of genomes (Table 1). Compared to the GTDB tree with roughly 65,000 genomes, this tree has three times more genomes *and* is inferred from 121× more characters (42.5 billion versus 350 million). Using simulations, we show that uDance can infer whole-genome phylogenies with 10,000 taxa more accurately than alternative methods while remaining computationally tractable. Moreover, uDance is designed to be easily used on large multi-node computing clusters.

uDance overview

The input to uDance is a backbone tree, a set of multiple sequence alignments (MSAs) of backbone sequences, and new (query) sequences (Fig. 1a). Alternatively, when a backbone tree is not available, uDance can select a set of backbone species with high diversity to reconstruct a backbone tree (see Methods). At a high level, uDance inserts the query sequences on the backbone tree independently and then refines the tree locally. The backbone is allowed to change based on the new information provided by the query sequences, but uDance also outputs an “incremental” tree with the backbone relationships fixed for users that require consistency between updates in their analysis. Since uDance aims for automatic analyses of large data, it has many built-in quality control and filtering strategies; it may decide a set of query sequences cannot be confidently put in the output tree (i.e., are unplacable) and some backbones need to be removed. When more sequences become available, the output from the previous iteration can be used as the input in the next iteration to incrementally grow the tree.

The uDance algorithm has five steps (Methods). First, it places query sequences onto the backbone tree (Fig. 1b) based on gene MSAs using the scalable APPLES-2 algorithm [26] that can insert a query genome, using 381 marker genes, into the Web of Life tree in under 16 seconds [26], making it feasible to place all available genomes. Then, it divides the resulting placement tree into multiple partitions using a clustering algorithm that controls cluster size and phylogenetic diversity. For every pair of adjacent partitions, the algorithm selects representative sequences from each partition considering both genome completeness and closeness to the other partition; it adds these representatives to the other partition as *outgroups*. Next, uDance infers a maximum likelihood (ML) gene tree (i.e., RAxML-NG [16]) per gene on the set of backbone, query, and outgroup sequences in each partition and then proceeds to infer a species tree per partition by summarizing the gene trees using ASTRAL [17]. Importantly, the species tree is constrained [27] to be compatible with the backbone topology for the outgroup species but (optionally and by default) allows the rest of the tree to change. These constraints create nearly-compatible subtrees and enable uDance to stitch them back into a single species tree at the end; in the process, some backbone or query sequences (those mixing with outgroup sequences) may become unplaceable, which are then removed from the final tree and added as queries for future updates. The gene tree and species tree inferences are computational bottlenecks and can be performed in a distributed fashion over multiple machines on a high-performance computing (HPC) cluster.

Accuracy of uDance in Simulations

To evaluate uDance, we simulated a 10,000-taxon 500-gene dataset (Methods) inspired by the Web of Life [6] with various degrees of gene tree discordance with the species tree representing horizontal gene transfer (HGT) events as well as incomplete lineage sorting

(ILS) (Table 2). We simulated ten model conditions by adjusting model parameters. We ran uDance in the *de novo* mode, where it chose and reconstructed a 1000-species backbone and then updated the backbone with the remaining 9000 species. We compared the uDance results with three alternative *de novo* phylogenomic approaches that can handle this size: gene tree inference with FastTree-2 [28] followed by species tree inference using (1) ASTRAL-MP [29] (FT2+ASTRAL in short) and (2) ASTRID [30] (FT2+ASTRID in short); (3) concatenation (concat) of all input MSAs into a single MSA and inferring a single tree using FastTree-2. Slower ML methods, such as RAxML-NG, could not be run on our 10,000 gene trees (across all replicates) each with 10,000 taxa or concatenations of 100–500 genes.

On simulated data, uDance exhibits low error and is more accurate than other methods (Fig. 2a). While increased gene tree discordance results in higher species tree error for all methods, uDance remains accurate in most cases, achieving the lowest mean normalized Robinson-Foulds (nRF) distances between the estimated and true species tree in all model conditions except HD-P5, where *estimated* gene trees are on average 78% different from the species tree. In model conditions with 100 genes, the mean nRF for uDance was never larger than 0.075 for all but the last dataset. FT2+ASTRID was the only method that completed on all datasets besides uDance within the resource limits of 48 hours and 125 GB memory but was the least accurate with a mean nRF larger than 0.09. Concat completed on all datasets with 100 genes and failed in eleven out of 20 replicates with 500 genes. Error decreased with larger number of genes for all methods. With 500 genes, nRF decreased from 0.047 to 0.02 on mid discordance (MD-500), and from 0.06 to 0.028 on high discordance (HD-500) for uDance. FT2+ASTRAL fails to return a tree for all or some replicates in all model conditions except MD-100 (we skipped running FT2+ASTRAL in LD-100 dataset due to its high computational demand).

When measured using the quartet distance (QD) metric, uDance remains the most accurate method in all model conditions except MD-100, where FT2+ASTRID, uDance, and FT2+ASTRAL exhibit similar levels of accuracy. In the first eight model conditions, uDance has QD error below 2%. On the HD-P4 condition with very high (70%) true discordance, uDance still has good accuracy (0.044 QD); only in the HD-P5 condition the error increases to 34%. By using divide-and-conquer, uDance computes ML gene trees using RAxML-NG with superior accuracy compared to FastTree-2 run on the full data (Extended Fig. 1).

Distributed computing allows uDance to take advantage of HPC clusters. It inferred the backbone and updated it on the HD-500 dataset (10,000 species and 500 genes) in 14 hours wall-time hours using 1140 CPU-hours on a 48-node cluster using up to 672 cores (Fig. 2bc). Here, 40% of the wall-clock time is for backbone inference, which would be saved when a backbone tree is available. With 100 genes, uDance used 236 CPU-hours, which is five times less than FT2+ASTRAL (Fig. 2c). More than 90% of the uDance CPU time is spent on the distributed RAxML-NG gene tree inference jobs; in contrast, for FT2+ASTRAL, the single ASTRAL-MP job consumes more than 99% of the total compute. Concatenation takes less CPU time (Fig. 2c) but more wall-clock time than uDance because FastTree-2 is single node and poorly parallelized (Fig. 2b). In terms of memory usage, uDance and FT2+ASTRID are more scalable, and both methods use less than 30G of memory.

Since uDance operates by dividing the full data set into smaller partitions, we tested the impact of the choice of parameters to ensure they do not impact results dramatically (Supplementary Table S1). By reducing backbone sizes, we can dramatically reduce the total running time (e.g., reducing from 1000 to 250 reduced the running time by 60%) at a small but noticeable cost to accuracy (from 6.1% to 7.3%). Using the true species tree as the backbone marginally decreased the error to 5.7%. Moreover, uDance was largely robust to the partition size in terms of accuracy but smaller partitions reduced the running time (e.g., going from 500 to 50 increased error from 6.1% to 6.8% and reduced the backbone update wall-time from 161 to 27 hours). Based on these results, in addition to the default method where we set the partition size as a sub-linear function of the number of genomes, we also designed a *fast* mode where partition sizes are fixed to 100 regardless of the input size. Testing the fast mode on a set of 5 – 1280 query genomes, we observed that the fast mode reduced the running by a factor of 3X to 8X (depending on the number of queries) while having minimal impact on accuracy (Extended Figure 2). Moreover, the accuracy tended to *improve* after additions with the update mode of uDance whereas it remained the same or degraded with the incremental model (when the backbone is fixed) or simple placement.

To test uDance on a growing tree, we simulated a replicate with 64,000 taxa and 100 genes similarly to the 10,000-taxon dataset. We started with 250 randomly selected genomes and doubled the tree size successively with randomly selected new genomes. As the tree size grew, the nRF error of all methods decreased, and uDance had lower nRF errors than FT2+ASTRAL and concat (Fig. 2d). In terms of QD, uDance started with a much lower error than other methods; as the tree grew, uDance maintained its low QD while other methods showed a downward trend. uDance had the lowest QD among all methods except on 32,000 leaves tree where it matched concat. FT2+ASTRAL did not complete for instances with ≥4,000 species within 48 hours wall-time and 120GB memory limit. Concat scaled better but had the nRF lowest accuracy and failed to finish within 48 hours wall-time on the largest dataset. When distributed over approximately 600 CPU cores, uDance was faster than both alternatives in wall-time though it used more CPU-time than concat (Fig. 2e). Finally, for memory, uDance scaled better with the increasing tree size compared to alternatives.

We also tested uDance in the presence of model violations by creating a simulation (HD-HET) under a codon model with parameters varying across the tree and obtained from the biological dataset (see Methods). In the single replicate that we were able to simulate, the species tree inferred by uDance had a substantially lower error (nRF .049; QD 0.002) than FT2+ASTRID (nRF 0.11; QD 0.016) and concat (nRF .086; QD 0.032), while FT2+ASTRAL failed to finish in time. While the lack of replicates makes it hard to make general statements, the better accuracy of uDance is consistent with the fact that it estimates a different GTR model for each subtree, allowing more variations; nevertheless, note that uDance subtrees do not necessarily coincide with changes in model parameters.

Phylogenomic Reconstruction of 200,000 Microbial Genomes

We obtained 656,574 Archaea and Bacteria genomes assemblies from NCBI available as of May 14, 2020. After several postprocessing steps such as duplicate and contamination removal (see Methods), we curated a 296,745-genome data set with MSAs for the 381 genes used in WoL tree [6] and seven additional core ribosomal genes for a total of 656,907 AA sites (Table 1). Using an automated procedure (see Supplementary Notes 3), we selected 6,056 genomes from key groups with low taxon sampling in the WoL tree and used uDance to update the WoL tree to include a total of 15,953 genomes (called the 16k) (Fig. 3b). We ran a second iteration of uDance with 280,792 remaining query genomes and the 15,953 backbone tree to obtain a tree with 199,330 genomes (called the 200k) (Fig. 3c). The remaining 97,415 genomes were removed due to being near-duplicates (90,693), unplaceable (4,174), or potentially chimeric, contaminated, or drastically incomplete genomes (2,548) (see Methods). Taxonomic decoration [31] of the 200k tree against GTDB [32] identified 140, 360, and 1060 phyla, classes, and orders, which were 72%, 107%, and 172% higher than the original WoL tree (called the 10k tree) (Extended Figure 3). Judging by NCBI labels, Bacterial and Archaeal representation increased 18 and 6-fold in the 200k versus the 10k tree (Fig. 3a). Groups with lower sampling (e.g. DPANN) had dramatic increases (e.g., 50-fold).

Comparison between growing microbial trees

All trees (10k, 16k, and 200k) demonstrated clear separation between Archaea and Bacteria domains (Fig. 3bc) and recapitulated main findings of Zhu et al. [6]. Candidate Phyla Radiation (CPR, also Patescibacteria in GTDB taxonomy) continued to form a monophyletic group at the base of Bacteria. Also, the length of the branch separating Archaea and Bacteria (A-B branch) remained relatively short (0.14, 0.27, and 0.18 in the 10k 16k, and 200k trees respectively). Phylum-level resolutions were similar across trees with some exceptions noted below (Fig. 3bc). The 200k tree had a quartet distance of 0.003 to the 16k tree, itself at distance 0.014 to the 10k tree, which was far lower than their discordance with concatenation-based trees released by GTDB (e.g., 0.084 comparing 16k to GTDB). Nevertheless, trees were locally different: 992 (9.8%) branches of the 10k tree changed in the 16k tree, and 1870 (11.7%) branches of the 16k tree changed in the 200k tree. Most conflicting branches were short (Figs. 4d and Extended Figure 4) and were not uniformly distributed across the tree (Figs. 4cf and Extended Figure 3). When induced to each of the largest phyla and super-phyla defined by NCBI, some groups (esp. DPANN, Asgard, Parcubacteria, Elusimicrobia, and Diaforarchaea) had moderately high quartet dissimilarity (0.043 to 0.065) between 16k and 200k trees while remaining 23 taxonomic groups were stable (with dissimilarity below 0.02). Moreover, topological changes were (Fig. 4c and Extended Figure 4) overabundant among deep branches (e.g., ≈10% of the changing branches had a depth between 0.35 and 1 compared to 1% of all branches), but the nine most basal branches remained unchanged in all trees.

The most striking changes were near the base of Bacteria. The base of non-CPR Bacteria was a clade including phyla Dictyglomota, Thermotogota, Caldisericota, and Coprothermobacterota in 10k and 16k trees (with 41 and 53 genomes, respectively). In the 200k tree, these phyla partially broke up and moved away from their basal position to unite with other phyla (e.g., Synergistota and Fusobacteria), potentially due to their better taxon sampling (more than 230 genomes in total). Instead, the large super-phyla Terrabacteria that contains 76,639 genomes (including Cyanobacteria, Actinobacterioda, Firmicutes) forms the most basal non-CPR bacterial clade. Our tree, in contrast to some new analyses [33], did not put CPR as part of Terrabacteria. In the 200k tree, unlike 10k and 16k trees, Fusobacteria and DST (Deinoccoccota, Synergistota, Thermotogota) split from CPR and Terrabacteria are placed between Terrabactera and what has been called Gracilicutes, a position that is similar to other recent analyses [33].

Comparisons to Published Taxonomies

The taxonomic decoration of successive trees using the GTDB taxonomy revealed many similarities and some differences (Fig. 3bc). For example, for taxa labeled Firmicutes by GTDB to be monophyletic in the 200k tree, 19 genomes assigned placeholder names (DTU030, DUMJ01, and UBP7) should be classified as Firmicutes. Similarly, reassigning a handful of genomes with placeholder names makes Proteobacteria (with >75,000 genomes) monophyletic, and changing the classification of only four out of 1,122 genomes named Planctomycetota by GTDB would make Planctomycetota a monophyletic phylum. The genomes decorated as seven Desulfobacterota groups (e.g., Desulfobacterota B) were monophyletic in the 10k tree but paraphyletic in the 16k (7 clades) and 200k tree (5 clades); nevertheless, all Desulfobacterota remain *close* to each other. Similar local paraphyly of GTDB names was observed for Gracilibacteria, Paceibacteria, Armatimonadota, Acidimicrobiia, Hydrogenedentota, Planctomycetota, Aenigmatarchaeota, Nanoarchaeota, Methanobacteriota, and Thermoplasmatota. Besides these local disagreements, two GTDB names, Myxococcota and Bipolaricaulota, were broken to far-apart sub clades in our trees. Bipolaricaulota broke into two groups, one near Chloroflexota and another basal to a large group of non-Terrabacteria. Myxococcota is paraphyletic (Myxococcota and Myxococcota A) in the published GTDB tree (Extended Figure 5) and may represents a truly paraphyletic group.

The consistency of uDance trees with major super-phyla of the NCBI taxonomy changed with successive updates (Fig. 3d). All uDance trees showed high consistency for several groups, including Terrabacteria, FCB, TACK, Microgenomates, and Parcubacteria. In some cases, increased taxon sampling reduced consistency with taxonomy (e.g., archaeal groups DPANN and Asgard), perhaps because of increased chances for non-monophyly with more genomes. Interestingly, some groups (e.g., PVC and FCB) became more consistent with every iteration. Yet others, such as Euryarchaeota, remained strongly paraphyletic in all trees. Overall, the GTDB tree had slightly higher levels of discordance with NCBI groups than the 200k tree (Fig. 3d), and it had similar levels of topological dissimilarity to all uDance trees (Extended Figure 3).

Gene tree conflicts, branch support, and diversity across the tree

We observed high levels of discordance between gene trees and the species tree (Fig. 4a) and wide variations across genes. Breaking genes into the ten largest functional categories of genes [34] revealed that DNA and RNA binding genes, which are involved in the core machinery of genetic information processing, had the highest similarity to the inferred species tree, with more than 0.80 quartet similarity on average (Fig. 4a). On the other hand, genes involved in peripheral functions such as transmembrane transport were most topologically distant to the species tree, as expected under HGT.

Examining partitions across the tree, we observed high heterogeneity. Some partitions (e.g., partitions 2 entirely made of Firmicutes and 14 mostly made of PVC and Spirochaetota) had elevated levels of discordance, while others (e.g., partition 6 made of Bacteroidota) had lowered discordance. uDance produces species and partition trees with branch support (local posterior probability) thanks to ASTRAL [35]. Partitions also manifest differences in terms of branch support, as evident from decreased support for partitions closer to the Archaea/Bacteria split in the 16k tree (Extended Figure 6A). For the 200k tree, support correlated with sequence diversity, which changes across partitions (Extended Figure 6B). The average branch length in a partition can be as low as 0.0002 (partition 15, which contains genomes from the Pseudomonadaceae family) or as high as 0.073 (partition 42, which spans several phyla including Desulfobacterota, Myxococcota, and Bdellovibrionota). Branch supports correlate significantly with the diversity of partitions (Pearson $r = 0.28$) and clearly reduce for often-abnormally-large partitions with mean branch length below 0.001 (Extended Figure 6B). As expected, going from 16k to 200k taxa reduces the proportion of highly supported branches as the alignment size is fixed (Fig. 4e); however, note that that 200k tree still has far more branches with high support (e.g., around 84000 branches have ≥ 95% support; see Extended Figure 6C). Models of evolution also vary across partitions. For example, clear differences were observed in across-site rate variation parameters across the tree (Extended Figure 7); for several genes, rates shifted substantially for specific partitions but not others. This observation was not specific to the microbial data set. Using uDance, we reanalysed a data set of 1,124 plant transcriptomes and 410 single-copy genes [36]. The best-fit model changes across partitions created by uDance and correlated with the partitions rather than genes (Fig. 4h). Gene tree discordance also varied not only across genes but also across the tree partitions (Fig. 4g).

Discussion

We showed that uDance, a divide-and-conquer approach for incrementally growing ultra-large whole-genome phylogenies, is more accurate than *de novo* species tree reconstruction via concatenation and the coalescent-based phylogenomic approach using ASTRAL. Since uDance is massively distributed and memory efficient, it is more scalable than concatenation, which did not manage to analyze some of our datasets given time and memory limits. While in practice we may be able to use weeks of analyses instead of the 48-hour limit imposed in our simulations, it remains the case that for limited amounts of time, uDance can handle larger datasets. In our simulations, which generally had high levels of HGT, uDance was also more accurate than the alternatives, including concatenation. However, it should be noted that with lower levels of HGT, concatenation may have a relative advantage. Also, we used concatenation with no partitioning and with a model that allowed only heterogeneity of rates across sites. Using partitioning or models that allow more forms of heterogeneity (the latter of which can also be incorporated in uDance) could improve its accuracy, especially for deep phylogenies.

Beyond improved accuracy, the main advantage of uDance is the continuous tree update process it enables. By creating a dataset with hundreds of thousands of unique genomes inferred from many hundreds of genes, we demonstrated it is possible to infer very large trees by leveraging calculations that have already happened in the past. We also showed that in updating trees, it is important to allow the backbone tree to change. This means that the use of the divide-and-conquer strategy requires care. When updating the tree in subsequent steps, if the rooting of the tree is kept fixed, the same branches may be selected as boundaries between subsets, keeping some branches stale. To avoid this scenario, we recommend changing the rooting of the tree from one iteration to another, as we did in our experiments. As the number of genomes and phylogenies computed increases, methodologies like uDance that natively support gradually growing trees will become increasingly important in practice. Finally, while uDance currently uses traditional phylogenetic inference methods in each step, recent advances in using deep learning [37, 38] for updating trees can be easily incorporated to open the path for a new paradigm where species trees are extended using data from single genes.

Our 200k tree is the largest microbial multilocus tree, considering the total number of species and genes. Since the accuracy of downstream microbiome analyses often depends on the density of the reference [39], trees like this have the potential to improve downstream microbiome analyses. The current tree includes only a tiny fraction of the estimated 10^{12} microbial species [40], and the ability to update it in future is crucial. Reassuringly, for the common species, going from 10k to 16k and then 200k trees showed relative stability at high level, captured by the quartet score, and only local changes to specific taxon resolution in successive trees. All trees were mostly consistent with GTDB, with most differences boiling down to renaming placeholder names in GTDB (e.g., DTU030). In a few cases, the broken monophyly of a large named group seemed explainable by misassignment of that name to a small number of genomes. Given the much better sampling of sites across the genome in our trees than GTDB (656,907 sites used for our trees, versus 41,084 sites in GTDB), it seems likely that these genomes were misclassified by GTDB or the taxon decoration step. Finally, the divide-and-conquer approach enabled us to study the heterogeneity of inferences across the tree and revealed that the models of evolution change across the tree, an important factor that is missed by the traditional single analyses encompassing the entire tree.

Methods

Workflow of uDance

The input is a backbone tree and MSAs of backbone and query sequences per gene. The uDance algorithm has several steps (Fig. 1) detailed below. Most steps include quality control strategies, which we present at the end. All constant values mentioned below are defaults and adjustable by the user (Supplementary Table S3).

Step 1. Placement—We concatenate all input MSAs, remove duplicate query sequences, and place the queries on the backbone tree using the scalable placement tool APPLES-2 [26] under the Jukes-Cantor model [42] for DNA and Scoredist [43] for amino acid (AA). Since APPLES-2 places each query independently, the relationship between the queries is not resolved in the resulting placement tree.

Step 2. Decomposition—To refine the placement tree, uDance uses divide-and-conquer. It divides the placement tree into several independent partitions which are processed independently and in parallel. Let the rooted placement tree from the previous step be denoted by $T = (V, E)$, representing vertices V (each with degree one or three, except the root), edges E with a length and a weight, and leaf set $\mathscr{L} \subset V$. The branch length of an edge $e = (u, v)$ is denoted by b_e. We define $P(e)$ to be the set of query sequences placed on an edge. The weight w_e of the edge e is $|P(e)|$ if e is an internal node and $|P(e)| + 1$ if the edge is terminal (leaf). We define a clustering of the edges of the tree T as a coloring of edges such that the set of edges with the same color are a connected (unrooted) subtree. We use $C(e)$ to denote the color of and edge $e = (u, v)$, writing $C(u)$ as a shorthand for $C(u, v)$, and use $Q(u)$ to denote the total number of placements and leaves that are descendant of u and share the same coloring.

Our decomposition algorithm is parameterized by a (soft) minimum threshold α of diversity (measured in phylogenetic total branch length in substitution units) and a (soft) limit S on the size of each cluster. The goal is to keep the size of each partition (including placements) below S *and* the diversity of each set (not including placements) above α. Since achieving both goals is impossible, we prioritize α, compromising on S when needed. By default, we set $S = \min(2500, 100\left\lceil\frac{8\sqrt{n} + 3\sqrt{m}}{100}\right\rceil)$ where n and m are the numbers of backbone and query sequences, respectively. Users can adjust S but we recommend keeping $S \leq 6,000$ to avoid a slow refinement step.

Algorithm (Supplementary Notes 2) assigns colors using a bottom-up traversal of the tree. For each node u, we mark it as a junction of neighboring colors if $Q(l) + w_l + Q(r) + w_r + w_u$ exceeds the desired partition size limit S but we make two exceptions to this rule: (1) either b_u, b_l, or b_r is shorter than 10^{-5} (signifying an unresolved branch); or (2) declaring the node as a junction creates a partition with a diameter (maximum pairwise distance between nodes in the partition) less than the threshold α. The set of edges with the same color defines a partition consisting of leaves incident to those edges and all placements on them. A partition may have more than S members for groups (e.g., species) with high taxon sampling. To avoid very large partitions that dominate running time to differentiate near identical genomes, we use a second threshold τ (6000 by default). If the number of all sequences in the query exceeds τ, we remove near duplicate query sequences using the following iterative graph thresholding algorithm.

For each abnormally large partition with m query sequences, we create an $m \times m$ adjacency matrix G where $G(x, y)$ gives the fraction of genes in the input dataset for which the gene sequences of x and y are identical. Starting from $\beta = 0.99$, we create a graph G_β where nodes x and y are connected if $G(x, y) \geq \beta$. We gradually decrease β until the number of connected components in G_β is less than τ. Finally, we choose the sequence with the highest gene occupancy from each connected component and remove other query sequences. We also output the information needed to add the removed genomes back as a polytomy at the end as the sister to their connected component.

Step 3. Augmenting partitions with representatives from neighbors—For an internal node u in a binary rooted tree, we denote left and right children of u with $l(u)$ and $r(u)$. We denote descendent leaves of u with the maximum gene occupancy and the minimum tree distance to u with $G(u)$ and $H(u)$, respectively. Let $\mathscr{L}(i)$ and $\mathscr{E}(i)$ be the set of backbone leaves and edges in a partition i; by construction, $|\mathscr{L}(i)| \geq 2, \forall i$. Partitions, which are disjoint in edges and leaves, are adjacent to each other at "junction" nodes. We define a scheme to make partitions overlap by adding to each partition some leaves from other partitions, which we call outgroup sequences. More formally, let V be a node, and let three edges adjacent to v be denoted with (u_j, v), $j \in \{1, 2, 3\}$ and be colored c_j; i.e., $(u_j, v) \in \mathscr{E}(c_j)$.

Note that v is a junction node if and only if $c_1 \neq c_2$ or $c_1 \neq c_3$. Without loss of generality, there are three categories of color arrangements around V: either (1) $c_1 \neq c_2 \neq c_3$, (2) $c_1 = c_3 \neq c_2$, or (3) $c_1 \neq c_2 = c_3$. In all cases, we first reroot T at node u_1. In the case (1) and (2), we select $H(l(u_2))$, $H(r(u_2))$, and, if not identical to the previous two, $G(u_2)$ (Extended Figure 8A). Note that the selected nodes are not always colored c_2 as there can be more partitions below u_2. These nodes, called $O_{c_2}(i)$, are added to the partition. We expand the list of selected outgroup sequences with the closest and the most occupant descendants of u_3 for case (1). However in the case (3), we select $H(u_2)$, $H(u_3)$ (thus, one from each child of v), and $G(v)$. Overall, based on the scenario, we add two to six outgroups to c_1 at the junction v. We repeat this procedure for c_2 and c_3 and for all junction nodes. Let $R(c) = \cup_{o \neq c} O_c(o)$ be nodes added *through* c to other partitions. We let $O(c) = \cup_o O_o(c)$ be the set of all outgroup sequences present in c.

Claim 1.: $R(c) \subseteq O(c) \cup \mathscr{L}(c)$.

Proofs of this and the following claims are available in online Supplementary Notes 1.

Step 4. Partition tree refinement—uDance infers a species tree on the set of all backbone and query sequences for each partition using ASTRAL-constrained [27]. First, it performs unconstraisned ML gene tree inference with the user-specified tool (RAxML-NG by default, RAxML-8, or IQTree-2) and phylogenetic model. Users also have the option to specify the number of starting trees and perform automatic model selection using ModelTest-NG [44]. Regardless of the inference tool used, uDance computes the gene tree branch supports with IQTREE-2's approximate Bayesian (aBayes) test. [45]. It then performs constrained species tree inference using ASTRAL-constrained using two separate constraint trees, *incremental* and *updates*. The *incremental* constraint is T induced to $\mathscr{L}(i) \cup O(i)$ while *updates* is T induced to $R(i) \cup O(i)$. The incremental option enforces the ASTRAL tree to fully match the backbone tree while the updates option allows both query and backbone species to freely move but outgroup relationships are fixed. uDance has three modes for tree refinement. The first option selects ASTRAL runs with incremental constraints for all partitions, resulting in a uDance tree that fully matches the backbone tree topology. The second option selects ASTRAL runs with updates-cons for all partitions. The third option (and the default mode) is *maximum-qs* approach: uDance picks the ASTRAL tree with a higher quartet score among the two ASTRAL runs for each partition. All partition species trees have branch support values (local posterior probability [35]) computed by ASTRAL.

We estimate branch lengths of the species tree in substitution unit using a new algorithm that we implemented as part of ASTRAL (v5.17.2) to transfer the median of quartet branch lengths from the gene trees to corresponding species tree branches.

This algorithm simply computes the median length of all internal branches of all quartets that are mapped exactly to a particular species tree branch across all gene trees. Note that when only a single gene is given, the same procedure is performed, skipping the ASTRAL step, and outputting the gene tree estimated using IQ-Tree with the corresponding constraints.

Step 5. Stitching—Let the refined trees for a partition c be called $A(c)$ and w be a junction node of tree T that is adjacent to colors c, o, o' where $c \neq o$ but o' may be equal to o or c. For both incremental and update constraints, we can prove that each such w corresponds to a node in $A(c)$. More formally:

Claim 2.: *When $o \neq o'$, the junction node w corresponds to a unique internal node v of the refined tree $A(c)$ such that:*

1. $S(u_2) \cap (R(c) \cup O(c)) = O_o(c)$
2. $S(u_j) \cap (R(c) \cup O(c)) \neq \emptyset, j \in 1, 3$

where $(u_j, v), j \in 1, 2, 3$ are three edges adjacent to v and $S(u_j)$ is the set of leaves on the side of u_j if we remove the edge (u_j, v).

Claim 3.: *When $o = o'$, the junction node w corresponds to a unique internal node v of the refined tree $A(c)$ such that:*

1. $(S(u_1) \cup S(u_2)) \cap (R(c) \cup O(c)) = O_o(c)$
2. $S(u_j) \cap (R(c) \cup O(c)) \neq \emptyset, j \in 2, 3$

where u_j and $S(u_j)$ are defined similarly to Claim 2.

Proofs of these claims (Supplementary Notes 1) give an algorithm for finding the junction node v. The definition of *unplaceable* nodes and follows naturally:

Definition: (Unplaceable). Let v be the node in $A(c)$ which corresponds to w in T and is adjacent to u_1, u_2, u_3 colored c, o, and o', respectively. Any leaf $x \in \mathscr{L}(c)$ placed closer to u_2 (and u_3 if $c \neq o'$) than v (i.e., $x \in S(u_2)$ and $x \in S(u_3)$ if $c \neq o'$) is mixing with outgroup o (or o') and is hence unplaceable.

Given these claims, we can design the following stitching procedure (Extended Figure 8BC). Perform a bottom-up traversal of T. On a junction node w, assume we have adjacent colors c, o, and o' such that $c \neq o$. Find the node v adjacent to u_1, u_2, u_3 corresponding to w in $A(c)$ using the procedure described in proofs of Claims 2 and 3. Note that this procedure reroots the tree at u_1 where $(u_1, v) \in \mathscr{E}(c)$. Mark nodes below u_2 (and u_3 if $o \neq o'$), including unplaceable leaves colored c, for removal from $A(c)$. We repeat this procedure for o' if it is different from o. Once all junctions are processed, we remove all marked nodes from all refined trees, denoted with $A'(c)$. The remaining refined trees $A'(c)$ and $A'(o)$ are simply merged at the mapped (v) junction nodes if they are adjacent to the same junction (w) node in T. This simple grafting is always possible:

Claim 4.: All refined trees with unplaceable nodes removed are compatible and can be grafted.

Proof. We follow the stitching procedure mentioned above. By Claims 2 and 3, we can always find junction nodes. As junctions are processed, we remove leaves belonging to other colors. Because any mixing between colors was erased by removing unplaceables, clearly, the remaining refined subtrees will have a single color and a clear mapping of junctions to the original tree T. Thus, the remaining refined trees can be simply grafted onto T at their junction nodes.

Note that while uDance can update a tree, the updates are not universal. Since uDance uses the spanning tree for stitching, topological updates are restricted to edges within partitions. We leave it to future work to extend the algorithm to allow updates to the partition spanning tree. □

Across all steps: Quality control—Several steps are used to ensure the quality of the data included.

Sequence quality.: Prior to the phylogenetic placement, to save time and memory, uDance filters out extremely gappy sites (≥95% gaps by default) in the input MSAs using TrimAl [46]. It then uses TAPER [47] to mask smaller errors in the MSAs. Before gene tree inference, uDance computes an induced MSA and removes all-gappy sites for each partition gene pair. If this MSA has fewer than 100 sites, the gene is discarded in that partition due to strong evidence that gene tree estimation from short fragmentary sequences is problematic [48]. In addition, any sequences with fewer than 75 base-pairs in each partition for each gene are filtered out from the induced MSA due to the same line of evidence. Finally, uDance infers a quick ML gene tree for each gene using FastTree-2 and performs outlier sequence detection and removal with TreeShrink [49] using its default parameters.

Backbone quality.: Errors in the backbone topology negatively affect the phylogenetic placement step because a misplaced sequence can attract queries of its kind to the wrong part of the tree. To catch these, uDance uses APPLES-2 to replace every sequence in the backbone in a leave-one-out fashion and measure the topological distance between the original and new location. When the distance is larger than $\log_2 n - 1$ where n is the number of backbone sequences, it remove these "suspect" sequences. We chose this equation because, in a balanced tree, it gives the height of the tree, which would be the maximum possible number of branches a leaf can move; for a typical tree that is neither fully balanced nor fully unbalanced, this choice gives a heuristic estimate of unreasonably far placements. uDance then re-estimates branch lengths of the backbone tree with suspect sequences pruned and inserts back suspect sequences using APPLES-2. It keeps those that no longer place no more than $\log_2 n - 1$ edges farther than the original position, which can happen because some of the suspect sequences may be error-free and unstable because of other erroneous sequences. Furthermore, we have observed that sequences with much lower occupancy than their neighbors are often misplaced. Thus, we use TreeCluster [50] to cluster the backbone tree with *max* clustering strategy (threshold 0.7). For each cluster, we perform 2-means clustering in the first step. Let C_1 and C_2 be the two clusters with centroids m_1 and $m_2, m_1 < m_2$. While $|C_1| < |C_2|$ and $1 - \frac{m_1}{m_2} \geq 0.6$, we filter out the minimum occupancy sequences from C_1 and return to step one. All filtered sequences are removed from the backbone.

Gene tree quality.: Given very poor alignments, short sequences, or low phylogenetic signal, a gene tree may fail almost entirely to resolve confidently. Before the species tree inference stage, uDance computes the median branch support for all the input gene trees and clusters them based on the median branch support using one-dimensional 2-means clustering. If the cluster with the smaller centroid has less than 20% of all the genes present in the partition and the difference between centroids of the two clusters is larger than 0.1, all genes in the cluster with the smaller centroid are discarded. uDance proceeds with contracting low support branches using the user-specified threshold. We use 0.33 and 0.66 for simulated and biological datasets respectively (note that aBayes branch supports range between 0.33 and 1).

Backbone inference—When a backbone tree is not available, uDance can infer one *de novo*. We designed a procedure called Mainlines to select a subset of n sequences in the backbone tree with high diversity. Mainlines begins with creating a concatenation MSA by selecting $\left\lceil \frac{l}{k} \right\rceil$ least gappy sites from each k gene MSA. Here, l is the number of sites in the concatenation alignment (5000 by default). Next, we use FastTree-2 to infer a tree using GTR+G (nt) or LG+G (aa) model. Mainlines uses this tree to subsample n backbone species from the entire data set. The selection is carried out by exploring the range of plausible threshold values for TreeCluster (max option) via binary search. The search stops when either a threshold that results in exactly n clusters is found or the threshold difference between two consecutive iterations is below 0.0001. After Mainlines determines the set of backbone sequences, uDance executes the phylogenomic inference pipeline, which infers ML gene trees (default using RAxML-NG) and an ASTRAL species tree. The only difference between the pipeline used during the backbone inference stage is that the ASTRAL search does not have tree constraints.

Distributed Implementation—We implement uDance using the workflow engine Snakemake [51]. uDance is flexible and largely configurable (see Table S3 for configurable options). uDance is supported in multiple operating systems (Linux, MacOS), is easy to install, and thanks to Snakemake, is easily deployable in distributed HPC platforms.

Simulations

We perform a set of simulations using SimPhy [52], starting from a default model condition (named HD-500) and deriving the other eight model conditions by adjusting simulation parameters. We simulate 10 replicates per model condition. Simulation parameters are chosen with reference to Web of Life microbial data set. The software tools and the version used in our experiments is given in Supplementary Table S4.

Default model condition (HD-500).—We simulate a 10,000 taxa species tree under the Birth-death process with fixed speciation and extinction rate of 5×10^{-7} and 4.16×10^{-7} respectively. The number of generations in the tree is fixed to 10^9. Each replicate has 500 gene trees and ILS and HGT constitute the two sources of gene tree discordance. Gene trees have 0.03 mean nRF distance to the species tree due to ILS prior to the introduction of HGT events. We set the rate of HGT events so that the average discordance across all replicates is 0.38 (nRF). We note that the amount of discordance due to HGT is much higher than ILS because we wanted to remain similar to our microbial reference dataset, WoL. The probability of a horizontal gene transfer event between two taxa is inversely proportional to their distance in the species tree. For each gene, we use INDELIBLE [53] to simulate multiple sequence alignments under GTR+G model. In each replicate, we draw two gene sequence length hyperparameters λ and ν from Uniform(5.5, 6.5) and Uniform(0.1, 0.2) respectively. The length of each gene sequence within a replicate is drawn from Lognormal (λ, ν). This parametric process results in lengths ranging between 169 and 869, and averages 406 base pairs.

We randomly delete a single consequent chunk of characters in each sequence through the following process in which the deleted range is more probable to be on tips of the sequence than the center. The ratio of the deletion length to the total length is drawn from Beta $(\rho, 1-\rho)$ where ρ is a hyperparameter drawn from Uniform(0.2,0.6). The location of the center of the deleted range is drawn from Beta(0.4,0.4) distribution where 0 and 1 represent the leftmost and rightmost eligible center locations respectively. Finally, we realign modified sequences using UPP [54].

Gene tree estimation error, measured by nRF between true gene and tree estimated with FastTree-2 under GTR+G model, depends on SimPhy sequence mutation rate besides the gene sequence length and alignment gappiness parameters. We adjust the mutation rate so that the average error is approximately 0.45. In particular, the overall mutation rate is 4×10^{-8} per generation and there are rate multipliers per gene, per species, and per gene/species, which ensure deviations from ultrametricity. With the average error set to 0.45 and true discordance around 0.38, the discordance from the estimated gene trees to the true species is around 0.58, which is similar to the levels of discordance observed on the WoL dataset.

Derived model conditions: We create another model condition named MD-500 from the previous one by (1) making two adjustments: (1) changing the distribution from which λ is drawn to Uniform(6, 7) and (2) decreasing the HGT rate so that gene tree discordance is 0.35. We generate MD-100 and HD-100 by selecting only the first 100 out of the 500 genes from MD-500 and HD-500 respectively. Finally, we derive five additional model conditions HD-P1, HD-P2, HD-P3, HD-P4, and HD-P5 by selecting the gene trees in ranges 1–100, 101–200, 201–300, 301–400, and 401–500 after all genes in HD-500 dataset are sorted low-to-high according to the nRF between their estimated gene tree and true species tree. We create another model condition named LD-100 from MD-500 by sorting its genes in increasing gene tree estimation error order and selecting the top 100 genes.

Serial dataset was simulated using the same SimPhy parameters in the first replicate of HD-500 dataset, except the number of species and genes are set to 64,000 and 100 respectively. Subsets of size 250×2^i, $i \in [0, 8]$ were formed by successively halving the number of species through the selection a random subset. uDance inferred a *de novo* backbone on subset size 250 and for other subsets, the backbone was the output of the preceding run. The backbone tree was re-rooted at the farthest node to the root in each iteration in order to add variability to the joint nodes (partition borders) in successive runs.

In order to test uDance under a heterogeneous model of evolution, we created HD-HETER dataset using a mixture of four codon models as follows. For the κ parameter of the codon models, we used the default value of 2.5 everywhere. For other parameters, we created a mixture. We first divided the WoL tree into subsets with a maximum diameter of 0.9 using TreeCluster. We chose the largest Archaea-only and CPR-only groups, and the largest two non-CPR bacterial clades. Next, we concatenated nucleotide alignments for genes p0001 – p0010 and counted each tri-nucleotide for each of the four clades. These give us four sets of codon frequencies, which we use. We also took four sets (Supplementary Table S2) of M5 codon model parameters provided by Yang *et al.* [55], who estimated the parameters for 12 proteins from various organisms. We matched each M3 model with one of the codon frequency sets, and this procedure gives us four models to choose from. To simulate sequences under these models, we used INDELIBLE, which can approximate the M5 model using the related M3 model with discrete rate heterogeneity distributions; we discretized M5 gamma distributions to 20 categories. We assigned the four codon models to the branches of the gene trees in the first replicate of HD-100 dataset using a simple algorithm: traverse the edges of the tree and start with a random model. With probability 0.05, the next edge is assigned a new model, and the next model is selected randomly among the three remaining models. Finally, we used the same alignment lengths for these trees as the full HGT dataset. Using the same procedure from HD-100, we added gaps to simulated sequences and computed a multiple sequence alignment with UPP.

uDance backbone size: Novelty of a query with respect to the backbone increases the placement error [26]. The number of backbone sequences and the backbone selection strategy affect the query novelty. To determine the relationship between the backbone size and query novelty in HD-100 dataset, we ran mainlines with two settings $N \in \{500, 1000\}$, where N is the number of subsampled sequences. Note that Mainlines uses TreeCluster-max to choose backbone sequences. We quantified query novelty via novelty score κ, which was defined as two times the terminal branch length (in the unit of the expected number of substitutions per site) of the query when placed on the true location on the backbone tree. We set a target $\kappa = 1$ for the limit for query novelty. With $N = 500$ and $N = 1000$, κ was less than one for more than 66% 97% of the query sequences (Extended Figure 9A). However, when backbone sequences were chosen randomly, only 73% satisfied $\kappa < 1$. Therefore, we proceeded with $N = 1000$.

Biological Data

Web of Life 2

Data preparation.: A total of 656,574 non-duplicate Bacterial and Archaeal genomes were retrieved from the NCBI genome database (RefSeq and GenBank [56]) on May 14, 2020 (which succeeded RefSeq release 200). Two sets of marker genes were independently inferred from the genomes. First, 400 global marker genes were inferred using PhyloPhlAn [57] v1 (commit 2c0e61a), with default parameters, on the amino acid sequences of the open reading frames (ORFs) predicted by Prodigal [58] v2.6.3, with default parameters. Of these, the 381 genes previously validated [58] were selected. Second, 37 core marker genes were inferred PhyloSift [59] v1.0.1 from the genome sequences, with default parameters. For 11 of the 37 core marker genes that were present in both the core and global marker sets, we selected the ones in the global set. Amino acid and nucleotide sequences of the marker genes were extracted using previously developed scripts. We retained only seven of 26 new core genes as we removed any gene with fewer than median sequence length 150 AAs among the curated genomes. The new curated set is a superset of all sequences in the WoL reference data set. We removed any genome with fewer than 66 marker genes (Extended Figure 9B). For each 381 marker gene, we aligned the query AA sequences onto the corresponding MSA from the WoL data set using UPP [54], masking alignment insertion sites. Since the WoL dataset did not include the 7 core ribosomal genes, we computed the backbone MSAs for those genes ourselves using UPP. We excluded marker gene p0127 from the analysis as UPP failed to finish in 48 hours. 1,412 genomes with GUNC [60] clade seperation score ≥ 0.50 and contamination portion ≥ 0.25 are suspected to be contaminated or chimeric and removed from the data set (Extended Figure 9C). After removing duplicate sequences that share identical AA sequences, the number of unique genomes in the data set reduced to 296,745.

Reconstruction of the 16k tree.: Using an automated procedure (see Supplementary Notes 3), we selected 6,056 species for insertion on the WoL tree. These sequences were chosen in a way that sought to increase taxon sampling of key groups with low sampling (see Fig. 3a). We performed two rounds of uDance (v1.1.0) to update the WoL tree with the selected sequences. We instructed uDance to use the entire global marker gene set at the phylogenetic placement stage. We ran uDance with partition size parameter 1,000, which resulted in 20 partitions. For gene tree estimation inside uDance, we opted to use RAxML-NG [16] with LG+G [61] model and three starting trees. Finally, in this uDance run, we set the low support branch contraction threshold to 0.9 and the minimum gene occupancy threshold to 30. 241 genomes were unplaceable in the output tree after the first run. The number of unplaceable sequences was less than 25 in all partitions except one partition where 187 genomes, mostly classified as members of Myxoccocota and Bdellovibrionota phyla, were dropped out. We ran a second round of uDance with partition size parameter 2,000 and re-inserted 172 of these 187 unplaceable sequences in the second round. We refer to the resulting tree with 15,956 genomes as the 16k tree.

Reconstruction of the 200k tree.: We performed one round of uDance (v1.6.0) where we updated the 16k tree with the remaining 280,792 query sequences in the data set. We did not add the queries that were unplaceable in the 16k run to the query set in this run. In order to speed up phylogenetic placement, we sorted the marker genes based on the quartet distance between their gene tree and the species tree in the WoL dataset and selected the top 68 marker genes. In an earlier study [26], we found that phylogenetic placement on the WoL dataset using 50 marker genes is nearly as accurate as using the full set. The cutoff number 68 was determined based on a trade-off between the number of Archaea-rich genes and the total number of genes in the selection (Extended Figure 9D). Unlike the 16k run, we used the 68 selected marker genes during the phylogenetic placement stage in uDance. We set the partition size to 2,500, which created 78 partitions. After near-duplicate removal in some large partitions, the total number of query and backbone sequences in all partitions combined equaled 201,316; thus, roughly 80,000 genomes that were nearly (but not fully) identical to many other genomes were dropped. We opted to use RAxML-NG with LG+G model and two starting trees inside uDance. Finally, in this uDance run, we set the low support branch contraction threshold to 0.66 and the minimum gene occupancy threshold to 30. The resulting tree, which we call the 200k tree, contains 199,330 sequences. 1,986 sequences were either unplaceable or were removed in one of the quality control steps in the workflow.

Taxonomy decoration.: In this analysis, we used NCBI (retrieved on 2020-07-01) and GTDB (release 207) taxonomy databases. Taxonomy decoration and consistency analysis are performed using tax2tree [31]. To compute the consistency of a tree with NCBI database, we first decorate the tree using NCBI taxonomy. When decorating using the NCBI database, we only performed assignments at phylum and super-phylum rank (also called clade. Examples are PVC, Parcubacteria, and DPANN). We removed any suffixes (such as _A, _B) of the names of paraphyletic ranks before the decoration with the GTDB database to prevent carrying over potentially incorrect paraphyletic groups in GTDB. In the case of phyla, only nine groups had such suffixes.

Visualization.: The trees in this study are visualized with iTOLv5 [62]. Unique colors were assigned to selected phyla and classes and according to the taxonomic decoration using the GTDB database. To display the 16k trees on a page, we collapsed the clades that represent a single phylum and with fewer than 400 leaves or that represent a single class. We matched the phyla and class-level visualization in the 200k tree to 16k tree. After collapsing, we grouped a clade of phyla or classes if each one had fewer than 20 and 40 members for the 16k and 200k trees respectively. We added numerical suffixes to the names of the paraphyletic ranks. We dropped names for the remaining clades that were assigned alpha-numeric temporary names (e.g., UBA3054) in GTDB.

1KP

In this experiment, we grew an existing plant phylogenetic tree using uDance. For the backbone tree, we used trees from the 1,000 plants (1KP) pilot study [63], which was inferred from AA characters on 94 plant species using ASTRAL. We extended the backbone tree using 410 AA alignments from the follow-up 1KP [36] study. We found the best model of evolution for each gene alignment partition using ModelTest-NG [44]. The model of evolution search space included all models with and without GAMMA rate heterogeneity but excluded invariant site models and the models that are not supported by both RAxML-NG and IQTree-2.

Methods Compared

We compare uDance with the following methods on the simulated HGT dataset.

The first method is the two-step phylogenomic approach where the first step is gene tree inference and the second step is species tree inference using ASTRAL-MP or ASTRID. We perform gene tree inference with FastTree-2 using GTR+G model of evolution. Systematic exploration of the phylogenomic approach using alternative gene tree inference tools such as IQTree-2 and RAxML-NG is not feasible on the HGT data set due to the computational cost of these tools.

The second approach is the concatenation-based species tree inference where we create a concatenation MSA (also called supermatrix) and perform phylogenetic inference with FastTree-2 using the GTR+G model. Once again, FastTree-2 is the only tool that can handle the large inputs in our dataset.

Extended Data

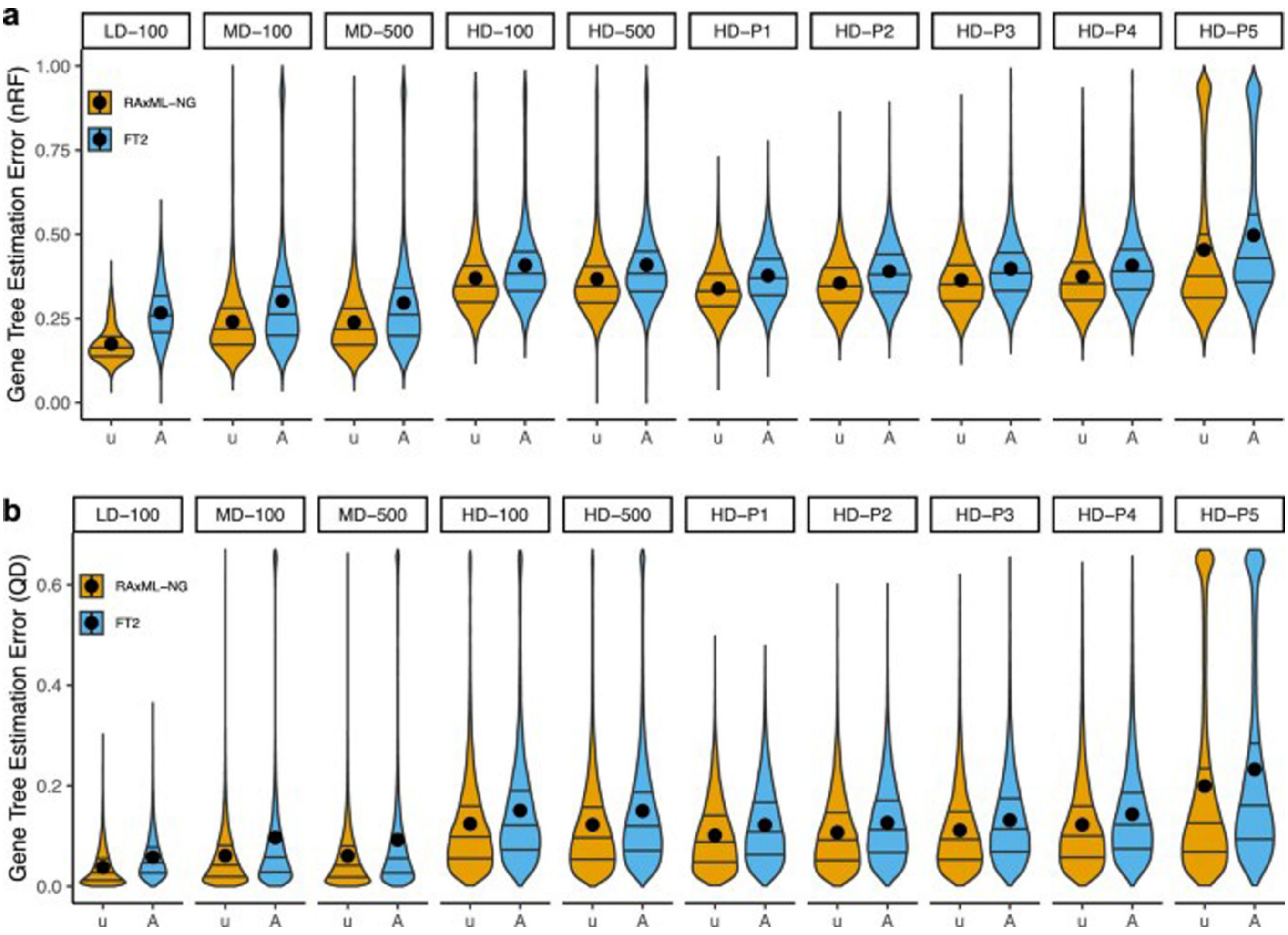


Figure Extended Figure 1:

(A) nRF and (B) QD distance between estimated and true gene trees for all partition-gene pairs in all model conditions in the simulated dataset. RAxML-ng is used inside uDance (u) and on subsets whereas FT2 (A) is computed on the full dataset. The calculation of errors is always on the subsets to obtain fair comparisons.

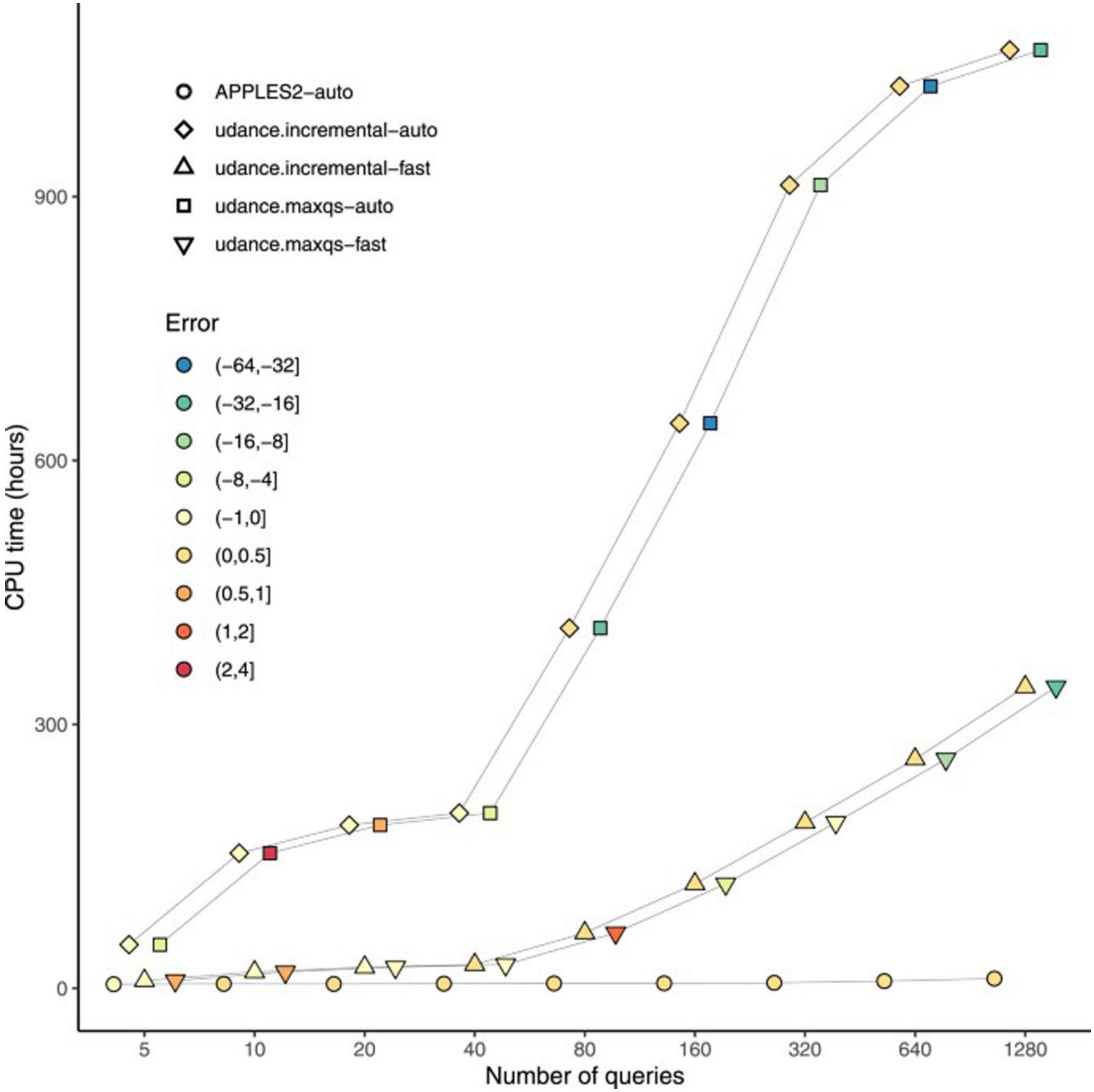


Figure Extended Figure 2:

Inserting a small number of queries on a large tree using uDance in auto and fast mode. We selected the 16,000 genomes from the analysis used in Figure 2g and updated it with $\left\{5 \times 2^i \mid 1 \leq i \leq 8, i \in \mathbb{Z}\right\}$ genomes using uDance in the auto (standard) and fast insertion mode (where the only difference is that partition sizes are set to 100). We measured the delta error for each query, which is defined as the change in RF distance between the true tree and inferred tree after placement of the query sequence. We show the mean delta error versus CPU time for various query set sizes. The running time grows slower with the −fast mode with out a significant sacrifice in accuracy. Whether the fast or the default modes is used, the accuracy is substantially higher when we allow the backbone tree to change. In fact, the accuracy *improves* after addition if the update mode is used whereas the accuracy stays the same or degrades with the incremental model or simple placement.

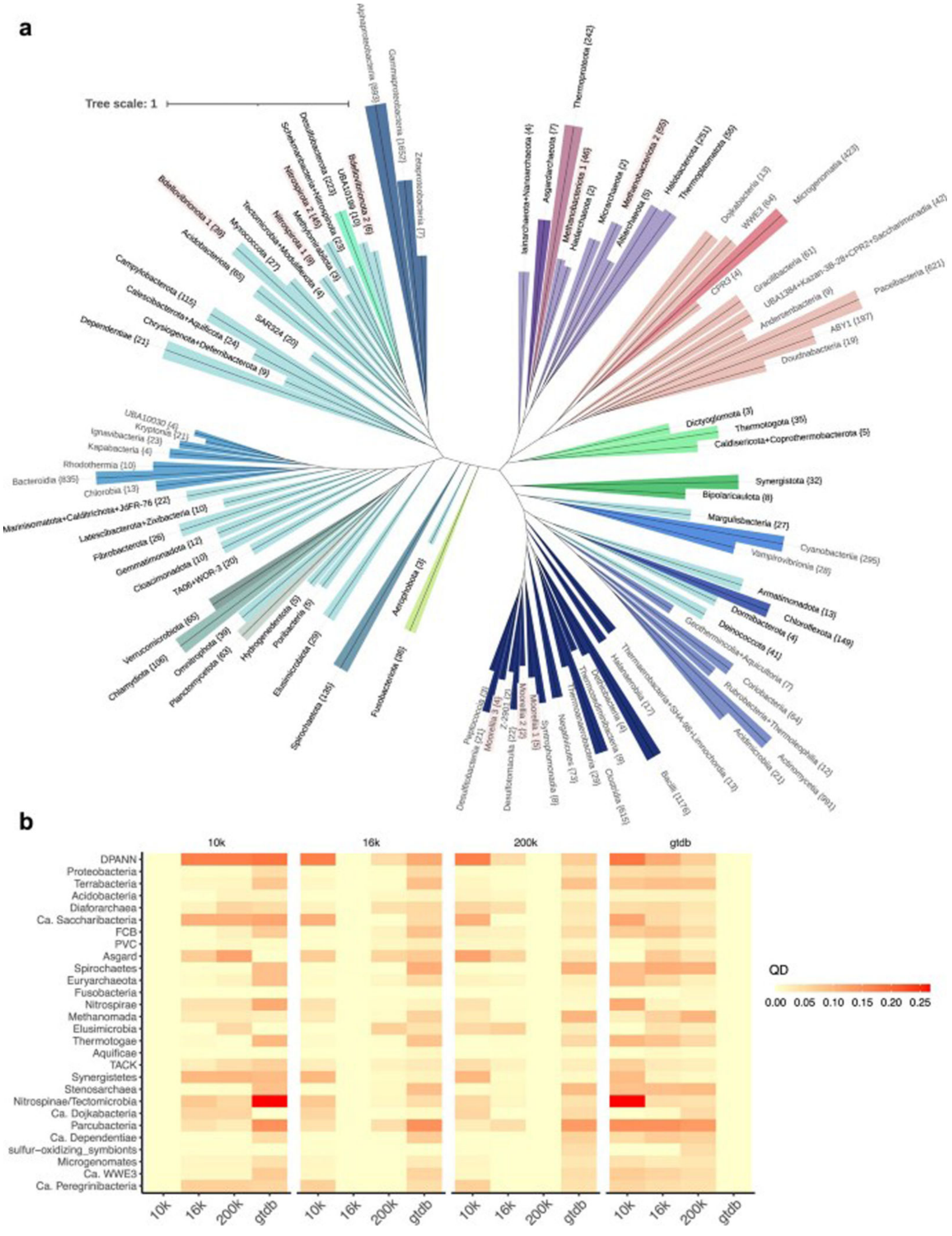


Figure Extended Figure 3:

A) The 10K ASTRAL tree decorated with GTDB taxonomy. **B)** All by all comparison between the 10k, 16k, 200k, and GTDB trees on NCBI defined phyla and super-phyla.

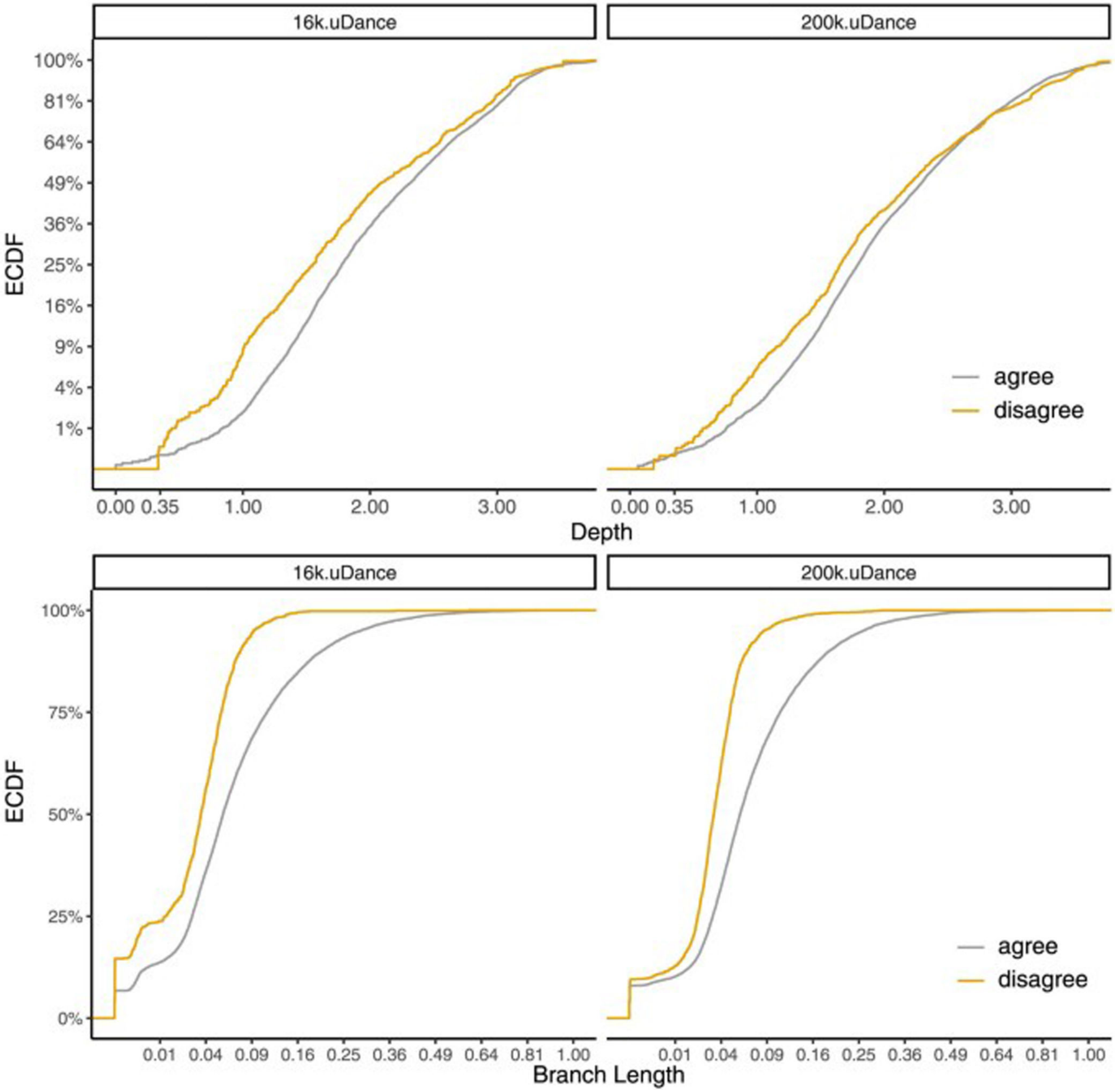


Figure Extended Figure 4:

ECDF of depth and the branch length of agreeing and disagreeing branches between the backbone and output phylogenies.

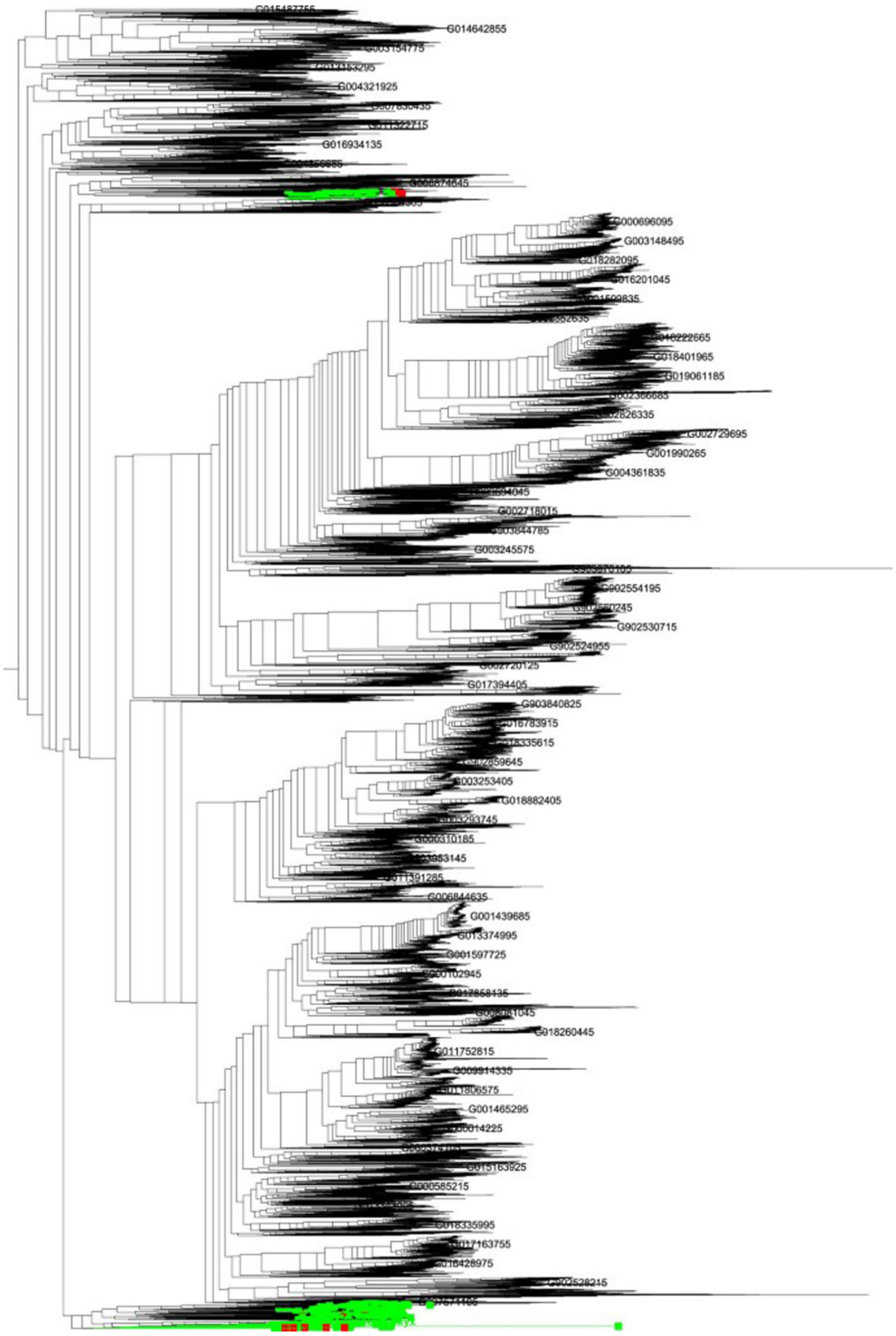


Figure Extended Figure 5:

The paraphyletic Myxococcota phylum on the GTDB phylogenetic tree. Green and red sequences represent the members of the phylum that are proximal Desulfobacteria and Proteobacteria respectively in the 200k tree.

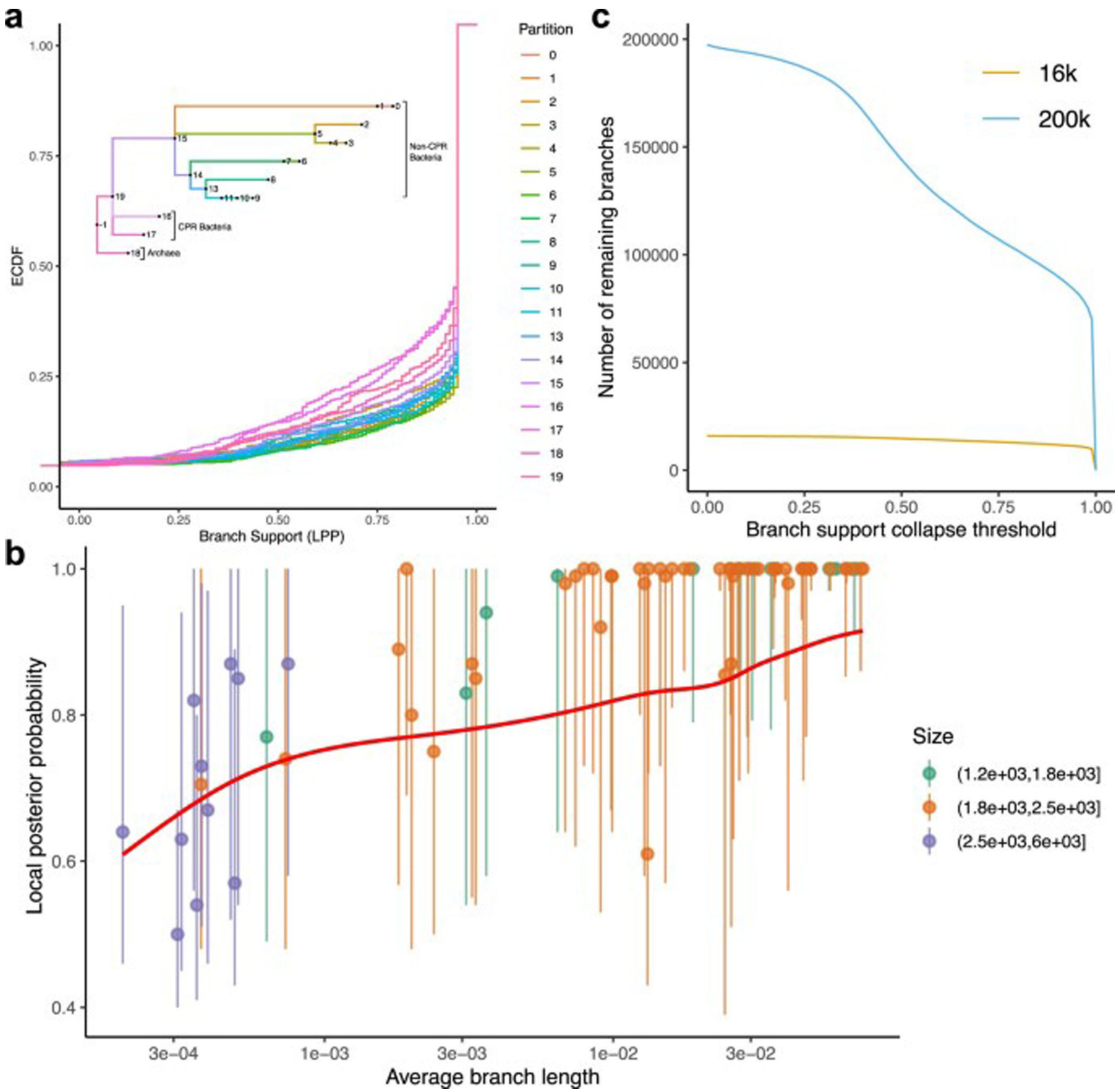


Figure Extended Figure 6:

A) ECDF of branch support (local posterior probability) across partitions of the 16k tree. B) Branch support (local posterior probability) versus the diversity (average branch length) of all 78 partitions in the 200k tree. The dot and the range indicates median and 0.25–0.75 quantiles. Three colors correspond to clusters that are unusually small, unusually large, or the expected size. 14 of 15 partitions with the lowest diversity are of size between 2,500 and 6,000. The largest partitions in the 200k tree are over-represented parts of the tree of life in the reference genomic library that did not break into smaller partitions by uDance because of their lower diversity. C) Number of uncollapsed branches vs branch support (localPP) collapsing threshold. The comparison of number of uncollapsed branches vs branch support collapsing threshold for 16k and 200k tree.

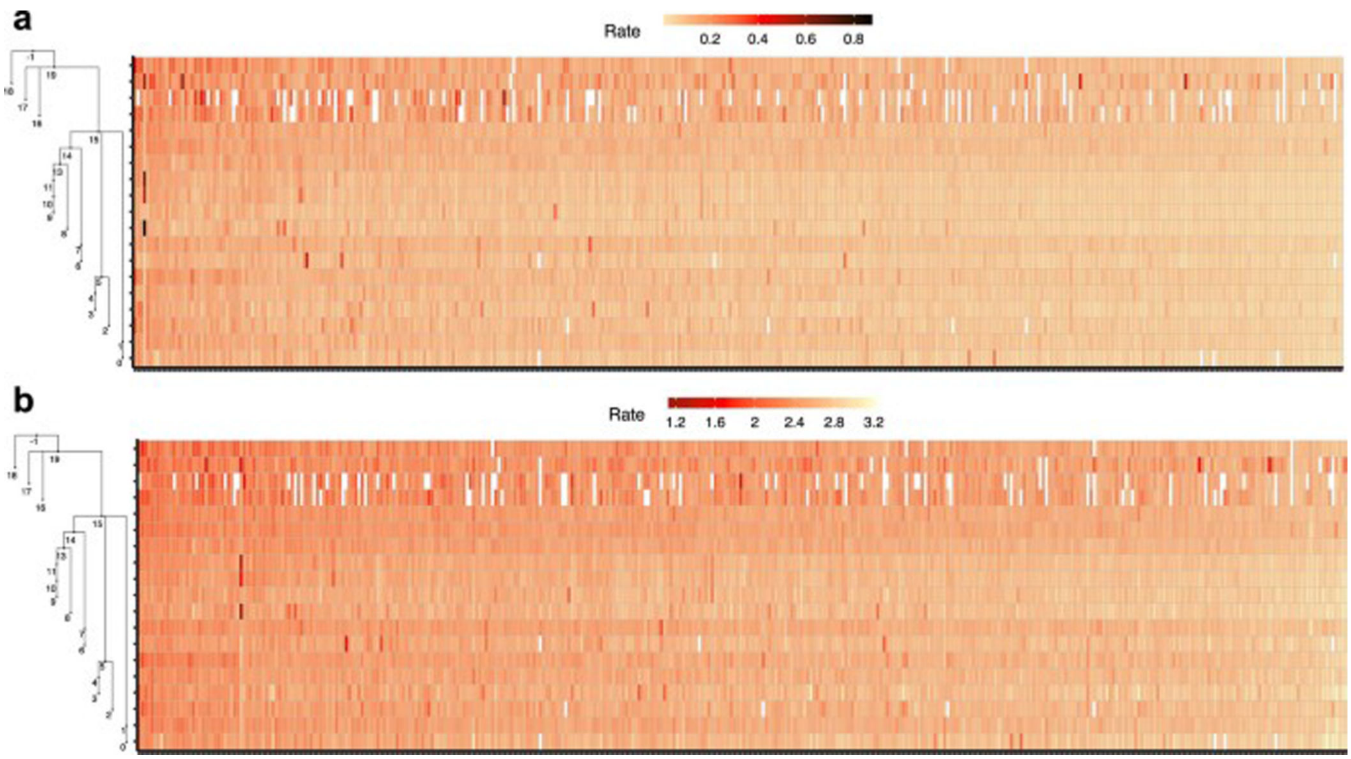
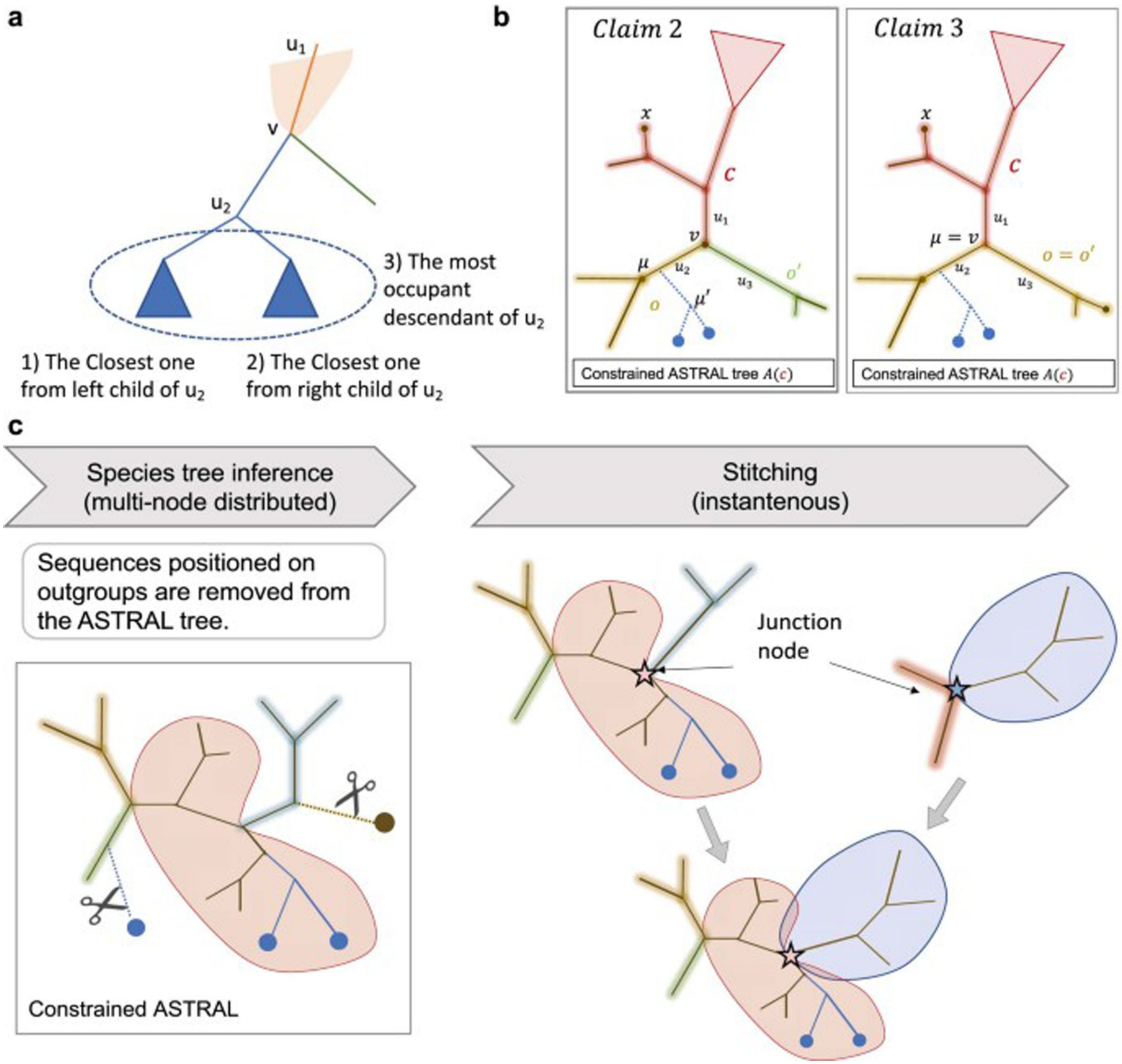


Figure Extended Figure 7:

Divide-and-conquer approach permits heterogeneity of model of evolution parameters across the tree. In discrete four-category LG+GAMMA model, the rate heterogeneity across sites is modeled by a discrete approximation of the Gamma distribution. The first and the fourth discrete rate of LG+G model for every partition and gene pair in the 16k tree are shown in (a) and (b), respectively. The partition hierarchy created by uDance is shown on the left. A blank cell indicates a missing gene in a partition.


a
u_1
v
u_2
3) The most occupant descendant of u_2
1) The Closest one from left child of u_2
2) The Closest one from right child of u_2
b
Claim 2
x
c
u_1
v
μ
u_2
u_3
o'
o
μ'
Constrained ASTRAL tree $A(c)$
Claim 3
x
c
u_1
$\mu = v$
u_2
u_3
$o = o'$
Constrained ASTRAL tree $A(c)$
c
Species tree inference (multi-node distributed)
Stitching (instantenous)
Junction node
Constrained ASTRAL

Sequences positioned on outgroups are removed from the ASTRAL tree.

Figure Extended Figure 8:

A) Outgroup taxa selection strategy. Two to three taxa are chosen from the partition c_2 (blue) to be added to the partition c_1 (orange). B) Finding junction node v in Constrained ASTRAL tree for color (partition) c. We illustrate the setup for Claim 2 and 3. C) Stitching happens at junction nodes. After removing taxa placed on outgroup branches, other subtrees can be stitched to this subtree without any need for conceptual merge, but simply replacing the connecting nodes.

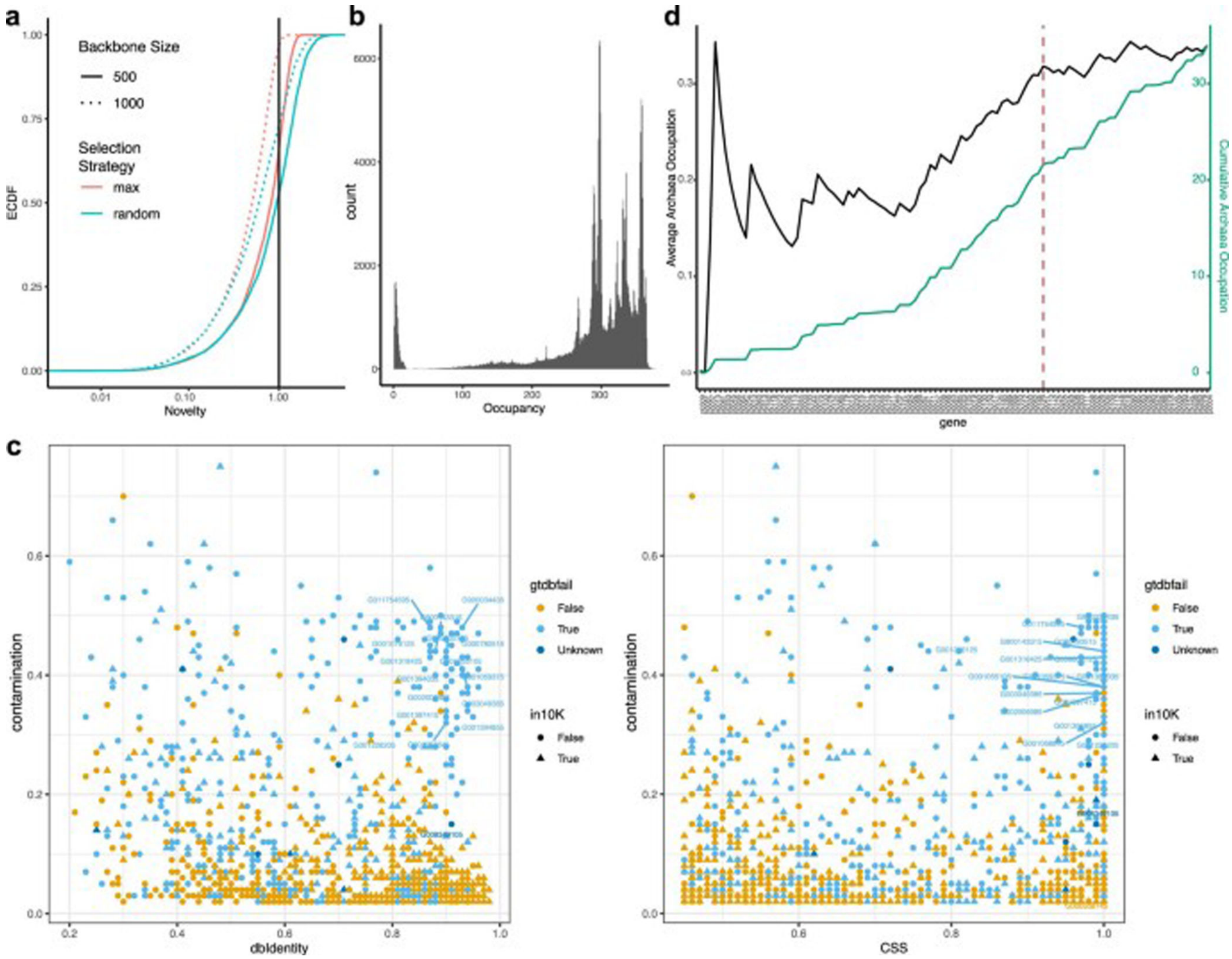


Figure Extended Figure 9:

A) Determining the backbone size in the simulated HD-100 dataset. ECDF of novelty of query sequences with respect to a backbone tree of N downsampled sequences induced from the full HGT dataset. With $N = 1000$ sequences selected using TreeCluster-max, for more than 95% of the query sequences, the novelty score is less than one. Novelty score is defined as two times the terminal branch length of the query when placed on the true location on the backbone tree. B) The distribution of number of marker genes per sequence in WoL2 dataset. C) Two dot plots comparing (1) contamination ratio-vs-CSS and (2) contamination ratio-vs-GUNC database identity for the species in the 16K tree that are ”chimeric” (CSS > 0.45). We colored each point based on whether the sequence passed QC in GTDB or not. Triangle points are the published WoL tree, and round points are the new 6K taxa we added in the 16K tree. In these figures, We annotated 17 taxa in the 16K tree that might be reducing the accuracy of uDance and APPLES-2 in large clusters (subtrees) that include some of the densely sampled species such as Salmonella, E. coli, TB, etc. The pattern is clear that these contaminated genomes can be characterized by a large contamination ratio, near 100% CSS, and high database identity. We do not remove high CSS taxa if their contamination percentage is low, since uDance performs whole-genomebased placement, and it's tolerant to low levels of contamination. Removing taxa satisfying both CSS ≥ 0.5 and Contamination ratio ≥ 0.25 removes 195 taxa from the 16K tree. 171 of them (87%) fail QC in GTDB. Of 195, 37 taxa are also present in WoL tree. 29 of these 37 don't pass GTDB QC. D) Determining "best" marker genes to be used with APPLES-2 in order to improve placement speed. We picked a local maxima of average Archaea occupancy at 68th marker gene, which also ensures that, on average, Archaea sequences have at least 20 marker genes. The set of Archaea used in computation of these two statistics are taken from WoL tree. The name of the genes (shown on x-axis) are not important and can be ignored.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgements

This work was supported by the National Science Foundation (NSF) grant IIS 1845967 to S.M and National Institutes of Health (NIH) grant 1R35GM142725 to S.M, NIH grants U19AG063744, U24DK131617 and DP1-AT010885, to R.K. and NSF grant RAPID 20385.09 to R.K. It was also supported by the 2020 UCSD Center for Microbiome Innovation Grand Challenge Award to M.B. Computations were partially performed using Expanse at San Diego Supercomputing Centre through allocations ASC150046 and BIO210103 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

Data Availability

Microbial genomes are publically available via RefSeq https://ftp.ncbi.nlm.nih.gov/refseq/release/. Microbial and simulated gene sequences/alignments, intermediate and output files from the analysis of biological and simulated data are openly available at Harvard Dataverse [64] https://doi.org/10.7910/DVN/BCUM6P. Microbial tree output files and postprocessing data is available at Zenodo [65] https://doi.org/10.5281/zenodo.8057941.

References

[1]. Gonzalez A. et al. Qiita: rapid, web-enabled microbiome meta-analysis. Nature Methods 15, 796–798 (2018). URL https://www.nature.com/articles/s41592-018-0141-9. [PubMed: 30275573]

[2]. Zhu Q. et al. Phylogeny-Aware Analysis of Metagenome Community Ecology Based on Matched Reference Genomes while Bypassing Taxonomy. mSystems 7, 1 (2022). URL https://journals.asm.org/doi/10.1128/msystems.00167-22.

[3]. Nayfach S, Shi ZJ, Seshadri R, Pollard KS & Kyrpides NC New insights from uncultivated genomes of the global human gut microbiome. Nature 568, 505–510 (2019). URL 10.1038/s41586-019-1058-x. [PubMed: 30867587]

[4]. DeSantis TZ et al. Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB. Appl. Environ. Microbiol 72, 5069–5072 (2006).URL http://aem.asm.org/cgi/content/abstract/72/7/5069http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1489311/. [PubMed: 16820507]

[5]. Quast C. et al. The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. Nucleic Acids Research 41, D590–D596 (2012). URL http://academic.oup.com/nar/article/41/D1/D590/1069277/The-SILVA-ribosomal-RNA-gene-database-project. [PubMed: 23193283]

[6]. Zhu Q. et al. Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea. Nature Communications 10, 5477 (2019). URL 10.1038/s41467-019-13443-4 http://www.nature.com/articles/s41467-019-13443-4.

[7]. Parks DH et al. A complete domain-to-species taxonomy for Bacteria and Archaea. Nature Biotechnology 38, 1079–1086 (2020). URL 10.1038/s41587-020-0501-8 https://www.nature.com/articles/s41587-020-0501-8.

[8]. Mirarab S, Nakhleh L. & Warnow T. Multispecies Coalescent: Theory and Applications in Phylogenetics. Annual Review of Ecology, Evolution, and Systematics 52, 247–268 (2021). URL https://www.annualreviews.org/doi/10.1146/annurev-ecolsys-012121-095340.

[9]. Davidson R, Vachaspati P, Mirarab S. & Warnow T. Phylogenomic species tree estimation in the presence of incomplete lineage sorting and horizontal gene transfer. BMC Genomics 16, S1 (2015). URL https://bmcgenomics.biomedcentral.com/articles/10.1186/1471-2164-16-S10-S1.

[10]. Maddison WP Gene Trees in Species Trees. Systematic Biology 46, 523–536 (1997). URL http://www.jstor.org/stable/2413694?origin=crossrefhttp://sysbio.oxfordjournals.org/cgi/content/abstract/46/3/523 http://sysbio.oxfordjournals.org/content/46/3/523.short.

[11]. Degnan JH & Rosenberg NA Gene tree discordance, phylogenetic inference and the multispecies coalescent. Trends in Ecology and Evolution 24, 332–340 (2009). URL http://www.cell.com/ecology-evolution/abstract/S0169-5347(09)00084-6 http://www.sciencedirect.com/science/article/pii/S0169534709000846. [PubMed: 19307040]

[12]. Gogarten JP, Doolittle WF & Lawrence JG Prokaryotic Evolution in Light of Gene Transfer. Molecular Biology and Evolution 19, 2226–2238 (2002). URL http://academic.oup.com/mbe/article/19/12/2226/997576. [PubMed: 12446813]

[13]. Creevey CJ, Doerks T, Fitzpatrick DA, Raes J. & Bork P. Universally Distributed SingleCopy Genes Indicate a Constant Rate of Horizontal Transfer. PLoS ONE 6, e22099 (2011). URL https://dx.plos.org/10.1371/journal.pone.0022099.

[14]. Yan Z, Smith ML, Du P, Hahn MW & Nakhleh L. Species Tree Inference Methods Intended to Deal with Incomplete Lineage Sorting Are Robust to the Presence of Paralogs. Systematic Biology 71, 367–381 (2022). [PubMed: 34245291]

[15]. Asnicar F. et al. Precise phylogenetic analysis of microbial isolates and genomes from metagenomes using PhyloPhlAn 3.0. Nature Communications 11, 2500 (2020). URL 10.1038/s41467-020-16366-7 http://www.nature.com/articles/s41467-020-16366-7.

[16]. Kozlov AM, Darriba D, Flouri T, Morel B. & Stamatakis A. RAxML-NG: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. Bioinformatics 35, 4453–4455 (2019). URL https://academic.oup.com/bioinformatics/article/35/21/4453/5487384. [PubMed: 31070718]

[17]. Mirarab S. et al. ASTRAL: genome-scale coalescent-based species tree estimation. Bioinformatics 30, i541–i548 (2014). URL http://bioinformatics.oxfordjournals.org/cgi/content/long/30/17/i541http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/btu462 https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btu462. [PubMed: 25161245]

[18]. Matsen FA, Kodner RB & Armbrust EV pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. BMC Bioinformatics 11, 538 (2010). URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-538. [PubMed: 21034504]

[19]. Rabiee M. & Mirarab S. INSTRAL: Discordance-aware Phylogenetic Placement using Quartet Scores. bioRxiv 432906 (2018). URL http://biorxiv.org/content/early/2018/10/02/432906.abstract.

[20]. Wedell E, Cai Y. & Warnow T. SCAMPP: Scaling Alignment-Based Phylogenetic Placement to Large Trees. IEEE/ACM Transactions on Computational Biology and Bioinformatics 20, 1417–1430 (2023). URL https://ieeexplore.ieee.org/document/9763324/. [PubMed: 35471888]

[21]. Barbera P. et al. EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences. Systematic Biology 68, 365–369 (2019). URL https://academic.oup.com/sysbio/article/68/2/365/5079844. [PubMed: 30165689]

[22]. Warnow T. Divide-and-Conquer Tree Estimation: Opportunities and Challenges, 121–150 (Springer International Publishing, Cham, 2019). URL 10.1007/978-3-030-10837-3_6.

[23]. Nelesen SM, Liu K, Wang L-S, Linder CR & Warnow T. DACTAL: divide-and-conquer trees (almost) without alignments. Bioinformatics 28, i274–i282 (2012). [PubMed: 22689772]

[24]. Huson DH, Nettles SM & Warnow TJ Disk-covering, a fast-converging method for phylogenetic tree reconstruction. Journal of computational biology 6, 369–386 (1999). URL http://www.ncbi.nlm.nih.gov/pubmed/10582573. [PubMed: 10582573]

[25]. Balaban M, Jiang Y, Balaban M, Zhu Q, McDonald D, Knight R. & Mirarab S. Generation of accurate, expandable phylogenomic trees with uDance. Github. https://github.com/balabanmetin/uDance (2023).

[26]. Balaban M, Jiang Y, Roush D, Zhu Q. & Mirarab S. Fast and accurate distance-based phylogenetic placement using divide and conquer. Molecular Ecology Resources 22, 1213–1227 (2022). URL https://onlinelibrary.wiley.com/doi/10.1111/1755-0998.13527. [PubMed: 34643995]

[27]. Rabiee M. & Mirarab S. Forcing external constraints on tree inference using ASTRAL. BMC Genomics 21, 218 (2020). URL https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-020-6607-z. [PubMed: 32299337]

[28]. Price MN, Dehal PS & Arkin AP FastTree-2 – Approximately Maximum-Likelihood Trees for Large Alignments. PLoS ONE 5, e9490 (2010). URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2835736&tool=pmcentrez&rendertype=abstract. [PubMed: 20224823]

[29]. Yin J, Zhang C. & Mirarab S. ASTRAL-MP: scaling ASTRAL to very large datasets using randomization and parallelization. Bioinformatics 35, 3961–3969 (2019). URL https://academic.oup.com/bioinformatics/article/35/20/3961/5418955. [PubMed: 30903685]

[30]. Vachaspati P. & Warnow T. ASTRID: Accurate Species TRees from Internode Distances. BMC Genomics 16, S3 (2015).

[31]. McDonald D. et al. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. The ISME journal 6, 610–618 (2012). [PubMed: 22134646]

[32]. Parks DH et al. A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. Nature Biotechnology 36, 996–1004 (2018). URL https://www.nature.com/articles/nbt.4229.

[33]. Coleman GA et al. A rooted phylogeny resolves early bacterial evolution. Science 372 (2021). URL https://www.science.org/doi/10.1126/science.abe0511.

[34]. Ashburner M. et al. Gene Ontology: tool for the unification of biology. Nature Genetics 25, 25–29 (2000). URL https://www.nature.com/articles/ng0500_25. [PubMed: 10802651]

[35]. Sayyari E. & Mirarab S. Fast Coalescent-Based Computation of Local Branch Support from Quartet Frequencies. Molecular biology and evolution 33, 1654–1668 (2016). URL http://mbe.oxfordjournals.org/content/early/2016/04/15/molbev.msw079.abstracthttp://mbe.oxfordjournals.org/lookup/doi/10.1093/molbev/msw079 https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/msw079. [PubMed: 27189547]

[36]. OneKP Initiative OTPT One thousand plant transcriptomes and the phylogenomics of green plants. Nature 574, 679–685 (2019). URL http://www.nature.com/articles/s41586-019-1693-2. [PubMed: 31645766]

[37]. Jiang Y, Balaban M, Zhu Q. & Mirarab S. DEPP: Deep Learning Enables Extending Species Trees using Single Genes. Systematic Biology 72, 17–34 (2023). URL https://academic.oup.com/sysbio/article/72/1/17/6575921. [PubMed: 35485976]

[38]. Jiang Y, Tabaghi P. & Mirarab S. Learning hyperbolic embedding for phylogenetic tree placement and updates. Biology 11 (2022). URL https://www.mdpi.com/2079-7737/11/9/1256.

[39]. Nasko DJ, Koren S, Phillippy AM & Treangen TJ RefSeq database growth influences the accuracy of k-mer-based lowest common ancestor species identification. Genome Biology 19, 165 (2018). URL https://genomebiology.biomedcentral.com/articles/10.1186/s13059-018-1554-6. [PubMed: 30373669]

[40]. Locey KJ & Lennon JT Scaling laws predict global microbial diversity. Proceedings of the National Academy of Sciences 113, 5970–5975 (2016). URL http://www.pnas.org/lookup/doi/10.1073/pnas.1521291113.

[41]. Fullam A. et al. proGenomes3: approaching one million accurately and consistently annotated high-quality prokaryotic genomes. Nucleic Acids Research (2023).

[42]. Jukes TH & Cantor CR Evolution of protein molecules. In Mammalian protein metabolism, Vol. III (1969), pp. 21–132 III, 21–132 (1969).

[43]. Sonnhammer EL & Hollich V. Scoredist: A simple and robust protein sequence distance estimator. BMC Bioinformatics 6, 1–8 (2005). [PubMed: 15631638]

[44]. Darriba D. et al. ModelTest-NG: A New and Scalable Tool for the Selection of DNA and Protein Evolutionary Models. Molecular Biology and Evolution 37, 291–294 (2020). URL https://academic.oup.com/mbe/article/37/1/291/5552155. [PubMed: 31432070]

[45]. Anisimova M, Gil M, Dufayard J-F, Dessimoz C. & Gascuel O. Survey of Branch Support Methods Demonstrates Accuracy, Power, and Robustness of Fast Likelihood-based Approximation Schemes. Systematic Biology 60, 685–699 (2011). URL https://academic.oup.com/sysbio/article/60/5/685/1644562. [PubMed: 21540409]

[46]. Capella-Gutierrez S, Silla-Martinez JM & Gabaldon T. trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses. Bioinformatics 25, 1972–1973 (2009). URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btp348. [PubMed: 19505945]

[47]. Zhang C, Zhao Y, Braun EL & Mirarab S. TAPER: Pinpointing errors in multiple sequence alignments despite varying rates of evolution. Methods in Ecology and Evolution 12, 2145–2158 (2021). URL https://onlinelibrary.wiley.com/doi/10.1111/2041-210X.13696.

[48]. Sayyari E, Whitfield JB & Mirarab S. Fragmentary Gene Sequences Negatively Impact Gene Tree and Species Tree Reconstruction. Molecular Biology and Evolution 34, 3279–3291 (2017). URL https://academic.oup.com/mbe/article/34/12/3279/4344836. [PubMed: 29029241]

[49]. Mai U. & Mirarab S. TreeShrink: fast and accurate detection of outlier long branches in collections of phylogenetic trees. BMC Genomics 19, 272 (2018). URL https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-018-4620-2. [PubMed: 29745847]

[50]. Balaban M, Moshiri N, Mai U, Jia X. & Mirarab S. TreeCluster: Clustering biological sequences using phylogenetic trees. PLOS ONE 14, e0221068 (2019). URL 10.1371/journal.pone.0221068 http://dx.plos.org/10.1371/journal.pone.0221068.

[51]. Mölder F. et al. Sustainable data analysis with Snakemake. F1000Research 10, 33 (2021). URL https://f1000research.com/articles/10-33/v2. [PubMed: 34035898]

[52]. Mallo D, De Oliveira Martins L. & Posada D. SimPhy : Phylogenomic Simulation of Gene, Locus, and Species Trees. Systematic Biology 65, 334–344 (2016). URL http://biorxiv.org/content/early/2015/06/30/021709.abstracthttp://sysbio.oxfordjournals.org/content/early/2015/12/04/sysbio.syv082.short?rss=1https://academic.oup.com/sysbio/article-lookup/doi/10.1093/sysbio/syv082. [PubMed: 26526427]

[53]. Fletcher W. & Yang Z. INDELible: A flexible simulator of biological sequence evolution. Molecular Biology and Evolution 26, 1879–1888 (2009). [PubMed: 19423664]

[54]. Nguyen N.-p. D., Mirarab S, Kumar K. & Warnow T. Ultra-large alignments using phylogenyaware profiles. Genome Biology 16, 124 (2015). URL http://genomebiology.com/2015/16/1/124https://genomebiology.biomedcentral.com/articles/10.1186/s13059-015-0688-z. [PubMed: 26076734]

[55]. Yang Z, Nielsen R, Goldman N. & Pedersen A-MK Codon-Substitution Models for Heterogeneous Selection Pressure at Amino Acid Sites. Genetics 155, 431–449 (2000). URL https://academic.oup.com/genetics/article/155/1/431/6047860. [PubMed: 10790415]

[56]. Haft DH et al. RefSeq: an update on prokaryotic genome annotation and curation. Nucleic Acids Research 46, D851–D860 (2018). URL http://academic.oup.com/nar/article/46/D1/D851/4588110. [PubMed: 29112715]

[57]. Segata N, Börnigen D, Morgan XC & Huttenhower C. PhyloPhlAn is a new method for improved phylogenetic and taxonomic placement of microbes. Nature Communications 4, 2304 (2013). URL https://www.nature.com/articles/ncomms3304.

[58]. Hyatt D. et al. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics 11, 119 (2010). URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-119. [PubMed: 20211023]

[59]. Darling AE et al. PhyloSift: phylogenetic analysis of genomes and metagenomes. PeerJ 2, e243 (2014). URL https://peerj.com/articles/243. [PubMed: 24482762]

[60]. Orakov A. et al. GUNC: detection of chimerism and contamination in prokaryotic genomes. Genome Biology 22, 178 (2021). URL https://genomebiology.biomedcentral.com/articles/10.1186/s13059-021-02393-0. [PubMed: 34120611]

[61]. Le SQ & Gascuel O. An Improved General Amino Acid Replacement Matrix. Molecular Biology and Evolution 25, 1307–1320 (2008). URL https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/msn067. [PubMed: 18367465]

[62]. Letunic I. & Bork P. Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation. Nucleic Acids Research 49, W293–W296 (2021). URL https://academic.oup.com/nar/article/49/W1/W293/6246398. [PubMed: 33885785]

[63]. Wickett NJ et al. Phylotranscriptomic analysis of the origin and early diversification of land plants. Proceedings of the National Academy of Sciences 111, 4859–4868 (2014). URL http://www.pnas.org/lookup/doi/10.1073/pnas.1323926111 http://www.pnas.org/cgi/content/long/111/45/E4859.

[64]. Balaban M, Jiang Y, Balaban M, Zhu Q, McDonald D, Knight R. & Mirarab S. Data for article: Generation of accurate, expandable phylogenomic trees with uDance. Harvard Dataverse. 10.7910/DVN/BCUM6P (2023).

[65]. Balaban M, Jiang Y, Balaban M, Zhu Q, McDonald D, Knight R. & Mirarab S. Post-processing data for article: Generation of accurate, expandable phylogenomic trees with uDance. Zenodo. 10.5281/zenodo.8057941 (2023).

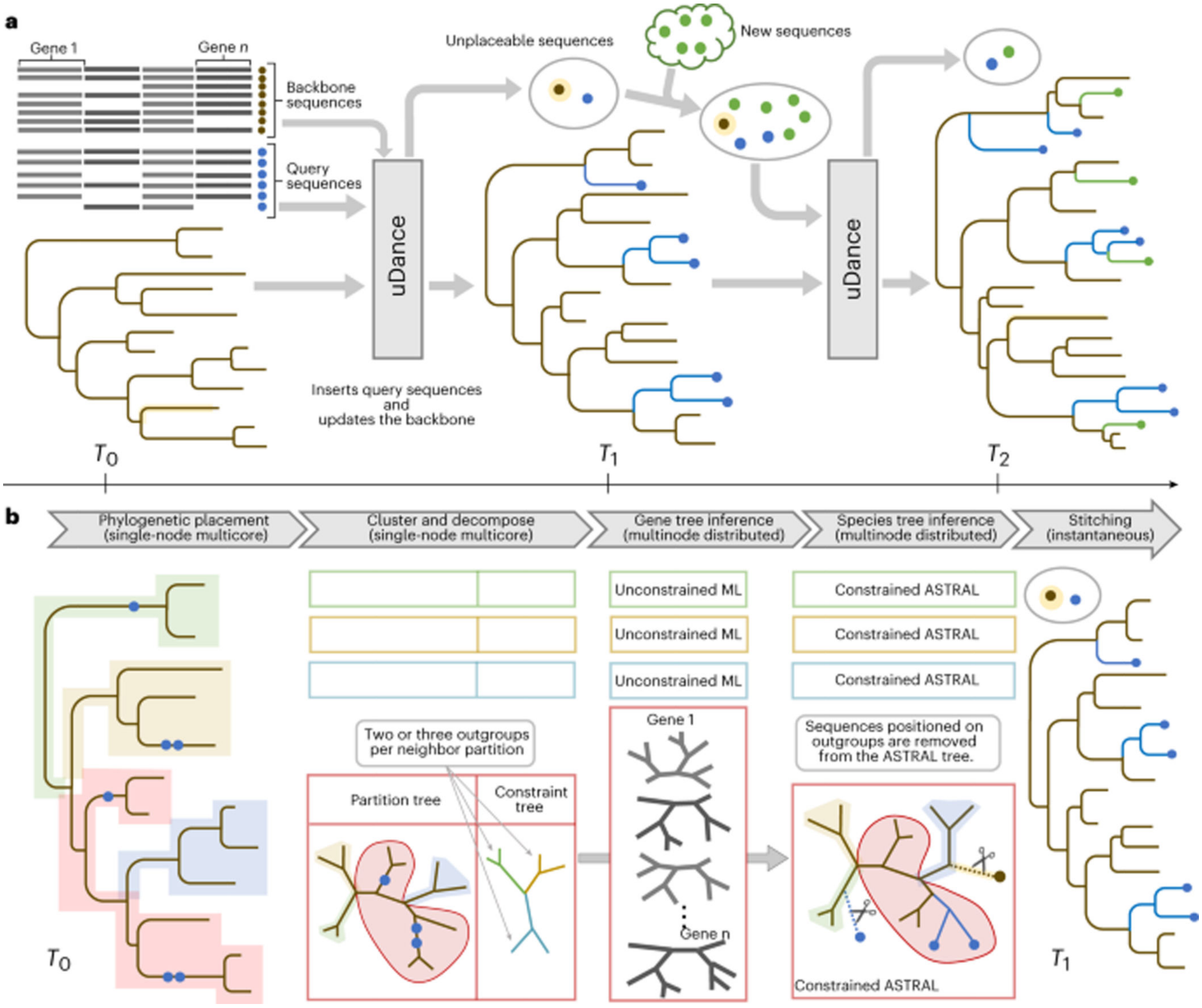

a
Gene 1
Gene n
Backbone sequences
Query sequences
Unplaceable sequences
New sequences
uDance
uDance
Inserts query sequences and updates the backbone
T_0
T_1
T_2
b
Phylogenetic placement (single-node multicore)
Cluster and decompose (single-node multicore)
Gene tree inference (multinode distributed)
Species tree inference (multinode distributed)
Stitching (instantaneous)
Unconstrained ML
Unconstrained ML
Unconstrained ML
Constrained ASTRAL
Constrained ASTRAL
Constrained ASTRAL
Two or three outgroups per neighbor partition
Partition tree
Constraint tree
Gene 1
Gene n
Sequences positioned on outgroups are removed from the ASTRAL tree.
Constrained ASTRAL
T_0
T_1

Figure 1:

uDance overview. a) Updates to the phylogenetic tree through time (T_0, T_1, and T_2) with new sequences arriving and each tree used as the backbone tree in the next step. Some sequences may be unplaceable and are added as query to the next iteration. b) Each update involves divide-and-conquer and several steps, most of which can be executed in a distributed fashion. The new sequences are independently added to the tree using phylogenetic placement. Then, the tree with placements is divided into partitions, and each partition plus enough outgroups from other partitions are reanalyzed to infer local gene trees and a local species tree. These species trees are next joined together using a constrained search that makes it possible to stitch back together the subtrees.

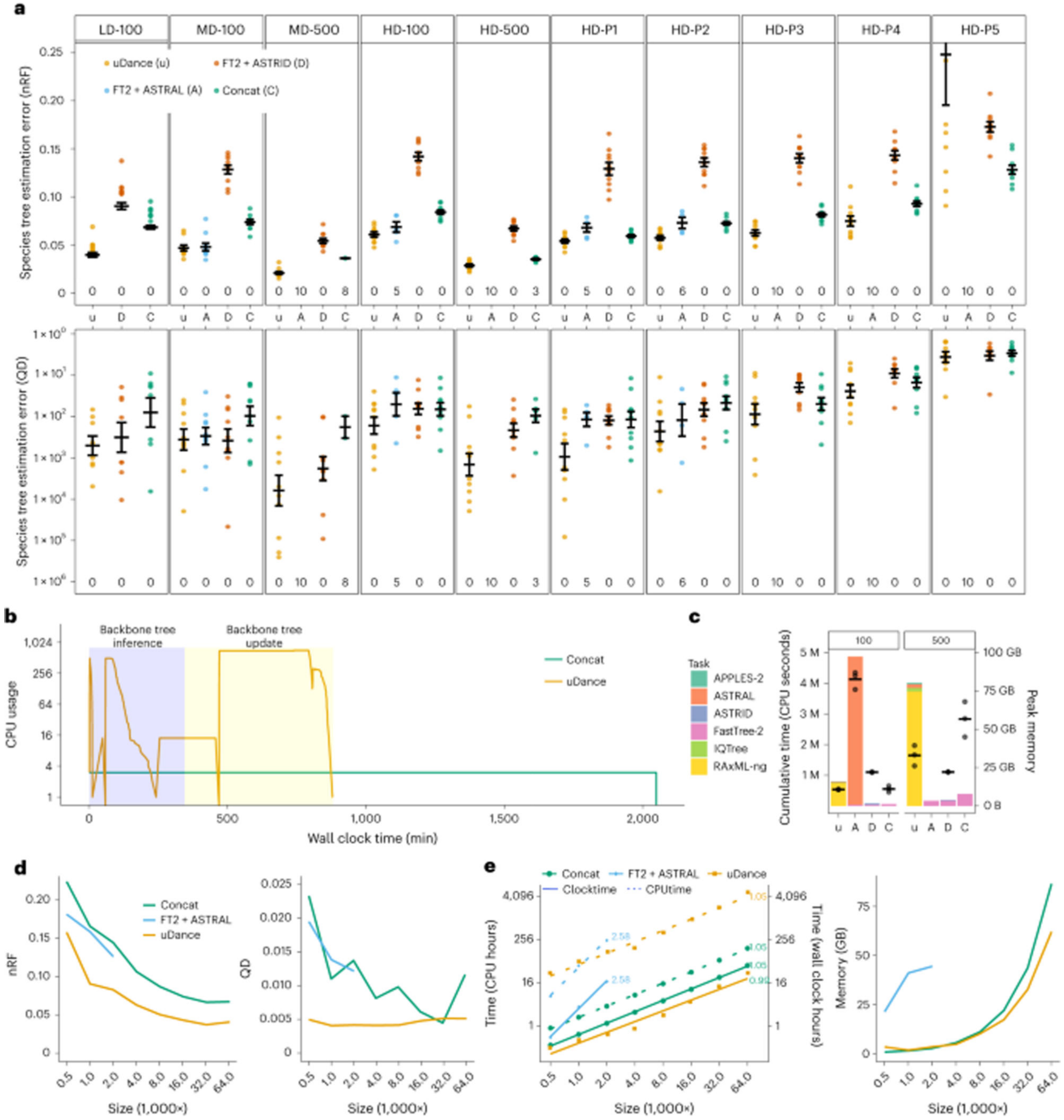


Figure 2:

Results on simulation data set: three model condition with low (LD), mid (MD), and high (HD) discordance with 100 and 500 genes. HD-P1 to HD-P5 are 100 gene subsets of HD-500 with successively higher levels of gene tree discordance. (a) normalized Robinson-Foulds (nRF) and Quartet distance (QD) between inferred and true species tree. We show all points, mean (long dash) and the first and third quartiles (whiskers) over *n* = 10 independent replicates. The number above x-axis indicates the number of replicates in which each method failed to return a tree in 2 days, given 125GB of memory and reduces *n*. (b) The timeline of CPU usage on replicate 7 in HD-100 data set. Maximum number of cores made available for uDance is 672. (c) Cumulative CPU time (bars) and peak memory (dots and the horizontal bar) used by each method in *n* = replicates where concatenation method completed on HD-500 and FT2+ASTRAL method completed on HD-100 data set (mean shown as dashed line). FT2+ASTRAL failed in HD-500 for all replicates. For uDance, running time calculations include the time spent on backbone estimation. (d) The result of serial (online) tree inference on simulated data, showing nRF and QD Error, (e) wall-clock time, CPU time, and memory use versus output tree size. Staring from 250-sequence *de novo* backbone, every succeeding tree uses the preceding uDance output tree as the backbone. Missing points indicate failure to finish in the allotted 48-hour wall-time (FT2 was allowed 49 hours in one case). CPU time is aggregated cores across all nodes. In log-log plots, the slope of the line is shown for each method. Experiments are performed on a HPC with 45 exclusive nodes, each with Intel(R) Xeon(R) 16 cores E5–2670 2.60GHz CPU and 128 GB RAM.

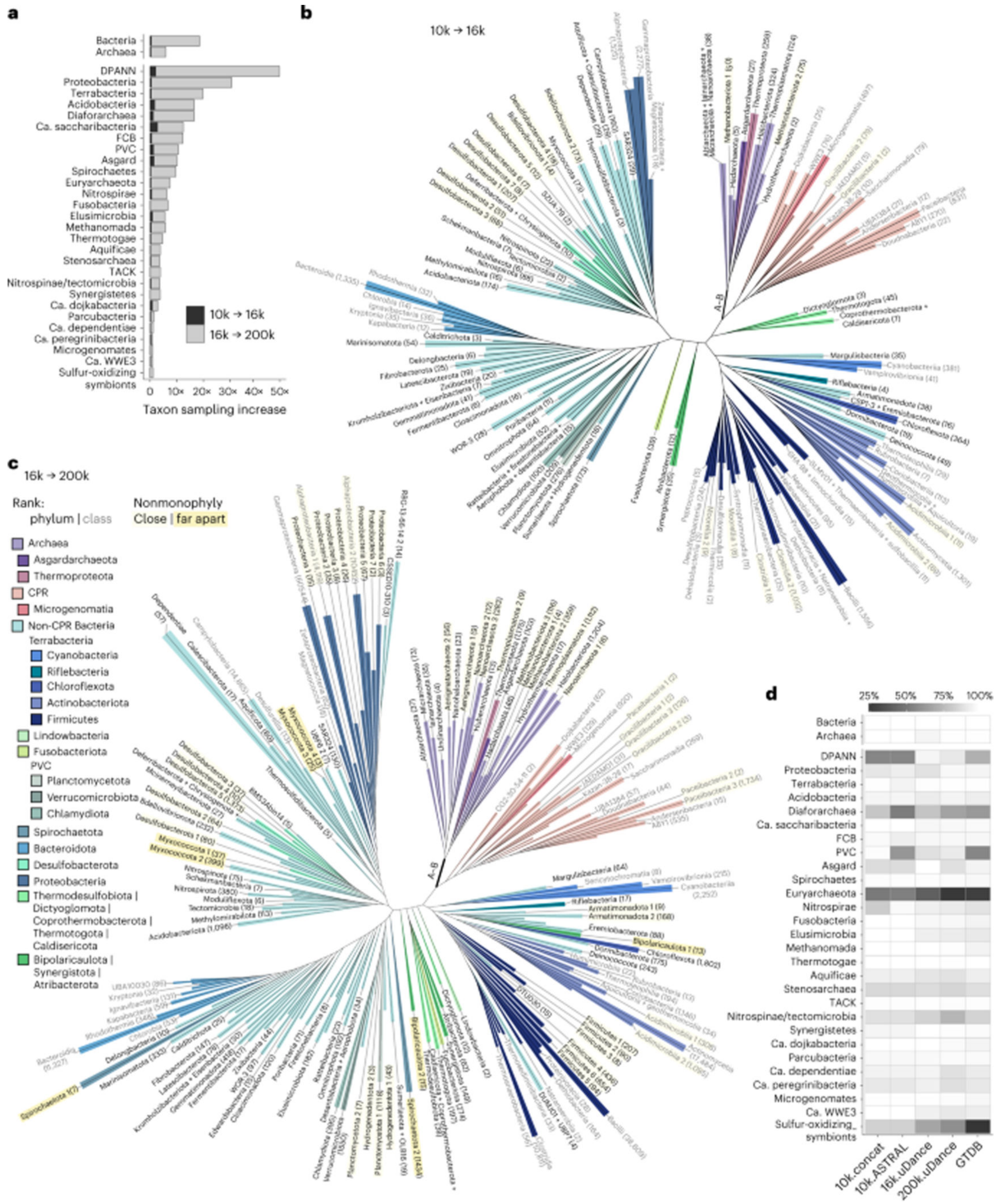


Figure 3:

New trees of microbial life. (a) Increased taxon sampling with respect to the 10k (Web of Life) tree. (b,c) 16k and 200k trees built using uDance. The tree is decorated with GTDB taxonomy v207 (with _A, _B, etc. suffixes removed) and collapses at phylum level for small phyla and class level for large phyla. Number of genomes in each group is shown, and taxon-holder names with less than 50 genomes were left unlabelled unless they disrupt monophyly of another phylum. Paraphyletic clades in our tree are given a numerical index. (d) Consistency of the large phyla and super-phyla in NCBI taxonomy database with various microbial phylogenies. In (a) and (d), we only show groups with at least 20 members in the 16K tree.

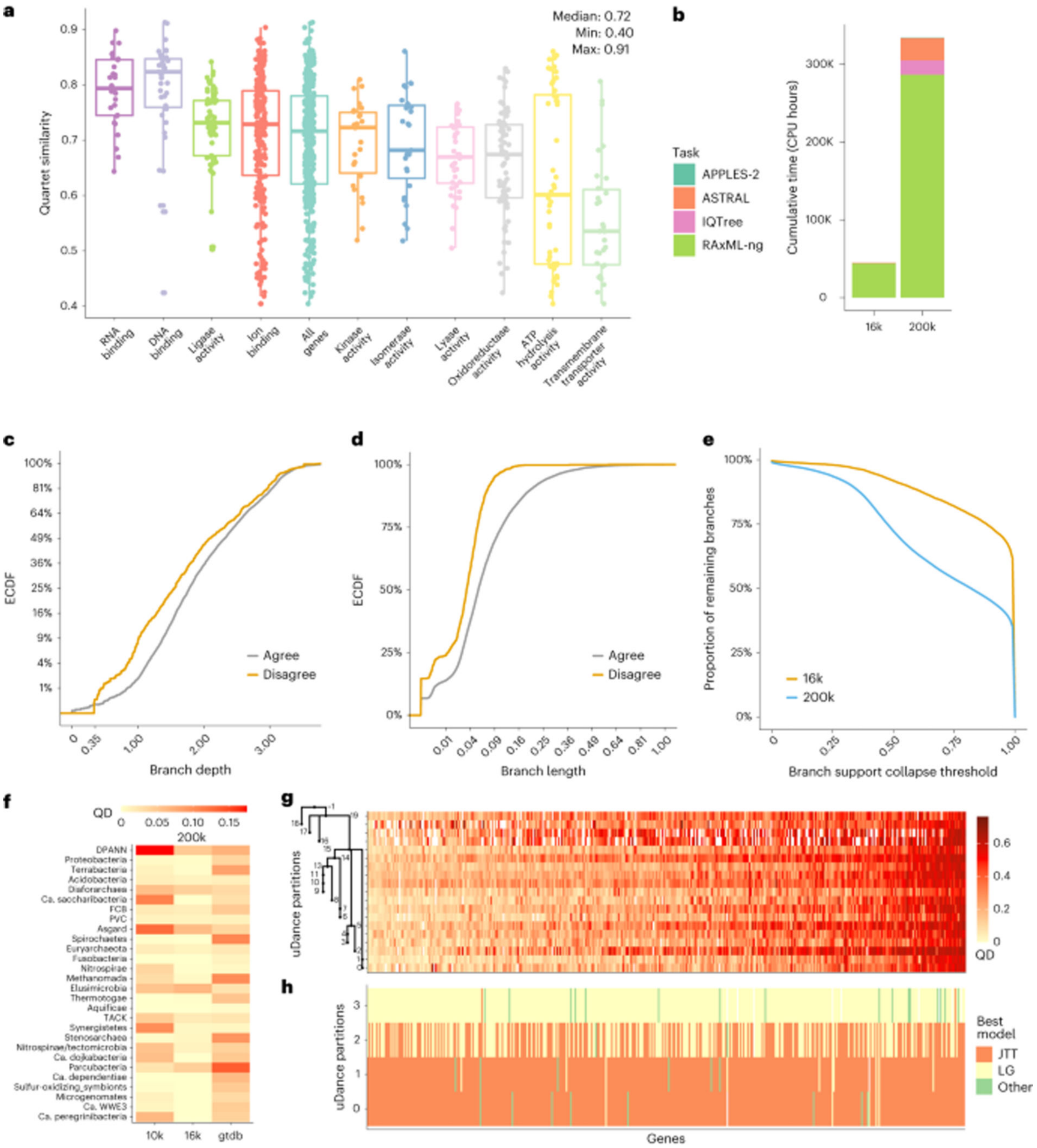


Figure 4:

(a) Quartet score between the 16k species tree and $n = 387$ independent gene trees for categorized by the gene function as well as "All" genes (b) Running time of individual steps. ECDF of (c) depth (root-to-node distance) and (d) the branch length, separated for branches agreeing and disagreeing between the backbone 10k and the output 16k trees. (e) The portion of branches with support above a threshold (x-axis) in uDance trees. (f) Quartet distance between the 200k tree and other trees, restricted to NCBI phyla and super-phyla. (g) Gene tree discordance for every partition (rows) and gene trees (columns) in the 16k tree with the partition spanning tree shown on the left. (h) Heterogeneity of the best fit model of evolution across partitions of the tree in 1,000 plant (1KP) dataset. Boxplots show median (centre), interquartile range; IQR (bounds of box and 1.5× IQR (whiskers).

Table 1:

The comparison between reference phylogenetic trees: uDance-WoL2 (presented here), Web of Life (WoL) [6], GTDB [32], PhyloPhlAn-3 [15], and proGenomes [41]. AA; amino acid. GTDB uses a distinct set of marker genes for Archaea shown in parentheses. proGenomes sequence alignments are not publicly available; instead, we report our estimates based on the longest gene sequences in the dataset.

	# Genomes	# Genes	# AA Sites in gene MSAs	# AA Filtered Sites used in phylogeny	Alignment size used in phylogeny
uDance-WoL2	199,330	387	656,907	213,090	42.5 billion
WoL	10,575	381	1,036,001	241,305	2.6 billion
proGenomes-3	41,171	40	34,107	34,107	1.4 billion
GTDB (v207)	65,703	120 (53)	41,084 (13,540)	5,036 (10,153)	350 million
PhyloPhlAn-3	17,672	400	1,872,710	4,522	80 million

Table 2:

Properties of the simulated dataset.

Condition	# genes	True ILS (nRF)	True HGT+ILS (nRF)	Gene tree error (nRF)	Total discord (nRF)	# Nuc Sites in gene MSAs
MD-500	500	0.03	0.42	0.39	0.58	1,425,470
MD-100	100					289,642
HD-500	500	0.03	0.44	0.54	0.68	258,706
HD-100	100					51,472
HD-P1	100	0.03	0.33	0.51	0.60	43,276
HD-P2			0.39	0.53	0.64	45,334
HD-P3			0.44	0.54	0.67	48,481
HD-P4			0.48	0.55	0.70	51,878
HD-P5			0.55	0.60	0.78	69,737
LD-100	100	0.03	0.43	0.26	0.52	173,447
HD-HET	100	0.03	0.43	0.37	0.57	153,443
Serial	100	0.03	0.45	0.23	0.52	51,726

Note: True ILS and HGT+ILS, average normalized RF distance between true genes trees and the true species tree contributed by ILS only and ILS+HGT; gene tree error and total discord: average normalized RF distance between gene trees estimated with FastTree-2 and the corresponding true gene trees and true species tree, respectively.