nature biotechnology

Article

https://doi.org/10.1038/s41587-023-01868-8

# Generation of accurate, expandable phylogenomic trees with uDance

In the format provided by the authors and unedited

# Supplementary Tables

Table S1: Exploring uDance parameter configuration.

| Condition | Configuration | nRF | ΔnRF | QD | ΔQD | $t_{upd}$ | $t_{tot}$ |
|---|---|---|---|---|---|---|---|
| Default setting | Estimated,1000,500 | 0.061 | 0 | 0.012 | 0 | 161 | 236 |
| Varying Backbone Size | Estimated,250, 500 | 0.073 | 0.012 | 0.018 | 0.006 | 91 | 94 |
| | Estimated,500, 500 | 0.064 | 0.003 | 0.013 | 0.001 | 96 | 107 |
| | Estimated,1000,500 | 0.061 | 0 | 0.012 | 0 | 161 | 236 |
| | Estimated,2000,500 | 0.059 | -0.002 | 0.01 | -0.002 | 89 | 254 |
| Varying Partition Size | Estimated,1000,50 | 0.068 | 0.007 | 0.014 | 0.002 | 27 | 102 |
| | Estimated,1000,100 | 0.064 | 0.003 | 0.012 | 0 | 41 | 115 |
| | Estimated,1000,250 | 0.062 | 0.001 | 0.012 | 0 | 77 | 152 |
| | Estimated,1000,500 | 0.061 | 0 | 0.012 | 0 | 161 | 236 |
| | Estimated,1000,1000 | 0.061 | 0.0005 | 0.013 | 0.001 | 289 | 364 |
| Varying Backbone Tree | Estimated,1000,500 | 0.061 | 0 | 0.012 | 0 | 161 | 236 |
| | True, 1000,500 | 0.057 | -0.004 | 0.0002 | -0.012 | 86 | 86 |

*Note:* nRF,e normalized Robinson-Foulds distance; ΔRF, the nRF for the alternative condition minus the default condition; QD, quartet distance; ΔQD, the QD for alternative condition minus the default condition; $t_{upd}$, runtime excluding the backbone computation in CPU hours; $t_{tot}$, total runtime in CPU hours. Lower the better for all measures. The default condition is repeated for each of comparison.

Table S2: M5 model parameters used in HD-HETER dataset (left) and Jensen-Shannon Divergence between stationary codon frequencies assigned to each model. Stationary codon frequencies (64 parameters per model) are inferred empirically.

| Model | Alpha | Beta |
|---|---|---|
| 1 (Archaea) | 0.204 | 3.781 |
| 2 (CPR) | 0.271 | 2.163 |
| 3 (Non-CPR 1) | 0.508 | 1.463 |
| 4 (Non-CPR 2) | 0.49 | 7.885 |

| Model | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0.0000 | 0.8099 | 0.6837 | 0.0909 |
| 2 | 0.8099 | 0.0000 | 0.0799 | 0.6739 |
| 3 | 0.6837 | 0.0799 | 0.0000 | 0.5773 |
| 4 | 0.0909 | 0.6739 | 0.5773 | 0.0000 |

Table S3: uDance configurable parameters.

| **Parameter** | **Description** |
|---|---|
| chartype | Amino-acid or nucleotide characters |
| backbone | Three backbone tree source options. (1) de-novo, (2) user tree, and (3) user list |
| resources.large_memory | Large memory jobs memory limit (MB) |
| resources.cores | Large memory jobs CPU cores limit |
| trim_config.percent_nongap | Sites with less non-gap fraction than below is removed |
| mainlines_config.n | Target number of backbone taxa in de novo inference |
| mainlines_config.length | concatenation alignment length |
| backbone_filtering | Backbone filtering is recommended if backbone contains misplaced or noisy sequences |
| apples_config.method | APPLES-2 placement mode |
| apples_config.filter | APPLES-2 -f parameter (filter diameter) |
| apples_config.base | APPLES-2 -b parameter (minimum observations) |
| apples_config.overlap | APPLES-2 minimum alignment overlap fraction |
| prep_config.edge_thr | Partition diameter limit |
| prep_config.cluster_size | Approximate partition size. Options are (1) auto, (2) fast, (3) user defined integer. |
| prep_config.sublength | Minimum partition alignment length. If not satisfied, the partition is discarded. |
| prep_config.pruneafter | Maximum partition size |
| prep_config.min_placements | The maximum number of placements occurred for the partition to be skipped to save running time |
| infer_config.method | Gene tree inference method (RAxML-ng, IQTree-2, or RAxML-8) |
| infer_config.numstart | The number of starting trees |
| infer_config.num_threads | The number of threads used in gene tree inference |
| refine_config.contract | Contract low support (IQTree-2 aBayes test) branches |
| refine_config.occupancy | Gene occupancy threshold for sequence inclusion |
| refine_config.outlier_sizelimit | The next two are 1D k-means-based (k=2) outlier gene detection parameter. Limit for outlier size fraction. |
| refine_config.outlier_difference | Centroid difference must be larger than this value to designate the first cluster as the outlier set |
| refine_config.infer_branchlen | Infer branch lengths in substitution unit using ASTRAL |

Table S4: The software packages and their versions used in uDance workflow

| **Software package** | **Version** |
|---|---|
| APPLES-2 | 2.0.11 |
| newick_utils [66] | 1.6 |
| trimal [46] | 1.4.1 |
| gappa [67] | 0.7.1 |
| pandas | 1.3.0 |
| seqkit | 2.1.0 |
| treeshrink | 1.3.9 |
| iqtree | 2.1.2 |
| ASTRID | 2.2.1 |
| upp | 2.0 |
| dendropy [68] | 4.5.2 |
| fasttree | 2.1.10 |
| raxml | 8.2.12 |
| raxml-ng | 1.1.0 |
| scipy | 1.9.0 |
| snakemake | 7.12.0 |

# Supplementary Notes

## 1 Selection of sequences for insertion on the WoL tree

We designed an automated procedure to select new sequences for insertion on the WoL tree. 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. Therefore, we selected the top 50 genes from the marker genes set sorted in increased order based on the quartet distance between their gene tree and the species tree in the WoL dataset. We concatenated the MSAs of the selected genes and placed all novel unique genomes on the WoL tree using APPLES-2. Next, we clustered the tree using TreeCluster [50] with threshold 0.3 and default clustering criterion. We then removed all clusters that include at least one genome from WoL tree and selected one genome with a copy of 16S gene per each remaining clusters, which now entirely consist of novel genomes. When there were many genomes with a 16S gene in a cluster, we broke the ties by choosing the genome with the highest gene occupancy. This automated procedure determined 6,056 sequences for insertion.

## 2 uDance Tree Partitioning Algorithm

**Algorithm 1:** Placement-weighted clustering. Definitions: $B(u)$ to be the length of the path from $u$ to the most distant *connected* descendant node in $v$ such that $C(v) = C(u)$.

**Input:** A tree $T^o = (V, E)$, a threshold $\alpha$, and partition size limit $S$

current_color $\leftarrow 0$
$C(e) \leftarrow -1$ **for** $e \in E$
$B(v) \leftarrow 0$ **for** $v \in V$
$Q(v) \leftarrow 0$ **for** $v \in V$
**for** $u \in$ *post order traversal of internal nodes of* $T^o$ **do**
    $l$,$r \leftarrow$ left and right child of $u$
    **if** $Q(l) + w_l + Q(r) + w_r + w_u \leq S$ ***or***
    $B(l) + b_l + B(r) + b_r < \alpha$ ***or***
    $b_l \leq 10^{-5}$ ***and*** $w_l > 0$ ***or***
    $b_r \leq 10^{-5}$ ***and*** $w_r > 0$ ***or***
    $b_u \leq 10^{-5}$ ***and*** $w_u > 0$ **then**
        $Q(u) \leftarrow Q(l) + w_l + Q(r) + w_r$
        $B(u) \leftarrow max(B(l) + b_l, B(r) + b_r)$
    **else if** $Q(l) + w_l + Q(r) + w_r \leq S$ **then**
        paint($l$,current_color); paint($r$,current_color)
        current_color += 1; $B(u) \leftarrow 0$
    **else if** $min(Q(l) + w_l, Q(r) + w_r) + w_u > S$ **then**
        paint($l$,current_color); paint($r$,current_color+1)
        current_color += 2; $B(u) \leftarrow 0$
    **else if** $Q(l) + w_l \geq Q(r) + w_r$ **then**
        paint($l$,current_color); current_color += 1
        $B(u) \leftarrow B(r) + b_r$
    **else**
        paint($r$,current_color); current_color += 1
        $B(u) \leftarrow B(l) + b_l$
**return** *Clusters of edges defined by their assigned color*
**Function** `paint(`*u,color*`)`**:**
    **for** $v \in$ *descendants of* $u$ **do**
        **if** $u \neq v$ ***and*** $C(v) = -1$ **then**
            $C(v) \leftarrow$ *color*
    **return**

## 3 Proof of claims

**Claim 5.** $R(c) \subseteq O(c) \cup \mathcal{L}(c)$.

*Proof.* Let $x$ be a leaf in $R(c)$. By definition, there exist $o$ such that $x \in O_c(o)$. Let $v$ be the junction node in T where $(u_1, v) \in \mathcal{E}(o)$ and $(u_2, v) \in \mathcal{E}(c)$. When $T$ rooted at $u_1$, $x$ must be descendent of $u_2$. The statement is trivially correct when $x \in \mathcal{L}(c)$. However, $x$ might belong to a different partition, which is a descendent of $c$. Let $j$ be the first edge on the path between $u_2$ and $x$ such that $j \in \mathcal{E}(c')$ and $c \neq c'$. The results follows from the fact that since $x$ is either the closest or the highest occupancy leaf under $u_2$, $x$ must also belong $O_{c'}(c)$.

□

**Claim 6.** *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 $(u_j, v)$.*

*Proof.* By definition of constrained search, $A(c)$ induces the constraint tree. To find $v$, first, find a leaf $x \in R(c)$ in $A(c)$, which is always possible by Claim 1 for both *incremental* and *updates* constraint trees. We root $A(c)$ at $x$ and find the MRCA $\mu$ of $O_o(c)$. Because the topology of $T$ for $O(c)$ is retained in $A(c)$ by constraints, $S(\mu) \cap (R(c) \cup O(c)) = O_o(c)$. Let $\mu'$ be the sibling node of $\mu$. $S(\mu') \cap (R(c) \cup O(c))$ may be empty set. Therefore, we iterate through the ancestors of $\mu$ until we find a node $v$ such that $l(v) \cap (R(c) \cup O(c)) = O_o(c)$ and $r(v) \cap (R(c) \cup O(c)) \neq \emptyset$. This procedure finds the node $v$ that corresponds to $w$ in $T$. In other words, both conditions given above are also satisfied by $w$ in $T$ provided that $o \neq o'$. □

**Claim 7.** *When $o = o'$, the junction node $w$ corresponds to a unique internal node $v$ of the refined tree $A(c)$ such that:*

1. $(S(u_2) \cup S(u_3)) \cap (R(c) \cup O(c)) = O_o(c)$
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.*

*Proof.* Root $A(c)$ similarly and find the MRCA $\mu$ of $O_o(c)$. Because the topology of $T$ for $O(c)$ is retained in $A(c)$ by constraints, $S(\mu) \cap (R(c) \cup O(c)) = O_o(c)$. $\mu$ satisfies the second property as well since if, without loss of generality, $l(\mu) \cap (R(c) \cup O(c)) = \emptyset$ was true, than the MRCA of $O_o(c)$ would be $l(\mu)$. Therefore $\mu = v$ and it corresponds to $w$ in $T$. □

## 4 nRF and QD error computation

We computed normalized Robinson-Foulds distance error using a wrapper script (https://github.com/balabanmetin/compareTrees) based on the `CompareTree.pl` tool developed and made available by the authors of FastTree-2. We computed quartet distance using tqDist [69].

# References


[66] Junier, T. & Zdobnov, E. M. The Newick utilities: high-throughput phylogenetic tree processing in the U ¡scp¿nix¡/scp¿ shell. *Bioinformatics* **26**, 1669–1670 (2010). URL https://academic.oup.com/bioinformatics/article/26/13/1669/200713.

[67] Czech, L., Barbera, P. & Stamatakis, A. Genesis and Gappa: processing, analyzing and visualizing phylogenetic (placement) data. *Bioinformatics* **36**, 3263–3265 (2020). URL https://academic.oup.com/bioinformatics/article/36/10/3263/5722201.

[68] Sukumaran, J. & Holder, M. T. DendroPy: a Python library for phylogenetic computing. *Bioinformatics* **26**, 1569–1571 (2010). URL http://www.ncbi.nlm.nih.gov/pubmed/20421198.

[69] Sand, A. *et al.* tqDist: a library for computing the quartet and triplet distances between binary or general trees. *Bioinformatics* **30**, 2079–2080 (2014). URL https://academic.oup.com/bioinformatics/article/30/14/2079/2390663.