Supplementary Material to ASTRAL: Genome-Scale Coalescent-Based Species Tree Estimation

S. Mirarab [1], R. Reaz [1], Md. S. Bayzid [1], T. Zimmermann [1], M.S. Swenson [2], and T. Warnow [1*]

[1]Department of Computer Science, The University of Texas at Austin, Austin TX 78712
[2]Department of Electrical Engineering, The University of Southern California, Los Angeles CA

Contents

*to whom correspondence should be addressed

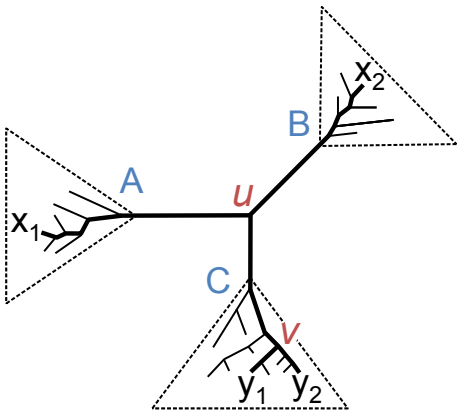


Figure 1: **Mapping induced quartet trees to a tripartition.** Each node u in an unrooted tree defines a tripartition $(A|B|C)$ of the set of taxa. Each induced quartet tree $x_1x_2|y_1y_2$ maps to two tripartitions. Here, we show how the quartet tree on x_1, x_2, y_1, y_2 maps to u and v. Node u is where the paths from x_1 and x_2 to either y_1 or y_2 first join each other. Similarly, node v is where the paths from y_1 and y_2 to either x_1 or x_2 first join each other.

1 ASTRAL Details

Each node u in an unrooted gene tree g divides the set of taxa into three distinct subsets (see Figure 1). We refer to this tripartition of taxa with the $Tri(u) = (A|B|C)$ notation, and refer to each of the three subsets as one "side" of the tripartition. Consider any arbitrary quartet tree $x_1x_2|y_1y_2$ induced by the tree g. The path from x_1 to y_1 and the path from x_2 to y_1 first join each other at some node u. It is easy to show that the paths from x_1 and x_2 to y_2 first join each other at the same exact node u. Thus, x_1 is on one side of $Tri(u)$ and x_2 is on another side, and $\{y_1, y_2\}$ are both on the third side of $Tri(u)$. Similarly, if paths from y_1 and y_2 are followed to either x_1 or x_2, they join each other first at some node v, and as long as the tree is fully resolved, we can easily show that $u \neq v$. Again we can show that y_1 is on one side of the tripartition $Tri(v)$, y_2 is on another side, and x_1, x_2 are on the third side. Thus,

Lemma 1. *For each quartet tree $x_1x_2|y_1y_2$ induced by an unrooted fully resolved (i.e., binary) tree, there are exactly two nodes u and v where $Tri(u)$ has x_1 on one side, x_2 on another side, and y_1, y_2 on the third side; and similarly, $Tri(v)$ has y_1 on one side, y_2 on another side, and x_1, x_2 on the third side.*

A consequence of Lemma 1 is that each quartet tree in each unrooted gene tree can be mapped uniquely to exactly two nodes. Furthermore, given any node u with tripartition $(A|B|C)$, the number of quartet trees that are mapped to u can be easily counted by choosing two leaves from one side, and one leaf from each remaining side. Therefore, the number of quartet trees mapped to a node u can be counted using

$$F(a,b,c) = \binom{a}{2}\binom{b}{1}\binom{c}{1} + \binom{a}{1}\binom{b}{2}\binom{c}{1} + \binom{a}{1}\binom{b}{1}\binom{c}{2} = \frac{abc(a+b+c-3)}{2}$$

where $Tri(u) = (A|B|C)$ and $a = |A|$, $a = |B|$, $a = |C|$. We consider a quartet tree to be mapped to a tripartition $(A|B|C)$ if and only if two of its leaves are in one side, and each of the other two leaves are in different sides (equivalently, all sides of the tripartition have a non-empty intersection with the quartet tree leaf set).

Using this result we can now define a dynamic programming algorithm to calculate a tree that satisfies the maximum number of quartet trees from a given set of input trees, subject to the constraints that all bipartitions of the output tree should come from an input set $\mathcal{X}$. Importantly, we can do this without even enumerating the set of all quartet trees and calculating their weights.

Support of quartet tree q for tree T_A: We begin by defining the support of a quartet tree q for a rooted binary tree T_A on taxon set A. Let $\mathcal{I}$ denote the set of all quartet trees induced by all the gene trees, and for each $q \in \mathcal{I}$, we define $weight(q)$ to be the number of gene trees that induce q (i.e., the number of times q appears in any of the input gene trees). Then, for a given rooted subtree T_A on clade A, we define $m(q, T_A)$ to be 0 if q cannot be induced by any tree that contains T_A as a subtree or if $|q \cap A| \leq 1$, and otherwise to be the number of nodes in T_A that q maps to (i.e., either 1 or 2). We define the total support of the tree T_A from quartet tree q to be the product of $m(q, T_A) \cdot weight(q)$, and denote this by $c(q, T_A)$.

Computing $C(A)$, the total support of the best tree T_A: We set

$$C(A) = \max_{T_A}\{\sum_{q \in \mathcal{I}} c(q, T_A)\},$$

where T_A is a rooted binary tree on taxon set A. That is, $C(A)$ is the largest total value that can be obtained for any rooted tree on A.

To compute this using dynamic programming, suppose T_A is a rooted tree on clade A, in which the root of T_A has major subcluster A' (thus, the

root of T_A has a child defining cluster A', and another child defining cluster $A-A'$). Suppose that T_A is optimal among all rooted trees on A that contain A' as a rooted subclade off the root. Now consider the total support of all quartet trees to T_A. Each such quartet tree that contributes support maps either to one or two nodes in T_A. Note that this total support is obtained by $C(A') + C(A - A') + W(A'|A - A'|S - A)$, where $W(A'|A - A'|S - A)$ is the total weight of all quartet trees that map to the root of T_A (i.e., to the tripartition $(A'|A - A'|S - A)$).

Dynamic Programming Algorithm: Thus, the dynamic programming algorithm is given by:

$$C(A) = \max_{A' \subset A; A' \in \mathcal{X}^*} (C(A') + C(A - A') + W(A'|A - A'|S - A))$$

where $\mathcal{X}^*$ is the set of clusters formed by taking halves of the bipartitions in $\mathcal{X}$. We set the boundary condition $C(\{x\}) = 0$.

Preprocessing: The preprocessing involves computing all allowed tripartitions, and then computing $W(N)$ for every allowed tripartition N, where $W(N)$ is the number of quartet trees in the gene trees that map to the tripartition $N = (X|Y|S - X - Y)$. To count these, we need to look at each individual tripartition from each individual gene tree, and count how many of its quartet trees are shared with the quartet trees defined by tripartition N.

In general, let $M = (A_1|A_2|A_3)$ and $M' = (B_1|B_2|B_3)$; we will show how to calculate the number of quartet trees in common between these two tripartitions. First, we calculate $n_{ij} = |A_i \cap B_j|$ for all i, j between 1 and 3.

We let $QI(M, M')$ denote the number of quartet trees that map to both tripartitions M and M", and note that

$$QI(M, M') =$$

$$F(n_{11}, n_{22}, n_{33}) + F(n_{11}, n_{23}, n_{32}) +$$
$$F(n_{12}, n_{21}, n_{33}) + F(n_{12}, n_{23}, n_{31}) +$$
$$F(n_{13}, n_{21}, n_{32}) + F(n_{13}, n_{22}, n_{31})$$

Now, letting N denote the tripartition $(A|A-A'|S-A)$ and $Tri(g)$ denote the set of tripartitions of a gene tree g, we can calculate W as

$$W(N) = \sum_{g \in G} \sum_{M \in Tri(g)} QI(N, M)$$

Note that for a rooted tree T on the full set of species S, there is no tripartition defined at the root. Hence, we define its total support using $C(S') + C(S - S')$, where $S' \subset S$ is the subcluster of one child of the root. Thus, $C(S) = \max_{S' \subset S}[C(S') + C(S - S')]$. Equivalently, we set $W(A|S - A|\emptyset) = 0$.

Finally, every quartet tree that supports a binary tree maps to two nodes in the tree, and so the best MQSST score that can be achieved is half of $C(S)$.

Running time analysis.

Theorem 1. *ASTRAL runs in $O(n^2kx^2)$ time, where n is the number of species, k is the number of gene trees, and x is the number of bipartitions in $\mathcal{X}$.*

Proof. The initialization involves computing the set of allowed tripartitions, and then the weight of each such allowed tripartition. The input set $\mathcal{X}$ contains just bipartitions, and so computing the set of allowed tripartitions is straightforward, taking $O(nx^2)$ time, and producing a set of $O(x^2)$ allowed tripartitions. To compute the weight for a given allowed tripartition, it needs to be compared to every triparition from the input set of gene trees, and each comparison takes $O(n)$ time. There are $O(nk)$ tripartitions from the input set of k gene trees, and so this step takes $O(n^2kx^2)$ time.

The algorithm then computes $C(A)$ for every $A \in \mathcal{X}^*$, but to do so it looks at every $A' \in \mathcal{X}^*$ that is a subset of A, and then performs $O(1)$ operations; therefore, this step takes $O(x)$ time per element of $\mathcal{X}^*$. Hence, to compute all $C(A)$ for all $A \in \mathcal{X}^*$ takes $O(x^2)$ time, once the weights of all tripartitions are computed. Hence the total time is $O(x^2 + n^2kx^2) = O(n^2kx^2)$. □

Corollary 1. *If ASTRAL is run in default mode, then $\mathcal{X}^*$ contains only the distinct clades in the input gene trees, and so $|\mathcal{X}^*|$ is $O(kn)$. Hence, in default mode ASTRAL runs in $O(n^4k^3)$ time.*

However, the running time is better expressed as a function of the number of *distinct* tripartitions among its input gene trees and inferred allowed tripartitions. Therefore, when the amount of ILS is very low, then the number of distinct tripartitions in the input gene trees is mostly impacted by estimation error rather than true gene tree conflict, and can be quite small; in such cases, ASTRAL can be very fast. But when ILS is high, or gene tree estimation error is high (for example, due to insufficient phylogenetic signal in the gene trees), then ASTRAL needs to compute weights for a larger set of tripartitions, and hence is slower.

2 Further Results

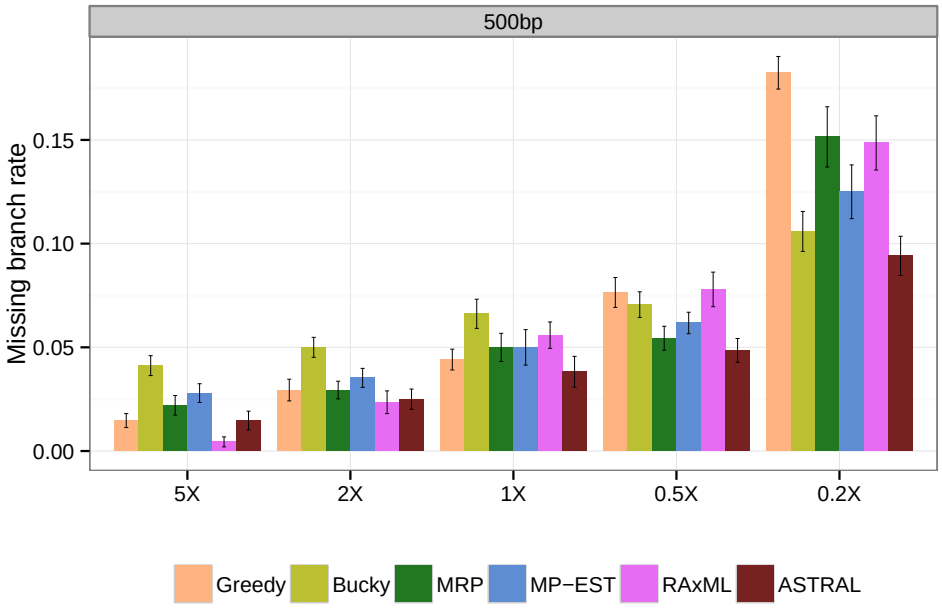


Figure 2: **Species tree estimation error on the simulated mammalian datasets with 37 genes and 200 genes with 500bp length, showing BUCKy-pop.** We show the missing branch rates for estimated species trees computed using summary methods (MRP, MP-EST, greedy, BUCKy-pop, and ASTRAL) as well as concatenation using RAxML maximum likelihood on the mammals simulated datasets of 200 genes of 500bp length, and varying ILS level. The summary methods are run on RAxML gene trees, BUCKy-pop is run on RAxML bootstrap gene trees. and CA-ML is run on the true alignments. Average and standard error shown based on 20 replicates.

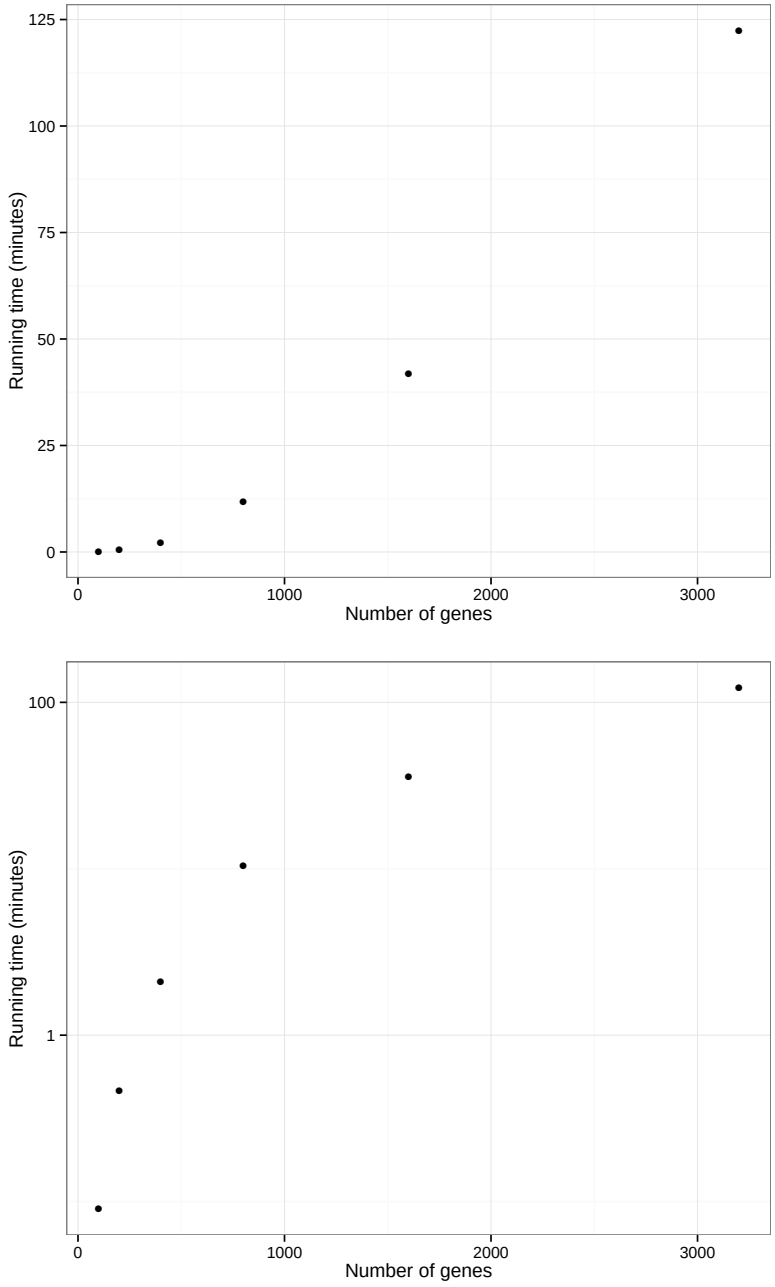


Figure 3: **Running time of ASTRAL as a function of the number of genes.** We show the running time for default ASTRAL on the mammals simulated datasets with varying number of true gene trees with much increased ILS level (0.2X). Top: normal scale; Bottom: log scale.

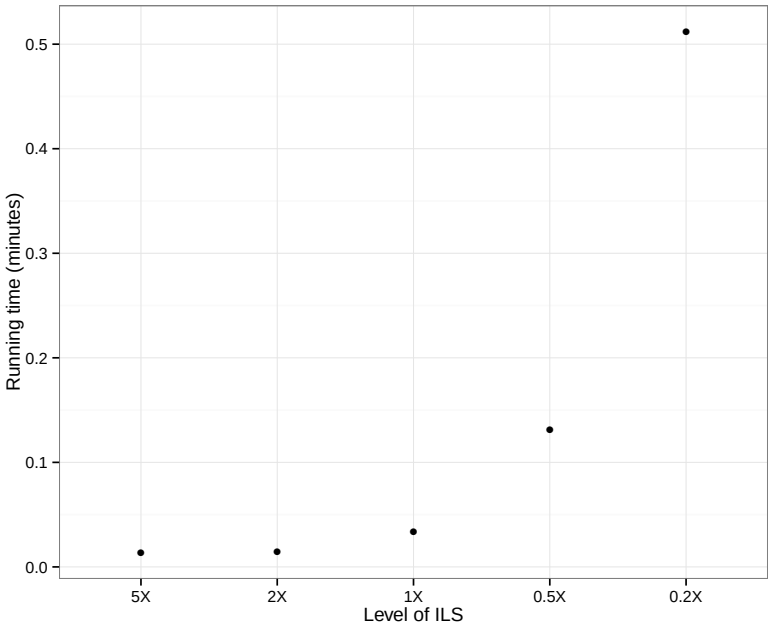


Figure 4: **Running time of ASTRAL as a function of the amount of gene tree incongruence** We show the running time for default ASTRAL on the mammals simulated datasets with varying levels of ILS with 200 genes of 500bp resolution. The running time of ASTRAL increases as the level of ILS is increased, because the set $\mathcal{X}$ is populated with more bipartitions when gene trees have high levels of ILS.

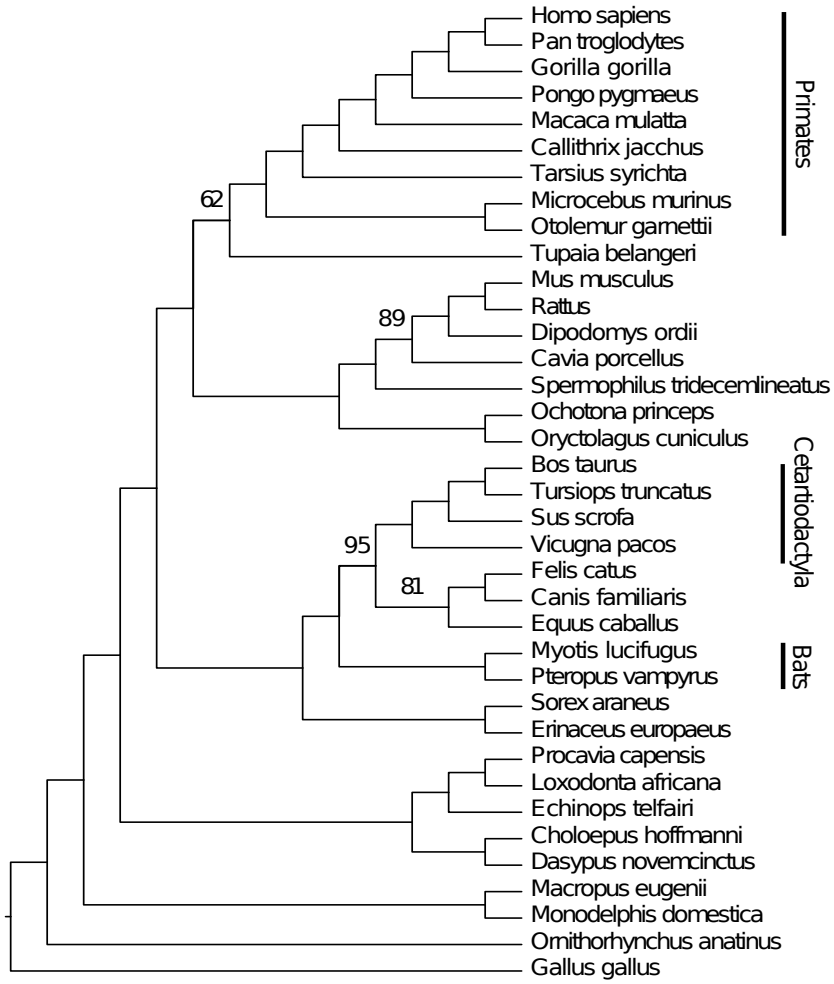


Figure 5: **MP-EST tree on the Song et al. dataset [3].** We show results of analyzing the biological mammalian dataset with 37 taxa and 424 genes using MP-EST. The original dataset had 447 genes; we removed 21 mislabelled genes and 2 genes that we found to be outliers. Bootstrap support values were obtained using the multi-locus bootstrapping procedure with site and gene re-sampling and 100 replicates (gene resampling is used to reproduce the procedure from Song et al). Our MP-EST tree is different from the one reported by Song et al. in the support obtained for the sister relationship of tree shrews and the primates (our tree recovers 62%, while [3] reports support above 95%). The exact cause of this difference is not clear to us. Branches without designation have 100% support.

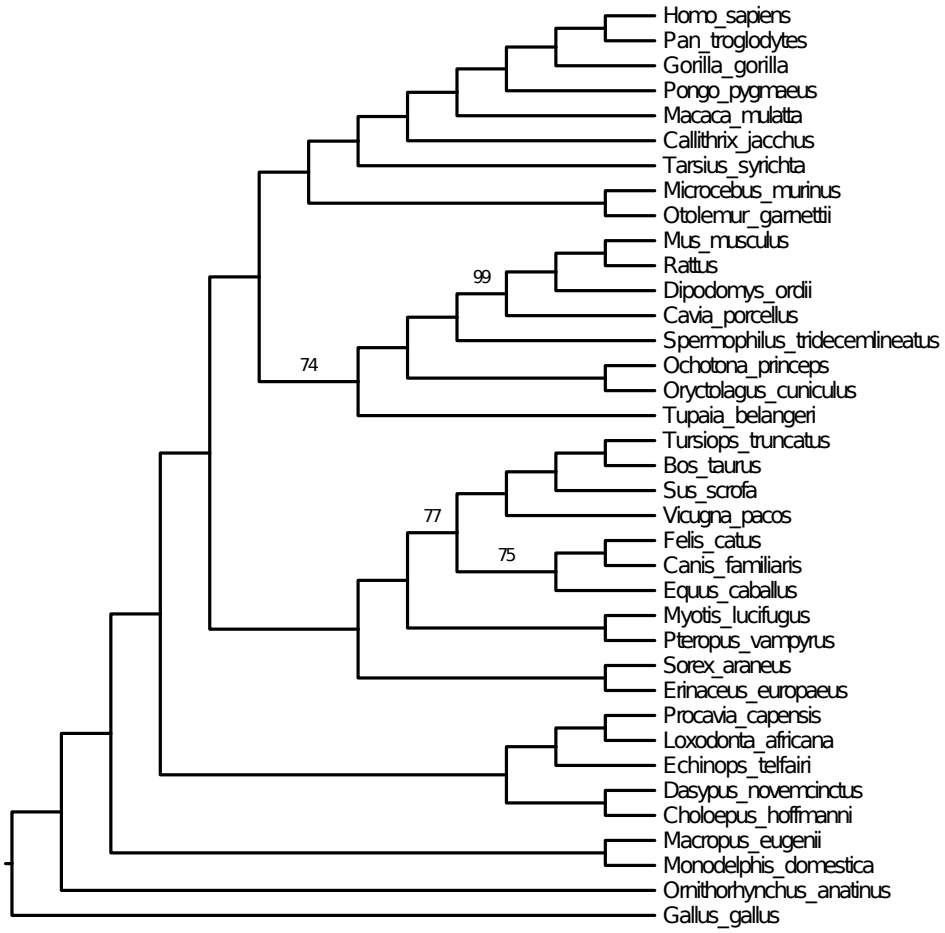


Figure 6: **ASTRAL tree on the Song et al. dataset [3].** We show results of analyzing the mammalian dataset with 37 taxa and 424 genes using ASTRAL. The original dataset had 447 genes; we removed 21 mis-labelled genes and 2 genes that we found to be outliers. Bootstrap support values were obtained using the multi-locus bootstrapping procedure with site and gene re-sampling and 100 replicates (gene resampling is used to reproduce the procedure from Song et al.) Branches without designation have 100% support.

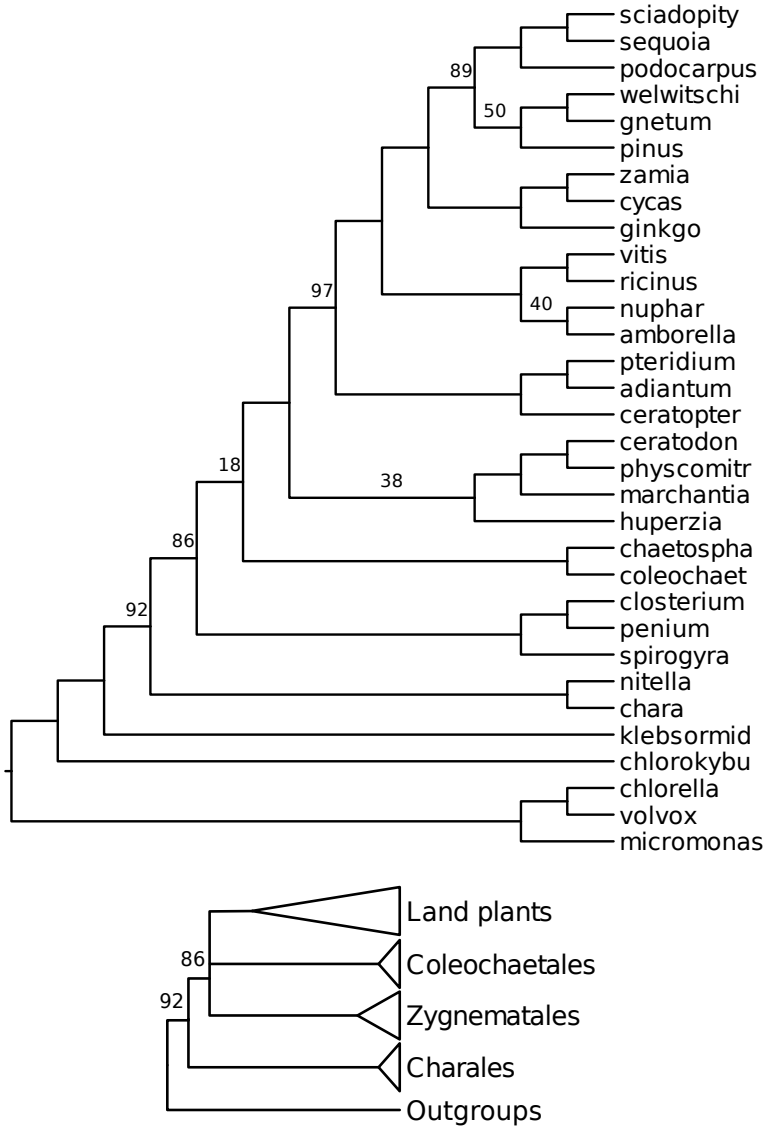
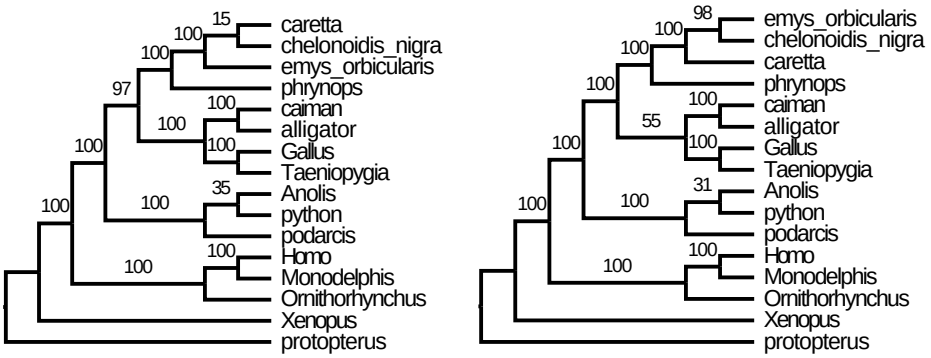


Figure 7: **ASTRAL tree on Zhong et al. dataset [7].** We analyzed a plant dataset with 32 species and 184 genes from [7]. Bootstrap support values were obtained using the multi-locus bootstrapping procedure with 100 replicates; values not shown indicate 100% support. The rooted ASTRAL tree (with bootstrap support values) is shown on top, and we show a cartoon version of the tree below. The cartoon version only shows the relationship between the 5 groups – land plants, Coleochaetales, Zygnematales, Charales, and the outgroups, after collapsing the branch with bootstrap support of 18%. Note that there are three possible sister groups to land plants: Coleochaetales, Zygnematales, or the two together (Zygnematales+Coleochaetales); however, Charlaes is strongly rejected as the sister group to land plants.



(a) ASTRAL on AA gene trees

(b) ASTRAL on DNA gene trees

Figure 8: **ASTRAL trees on Chiari et al. dataset [1].** We analyzed an Amniota dataset with 16 species and 248 genes from [1] using the exact version of ASTRAL on both AA and DNA gene trees. Bootstrap support values have been obtained using the multi-locus bootstrapping procedure with 100 replicates. The two trees are highly similar (differing topologically only in one very low support branch), but display some some differences in bootstrap support.

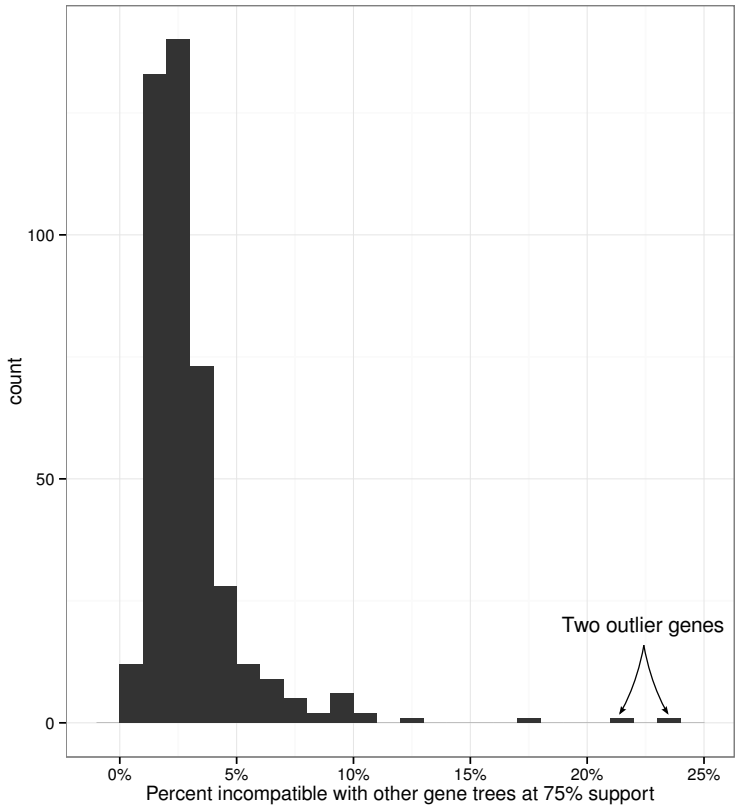


Figure 9: **Outlier mammalian genes.** The histogram shows the distribution of average percentage of branches that each gene had in conflict with other gene trees with at least 75% support. Two genes, with IDs 232 and 209, had on average more than 20% of their edges in conflict with other gene trees with bootstrap support higher than 75%. We suspect these two gene trees have undetected problems. Hence, we removed these two genes from the dataset (in addition to 21 mis-labelled genes).

Method	bestML	All BS
CA-ML	0.057	
ASTRAL	0.061	**0.052**
GC	0.064	0.056
MRP	0.064	0.055

Table 1: Average FN rates (over 20 replicates) of different methods on the 100-taxon 25-gene simulated datasets.

3 Experimental Details

3.1 Extra trees for Zhong *et al.* biological dataset

Many of the gene trees in the Zhong *et al.* biological dataset were incomplete. In this kind of input, the default setting for $\mathcal{X}$ to be the bipartitions defined by gene trees is insufficient. We therefore used bipartitions drawn from the following set of estimated trees to add to the set $\mathcal{X}$: the MP-EST tree from [7], the STAR tree from [4], 50 bootstrap trees from concatenation with RAxML (we computed), 50 parsimony trees on the concatenated alignments (we computed), and 100 bootstrap MRL [2] trees (we computed). (See below for information about MRL.)

For the mammals dataset, since all gene trees were complete, we did not need to add extra bipartitions to the set $\mathcal{X}$.

3.2 Methods and Commands

3.2.1 Gene tree estimation

RAxML version 7.3.5 [5] was used to estimate gene trees. The following command was used for estimating the best ML trees.

```
 raxmlHPC-SSE3 -m GTRGAMMA -s [input_file] -n [a_name]
-N 20 -p [random_seed_number]
```

The following command was used for bootstrapping.

```
 raxmlHPC-SSE3 -m GTRGAMMA -s [input_file] -n [a_name] -N 200
-p [random_seed_number] -b [random_seed_number]
```

3.2.2 ASTRAL

We ran version 3.1.1 of ASTRAL (corresponding to the github commit `fb21c0ce6140e9e238575356bc174c88c6cfc597` from March 6th on `https://github.com/smirarab/ASTRAL` with the following command:

```
java -jar astra_3.1.1.jar -wq -in [input_tree]
```

Where the exact version of ASRAL was used, we ran it with the following command:

```
java -jar astra_3.1.1.jar -wq -in [input_tree] -xt
```

To add new bipartitions to $\mathcal{X}$, we used it with the following command:

```
java -jar astra_3.1.1.jar -wq -in [input_tree] -ex [extra_tree_files]
```

3.2.3 BUCKy-pop

We ran BUCKy with the default settings, except for the number of generations that we changed from 100K to one million. The following command was used to run BUCKy.

```
bucky -n <numberOfGenerations> -o <outputFileRoot> <inputFiles>
```

3.2.4 MRP and MRL

MRP trees are built using a custom Java program available at `https://github.com/smirarab/mrpmatrix`. The following command was used to create the MRP matrix.

```
java -jar mrp.jar [input_file] [output_file] NEXUS
```

We used the default heuristic in PAUP* (v. 4. 0b10) [6] for maximum parsimony. This heuristic first generates an initial tree through random sequence addition and then uses Tree Bisection and Reconnection (TBR) moves to reach a local optimum. This process is repeated 1000 times, and the most parsimonious tree is returned. When multiple trees have the same maximum parsimony score, the greedy consensus of those trees is returned. The following shows the PAUP* commands used.

```
begin paup;
set criterion=parsimony maxtrees=1000
increase=no;
hsearch start=stepwise addseq=random
nreps=100 swap=tbr;
filter best=yes;
savetrees file = <treeFile> replace=yes
format=altnex;
contree all/ strict=yes
treefile = <strictConsensusTreeFile>
replace=yes;
tcontree all/ majrule=yes strict=no
treefile = <majorityConsensusTreeFile>
replace=yes;
contree all/ majrule=yes strict=no
le50=yes
treefile = <greedyConsensusTreeFile>
replace=yes;
log stop;
quit; end;
```

MRL stands for "Matrix Representation with Likelihood", and is the supertree

method obtained by running two-state symmetric maximum likelihood on the MRP matrix [2]. We computed maximum likelihood trees on the same MRP matrix using RAxML under the two-state maximum likelihood model, to obtain MRL (matrix representation with likelihood) trees.

3.2.5 Concatenation

We used RAxML version 7.3.5 to create the parsimony starting trees:

```
raxmlHPC-SSE3 -y -s supermatrix.phylip -m GTRGAMMA
-n [a_name] -p [random_seed_number] -s [alignment]
```

We then used RAxML-light version 1.0.6 with the following command to search for the ML tree.

```
raxmlLight-PTHREADS -T 4 -s supermatrix.phylip -m GTRGAMMA -n name
-t [parsimony_tree] -s [alignment]
```

References

[1] Y Chiari, V Cahais, N Galtier, and F Delsuc. Phylogenomic analyses support the position of turtles as the sister group of birds and crocodiles (archosauria). *BMC Biology*, 10:65, 2012.

[2] N. Nguyen, S. Mirarab, and T. Warnow. MRL and SuperFine+MRL: new supertree methods. *J. Algorithms for Molecular Biology*, 7(3), 2012.

[3] S Song, L Liu, S V Edwards, and S Wu. Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model. *Proceedings of the National Academy of Sciences of the United States of America*, 109(37):14942–14947, 2012.

[4] M S Springer and J Gatesy. Land plant origins and coalescence confusion. *Trends in Plant Science*, 19(5):267–269, 2014.

[5] A. Stamatakis. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.

[6] D.L. Swofford. *PAUP*: Phylogenetic analysis using parsimony (* and other methods). Ver. 4.* Sinauer Associates, Sunderland, Massachusetts, 2002.

[7] B Zhong, L Liu, Z Yan, and D Penny. Origin of land plants using the multispecies coalescent model. *Trends in Plant Science*, 18(9):492–495, 2013.