Syst. Biol. 61(2):214–227, 2012

DOI:10.1093/sysbio/syr092
Advance Access publication on September 20, 2011

SuperFine: Fast and Accurate Supertree Estimation

M. SHEL SWENSON[1,*], RAHUL SURI[1], C. RANDAL LINDER[2], AND TANDY WARNOW[1]

[1]*Department of Computer Science, The University of Texas at Austin, Austin, TX, USA; and*
[2]*Section of Integrative Biology, School of Biological Sciences, The University of Texas at Austin, Austin, TX, USA;*
[*]*Correspondence to be sent to: The University of Texas at Austin, 1 University Station C0500, Austin, TX, USA; E-mail: shelswenson@gmail.com.*

Received 2 August 2010; reviews returned 11 November 2010; accepted 26 May 2011
Associate Editor: Tiffani Williams

Abstract.—Many research groups are estimating trees containing anywhere from a few thousands to hundreds of thousands of species, toward the eventual goal of the estimation of a Tree of Life, containing perhaps as many as several million leaves. These phylogenetic estimations present enormous computational challenges, and current computational methods are likely to fail to run even on data sets in the low end of this range. One approach to estimate a large species tree is to use phylogenetic estimation methods (such as maximum likelihood) on a supermatrix produced by concatenating multiple sequence alignments for a collection of markers; however, the most accurate of these phylogenetic estimation methods are extremely computationally intensive for data sets with more than a few thousand sequences. Supertree methods, which assemble phylogenetic trees from a collection of trees on subsets of the taxa, are important tools for phylogeny estimation where phylogenetic analyses based upon maximum likelihood (ML) are infeasible. In this paper, we introduce SuperFine, a meta-method that utilizes a novel two-step procedure in order to improve the accuracy and scalability of supertree methods. Our study, using both simulated and empirical data, shows that SuperFine-boosted supertree methods produce more accurate trees than standard supertree methods, and run quickly on very large data sets with thousands of sequences. Furthermore, SuperFine-boosted matrix representation with parsimony (MRP, the most well-known supertree method) approaches the accuracy of ML methods on supermatrix data sets under realistic conditions. [Algorithms; maximum likelihood; MRP; phylogenetics; simulation; supertrees.]

Reconstruction of phylogenetic trees presents substantial computational difficulties. High-throughput sequencing projects have enabled the collection of data for many sets of species, but joint analysis of these data can present certain practical difficulties due to the number of taxa involved and the amount of sequence data. In some cases, alignments of multiple-gene data sets for overlapping sets of taxa can be concatenated into a single supermatrix (where the sequences that are missing for some taxa are coded as missing data), and this supermatrix can then be analyzed using phylogenetic estimation methods such as maximum likelihood (ML). Although progress has been made on developing very fast ML heuristics for 10^4 or even 10^5 taxa (Price et al. 2010), the most accurate of the ML methods, RAxML (Stamatakis 2006) and GARLI (Zwickl 2006), are much slower and therefore cannot analyze data sets with many tens of thousands of sequences without extensive use of supercomputers. In addition, accurately aligning a large number of sequences is itself a computationally intensive problem (NP-hard in some formulations; Wang and Jiang 1994), and the most accurate methods are unable to run on very large data sets (Liu et al. 2010).

Supertree methods construct trees from smaller trees for overlapping subsets of the taxa. These are an appealing alternative to supermatrix analyses for large data sets because they do not require phylogenetic analysis of a large sequence alignment. Many supertree methods have been developed, see Bininda-Emonds (2004) for an overview of early methods, and also Baum and Ragan (2004), Burleigh et al. (2004), Chen et al. (2006), Cotton and Wilkinson (2007), Steel and Rodrigo (2008), Bansal et al. (2009), Ranwez et al. (2010), and Swenson et al. (2010a, 2010b). Of these methods, matrix representation with parsimony (MRP; see Baum 1992; Ragan 1992), is by far the most frequently used. We note that MRP is NP-hard (Foulds and Graham 1982), and so methods for MRP are based upon heuristics and are not guaranteed to produce optimal solutions.

Studies comparing different supertree methods have found that MRP and some other supertree methods, for example, Minflip (Chen et al. 2006), Quartets MaxCut (QMC) (Snir and Rao 2010), and Quartet Imputation (Holland et al. 2007), produce reasonably accurate trees. However, only MRP is both highly accurate and capable of being run successfully on data sets containing more than a few hundred taxa (Swenson et al. 2010a, 2010b). Thus, MRP is a popular supertree method that can run on large data sets and that has been shown to produce trees that match or improve upon the accuracy of other supertree methods. However, MRP can return supertrees that have relationships that are contradicted by all the input trees (Bininda-Emonds and Bryant 1998; Pisani and Wilkinson 2002; Bininda-Emonds 2003; Wilkinson et al. 2004, 2005), a property that is clearly undesirable. In addition, MRP is not statistically consistent, in the sense that for some distributions on input (source) trees, MRP is not guaranteed to converge to the true supertree as the number of source trees increases (Steel and Rodrigo 2008). Thus, although MRP has generally outperformed other supertree methods in terms of topological accuracy and/or scalability, it fails to have other desirable properties.

The major alternative to supertree methods is combined analysis (also known as "supermatrix analysis"). In a combined analysis, alignments for different markers are concatenated, and a phylogeny is then estimated on the resultant supermatrix. Under the assumption that all

markers evolve down the same tree (albeit potentially with different branch lengths), Swenson et al. (2009) and Swenson, Barbançon, et al. (2010) showed that combined analysis using ML produces more accurate trees than MRP and other supertree methods. Therefore, development of fast supertree methods that match, or at least more nearly approach, the accuracy of combined analyses would be very useful for phylogenetic inference on large numbers of taxa and in assembling the tree of life.

With this set of issues in mind, we have developed SuperFine, a new approach for supertree inference. Because SuperFine is designed to work with any existing supertree method, it is a "meta-method" (Huson, Nettles, et al. 1999; Huson, Vawter, et al. 1999; Warnow et al. 2001; Roshan et al. 2004a, 2004b; Moret et al. 2005; Warnow 2006). Generally, meta-methods are algorithms that work with any arbitrary "base" method for the problem they are designed to solve. SuperFine has two steps. In the first step, it produces an initial, incompletely resolved supertree, using an existing method called the strict consensus merger (SCM) (Huson, Nettles, et al. 1999; Roshan et al. 2004b) (an extension of the strict consensus tree method of Day 1985), applied to two trees at a time until all the trees are merged into a single tree. The second step refines the SCM tree using the base supertree method and the input source trees.

We have tested SuperFine with two different base supertree methods, MRP and QMC. Thus, SuperFine+MRP refers to SuperFine used with MRP in the resolution step, and similarly SuperFine+QMC refers to SuperFine used with QMC in the resolution step. SuperFine+MRP is, in a sense, a heuristic for the MRP optimization problem; however, unlike general MRP heuristics, it constrains the search to only those trees that refine the SCM tree computed in the first step. Thus, SuperFine+MRP searches for solutions to MRP but only in the space of trees that refine the SCM tree. Similarly, SuperFine+QMC is a heuristic for the quartet satisfiability optimization problem, with the search also constrained to refinements of the SCM tree. This two-step approach ensures that the tree it returns will contain at least those splits that are present in the SCM tree it computes, and reduces the time taken to find its solution.

Our extensive study on large simulated data sets shows that SuperFine+MRP and SuperFine+QMC (SuperFine based upon MRP and QMC, respectively) yield more accurate supertrees compared with MRP or QMC alone. We also compare SuperFine+MRP and SuperFine+QMC to a number of other supertree methods alone (i.e., without SuperFine), and find that none of those other methods produces supertrees that are as accurate as those produced by the SuperFine-enhanced versions of MRP and QMC. Both SuperFine+MRP and SuperFine+QMC are reasonably efficient, but SuperFine+MRP runs particularly quickly (finishing in well under an hour on all biological data sets inputs we studied, and in 3 h on the 1000 taxon simulated data sets), and can analyze larger data sets than can SuperFine+QMC. Finally, we also show that both SuperFine+MRP and SuperFine+QMC approach the accuracy of combined analysis using ML under realistic model conditions.

Materials and Methods

The SuperFine Meta-method

The input to the SuperFine algorithm is a set of phylogenetic trees (called "source trees") for overlapping subsets of the full set of taxa for which a supertree is to be reconstructed, and the base supertree method. Source trees need only include a topology (i.e., branch lengths are not required), and they do not have to be fully resolved or rooted.

SuperFine uses a novel two-stage algorithmic strategy: the first stage produces a very conservative and typically highly unresolved estimate of the supertree, and the second stage uses the given base supertree method and the source trees to refine the tree (Fig. 1).

Stage 1: Strict Consensus Merger.—SuperFine's first stage merges the source trees, two at a time, using the SCM technique. SuperFine is using DendroPy (Sukumaran and Holder 2010) for the strict consensus merger (as well as for other tree operations). The SCM of two trees first contracts any branch in either tree that conflicts with the other tree and then superimposes the trees, contracting additional branches if there is ambiguity about how to superimpose the trees (Fig. 2 and Appendix). Therefore, the branches in the SCM tree are supported by at least one tree and are not contradicted by either of the trees. After merging the two source trees into one tree, we repeat the process on a new pair of source trees until all the source trees have been merged into one tree.

Although the SCM of two trees is deterministic, the order in which three or more trees are merged can affect the resultant supertree. We experimented with four different rules for picking the next pair of trees to be merged, including three rules that use the number of taxa in common between two trees, and one rule that uses the number of taxa unique to one or the other tree. We found that the three methods that focus on maximizing (in various ways) the number of shared taxa gave better results than the method that tried to minimize the number of unique taxa. Differences in outcomes between the three methods that sought to maximize the number of shared taxa were rather small. We therefore picked one—maximum backbone number—as our criterion. This method computes the number of taxa in common between every pair of trees, and merges the two that have the largest intersection. If there are ties, then the first pair found that achieves the maximum is merged. For more details, see Appendix and Swenson (2008).

Stage 2: Refining polytomies.—The next stage iterates over the polytomies in the SCM supertree, resolving each polytomy based on the topologies of the source trees, and using the given base supertree method. Let $\mathcal{T}$

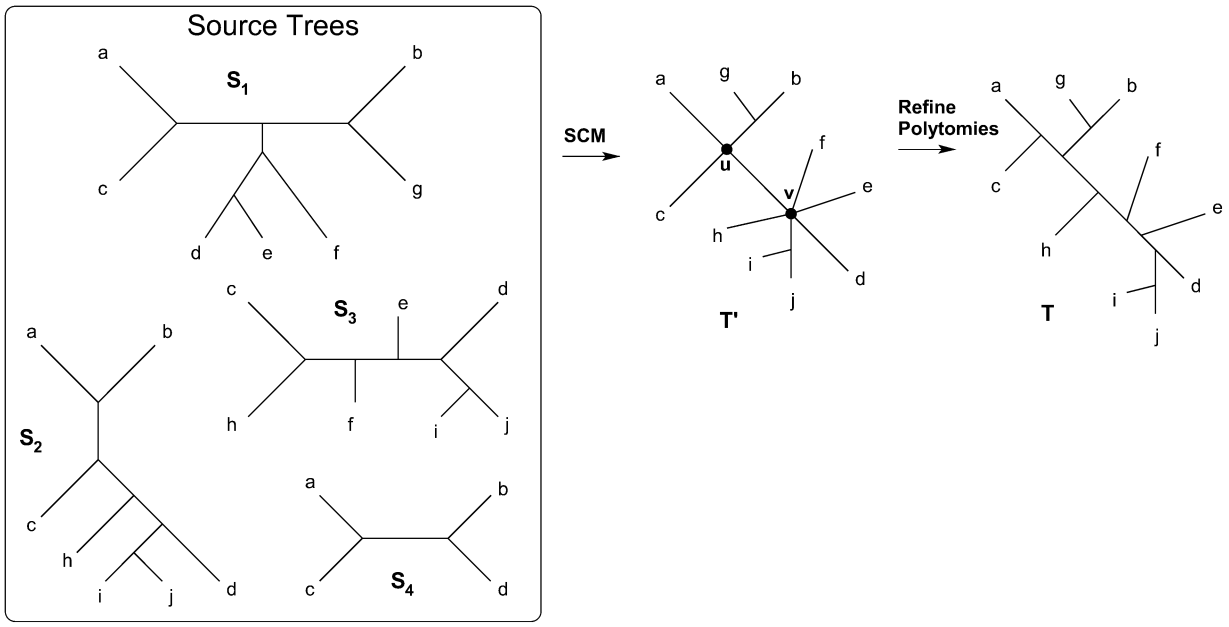

FIGURE 1. Schematic representation of the algorithmic strategy of SuperFine+MRP. Source trees S_1–S_4 are combined pairwise to produce an SCM tree, which retains only internal branches that are compatible with all of the source trees. Each polytomy in the SCM tree is then refined by running MRP on modified source trees (see text).

be a set of source trees, let T be an SCM supertree on $\mathcal{T}$, and let $L(T)$ denote the taxon-set of T. Let v be a node of degree d in T such that $d \geq 4$ (i.e., v is a polytomy of T). The polytomy v is refined, producing a tree T' that is a refinement of T, using the following procedure.

1) Root T at v, and let $v_1, \ldots, v_d$ be the children of v and $T_1, \ldots, T_d$ be the subtrees rooted at $v_1, \ldots, v_d$, respectively (Fig. 3a).
2) Compute a set $\mathcal{T}_v$ of re-encoded source trees based on $T_1, \ldots, T_d$. Let $\phi : L(T) \to \{1, \ldots, d\}$ be defined by $\phi(x) = i$ for $x \in L(T_i)$. Note that for every $x \in L(T)$, x is a taxon in exactly one of these d subtrees; thus, ϕ is well defined. Using this mapping, relabel the taxa of the source trees (Fig. 3b). Then, for each source tree, recursively delete any sibling pairs (or groups of sibling taxa if a source tree is nonbinary) that share label l and attach a single

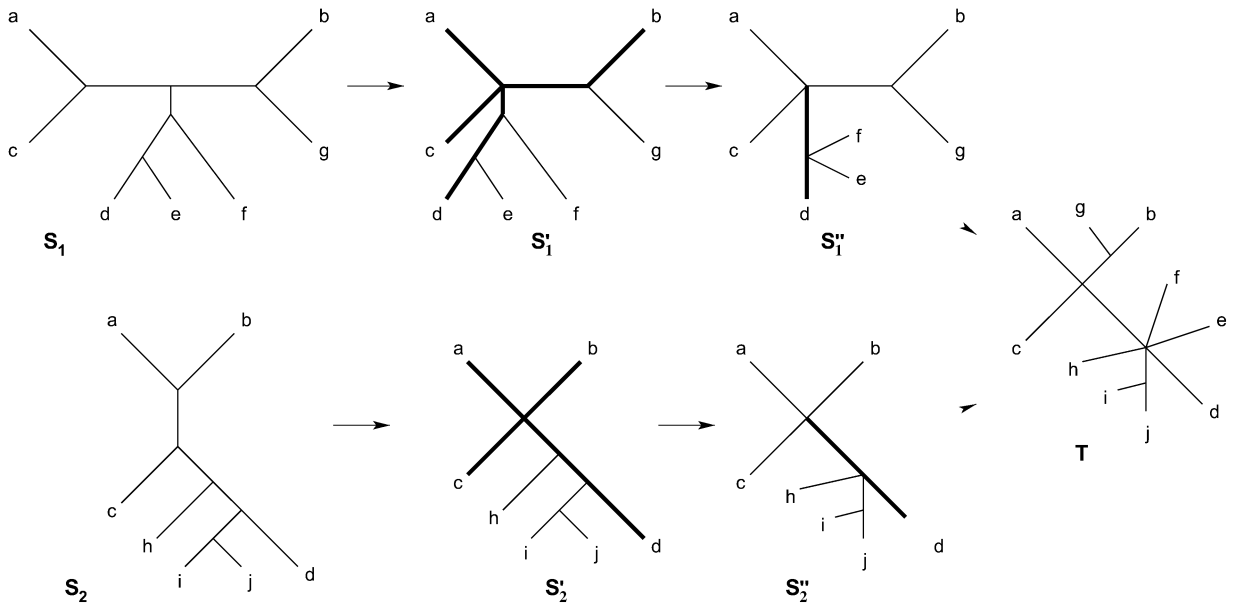

FIGURE 2. SCM of two trees S_1 and S_2. In S_1' and S_2', the strict consensus of S_1 and S_2 restricted to their common taxon set is shown in bold. In S_1'' and S_2'', the branches that are involved in a collapsing of a path in S_1' or S_2' are shown in bold. T is the SCM tree of S_1 and S_2.

taxon labeled l at the node where the siblings were attached (Fig. 3c). By Theorem 1.4 (Appendix), applying this process to any source tree in $\mathcal{T}$ will result in a re-encoded tree with at most one taxon with each label. Thus, each member of $\mathcal{T}_v$ is a phylogenetic tree whose taxon-set is a subset of $\{1, \ldots, d\}$.

3) Apply the base supertree method to $\mathcal{T}_v$ to obtain a tree T^* taxon labeled by the set $\{1, \ldots, d\}$ (see Figs. 3d and 3e for this step using MRP as the base supertree method).

4) Construct T' by attaching each T_i onto T^*, replacing taxon i in T^* with T_i, for each $i \in \{1, \ldots, d\}$ (Fig. 3f).

A few points are worth noting about this technique. First, the order in which polytomies are refined does not impact the outcome of the algorithm. Second, because the supertree method for resolving polytomies is applied to profiles of re-encoded source trees, each of which has at most one taxon with each label, each (Stage 2, Step 3) supertree analysis is performed on source trees with at most d taxa, where d is the maximum degree of any node in the SCM tree produced in the first stage. As a result, the running time of the refinement step is largely determined by the maximum degree of any node in the SCM tree. When that maximum degree is not too large, the refinement step can run quite quickly, *even* when the base supertree method used to resolve the polytomies is generally computationally intensive. Finally, Theorem 1.1 (Appendix) states that the SCM tree never has relationships that are violated by *any* source tree. Thus, the SuperFine method begins with a supertree that has good theoretical properties.

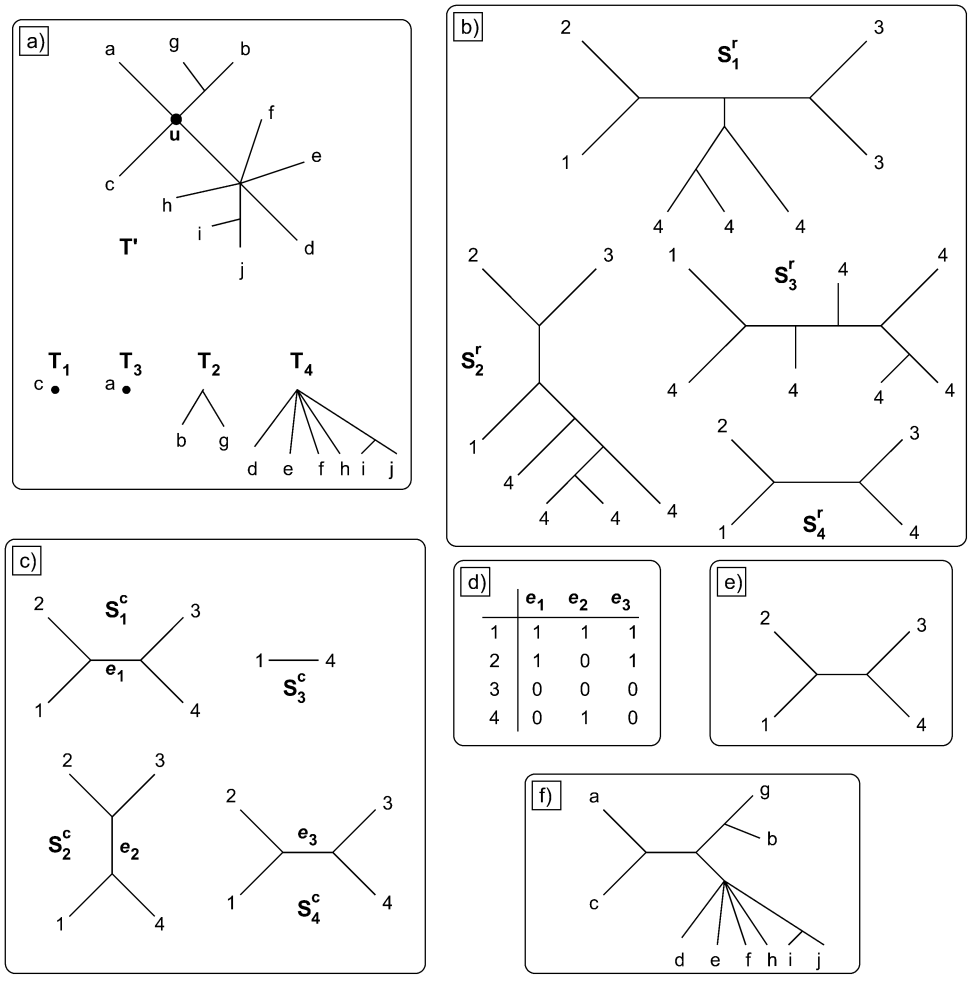

FIGURE 3. Schematic representation of the second step of the algorithmic strategy of SuperFine+MRP, in which we refine the SCM tree produced in the first step. The steps here refer to the SCM tree T', polytomy u, and source trees shown in Figure 1. a) The deletion of the polytomy u from the tree T' partitions T' into four rooted trees, T_1, T_2, T_3, and T_4. b) The leaves in each of the four source trees are relabeled by the index of the tree T_i containing that leaf, producing relabeled source trees S_1^r, S_2^r, S_3^r, and S_4^r. For example, the relabeled version of $S_4 = ac|bd$ is $S_4^r = 12|34$. c) Each S_i^r is further processed by repeatedly replacing sibling nodes with the same label, until no two siblings have the same label; this results in trees S_1^c, S_2^c, S_3^c, and S_4^c. d) The MRP matrix is shown for the four source trees, including only the parsimony informative sites; thus, S_3^c does not contribute a parsimony informative site and is excluded. e) The result of the MRP analysis on the matrix given in (d). f) The tree resulting from identifying the root of each $T_i, i = 1, 2, 3, 4$, with the node i in the tree from (e).

SuperFine+MRP.—SuperFine+MRP is SuperFine with an MRP heuristic used to refine the polytomies in the SCM tree. We experimented with various ways of running the MRP heuristic, and picked the parsimony ratchet (Nixon 1999) implemented for PAUP* (Swofford 2002) because it produced good results. We ran the parsimony ratchet with 100 random samples (with replacement) of the input sequences, each sample analyzed with TBR branch swapping, saving the best 201 trees. We returned the greedy (extended majority) consensus of the best MRP trees found in each analysis.

SuperFine+QMC.—SuperFine+QMC is SuperFine with the QMC method, used as a supertree method, to refine the polytomies in the SCM tree. QMC is a heuristic that takes a set of quartet (four-leaf) trees as input, and attempts to find a tree on the full set of taxa that agrees with the maximum number of its input quartet trees (an NP-hard problem) (Jiang et al. 2001). QMC uses a divide-and-conquer technique, combined with randomness, to produce a solution to the optimization problem that is not guaranteed to be optimal, but which performs well in practice. In order to run QMC as a supertree method, we replaced every source tree with its set of induced quartet trees and computed the union of these sets. We then applied the QMC heuristic in default mode to produce a tree on the full set of taxa. By design, this use of QMC involves computing the full set of quartet trees for every source tree. When all source trees are small, this method is reasonably fast; however, when source trees are not small, the representation of the source trees as quartet trees can be prohibitively expensive. However, on those supertree problems for which QMC can be used as a supertree method, it produced supertrees that matched or improved upon the topological accuracy of MRP (Swenson et al. 2010a, 2010b).

By contrast, SuperFine+QMC only needs to apply QMC to sets of quartet trees generated when analyzing polytomies. By construction, when SuperFine+QMC resolves a polytomy of degree d, it applies QMC to a collection of re-encoded source trees, each on at most d leaves. Thus, SuperFine+QMC can be applied to larger data sets than QMC can be applied to, as long as the polytomies are not of too high degree.

Other methods.—We compare SuperFine+MRP and SuperFine+QMC to MRP, MinFlip (Chen et al. 2006), PhySIC (Ranwez et al. 2007), SFIT (Creevey and McInerney 2005), Q-imputation (Holland et al. 2007), Robinson–Foulds Supertree (RFS; Bansal et al. 2009), and QMC used as a supertree method (as described above). We ran the MRP analysis using the same parsimony ratchet analysis as we used in SuperFine+MRP, and we ran QMC (as a supertree method) using the same QMC analysis as used in SuperFine+QMC. We ran MinFlip, PhySIC, SFIT, Q-Imputation, and RFS in their default settings.

Finally, we also performed a combined analysis using RAxML in its default (and accurate) setting to infer a GTR+Gamma ML tree for the simulated data sets. We did not perform a partitioned analysis on the concatenated alignment, and this potentially reduces the accuracy of the combined analysis tree.

We omitted methods that are not guaranteed to produce "plenary" supertrees (where a plenary supertree is one that includes all the taxa in the input source trees). Thus, we omitted PhySIC_IST, which failed to produce a plenary supertree in our studies. We also omitted methods that had not been shown to be promising in comparison with MRP, or which were not available in software (such as the ML supertree, Steel and Rodrigo 2008 or the majority rule supertree, Cotton and Wilkinson 2007).

All but one of the empirical data sets we examined included roots for their source trees, enabling us to analyze these data sets using methods that require rootings (i.e., MinFlip, PhySIC, and RFS) without having to estimate the location of the root. However, our simulation protocol produces unrooted source trees. In these cases, in order to be able to compute supertrees using these methods, we rooted each source tree in the simulated data sets on the midpoint of its longest path, based upon the ML branch lengths. This technique is one of the standard ways to locate the root, but it has the potential to introduce error into the rootings, and hence may reduce the accuracy of the supertrees estimated using these methods.

For methods, such as MRP, that returned more than one supertree for a given data set, we show results for the greedy consensus of all such trees. The greedy consensus builds a consensus tree by adding the splits in the input trees to the majority consensus tree, according to the frequency with which the split appears, until no additional split can be added. Thus, the greedy consensus is a refinement of the majority consensus.

For the combined analysis, the source tree data sets were concatenated into a supermatrix, and ML trees were inferred on the concatenated data sets using RAxML (Stamatakis 2006), version 2.2.0. Details regarding software versions and commands we used are given in online Appendix 1 (available at http://www.sysbio.oxfordjournals.org).

Simulated data sets

We used the simulated source tree data sets from Swenson et al. (2009) and Swenson, Barbançon, et al. (2010). These have realistic patterns of missing data, reflecting both biological processes and taxon sampling strategies used by systematists in phylogenetic studies. Swenson et al. (2009) and Swenson, Barbançon, et al. (2010) simulated evolution of genes down the model trees, modeling the birth and death (gain and loss) of each gene. Two types of source trees were generated on the model trees: *clade-based* source trees (each tree being a dense sample within a specific clade of the model tree), and *scaffold* source trees (a random sampling of a proportion of the taxa throughout the model tree). Scaffold trees are "backbone trees" used to relate the

clade-based source trees to one another and are similar to higher level taxonomic trees that provide the relationships among lower level taxonomic groups for which clade-based trees are produced. The proportion of taxa from the model tree that is sampled in the scaffold tree (called the *scaffold density*) is known to have a substantial impact on supertree estimation accuracy when using MRP (Swenson et al. 2009; Swenson, Barbançon, et al. 2010), with supertrees generally being more accurate when scaffold trees are more densely sampled. We produced scaffold trees of four densities (20%, 50%, 75% and 100%), in order to include a range of conditions that include ones typical of systematic studies (low scaffold densities) as well as ones that might favor supertree analyses.

We generated supertree data sets with 100, 500, and 1000 taxa, with each subtree data set input consisting of a number of clade-based source trees and one scaffold-based source tree. In order to assess the statistical significance of our results, we produced 30 replicates for each supertree input condition with 100 or 500 taxa, and 10 replicates for each supertree input condition with 1000 taxa. The details for how we generated the supertree data sets differ slightly for the different numbers of taxa. We begin our description with how we generated the 100-taxon supertree data sets, and then describe the generation process for the 500- and 1000-taxon data sets.

Each 100-taxon supertree problem input consists of five clade-based source trees and one scaffold-based source tree. Each clade-based source tree is produced by a RAxML analysis of a matrix produced by concatenating three different nonuniversal gene data sets, and each scaffold-based source tree is produced by a RAxML analysis of a matrix produced by concatenating four universal gene data sets. Thus, each 100-taxon supertree problem has six source trees, based in total on 19 genes.

We now briefly describe how we generate the gene data sets (see Swenson et al. 2009; Swenson, Barbançon, et al. 2010 for details). Each gene data set (whether universal or non-universal) consisted of sequences all of length 500, and were produced by simulating GTR+Gamma evolution down a model tree with the desired number of taxa; however, the GTR+Gamma parameters (branch lengths and substitution matrices) differ slightly between the different gene data sets. The model trees are generated using a pure-birth process tree using r8s, and then the branch lengths are modified randomly to deviate the tree from ultrametricity.

Biological data sets

The biological data sets we used were from four published supertree studies and one combined analysis study: temperate herbaceous papilionoid legumes (THPL, 558 taxa, 19 source trees, see Wojciechowski et al. 2000), comprehensive papilionoid legumes (CPL, 2228 taxa, 39 source trees, see McMahon and Sanderson 2006), marsupials (267 taxa, 158 source trees, see Cardillo et al. 2004), placental mammals (116 taxa, 726 source trees, see Beck et al. 2006), and seabirds (121 taxa, 7 source trees, see Kennedy and Page 2002). The biological source trees were produced using a variety of phylogeny estimation methods, including distance-based, parsimony, and likelihood methods. In all cases, we used the source trees provided in the original supertree studies, modified (when necessary) to account for ambiguously identified taxa; see online Appendix 2 for further details. All but one of these biological data sets came with rooted source trees, produced using outgroups by the authors of the studies; we used these rootings in order to produce supertrees for these data sets using the methods that require rooted source trees. The remaining data set (CPL) came with alignments for each marker; we computed RAxML trees on each of these alignments to produce the source trees for our analyses. Since the data set did not have an outgroup, we rooted the source trees using the midpoint method.

Measurements

Results on the simulated data were assessed using various criteria. Most importantly, we examined topological accuracy using false-positive (FP), false-negative (FN), and Robinson–Foulds (RF) error rates of the inferred trees compared with the model trees. The FP rate is the proportion of internal branch appearing in the inferred tree that are not in the model tree, and the FN rate is the proportion of internal branches in the model tree that are missing from the inferred tree. For those cases where the number of internal branches was 0, we set the corresponding error rate to 0. Finally, the RF rate is the average of these two values. When the estimated and model trees are binary, all three rates are the same.

In general, evaluation of supertree methods on biological data sets is difficult for several reasons. Most importantly, the true tree is generally not known, so absolute accuracy cannot be determined. Some studies have used measures of topological distance (typically the RF distance) to the source trees as a proxy for topological accuracy (e.g., Snir and Rao 2010 and Bansal et al. 2009). However, as noted in Swenson et al. (2010a, 2010b), topological distance is only weakly correlated with topological accuracy. Specifically, on simulated data, Swenson et al. (2010a) and Swenson et al. (2010b) found that Spearman rank correlations between total topological distance to the source trees and topological error with respect to the true tree were below 60% for all three ways of defining the topological distance. Therefore, only when two supertree methods have a relatively large difference in topological distances to the source trees is it likely the method with the smaller distance has a definite improvement in topological accuracy (Swenson et al. 2010a, 2010b). For this reason, although we present total topological distance measures for evaluating the supertree methods on

biological data, we are cautious in interpreting the topological distances to source trees.

For the biological data sets, we report the total topological distance to the source trees using three measures: SumFN (total number of branches in source trees missing from the supertree), SumFP (total number of branches in the supertree that are not in the source trees), and SumRF (total bipartition distance). Each measure is expressed as a percentage of the maximum possible, and so varies between 0 and 100. Although the SumRF distance is the more typical measure,it penalizes for resolution in the supertree that is not explicitly present in the source trees. SumRF is therefore, only appropriate when all source trees are completely resolved. Furthermore, SumFP is optimized by a tree without any internal branches (i.e., star trees). Since biological supertree data sets often have unresolved source trees, SumFN is a better metric because it properly handles incomplete resolution in the source trees. Furthermore, when the source trees and supertrees are all completely resolved, then SumRF, SumFN, and SumFP are all identical; thus, there is no advantage to using SumRF instead of reporting SumFN and SumFP.

We evaluated the performance of all estimated trees, that is, the SCM and other supertree methods, and trees estimated using combined analysis using ML. We evaluated the accuracy of the estimated supertrees using the three topological error measures (FN, FP, and RF rates), and also computed the topological distances of these supertrees to source trees. We computed topological distances between estimated supertrees for each biological data set (see online Appendix 3). Finally, the resolution of the SCM tree determines how different the SuperFine-boosted methods are from their base supertree methods, and so we also evaluated the resolution of the SCM tree.

RESULTS

Performance of SCM

Results on simulated data.—Figure 4 compares the FN and FP rates of SCM, SuperFine+MRP, and CA-ML on the 1000-taxon simulated data sets. SCM has much lower FP rates than the other methods (about 5%, as compared with 10–15% for the other methods), but also has much higher FN rates. The low FP rates show that almost all of the branches in the SCM tree are in the true tree. This is not surprising since the FPs in the SCM tree are constrained to those that at least one source tree has, and all the source trees support. Although the second stage usually resolves the tree further, it will never undo these universally supported (and highly accurate) bipartitions. The higher FN rates (18–23%) produced by the SCM tree, in comparison with the other methods (10–15%), show that the SCM tree is only partially successful at estimating the true tree. On the other hand, the SCM tree is quite well resolved (the resolution varies between 80% and 85%).

Results on biological data.—SCM supertrees computed on empirical data sets had various levels of resolution. One data set (placental mammals) had a very poorly resolved SCM tree, with one polytomy of degree 115 of a possible 116; the least resolved other tree (Marsupials) had one polytomy of degree 200 of a possible 267, but all the others were much more resolved. Thus, the SCM tree was less well resolved on the biological data sets than on the simulated data sets, but was well resolved on some.

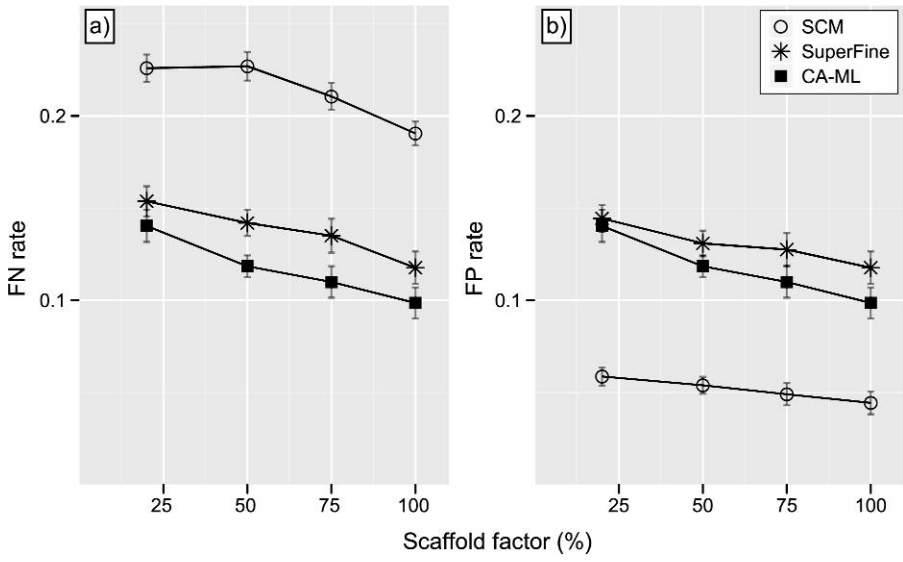

FIGURE 4. Comparison of SCM, SuperFine+MRP, and combined analysis using maximum likelihood (CA-ML) on simulated 1000-taxon data sets, as a function of the scaffold factor (proportion of the taxa in the scaffold data set). Topological error is given by (a) the FN rate, which is the proportion of internal branches in the true tree missing from the estimated tree, and (b) FP rate, which is the proportion of internal branches in the true estimated tree that are not in the true tree. Each point shows the average of 10 data sets and a standard error bar.

Performance of Superfine+MRP

We now explore the performance of SuperFine+MRP in comparison with other supertree methods and with CA-ML. We do not show results for all methods on all data sets for the following reasons. First, some methods either failed to run (due to memory requirements) or failed to complete within 2 weeks; these results are excluded. Second, for the 1000-taxon data sets, we only ran methods that had acceptable accuracy on the 500-taxon data sets; therefore, SFIT and PhySIC were excluded. We attempted to run Q-imputation on the 500-taxon data sets, but it failed to run on any of these data sets; we therefore did not attempt to run it on the 1000-taxon data sets. Also, in order to make fair comparisons of the methods, we only included the results from data sets on which all of the included methods completed. Finally, we discuss QMC and SuperFine+QMC in a later section.

Topological error and running time on simulated data.—We compared SuperFine+MRP with SFIT, PhySIC, Q-Imputation, MRP, MinFlip, RFS, and CA-ML, except as noted above, and focus the discussion on the missing branch (FN) rate. Results on simulated data showed several clear trends. First, for all scaffold factors and taxon numbers, the most accurate supertree method was consistently SuperFine+MRP, although the degree of improvement over the other methods varied with these parameters (Fig. 5 shows results for MRP, SuperFine+MRP, MinFlip, RFS, and CA-ML on 1000-taxon data sets; results for the other methods and for other numbers of taxa are given in online Appendix 3). Usually MRP was the second most accurate supertree method, although on a few model conditions (the 100-taxon data sets with scaffold densities of less than 100%) Q-imputation was the second most accurate supertree method.

Our results also show that SuperFine+MRP produced trees that were close in topological accuracy to combined analyses using ML. For the 500- and 1000-taxon data sets at 20% and 50% scaffold density, SuperFine+MRP's FP rates were statistically indistinguishable from those of the combined analyses, and for the same data set sizes, SuperFine+MRP's FN rates nearly matched those of the combined analyses.

As expected, the scaffold density had a large impact on the topological accuracy of the methods, with the most accurate results achieved on data sets with dense scaffolds. However, as scaffold density decreased, SuperFine+MRP's accuracy degraded at a much slower rate than the other supertree methods.

With respect to running time, SuperFine+MRP was always as fast or faster than other supertree methods, and it was consistently faster than the combined analyses even when the time to generate the source trees was taken into account (Fig. 5 gives results for 1000 taxon data sets, and online Appendix 3 provides running time information on the 100- and 500-taxon data sets). For the 1000-taxon simulated data sets, SuperFine+MRP typically ran in less than 3 h, whereas CA-ML often required more than a day to complete.

Distance to source trees and running time on biological data.—We examine SuperFine+MRP in comparison with MRP, RFS, MinFlip, Q-imputation, SFIT, and PhySIC (the performance of SuperFine+QMC is examined later). We report SumFN topological distances in Table 1, with SumFP and SumRF results reported in online Appendix 3.

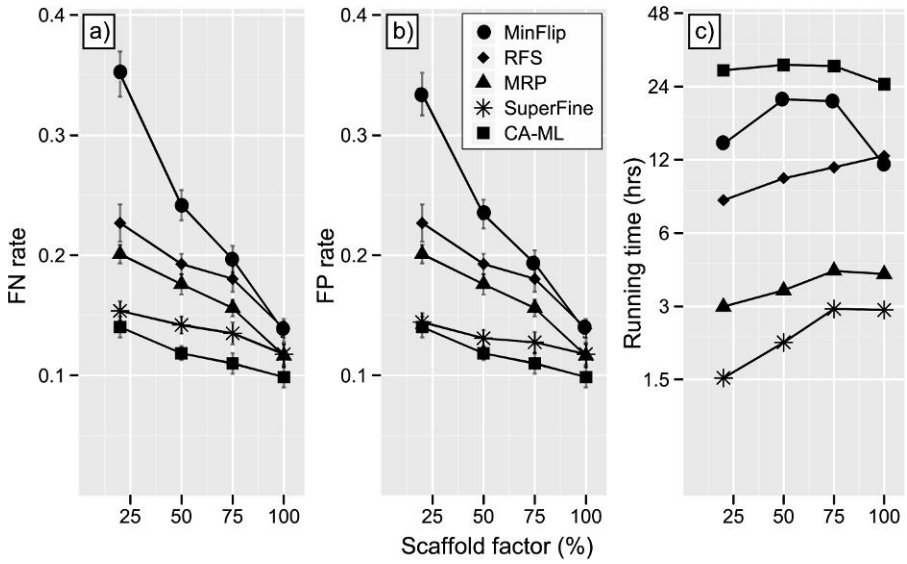

FIGURE 5. Comparison of MinFlip, RFS, MRP, SuperFine+MRP, and CA-ML on simulated 1000-taxon data sets. Topological accuracy is given by (a) normalized FN and (b) FP rates. Running time (c) is given in hours on a logarithmic scale; for the supertree methods, running time shown includes the time needed to calculate ML source trees using RAxML. Each point shows the average of 10 data sets and a standard error bar.

TABLE 1. Comparison of supertree methods on biological data sets with respect to SumFN and CPU time in hours (in parentheses)

	THPL	Seabirds	Placental mammals	CPL	Marsupials
SuperFine+MRP	**15** (0.011)	**13** (0.001)	**36** (0.157)	**33** (0.537)	**26** (0.053)
SuperFine+QMC	**17** (0.023)	**13** (0.001)	**38** (0.101)	F	F
RFS	19 (0.609)	**12** (0.003)	**36** (0.362)	**31** (123.082)	**26** (0.866)
MRP	20 (0.512)	15 (0.003)	**36** (0.058)	**33** (13.944)	**26** (0.078)
MinFlip	34 (1.093)	19 (0.003)	40 (0.121)	38 (302.269)	34 (0.335)
Q-imputation	29 (90.803)	14 (1.615)	F	F	F
SFIT	F	62 (1.053)	48 (108.686)	F	76 (111.879)
PhySIC	100 (0.152)	100 (0.001)	100 (0.001)	F	100 (0.007)

Notes: SumFN is the sum of normalized FN error rates to source trees, given as a percentage; the best scores (within 2%) for each data set are given in bold. F indicates the method failed to complete within 2 weeks. THPL refers to the temperate herbaceous papilionoid legumes data set, and CPL refers to the comprehensive papilionoid legumes data set.

SuperFine+MRP, MRP, and RFS tended to produce supertrees with smaller topological distances (both SumFN and SumRF) to the source trees than these other supertree methods. MinFlip and Q-imputation were slightly worse than these three, and PhySIC and SFIT were much worse. Because the methods that had poor topological accuracy on the simulated data also had substantially larger SumFNs than the methods that had good topological accuracy on the simulated data, it is likely that the methods that had worse SumFN and SumRF scores are simply less accurate supertree methods and did not produce reasonably accurate supertrees.

Comparisons between methods with relatively close topological distances to the source trees are difficult, since topological distance to source trees is only weakly correlated with topological error. Therefore, it is difficult to compare SuperFine+MRP, RFS, and MRP on the biological data sets. However, on the THPL data set, because of the large difference in the SumFN topological distance to source trees, it seems likely that MRP was less accurate than SuperFine+MRP and RFS.

The supertree methods fell into three groups with respect to running time (Table 1). SFIT and Q-Imputation were the slowest, failing to complete on several data sets, and taking the longest for those data sets for which they did complete. The fastest methods were SuperFine+MRP, MRP, and PhySIC, which completed substantially faster than the remaining methods, RFS and MinFlip. A comparison between the running times of SuperFine+MRP, MRP, and PhySIC shows that all completed quickly (in under an hour) on the relatively easy to analyze data sets, but that their running times were highly distinguished on CPL, the data set that presented the largest computational challenge to the supertree methods. For this data set, SuperFine+MRP finished in a little more than half an hour, while MRP took almost 14 hours, and PhySIC failed to complete. Thus, with respect to running time, SuperFine+MRP was the fastest method on the data sets we analyzed.

MRP scores on biological data.—We examined the MRP scores produced by SuperFine+MRP and MRP on these empirical data sets (Table 2).

On two of the data sets, SuperFine+MRP produced better MRP scores, on two SuperFine+MRP matched the MRP score, and on one data set, SuperFine+MRP produced a worse MRP score. Because SuperFine+MRP frequently produced trees that had better scores than MRP, this suggests that the MRP heuristic we used (the parsimony ratchet in PAUP*) does not work well with large MRP inputs (partial binary matrices containing many "?"s). It seems possible that better solutions to MRP might be obtained by using other MP software, such as TNT (Goloboff et al. 2008). However, the better topological accuracy that we obtain is also potentially the result of restricting the search for MRP solutions to only those trees that refine the tree produced in the first step of the SuperFine method.

Performance of SuperFine+QMC

We now explore SuperFine+QMC and compare its performance with QMC and other supertree methods

TABLE 2. Comparison of supertree methods on biological data sets with respect to MRP scores

	THPL	Seabirds	Placental mammals	CPL	Marsupials
SuperFine+MRP	**858**	**206**	**8809**	5488	**2112**
SuperFine+QMC	918	209	8893	F	F
RFS	1112	208	8855	6568	2140
MRP	902	211	**8809**	**5483**	**2112**
MinFlip	1064	218	9232	6056	2284
Q-imputation	1051	212	F	F	F
SFIT	F	481	10160	F	4822
PhySIC	5191	961	25,790	F	7537

Notes: For methods that return more than one tree, the best MRP score produced by any tree is shown. Best scores for each data set are given in bold. F indicates the method failed to complete within 2 weeks.

we explored. We use the same simulated and empirical data sets here as in our evaluation of SuperFine+MRP.

Topological error and running time on simulated data sets.—Figure 6 presents the results of a comparison of QMC with SuperFine+QMC and with SuperFine+MRP with respect to missing branch (FN) error rate on the simulated data sets; on these data sets, QMC fails to analyze the 500 and 1000-taxon data sets, whereas SuperFine+QMC and SuperFine+MRP succeed in analyzing all these data sets. Thus, SuperFine+QMC is able to analyze much larger data sets than QMC.

A comparison of QMC with SuperFine+QMC on the 100-taxon data sets on which both methods could be run showed that SuperFine+QMC produced more accurate trees than QMC on all but the 100% scaffold density data sets where both methods performed equally well. SuperFine+QMC was also faster than QMC (Fig. 7). Thus, SuperFine+QMC yields dramatic advantages over QMC with respect to topological accuracy, running time, and scalability. Furthermore, SuperFine+QMC produced supertrees of almost exactly the same topological accuracy as those produced by SuperFine+MRP, and in about the same amount of time, and thus (like SuperFine+MRP) outperforms the other supertree methods with respect to topological accuracy.

Distance to source trees and running time on biological data sets.—SuperFine+QMC failed to analyze some of the biological data sets (Marsupials and CPL) because of the polytomy degrees in the SCM trees for these data sets. The Marsupials SCM tree has a polytomy of degree 200, and the CPL data set has a polytomy of degree 532. However, SuperFine+QMC succeeded in analyzing the THPL and placental mammals data sets, which also had large polytomies (degree 95 and 115, respectively). Thus, SuperFine+QMC was able to analyze some large biological data sets, but not all, and the limitation is the maximum degree of the SCM tree. On the three biological data sets for which SuperFine+QMC was able to run, it produced trees that were on average as close to the source trees as SuperFine+MRP, using all three criteria (SumFN, SumRF, and SumFP, see Table 1 and online Appendix 3). Also, although SuperFine+QMC did not complete on all the biological data sets, when it did finish an analysis, it completed in minutes. Thus, SuperFine+QMC was as fast as SuperFine+MRP, and used less time than the other supertree methods on these data sets.

Finally, trees computed by SuperFine+MRP and SuperFine+QMC had very close total topological distances to source trees on the data sets on which both methods succeeded in running. Thus, on those data sets that SuperFine+QMC was able to analyze, the results it obtained were close to those obtained by SuperFine+MRP, and hence generally superior to those obtained by other supertree methods. However, SuperFine+QMC is unable to analyze some data sets, for which the SCM tree produces large degree polytomies. Therefore, SuperFine+QMC is a distinct improvement on QMC (its base method) and many other supertree methods, but cannot analyze the full range of data sets that SuperFine+MRP or MRP can.

DISCUSSION

The experiments using simulated data reported here indicate that SuperFine+MRP and SuperFine+QMC are more accurate than current supertree methods, and in particular more accurate than their respective base methods. Thus, SuperFine is a boosting technique for supertree methods, and provides substantial

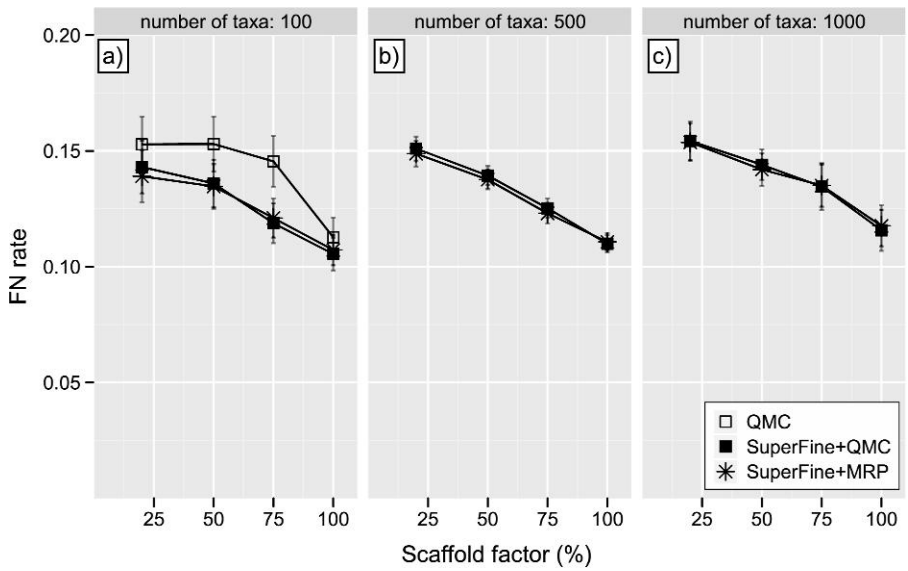

FIGURE 6. FN rate (mean with standard error bars) for QMC and SuperFine+QMC supertree reconstructions on simulated data sets with (a) 100, (b) 500, and (c) 1000 taxa, as a function of the scaffold factor. QMC fails to run on the 500- and 1000-taxon data sets.

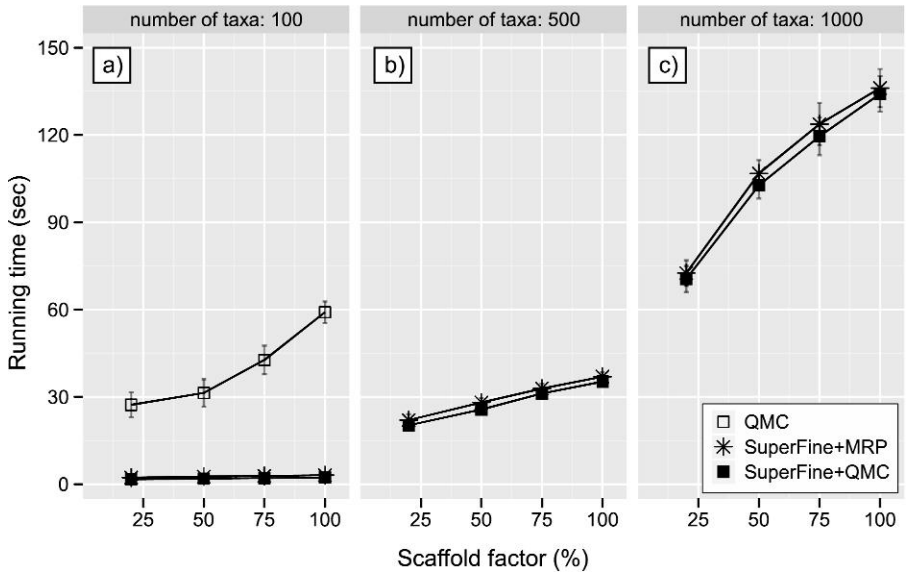

FIGURE 7. Running time (mean with standard error bars) for QMC and SuperFine+QMC supertree reconstructions on simulated data sets with (a) 100, (b) 500, and (c) 1000 taxa, as a function of the scaffold factor. QMC fails to run on the 500- and 1000-taxon data sets.

improvement for the base methods with respect to topological accuracy while also increasing the size of data sets that can be analyzed. Although SuperFine+MRP and SuperFine+QMC have very close performance (in terms of topological accuracy on simulated data and total topological distance to source trees for the biological data), SuperFine+MRP is generally more robust: it can analyze larger data sets than SuperFine+QMC, and does so quite efficiently. Thus, the major contribution of this paper, in terms of a single supertree method, is SuperFine+MRP.

On simulated data, Superfine+MRP comes close to the topological accuracy of combined analysis when the supertree has larger numbers of taxa and the scaffold densities are sparse. This is a condition that is likely fairly realistic for large supertree studies since biologists usually do not sample thoroughly when producing scaffold trees that determine the relationships among higher level taxonomic groups. Thus, we conjecture that under realistic conditions, SuperFine+MRP will have an advantage over other supertree methods. Furthermore, given SuperFine+MRP's increasing running time advantage as the number of taxa increases, it is likely to be the preferred method for trees having over 500 taxa.

For the biological data sets, our results show that SuperFine+MRP produces trees that are as close to the source trees as current methods that have been shown to previously be the most accurate, for example, MRP. Furthermore, SuperFine+MRP has a computational advantage over the other competitive supertree methods in that it can analyze large data sets reasonably efficiently. However, the relative accuracy between methods is more difficult to assess, due to the lack of an optimality criterion that has been demonstrated to reliably correlate with supertree topological accuracy.

The speed of SuperFine+MRP results from two features. First, the SCM technique is very fast, so that the first stage completes quickly. Second, polytomies are refined quickly because the re-encoding of the source trees as smaller trees (each with no more leaves than the degree of the polytomy) reduces the resolution of each polytomy to very small supertree problems. In addition, the parsimony ratchet implementation we used for MRP on the re-encoded source trees runs quickly, except for very large polytomies. In fact, even these supertree problems are less computationally intensive to compute than the MRP analysis of the original collection of source trees (unless the SCM tree is completely unresolved).

Thus, SuperFine+MRP improves on prior supertree methods, yielding more accurate trees under many realistic conditions and doing so quite efficiently. It is also the first supertree method to nearly match the accuracy of ML analyses of supermatrices. The speed and accuracy of SuperFine+MRP makes it a useful tool for large-scale phylogeny estimation, enabling significantly more accurate phylogenetic analyses of large multimarker data sets with high rates of missing data.

However, the improvement provided by SuperFine-boosting depends upon the resolution of the SCM tree. In particular, since it is possible for an SCM tree to be completely unresolved, SuperFine+MRP inherits all the negative properties that MRP has, including statistical inconsistency and the ability to have relationships that violate all the source trees. Similarly, SuperFine+QMC will also inherit all the negative properties for QMC. On the other hand, because these methods must refine the SCM tree, this reduces the opportunity to have contradictory splits, and it also ensures that some splits show up in the final supertree. Thus, even though Super Fine-boosted supertree methods cannot be guaranteed

to have good theoretical performance, it is not unsurprising to see the improved topological accuracy that results.

The impressive performance we observed for SuperFine thus depends (at least in part) on the ability of the SCM method to return a reasonably resolved initial estimate of the true tree that has a low FP rate. However, if many estimated gene trees are used, and if these do not have reasonably high accuracy, then the SCM tree itself may fail to be well resolved (and could even be a star). In this case, SuperFine boosting will fail to provide any advantage over the base supertree method. One obvious way to address this issue is to modify SuperFine so that the first step (producing the initial tree) is achieved using some technique other than SCM, which can retain features that are relatively common in the input set of source trees, even if not universally shared.

Finally, we note that although SuperFine-boosted methods have excellent accuracy and are quite fast, they are not the fastest of the existing supertree methods. The choice between different methods will thus need to take into account the relative importance between speed and accuracy. For those cases where accuracy is important, SuperFine provides a very substantial improvement over other methods, for many realistic conditions.

Supplementary Material

Supplementary Appendices can be found at http://www.sysbio.oxfordjournals.org/ and SuperFine source code is available through Dryad at (DOI:10.5061/dryad.879st).

Funding

This work was supported by the US National Science Foundation (DEB0733029 to M.S.S., C.R.L., and T.W.; ITR0331453 to M.S.S., R.S., and T.W.; ITR0121680 to C.R.L., M.S.S., and T.W.; EIA0303609 to T.W.; and IGERT0114387 to M.S.S.), and by the John Simon Guggenheim Foundation and Microsoft Research New England to T.W.

Acknowledgments

We thank Serita Nelesen for valuable feedback on initial implementations, and the reviewers for recommendations on additional experiments and suggested clarifications. The source code, simulated data sets, true alignments, and true trees are available at http://www.cs.utexas.edu/~phylo/software/superfine/submission/. Accessed 2 October 2011.

References

Bansal M., Burleigh J.G., Eulenstein O., Fernández-Baca D. 2009. Robinson-Foulds supertrees. Algorithms Mol. Biol. 5:18.

Baum B.R. 1992. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. Taxon. 41:3–10.

Baum B.R., Ragan M.A. 2004. The MRP method. In: Bininda-Emonds O.R.P., editor. Phylogenetic Supertrees: combining information to reveal The Tree Of Life. Dordrecht (the Netherlands): Kluwer Academic. p. 17–34.

Beck R.M.D., Bininda-Emonds O.R.P., Cardillo M., Liu F.G.R., Purvis A. 2006. A higher-level MRP supertree of placental mammals. BMC Evol. Biol. 6:93.

Bininda-Emonds O.R.P. 2003. Novel versus unsupported clades: assessing the qualitative support for clades in MRP supertrees. Syst. Biol. 52:839–848.

Bininda-Emonds O.R.P. 2004. Phylogenetic Supertrees: combining information to reveal The Tree Of Life. Dordrecht (the Netherlands): Kluwer Academic (Computational Biology).

Bininda-Emonds O.R.P., Bryant H.N. 1998. Properties of matrix representations with parsimony analyses. Syst. Biol. 47:497–508.

Burleigh J.G., Eulenstein O., Fernández-Baca D., Sanderson M.J. 2004. MRF supertrees. In: Bininda-Emonds O. R. P., editor. Phylogenetic Supertrees: combining information to reveal The Tree Of Life. Dordrecht (the Netherlands): Kluwer Academic. p. 65–86.

Cardillo M., Bininda-Emonds O.R.P., Boakes E., Purvis A. 2004. A species-level phylogenetic supertree of marsupials. J. Zool. 264: 11–31.

Chen D., Eulenstein O., Fernández-Baca D., Sanderson M.J. 2006. Minimum-flip supertrees: complexity and algorithms. IEEE/ACM Trans. Comp. Biol. Bioinform. 3:165–173.

Cotton J.A., Wilkinson M. 2007. Majority-rule supertrees. Syst. Biol. 56:445–452.

Creevey C., McInerney J. 2005. Clann: investigating phylogenetic information through supertree analyses. Bioinformatics. 21:390–392.

Day W.H.E. 1985. Optimal algorithms for comparing trees with labeled leaves. J. Classif. 2:7–28.

Foulds L.R. Graham R.L. 1982. The Steiner problem in phylogeny is NP-complete. Adv. in Appl. Math. 3:299.

Goloboff P., Farris J., Nixon K. 2008. TNT, a free program for phylogenetic analysis. Cladistics. 24:774–786.

Holland B., Conner G., Huber K., Moulton V. 2007. Imputing supertrees and supernetworks from quartets. Syst. Biol. 56:57–67.

Huson D., Nettles S., Warnow T. 1999. Disk-covering, a fast converging method for phylogenetic tree reconstruction. J. Comput. Biol. 6:369–386.

Huson D., Vawter L., Warnow T. 1999. Solving large scale phylogenetic problems using DCM2. In: Lengauer T., Schneider R., Bork P., Brutlag D.L., Glasgow J.I., Mewes H.W., Zimmer R., editors. Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology (ISMB'99). Association for the Advancement of Artificial Intelligence Press.

Jiang T., Kearney P., Li M. 2001. A polynomial-time approximation scheme for inferring evolutionary trees from quartet topologies and its applications. SIAM J. Comput. 30:1924–1961.

Kennedy M., Page R. 2002. Seabird supertrees: combining partial estimates of procellariiform phylogeny. Auk. 119:88–108.

Liu K., Linder C.R., Suri R., Warnow T. 2010. Multiple sequence alignment: a major challenge to large-scale phylogenetics. PLoS Curr. Accessed 18 November 2011.

McMahon, M., Sanderson M. 2006. Phylogenetic supermatrix analysis of GenBank sequences from 2228 papilionoid legumes. Syst. Biol. 55:818–836.

Moret B., Tang J., Warnow T. 2005. Reconstructing phylogenies from gene-content and gene-order data. In: Gascuel O., editor. Mathematics of evolution and phylogeny. Oxford (UK): Oxford University Press. p. 321–352.

Nixon K.C. 1999. The parsimony ratchet, a new method for rapid parsimony analysis. Cladistics. 15:407–414.

Pisani D., Wilkinson M. 2002. Matrix representation with parsimony, taxonomic congruence, and total evidence. Syst. Biol. 51:151–155.

Price M.N., Dehal P.S., Arkin A.P. 2010. FastTree 2—approximately maximum-likelihood trees for large alignments. PLoS One 5:e9490. doi:10.1371/journal.pone.0009490.

Ragan, M. A. 1992. Phylogenetic inference based on matrix representation of trees. Mol. Phylogenet. Evol. 1:53–58.

Ranwez V., Berry V., Criscuolo A., Fabre P., Guillemot S., Scornavacca C., Douzery E. 2007. PhySIC: a veto supertree method with desirable properties. Syst. Biol. 56:798–817.

Ranwez V., Criscuolo A., Douzery E.J. 2010. SuperTriplets: a triplet-based supertree approach to phylogenomics. Bioinformatics. 26:i115–i123.

Roshan U., Moret B., Williams T., Warnow T. 2004a. Performance of supertree methods on various dataset decompositions. In: Bininda-Emonds O.R.P., editor. Phylogenetic Supertrees: combining information to reveal The Tree of Life. Kluwer Academics. (Computational Biology; vol. 3), (Andreas Dress, series editor).
Roshan U., Moret B., Williams T., Warnow T. 2004b. Rec-I-DCM3: a fast algorithmic technique for reconstructing large phylogenetic trees. In: Brennder S., Schwartz J., Altman R., Kohane I., Toga A., Kikinis R., editors. Proceedings of the 3rd Computational Systems Biology Conference (CSB'05) Proceedings of the IEEE. Los Alamitos (CA): IEEE Computer Society. p. 98–109.
Snir S., Rao S. 2010. Quartets MaxCut: a divide and conquer quartets algorithm. IEEE/ACM Trans. Comput. Biol. Bioinform. 7:704–718.
Stamatakis A. 2006. RAxML-NI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics. 22:2688–2690.
Steel M., Rodrigo A. 2008. Maximum likelihood supertrees. Syst. Biol. 57:243–250.
Sukumaran J., Holder M.T. 2010. Dendropy: a Python library for phylogenetic computing. Bioinformatics. 26:1569–1571.
Swenson M. 2008. Phylogenetic supertree methods [dissertation]. Austin (TX): The University of Texas at Austin.
Swenson M.S., Barbançon F., Linder C.R., Warnow T. 2009. A simulation study comparing supertree and combined analysis methods using SMIDGen. In: Salzberg S., Warnow T., editors. Proceedings of the 2009 Workshop on Algorithms in Bioinformatics (WABI). Berlin-Heidelbeg (Germany): Springer. p. 333–344.
Swenson M.S., Barbançon F., Linder C.R., Warnow T. 2010. A simulation study comparing supertree and combined analysis methods using SMIDGen. Algorithms Mol. Biol. 5. doi:10.1186/1748-7188-5-8.
Swenson M.S., Suri R., Linder C.R., Warnow T. 2010a. An experimental study of Quartets MaxCut and other supertree methods. In: Moulton V., Singh M., editors. Proceedings of the 2010 Workshop on Algorithms in Bioinformatics (WABI). Berlin-Heidelberg (Germany): Springer. p. 288–299.
Swenson M.S., Suri R., Linder C.R., Warnow T. 2010b. An experimental study of Quartets MaxCut and other supertree methods. Algorithms Mol. Biol. 6:1–11.
Swofford D. 2002. PAUP*: phylogenetic analysis using parsimony (*and other methods). Version 4. Sunderland (MA): Sinauer Associates.
Wang L., Jiang T. 1994. On the complexity of multiple sequence alignment. J. Comput. Biol. 1:337–348.
Warnow T. 2006. Large-scale phylogenetic reconstruction. In: Aluru S., editor.) Handbook of Computational Biology. Boca Raton (FL): Chapman and Hall. (CRC Computer and Information Science Series). p. 21.1–21.23.
Warnow T., Moret B.M., St. John K. 2001. Absolute convergence: true trees from short sequences. In: Rao Kosaraju S., editor. Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA 01). Philadelphia (PA): SIAM (Society for Industrial and Applied Mathematics). p. 186–195.
Wilkinson M., Pisani D., Cotton J., Corfe I. 2005. Measuring support and finding unsupported relationships in supertrees. Syst. Biol. 54:823–831.
Wilkinson M., Thorley J.L., Pisani D.E., Lapointe F.J., McInerney J.O. 2004. Some desiderata for liberal supertrees. In: Bininda-Emonds O. R. P., editor. Phylogenetic Supertrees: combining information to reveal The Tree Of Life. Dordrecht (the Netherlands): Kluwer Academic. p. 227–246.
Wojciechowski M., Sanderson M., Steele K., Liston A. 2000. Molecular phylogeny of the "temperate herbaceous tribes" of papilionoid legumes: a supertree approach. Adv. Legume Syst. 9:277–298.
Zwickl D. 2006. GARLI download page. Website http://www.bio.utexas.edu/faculty/antisense/garli/Garli.html. (Accessed 2 October 2011).

APPENDIX

Detailed description of the SCM

The SuperFine algorithm proceeds in two stages: (1) compute a supertree from source trees, using the SCM (Huson, Nettles, et al. 1999; Roshan et al. 2004b; Roshan et al. 2004a), and (2) refine each polytomy in the SCM supertree, where a polytomy is a node of degree greater than three. Resolving a single polytomy involves re-encoding each source tree, and then applying the base supertree method to that set of re-encoded source trees, and finally adding edges to refine the polytomy based on the supertree that has been computed. The default base supertree method is MRP (Baum 1992; Ragan 1992; Baum and Ragan 2004), but other methods can be used.

The SCM is the foundation of the SuperFine method, and of interest in its own right. Because we prove theorems about the SCM, although the method has already been published (Huson, Nettles, et al. 1999; Huson, Vawter, et al. 1999; Roshan et al. 2004b), we describe how the SCM is computed in some detail.

Stage 1: the strict consensus merger.—SCM constructs a supertree from a set of source trees by merging pairs of trees until only a single tree remains. The merger of two trees begins with the strict consensus (Day 1985) of the induced subtrees of the two trees on the intersection of their taxon sets. The remaining taxa in the union of the two taxon sets are added to this consensus tree in such a way that they do not contradict any relationships in either of the trees. We define this process formally below.

First:

- Let $L(T)$ denote the taxon set of a phylogenetic tree T.
- Let $T|_X$ denote the induced subtree of T on the taxon-set X.
- Let $E(T)$ denote the edge set of the tree T.
- Let $\Sigma(T)$ denote the set of bipartitions induced by the internal edges of T.
- We say that T refines T', denoted $T' \leq T$, if $\Sigma(T') \subseteq \Sigma(T)$.
- Let Y be a proper subset of $L(T)$ and let e be an edge in $E(T|_Y)$. Thus, e defines a bipartition $A|B$ of Y. Since Y is a proper subset of $L(T)$, there is at least one and possibly several edges in T defining bipartitions $A'|B'$ that "extend" the biparition $A|B$, meaning $A \subseteq A'$ and $B \subseteq B'$. It is easy to see that the set of all such edges in T whose bipartitions extend $A|B$ forms a path in T; this is the "path corresponding to e". Thus, e corresponds to a path $p = (v_1, \ldots, v_l)$ in T such that for every $i \in \{1, \ldots, l-1\}$, the bipartition $A'|B'$ on $L(T)$ induced by the edge (v_i, v_{i+1}) satisfies $A \subseteq A'$ and $B \subseteq B'$.

The SCM tree T of two trees T_1 and T_2, such that $|L(T_1) \cap L(T_2)| \geq 3$, is defined formally as follows (depicted in Fig. 2).

Let $X = L(T_1) \cap L(T_2)$, and let T_1' and T_2' be maximally refined trees such that $T_1' \leq T_1$, $T_2' \leq T_2$, and $T_1'|_X = T_2'|_X$.

Let $T' = T'_1|_X = T'_2|_X$. (Note that T' is the strict consensus tree of $T_1|_X$ and $T_2|_X$.)

Then, suppose edge $e \in E(T')$ corresponds to a path of length greater than one in both T'_1 and T'_2. For each $i \in \{1, 2\}$, we modify the trees T'_i as follows. Let $e_1, \ldots, e_l$ be the path in T'_i, that corresponds to e in T'. Collapse all edges e_j in this path such that $1 < j < l$. Rename the common vertex of e_1 and e_l by v_e. (Note that both trees now have a vertex with the same name, v_e.) After performing this process for each edge of T', merge the resulting (potentially collapsed) trees T''_1 and T''_2 into a single tree T by modifying T' using the following process. For each edge $e \in E(T')$ that corresponds to a path of length greater than one in both T'_1 and T'_2 (i.e. an edge now corresponding to a path of length two in T''_1 and T''_2), subdivide e with a vertex v_e. Attach to v_e that vertex any subtree of T''_1 or T''_2 that is not in T' and is rooted at v_e in either T''_1 or T''_2. Now for any edge in $e \in E(T')$ that corresponds to a path p of length greater than one either in T'_1 or in T'_2, subdivide e with as many vertices as there are internal vertices on that corresponding path. For each internal vertex v of p, attach the subtree rooted at that vertex in T'_i to the corresponding newly constructed vertex subdividing e. Notice that by construction, $T|_{L(T_1)}$, and $T|_{L(T_2)}$ are simply contractions of T_1 and T_2, respectively; therefore $T|_{L(T_1)} \leq T_1$, and $T|_{L(T_2)} \leq T_2$.

We use the term "SCM supertree" to refer to the result of consecutive strict consensus mergers of pairs of trees from a set of trees, such that each pair of trees being merged have at least three taxa in common.

Theoretical Results

The main result of the section is Theorem 1.4, that the relabeling of source trees with labels drawn from $1, 2, \ldots, d$, where d is the degree of a polytomy, produces at most one taxon of each label. However, we also prove that the SCM tree has the noncontradiction property of Ranwez et al. (2007), in Theorem 1.1 with respect to splits.

Theorem 1.1 *Let $\mathcal{T}$ be a collection of trees and let T be a SCM supertree of $\mathcal{T}$. Then for every $t \in \mathcal{T}$, $\Sigma(T|_{L(t)}) \subseteq \Sigma(t)$. Hence, the SCM supertree has the "noncontradiction property" of Ranwez et al. (2007).*

Proof. We prove this by induction on the cardinality of $\mathcal{T}$. By construction, the result holds for $|\mathcal{T}| = 2$. Assume $|\mathcal{T}| = k$, and that the result holds for sets of $k - 1$ trees. Let t be a member of $\mathcal{T}$, and consider the last two trees T_1 and T_2 to be merged to create the final tree T. Then one of the following must be true: t=T_1, t=T_2, or either T_1 or T_2 is the strict consensus merger of some set of trees $\mathcal{T}'$ that includes t. Our base case shows that $\Sigma(T|_{L(T_1)}) \subseteq \Sigma(T_1)$, and $\Sigma(T|_{L(T_2)}) \subseteq \Sigma(T_2)$. Thus, if $t = T_1$ or $t = T_2$, then $\Sigma(T|_{L(t)}) \subseteq \Sigma(t)$. Now suppose, without loss of generality, that T_1 is the strict consensus merger of a set of trees $\mathcal{T}'$ such that $t \in \mathcal{T}'$. Then $|\mathcal{T}'| < k$ and by the induction hypothesis $\Sigma(T_1|_{L(t)}) \subseteq \Sigma(t)$, and the result follows. □

The following corollary of Theorem 1.1 is immediate.

Corollary 1.2 *Let $\mathcal{T}$ be a collection of trees, let T be a SCM supertree of $\mathcal{T}$, and let v be a vertex of T. Let u be a vertex adjacent to v, and let T' be the connected component of $T - \{u, v\}$ (the tree obtained by deleting the edge $\{u, v\}$ from T, but not the endpoints) that contains u. Then, for any $t \in \mathcal{T}$, one of the following three conditions holds: $L(t) \subseteq L(T')$, $L(t) \cap L(T') = \emptyset$, or $L(t) \cap L(T')|L(t) - L(T') \in \Sigma(t)$.*

Proof. Let $\mathcal{T}$ be a collection of trees, T a SCM supertree of $\mathcal{T}$, v a vertex of T, u a vertex adjacent to v, and T' the connected component of $T - \{u, v\}$ that contains u. Thus $L(T') \subset L(T)$. Now consider $t \in \mathcal{T}$. Suppose that t fails to satisfy the first two of the three conditions above: thus, $L(t) \subseteq L(T')$ and $L(t) \cap L(T') \neq \emptyset$. Then $L(t) \cap L(T')$ and $L(t) - L(T')$ are both nonempty. By the definition of T', $L(t) \cap L(T')|L(t) - L(T')$ is a split of $T|_{L(t)}$ (the subtree of T induced by $L(t)$). Therefore, by Theorem 1.1, $L(t) \cap L(T')|L(t) - L(T') \in \Sigma(t)$. □

As we show in the main text, this corollary is used in the refinement stage of the SuperFine algorithm. The following lemma is an immediate result of Corollary 1.2.

Lemma 1.3 *Let $\mathcal{T}$, T, v, and ϕ be as in the description of the refinement stage of the SuperFine algorithm. Then for any $i \in \{1, \ldots, d\}$ and $t \in \mathcal{T}$, one of the following conditions holds: $L(t) \subseteq \phi^{-1}(i)$, $L(t) \cap \phi^{-1}(i) = \emptyset$, or $L(t) \cap \phi^{-1}(i)|L(t) - \phi^{-1}(i) \in \Sigma(t)$.*

Proof. Let $\mathcal{T}$, T, v, and ϕ be as defined in the description of the refinement stage of the SuperFine algorithm. Additionally, let $v_1, \ldots, v_d$ and $T_1, \ldots, T_d$ be as defined in the same description. Consider $i \in \{1, \ldots, d\}$ and $t \in \mathcal{T}$. Then $\phi^{-1}(i) = L(T_i)$, and the result follows directly from Corollary 1.2 with $u = v_i$ and $T' = T_i$. □

The following result follows easily:

Theorem 1.4 *Source trees relabeled and collapsed using the process described in step 2) of the refinement stage of the SuperFine algorithm have at most one taxon with each label.*

Proof. Again let $\mathcal{T}$, T, v, $T_1, ..., T_d$, and ϕ be as defined in the description of the refinement stage of the SuperFine algorithm. Consider $i \in \{1, \ldots, d\}$ and $t \in \mathcal{T}$. Then by Lemma 1.3, we are in one of three cases: $L(t) \subseteq \phi^{-1}(i)$, $L(t) \cap \phi^{-1}(i) = \emptyset$, or $L(t) \cap \phi^{-1}(i)|L(t) - \phi^{-1}(i) \in \Sigma(t)$.

Case 1: ($L(t) \subseteq \phi^{-1}(i)$): All leaves of t are labeled i and thus collapse to a single leaf.

Case 2: ($L(t) \cap \phi^{-1}(i) = \emptyset$): No leaves of t are labeled i.

Case 3: ($L(t) \cap \phi^{-1}(i)|L(t) - \phi^{-1}(i) \in \Sigma(t)$): In this case the collapsing process replaces the subtree T_i with a single leaf labeled i.

Thus, the result holds. □