

Robustness of topological supertree methods for reconciling dense incompatible data

Stephen J. Willson
 Department of Mathematics
 Iowa State University
 Ames, IA 50011 USA
 swillson@iastate.edu
 tel 515-294-7671
 FAX 515-294-5454

Abstract—Given a collection of rooted phylogenetic trees with overlapping sets of leaves, a compatible supertree S is a single tree whose set of leaves is the union of the input sets of leaves and such that S agrees with each input tree when restricted to the leaves of the input tree. Typically with trees from real data, no compatible supertree exists, and various methods may be utilized to reconcile the incompatibilities in the input trees. This paper focuses on a measure of robustness of a supertree method called its “radius” R . The larger the value of R is, the further the data set can be from a natural correct tree T and yet the method will still output T .

It is shown that the maximal possible radius for a method is $R = 1/2$. Many familiar methods, both for supertrees and consensus trees, are shown to have $R = 0$, indicating that they need not output a tree T that would seem to be the natural correct answer. A polynomial-time method Normalized Triplet Supertree (NTS) with the maximal possible $R = 1/2$ is defined. A geometric interpretation is given, and NTS is shown to solve an optimization problem. Additional properties of NTS are described.

Index Terms—graph algorithm, phylogenetic tree, supertree, rooted triple

I. INTRODUCTION

A phylogenetic tree T on a collection X of taxa is a tree that seeks to represent an evolutionary history including all the taxa in X . The members of X correspond to the leaves of the tree, and the interior vertices correspond to hypothesized common ancestors from which speciation events occurred. The arcs indicate direct parent/child relationships between the vertices. Reconstructing phylogenetic trees from data is a fundamental problem of phylogenetics.

A recurring situation is that different researchers, using a variety of data, collections of taxa, and methods, together create a set \mathcal{D} of phylogenetic trees T_i with different leaf sets X_i . An important problem then becomes to create a single tree T with the leaf set $X = \cup X_i$ which combines the information in the various input trees T_i . If T exhibits all the trees T_i , it may be called a “compatible supertree”. Typically, however, the input trees are incompatible so that no single tree T contains all the input trees T_i . In this situation, T must reconcile incompatible data in some manner, and it may be called an “approximate supertree”. The term “supertree” refers to both situations.

Producing a “tree of life” is of necessity the computation of a supertree [9].

There exist a great many supertree methods. The book [10] contains a fine collection of articles on the subject, including overviews of supertrees and their limitations.

The most commonly utilized supertree method is Matrix Representation with Parsimony (MRP). This method was suggested by Baum [5] and Ragan [24]; for more information see [26] and [8]. Suppose we are given data consisting of a collection \mathcal{D} of rooted trees in which each leaf is labelled by a taxon. The topological information about each tree in \mathcal{D} is encoded into a new character matrix, typically with many entries corresponding to missing data. The computationally intensive method of maximum parsimony is then applied to this new character matrix. If a compatible supertree exists, it corresponds to a maximum parsimony tree. Typically, whether or not a compatible supertree exists, there are many different maximum parsimony trees. Some consensus tree is then presented as a summary of the maximum parsimony trees. MRP has been utilized for some very large datasets, such as a supertree of the mammals [11]. MRP is appealing because of its ease of implementation. Nevertheless, since it reduces the supertree problem to the NP-hard problem of maximum parsimony [18], it is inherently slow for dealing with large numbers of taxa, and it must rely on heuristics rather than proven methods.

MRP has the advantage of constructing either unrooted or rooted supertrees. Many methods, however, including those discussed at length in this paper, construct only rooted supertrees. Rooted trees are more biologically relevant than unrooted trees since it is believed that all life on earth has a common origin; unrooted trees express ignorance of the root. Biologists often produce unrooted trees and then later consider the position of the root. Moreover, [30] shows that there can be no unrooted supertree method satisfying certain desirable properties.

The fast procedure BUILD described in [2] finds a compatible rooted supertree from input rooted trees provided there are no incompatibilities. Fast generalizations in the presence of incompatibilities are provided in, for example, [27], [22], [29], and [25]. Methods based on BUILD have the advantage of speed since they typically have polynomial-time complexity. Some methods have the disadvantage that they may produce trees that resemble Adams consensus trees [1] in that biologists may interpret the graphs as containing nestings, not clades, and hence the results may not be interpretable as the depiction of historical evolutionary events.

The new procedure described in this paper will involve a modification of BUILD in order to construct a rooted supertree.

Any supertree method has two roles. The first role is that of

extrapolation, in which relationships are inferred which might exist in no input tree. For example, it may happen that no single input tree T_i contains the taxa a , b , and c together, yet the supertree infers a relationship among them. The second role is reconciliation of incompatible data, in which different input trees contradict each other, and some kind of choice must be made between them.

This paper focuses on the second role, that of reconciliation of incompatible data. To study this role, an extreme situation is studied in which the collection \mathcal{D} of input trees is *dense*, i.e., for every a , b , and c in X there exists an input tree T_i in \mathcal{D} containing a , b , and c .

Phylogenetic information in a rooted tree is carried by rooted triplets of the form $ab|c$, wherein taxa a and b are clustered by comparison with taxon c . Rooted triplets are an especially reliable carrier of phylogenetic information. Degnan and Rosenberg [14] show that under coalescent models, rooted triplets should be accurately reconstructed (whereas for trees with 5 or more taxa the most likely gene tree in some circumstances differs from the species tree). It is therefore plausible to try to utilize rooted triplets in creating supertrees.

In this paper, a quantity called the “robustness radius” R is defined for each rooted supertree method. An exact definition of R is given in Section 4. Here we illustrate its meaning. Suppose that a particular method has $R = 1/10$. Suppose a collection \mathcal{D} of input trees is dense, and suppose that T is a binary rooted tree such that for every rooted triplet $ab|c$ expressed in T , at least $(1 - R) * 100\% = 90\%$ of the input trees containing $\{a, b, c\}$ express the rooted triplet $ab|c$. Then we expect that the supertree method should output T as the supertree. This is a natural condition, since it would appear that for each collection $\{a, b, c\}$ the data are strongly supporting the rooted triplet $ab|c$ found in T . Moreover, to say that $R = 1/10$ is to assert also that $1/10$ is the largest number for which this condition holds. The condition would then fail if $R = 11/100$, whence there exists a collection \mathcal{D} of input trees and a binary rooted tree T such that for every rooted triplet $ab|c$ expressed in T , at least 89% of the input trees containing $\{a, b, c\}$ express the rooted triplet $ab|c$, yet the supertree method does not output T as the supertree.

Theorem 4.1 shows that for any rooted supertree method, $R \leq 1/2$. A new supertree method called Normalized Triplet Supertree (NTS) is shown to have the optimal robustness radius $R = 1/2$ (Theorem 4.2). Much of the remainder of the paper shows that NTS possesses many other desirable properties.

What is most surprising to the author is that many familiar supertree methods have robustness radius $R = 0$. Having $R = 0$ means that for every $\epsilon > 0$ there exists a dense collection \mathcal{D} of input trees and a binary tree T such that for every rooted triplet $ab|c$ in T we have that at least $(1 - \epsilon) * 100\%$ of the input trees containing $\{a, b, c\}$ express the rooted triplet $ab|c$, yet T is not output as the answer. Less formally, if $R = 0$ then there exist examples with overwhelming evidence for every rooted triple $ab|c$ in T , yet T is not output as the answer. Theorem 5.1 shows that both MinCutSupertree [27] and Modified MinCutSupertree [22] have robustness radius 0. Furthermore, there exist examples that show that the robustness radius R for MRP satisfies $R \leq 0.01$, and I conjecture that for MRP, $R = 0$. This fact undermines confidence in MRP, the most commonly used supertree method.

Consensus methods always have as input a collection \mathcal{D} that is dense since all the input trees have the same leaf set. Theorem

5.2 shows that the robustness radii for the strict, majority rule, and Adams consensus trees [28], [17] all satisfy $R = 0$. These methods thus have paradoxical results on some datasets. By contrast, if NTS (or another method of radius 1/2) is utilized as a consensus method, these paradoxes are avoided.

Theorem 3.2 shows that, if X has n members and \mathcal{D} has m input trees, then NTS may be computed in time $O(n^4m)$. The complexity can likely be improved, but this result shows that NTS can be computed in polynomial time. This fact gives NTS a practical advantage over MRP. Section 6 investigates further properties of NTS.

Section 7 gives a geometric interpretation of NTS in terms of a hypercube H_X . Roughly, each input dataset \mathcal{D} in which each taxon is a member of X gives rise to a unique point $spt_{\mathcal{D}}$ in H_X , and similarly each rooted phylogenetic X -tree T gives rise to a unique point spt_T in H_X . The robustness radius R of a method is naturally expressed in terms of the l_{∞} norm on H_X . It is shown (Theorem 7.3) that if S is the output of NTS given \mathcal{D} , then S satisfies a geometric optimization problem in H_X .

Suppose that a supertree method on the dataset \mathcal{D} outputs a tree T . Suppose in H_X that we have $\|spt_{\mathcal{D}} - spt_T\|_{\infty} = \alpha$. Section 8 gives an interpretation in terms of the natural measure on H_X . The smaller the value of α , the more strongly the data in \mathcal{D} support the tree T .

Section 9 concludes the paper with a biological example.

Finding robust methods for building trees is a central problem in phylogenetics. The approach in this paper seeks to identify the rooted triplets that are most strongly supported in the data for the problem of reconstructing rooted trees. Analogously for the problem of reconstructing unrooted trees, there have been several approaches for identifying quartets that are most strongly supported. The methods of quartet cleaning [21], [7] concern ways to use internal evidence to identify the quartets that are most strongly supported. In particular, they find an upper bound on the number of quartets that can be corrected which involves a factor 1/2, analogous to the maximal possible robustness radius in this paper. The works of Erdős, Steel, Székely and Warnow on the Short Quartet Method [15], [16] seek to reconstruct trees via short quartets, which are more likely to be accurate, and thereby infer a phylogeny by methods requiring shorter DNA sequences. Berry and Gascuel [6] consider the problem, given a set Q of quartets, of finding a maximum tree-like subset Q^* of Q . Atteson [3] deals with measuring the robustness of the neighbor-joining method, and his method for measuring robustness resembles the tool utilized in this paper.

The basic construction of NTS is a recursive computation of threshold graphs similar to those described in, for example, [19], [20], p. 157, and [4]. It differs in that the weights on an edge $\{a, b\}$ utilized at each stage may change during the reconstruction process depending on the context.

II. FUNDAMENTALS

A. Rooted trees

Let X be a finite nonempty set. A *rooted tree* (T, X) with leaf set X is a collection T of subsets U of X such that

- 1) $X \in T$.
- 2) If U and V are in T , either $U \subseteq V$ or $V \subseteq U$ or $U \cap V = \emptyset$. (nesting)
- 3) For each $x \in X$, the singleton $\{x\}$ lies in T .

4) The empty set is not a member of T .

We may also say T is a *phylogenetic X -tree*. Each member of T will be called a *cluster* in T . The cluster X and the singleton clusters $\{x\}$ are called *trivial*; all other clusters are *nontrivial*. The *star* tree contains all the trivial clusters but no other clusters.

A rooted tree as defined above is also a rooted tree in the usual graph-theoretic sense in terms of vertices and arcs. Figure 1C shows the graphical version of the rooted tree with $X = \{a, b, c, d, e\}$ and nontrivial clusters $\{a, b, c, e\}$, $\{a, e\}$, and $\{b, c\}$. Explicitly, the rooted tree (T, X) corresponds to the directed graph (V, A) in which the vertex set $V = T$. If (T, X) is a rooted tree, $B \in T$, $C \in T$, $C \subset B$, $B \neq C$, and there is no $D \in T$ with $C \subset D \subset B$, $B \neq D$, $C \neq D$, then there is an arc (B, C) directed from B to C . All members of the set A of arcs arise in this manner. If (B, C) is an arc, we say that C is a *child* of B and B is a *parent* of C . We call X the *root*. Each member of T other than the root has a unique parent. Each singleton set $\{x\}$ is a *leaf* and has no child. Each member of T that is not a leaf has at least two children. A rooted tree T is *binary* if each cluster that is not a leaf has exactly two children.

If T is a rooted tree with leaf set X and X' is a nonempty subset of X , then the *restriction* of T to X' , denoted $T|X'$, consists of the collection of sets $U \cap X'$ such that

- (i) $U \in T$, and
- (ii) $U \cap X'$ is nonempty.

It is easy to see the following result, whose proof is omitted:

Lemma 2.1: $T|X'$ is a rooted tree with leaf set X' .

Write $ab|c$ for the rooted tree with leaf set $\{a, b, c\}$ and with clusters $\{a, b, c\}$, $\{a\}$, $\{b\}$, $\{c\}$, and $\{a, b\}$. We call $ab|c$ a *resolved rooted triplet* on $\{a, b, c\}$. Note $ab|c = ba|c$. Given distinct elements a , b , and c , there exist three distinct resolved rooted triplets $ab|c$, $ac|b$, and $bc|a$ on $\{a, b, c\}$. Write abc for the rooted tree with leaf set $\{a, b, c\}$ and with clusters $\{a, b, c\}$, $\{a\}$, $\{b\}$, and $\{c\}$. We call abc the *star tree* on a , b , and c or the *unresolved rooted triplet* on $\{a, b, c\}$. If X is a set, then $RT(X)$ denotes the set of all resolved rooted triplets on $\{a, b, c\}$ such that $\{a, b, c\} \subseteq X$. If $|X| = n$, then $RT(X)$ contains $3\binom{n}{3}$ members since there are $\binom{n}{3}$ subsets with three members, and each has 3 resolved rooted triplets.

If (T, X) is a rooted tree and $\{a, b, c\} \subseteq X$, then say $ab|c$ is in T or $ab|c \in T$ if $T| \{a, b, c\} = ab|c$. Equivalently, $ab|c$ is in T iff T contains a cluster U such that $\{a, b\} \subseteq U$ but $c \notin U$. Similarly abc is in T if $T| \{a, b, c\} = abc$. Equivalently, abc is in T iff every cluster U of T that contains two members of $\{a, b, c\}$ also contains the third.

A *rooted tree X -family* \mathcal{D} is a finite collection of rooted trees (T_i, X_i) for $i = 1, \dots, k$, where T_i is a rooted tree on the leaf set X_i and $X = \cup X_i$. A rooted tree (T, X) is a *compatible supertree* for \mathcal{D} provided for $i = 1, \dots, k$ it is true that $T|X_i$ contains T_i . Note that it is possible that $T|X_i$ is more highly resolved (contains more members) than T_i . Equality is not required because in typical biological applications a lack of resolution is interpreted merely as inadequate information about the true resolution. In general, a *supertree* for \mathcal{D} is a rooted tree (T, X) .

B. Minimal threshold trees

This subsection describes the computation of minimal threshold trees. This method will be essential in the calculation of NTS. The technique is a variant on minimal threshold methods described in [19], [20], and [4].

A *graph* $G = (V, E)$ consists of a finite set V whose elements are *vertices* and set E of *edges* each of which is a set $\{u, v\}$ consisting of two distinct elements of V . There are no loops and no multiple edges. A *subgraph* $G' = (V', E')$ of G is a graph such that $V' \subseteq V$ and $E' \subseteq E$. If $G' = (V', E')$ and $G'' = (V'', E'')$ are subgraphs of G , say $G' \leq G''$ if $V' \subseteq V''$ and $E' \subseteq E''$. It is easy to see that \leq is a partial order on the set of subgraphs of G .

The graph $G = (V, E)$ is *complete* if E consists of all 2-subsets of V . Equivalently, G is complete if for all distinct u and v in V , $\{u, v\} \in E$.

If a and b are distinct vertices, a *path* in $G = (V, E)$ from a to b is a sequence $a = v_0, v_1, \dots, v_k = b$ of distinct vertices such that for all i , $1 \leq i \leq k$, $\{v_{i-1}, v_i\} \in E$. A graph G is *connected* if for each pair a and b of distinct vertices there exists a path from a to b . A graph G is *disconnected* if it is not connected. A subgraph $G' = (V', E')$ of G is a *component* of G provided that G' is connected and there is no subgraph $G'' = (V'', E'')$ of G such that $G' \leq G''$, $G' \neq G''$, and G'' is connected. Thus a component of G is a maximal connected subgraph of G .

A *weighting* w on a graph $G = (V, E)$ is a function $w : E \rightarrow \mathbb{R}_{\geq 0}$ where $\mathbb{R}_{\geq 0}$ is the set of nonnegative real numbers. If $G = (V, E)$ is a graph with weighting w and τ is a nonnegative real number, G_τ will denote the subgraph (V, E_τ) of G where $E_\tau = \{e \in E : w(e) > \tau\}$. If $\tau < \tau'$ then it is immediate that $E_{\tau'} \subseteq E_\tau$. Consequently, if $\tau < \tau'$, then each component of $G_{\tau'}$ is a subgraph of some component of G_τ . Observe $E_0 = \{e \in E : w(e) > 0\}$.

Suppose $G = (V, E)$ is a graph with weighting w . If $|V| \geq 2$ and G_0 is connected, let $\tau(G) := \inf\{\tau \in \mathbb{R}_{\geq 0} : G_\tau \text{ is disconnected}\}$. Observe that if $\tau > w(e)$ for all $e \in E$, then G_τ is disconnected since $E_\tau = \emptyset$, whence $\tau(G)$ is well-defined. Moreover, $G_{\tau(G)}$ is disconnected since if it were connected we could select $\epsilon > 0$ such that no value $w(e)$ lies in the open interval $(\tau(G), \tau(G) + \epsilon)$ whence for $\tau \in (\tau(G), \tau(G) + \epsilon)$ we would have $E_{\tau(G)} = E_\tau$ whence G_τ is also connected, contradicting the definition of $\tau(G)$. If $|V| \geq 2$ and G_0 is disconnected, define $\tau(G) = 0$. Call $\tau(G)$ the *minimal disconnection threshold* for G . If $\tau(G) > 0$ and $0 \leq \tau < \tau(G)$, then G_τ is connected.

Suppose X is a finite set. If U is a nonempty subset of X , let $K(U) = (U, E)$ be the complete graph with vertex set U . Suppose, for all nonempty subsets U of X such that $|U| > 2$, there is a weighting w_U on the edges of $K(U)$. The *minimal threshold tree* T for (X, w) is defined as the smallest collection T of subsets of X such that the following three conditions hold:

- 1) $X \in T$.
- 2) If $U \in T$ has exactly two elements, $U = \{u, v\}$, then $\{u\} \in T$ and $\{v\} \in T$.
- 3) If $U \in T$ and $|U| \geq 3$, form $K(U)$ with weighting w_U . Let $\tau(U) := \tau(K(U))$ be the minimal disconnection threshold for $K(U)$. Then $K(U)_{\tau(U)}$ is disconnected. For each component C of $K(U)_{\tau(U)}$, the set of vertices of C is in T . More explicitly, for each component $C = (U', E')$ of $K(U)_{\tau(U)}$, the set U' of vertices of C lies in T .

Note the use of the simpler notation $\tau(U)$ for $\tau(K(U))$.

Figure 1 shows an example of the calculation of a minimal threshold tree. Section 3 will show how the weights in this example were determined. Figure 1A exhibits a graph $K(X)$ for $X = \{a, b, c, d, e\}$. The weighting w_X is $w_X(\{a, b\}) = 1$, $w_X(\{a, d\}) = 1/3$, etc. Edges with weight 0 are omitted. The

minimal disconnection threshold is $\tau(K(X)) = \tau(X) = 2/3$ since the inclusion only of edges with weight greater than $2/3$ disconnects the graph, while for $\epsilon > 0$, the inclusion of edges with weight greater than $2/3 - \epsilon$ does not disconnect the graph. Hence the components of $K(X)_{2/3}$ are $\{d\}$ and $U = \{a, b, c, e\}$. Figure 1B exhibits $K(U)$ with the weights w_U . Again, edges with weight 0 are omitted. Note that $w_U(\{a, b\}) = 1/2$ while $w_X(\{a, b\}) = 1$, as described in Section 3. Then the minimum disconnection threshold is $\tau(U) = 1/2$ and the components of $K(U)_{1/2}$ are $\{a, e\}$ and $\{b, c\}$. Figure 1C shows the minimal threshold tree exhibiting the nontrivial clusters $\{a, b, c, e\}$, $\{a, e\}$, and $\{b, c\}$ which were identified by the construction.

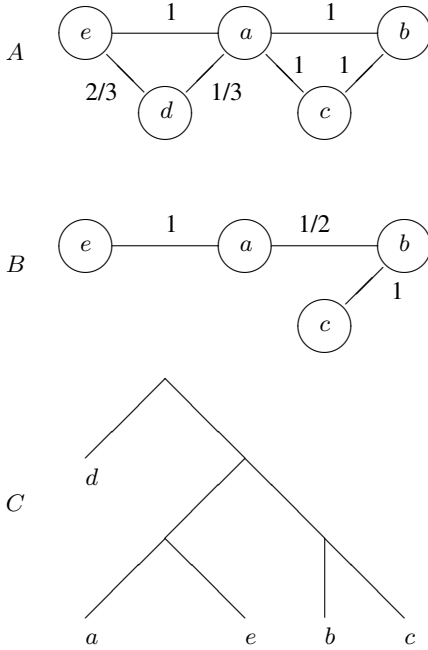


Fig. 1. Calculation of a minimal threshold tree. A shows $K(\{a, b, c, d, e\})$. Weights are shown for each edge. Edges with weight 0 are omitted. The graph disconnects with minimal threshold $2/3$. B shows $K(\{a, b, c, e\})$, which disconnects with minimal threshold $1/2$. C shows the minimal threshold tree.

For each $x \in X$, the singleton set $\{x\} \in T$, and the set X itself is in T , and constitutes its root. Hence the minimal threshold tree is in fact a rooted tree with leaf set X . Observe also that in the construction, given U , one may as well form not the complete graph $K(U) = (U, E)$ but a graph (U, E') where $E' = \{e \in E : w(e) > 0\}$ since only edges with positive weight influence the construction.

III. THE NORMALIZED TRIPLET SUPERTREE (NTS)

This section defines the Normalized Triplet Supertree (NTS) and describes its basic properties. The NTS is computed as a minimal threshold tree where the weightings are found in a certain manner. The computation will be shown to be polynomial-time. Of considerable interest will be the fact that when the input data are sufficiently “close” to those of a binary tree T , then the NTS will necessarily be T .

Let $\mathcal{D} = \{(T_i, X_i) : i \in \Lambda\}$ be a rooted tree X -family. Let $RT(X)$ be the set of resolved rooted triplets from X . Define

$spt_{\mathcal{D}} : RT(X) \rightarrow \mathbb{R}$ as follows: For each distinct a, b, c in X let

$$\text{den}(a, b, c) = |\{i \in \Lambda : \{a, b, c\} \subseteq X_i\}|,$$

$$\text{num}(ab|c) = |\{i \in \Lambda : ab|c \in T_i\}|,$$

$$spt_{\mathcal{D}}(ab|c) = \begin{cases} \frac{\text{num}(ab|c)}{\text{den}(a, b, c)} & \text{if } \text{den}(a, b, c) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Call $spt_{\mathcal{D}}(ab|c)$ the *normalized triplet support* for $ab|c$ in \mathcal{D} .

Lemma 3.1: Let \mathcal{D} be a rooted tree X -family. For all distinct a, b, c in X

- (1) $0 \leq spt_{\mathcal{D}}(ab|c) \leq 1$,
- (2) $spt_{\mathcal{D}}(ab|c) = spt_{\mathcal{D}}(ba|c)$,
- (3) $spt_{\mathcal{D}}(ab|c) + spt_{\mathcal{D}}(ac|b) + spt_{\mathcal{D}}(bc|a) \leq 1$.

The proof is immediate.

Suppose $U \subseteq X$ and $|U| \geq 3$. Given distinct $a \in U$, $b \in U$, define

$$mspt_U(a, b) = \max\{spt_{\mathcal{D}}(ab|c) : c \in U, c \neq a, c \neq b\}.$$

We call $mspt_U(a, b)$ the “maximum normalized triplet support for $\{a, b\}$ on U .”

The *Normalized Triplet Supertree (NTS)* for \mathcal{D} is defined to be the minimum threshold tree for $(X, mspt_U)$ where U ranges over all subsets of X such that $|U| \geq 3$. More explicitly, if S denotes the NTS for \mathcal{D} , then S is the smallest collection of subsets of X such that

- 1) $X \in S$.
- 2) If $U \in S$ has exactly two elements, $U = \{u, v\}$, then $\{u\} \in S$ and $\{v\} \in S$.
- 3) If $U \in S$ and $|U| \geq 3$, form the complete graph $K(U)$ with vertex set U , where each edge $\{u, v\}$ has weighting $mspt_U(u, v)$. Let $\tau(U) = \tau(K(U))$ be the minimal disconnection threshold for $K(U)$. Then $K(U)_{\tau(U)}$ is disconnected. For each component $C = (U', E')$ of $K(U)_{\tau(U)}$, $U' \in S$.

Observe that we may omit any edge $\{a, b\}$ from $K(U)$ such that $mspt_U(a, b) = 0$, since $\tau(U) \geq 0$.

We call the graph $K(U)$ with weighting $mspt_U$ the *Aho graph* of U .

For example, suppose $X = \{a, b, c, d, e\}$ and \mathcal{D} consists of the rooted triplets $ab|c$, $ab|d$, $ac|d$, $ad|e$, $ae|b$, $bc|a$, $bc|e$, $de|a$, and $de|a$. Note that \mathcal{D} is a multiset and $de|a$ occurs twice. Consider $K(X)$. Then $\text{den}(a, d, e) = 3$ since three input triplets contained a, d , and e . Of these, two are $de|a$, whence $\text{num}(de|a) = 2$, and one is $ad|e$, whence $\text{num}(ad|e) = 1$. Hence $spt_{\mathcal{D}}(de|a) = 2/3$, $spt_{\mathcal{D}}(ad|e) = 1/3$, and $spt_{\mathcal{D}}(ae|d) = 0$. In like manner $spt_{\mathcal{D}}(ad|b) = 0$ and $spt_{\mathcal{D}}(ad|c) = 0$. Hence $mspt_X(a, d) = \max\{spt_{\mathcal{D}}(ad|b), spt_{\mathcal{D}}(ad|c), spt_{\mathcal{D}}(ad|e)\} = \max\{0, 0, 1/3\} = 1/3$.

Figure 1 shows the computation of the NTS for this example. The preceding paragraph calculated the weight $w_X(\{a, d\}) = mspt_X(a, d) = 1/3$ in Figure 1A. In a similar manner the weights of all edges in Figure 1A are computed. As in Section 2.2 the minimal threshold is found to be $2/3$, whence the clusters $\{d\}$ and $\{a, b, c, e\}$ belong to the NTS. In Figure 1B we have $K(U)$ for $U = \{a, b, c, e\}$. The weights of the edges are found as in the preceding paragraph. Note that no rooted triplet containing d is used in the computation of $mspt_U(a, b)$. Hence $mspt_U(a, b) = \max\{spt_{\mathcal{D}}(ab|c), spt_{\mathcal{D}}(ab|e)\} = \max\{1/2, 0\} = 1/2$ gives the weight of edge $\{a, b\}$ in Figure 1B. By contrast for Figure 1A the weight of the edge $\{a, b\}$ is instead $mspt_X(a, b) =$

$\max\{spt_{\mathcal{D}}(ab|c), spt_{\mathcal{D}}(ab|d), spt_{\mathcal{D}}(ab|e)\} = \max\{1/2, 1, 0\} = 1$. The minimal threshold is $\tau(U) = 1/2$. Figure 1C exhibits the NTS.

In general, the computation of the components of $K(U)_{\tau(U)}$ may be performed as follows. If $K(U)_0$ is disconnected, then $\tau(U) = 0$ and the components of $K(U)_0$ may be found directly. If $K(U)_0$ is connected, note that $K(U)_1$ is disconnected since for all $\{u, v\}$, $mspt_U(u, v) \leq 1$. Let $I_0 = [0, 1]$. Suppose $I_i = [a_i, b_i]$ is an interval such that $K(U)_{a_i}$ is connected and $K(U)_{b_i}$ is disconnected. It follows that $\tau(U) \in (a_i, b_i]$. Let $c_i = (a_i + b_i)/2$. If $K(U)_{c_i}$ is connected, let $I_{i+1} = [c_i, b_i]$, whereas if $K(U)_{c_i}$ is disconnected, let $I_{i+1} = [a_i, c_i]$. Define $[a_{i+1}, b_{i+1}] = I_{i+1}$. Then I_{i+1} is an interval with half the length of I_i such that $K(U)_{a_{i+1}}$ is connected and $K(U)_{b_{i+1}}$ is disconnected.

The minimum disconnection threshold $\tau(U)$ is a value $mspt_U(u, v)$ for some u and v in U . Suppose \mathcal{D} contains m input rooted trees. Then $mspt_U(u, v)$ is a rational number with denominator at most m . Any two distinct rational numbers with denominator at most m differ by at least $1/m^2$ since

$$|a/b - c/d| = |ad - bc|/|bd| \geq 1/|bd| \geq 1/m^2.$$

Thus when the length of $I_i = [a_i, b_i]$ is less than $1/m^2$, then I_i contains precisely one rational number a/b with $b \leq m$, and this value must be $\tau(U)$. In particular, it follows that $K(U)_{b_i} = K(U)_{\tau(U)}$ since if e is an edge and $mspt_U(e) > \tau(U)$, then $mspt_U(e)$ has form a/b with $b \leq m$, whence $mspt_U(e) > b_i$.

The following result shows that the computation of the NTS is fast:

Theorem 3.2: Let $\mathcal{D} = \{(T_i, X_i) : 1 \leq i \leq m\}$ be a rooted tree X -family, where $|X| = n$.

- (1) The computation of $spt_{\mathcal{D}}(xy|z)$ for all x, y, z in X takes time $O(n^4m)$.
- (2) The computation of the normalized triplet supertree S takes time $O(n^4m)$.

Proof: For (1), for each input tree T_i , there are $O(n)$ clusters. For each cluster U , every possible x and y in U , $z \notin U$ lead to a rooted triplet $xy|z$, so the computation of such triplets takes time $O(n^3)$ for each of $O(n)$ clusters. Hence the time to compute the rooted triplets for each input tree is $O(n^4)$ and the total time to find all the rooted triplets is $O(n^4m)$. From these counts of all rooted triplets, the time to compute $spt_{\mathcal{D}}(xy|z)$ is another $O(n^3)$. Hence the total time required is $O(n^4m) + O(n^3) = O(n^4m)$. Note, however, that if each input tree T_i is instead described merely by its $O(n^3)$ rooted triplets, then the time can be reduced to $O(n^3m)$.

For (2), suppose all $spt_{\mathcal{D}}(xy|z)$ are known. For each cluster $U \in S$ and for each x and y in U , the computation of $mspt_U(x, y)$ takes time $O(n)$. Hence the computation of $K(U)$ including its weights takes time $O(n^3)$. Each weight $mspt_U(x, y)$ is a rational number between 0 and 1 with denominator at most m since no 3-subset $\{x, y, z\}$ occurs more than m times. Computation of the minimum threshold $\tau(U)$ by bisection hence requires at most $O(\log_2(m))$ steps to shorten the interval containing $\tau(U)$ to a length less than $1/m^2$. Each such step requires the evaluation of connectivity of $K(U)_{\gamma}$ for some γ , which can be done by a depth-first search requiring time $O(n^2)$. Hence for each U , the computation of the children of U can be accomplished in time $O(n^2 \log_2(m))$. It follows that the computation of $K(U)$ and the children of U can be done in time $O(n^3) + O(n^2 \log_2(m))$. Since S has $O(n)$ clusters, the total time to compute S after all

$spt_{\mathcal{D}}(xy|z)$ are known is $O(n^4) + O(n^3 \log_2(m))$.

Hence the total time to find S is $O(n^4m) + O(n^4) + O(n^3 \log_2(m)) = O(n^4m)$. ■

One advantage of the NTS is that sometimes we can predict the NTS. This result will require the following definition:

Let (T, X) be a rooted tree. Define $spt_T : RT(X) \rightarrow \mathbb{R}$ as follows: For each resolved rooted triplet xyz let

$$spt_T(xy|z) = \begin{cases} 1 & \text{if } xyz \text{ is in } T \\ 0 & \text{else} \end{cases}$$

Thus spt_T is an indicator function which, for each resolved rooted triplet, tells whether the triplet is present in T .

It is known ([28], p. 119) that no two distinct rooted trees have exactly the same sets of resolved rooted triplets. It follows that if (T, X) and (W, X) are distinct rooted trees then $spt_T \neq spt_W$.

The first main theorem (Theorem 3.5) is that if for all x, y , and z , $spt_T(xy|z)$ and $spt_{\mathcal{D}}(xy|z)$ are close enough, then the NTS will be equal to T . Its proof requires the following lemma:

Lemma 3.3: Let \mathcal{D} be a rooted tree X -family. Suppose that (T, X) is a rooted tree. Assume that for all distinct x, y, z in X we have

$$|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| < 1/2.$$

Suppose that the tree T has a cluster U with exactly k children A_1, \dots, A_k , $k > 1$. Suppose $W \subseteq X$ is the union of at least two members of A_1, \dots, A_k ; by renumbering say $W = A_1 \cup A_2 \cup \dots \cup A_m$ with $m > 1$. Let $K(W)$ be the Aho graph of W . Then

- (1) $K(W)_{1/2}$ is disconnected and its components are exactly A_1, \dots, A_m .
- (2) If $\tau(W)$ is the minimal disconnection threshold, then each component of $K(W)_{\tau}$ is a union of some collection of A_1, \dots, A_m .

Proof: Suppose a and b are in A_j , $j \leq m$. We show that a and b lie in the same component of $K_{1/2}(W)$. Assume without loss of generality that $j = 1$ and choose $c \in A_2$. Then $ab|c$ is in T . Hence $spt_T(ab|c) = 1$, whence $spt_{\mathcal{D}}(ab|c) > 1/2$. Hence $mspt_W(a, b) > 1/2$ and there is an edge $\{a, b\}$ in $K(W)_{1/2}$, whence a and b are in the same component of $K(W)_{1/2}$.

Suppose a and b are in different A_j 's, $j \leq m$. We show that a and b lie in different components of $K(W)_{1/2}$. For every $c \in W$ distinct from a and b , we have $spt_T(ab|c) = 0$ since a and b are in different A_j 's. Hence $spt_{\mathcal{D}}(ab|c) < 1/2$. It follows that $mspt_W(a, b) < 1/2$, whence $K(W)_{1/2}$ does not contain an edge $\{a, b\}$.

It follows that there are no edges between members of different A_j 's. Hence the components of $K(W)_{1/2}$ are exactly A_1, \dots, A_m . This proves (1).

For (2), note since $K(W)_{1/2}$ is disconnected, it follows that $\tau(W) \leq 1/2$. Since decreasing the threshold τ from $1/2$ to $\tau(W)$ can only add edges to $K(W)_{1/2}$, it follows that for each i , A_i is completely contained in some component of $K(W)_{\tau(W)}$. This proves (2). ■

Corollary 3.4: Let \mathcal{D} be a rooted tree X -family. Suppose that (T, X) is a rooted tree. Assume that for all distinct x, y, z in X we have

$$|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| < 1/2.$$

Suppose that the tree T has a cluster U with exactly 2 children A_1 and A_2 . Let $\tau = \tau(U)$ be the minimal disconnection threshold. Then $K(U)_{\tau}$ has exactly the two components A_1 and A_2 .

Proof: By Lemma 3.3 (2), A_1 is completely contained in some component of $K(U)_\tau$, and A_2 is completely contained in some component of $K(U)_\tau$. If these were the same component of $K(U)_\tau$, then there would be only one component, contradicting that $K(U)_\tau$ is disconnected. Hence the components of $K(U)_\tau$ must be exactly A_1 and A_2 . ■

Theorem 3.5: Let \mathcal{D} be a rooted tree X -family. Suppose that (T, X) is a binary rooted tree. Assume that for all distinct x, y, z in X we have

$$|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| < 1/2.$$

Let S denote the NTS. Then $S = T$.

Proof: In T the cluster X has exactly two children A_1 and A_2 since T is binary. By Cor 3.4 it follows that the children of X in S are also A_1 and A_2 . Generally, if U is a cluster of both T and S and U is not a leaf of T , then U has exactly two children since T is binary, and by Cor 3.4, U has the same children in S . The result follows by induction. ■

If T is not binary, then Theorem 3.5 does not apply and in fact even if $|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| < 1/2$ the NTS S need not equal T . The following Theorem 3.6 shows that S is nevertheless closely related to T in this case. Note that Theorem 3.6 reduces to Theorem 3.5 if T is binary.

Theorem 3.6: Let \mathcal{D} be a rooted tree X -family. Suppose that (T, X) is a rooted tree. Assume that for all distinct x, y, z in X we have

$$|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| < 1/2.$$

Let S denote the NTS. Each cluster W of S has one of the following forms:

- (1) W is a cluster of T ; or
- (2) there exists a cluster U of T with children A_1, \dots, A_k in T ($k > 2$), and there is a subcollection A_{i_1}, \dots, A_{i_m} for some m satisfying $m < k$ such that $W = \cup[A_{i_j} : j = 1, \dots, m]$.

Proof: This result is proved by induction using Lemma 3.3. The details are omitted. ■

The final result of this section shows that if T is not binary but $|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| < 1/2$, an alternative construction is still guaranteed to yield T .

Theorem 3.7: Let \mathcal{D} be a rooted tree X -family. Suppose that (T, X) is a rooted tree. Assume that for all distinct x, y, z in X we have

$$|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| < 1/2.$$

Define S to be the smallest collection of subsets of X such that the following hold:

- (1) $X \in S$.
- (2) If $U \in S$ has exactly two elements, $U = \{u, v\}$, then $\{u\} \in S$ and $\{v\} \in S$.
- (3) If $U \in S$ and $|U| \geq 3$, form the complete graph $K(U)$ with vertex set U , where each edge $\{u, v\}$ has weight $mspt_U(u, v)$. For each component C of $K(U)_{1/2}$, the set of vertices of C is in S .

Then $S = T$.

Proof: By Lemma 3.3(1), the components of $K(X)_{1/2}$ are precisely the children of X in T . Assume inductively that each member so far placed into S is a cluster of T . Suppose $U \in S$. Then by Lemma 3.3(1), the components of $K(U)_{1/2}$ are precisely the children of U in T . Hence the result follows by induction. ■

IV. ROBUSTNESS RADIUS

This section defines the ‘‘robustness radius’’ R of a supertree method. The radius yields a way to measure how reliably the method behaves when given dense data such that all rooted triplets favor a particular binary tree T . We prove that the best possible robustness radius is $R = 1/2$ and that for NTS the radius is exactly $1/2$. Hence NTS is optimal by that criterion. While there are other methods with optimal robustness radius $1/2$, Section 5 will show that many familiar methods are suboptimal by that criterion.

Following Atteson [3], say that the l_∞ radius R of a method of supertree construction is α provided that,

- 1) whenever \mathcal{D} is a rooted tree X -family and (T, X) is a binary rooted tree such that, for all distinct a, b, c in X ,

$$|spt_{\mathcal{D}}(ab|c) - spt_T(ab|c)| < \alpha$$

then the method outputs T ; and

- 2) for every $\beta > \alpha$ there exists a binary rooted tree (T, X) and a rooted tree X -family \mathcal{D} such that, for all distinct a, b, c in X ,

$$|spt_{\mathcal{D}}(ab|c) - spt_T(ab|c)| < \beta$$

but the method does not output T .

Less formally we call such α the *robustness radius* R .

If $R > 0$, then whenever $spt_{\mathcal{D}}$ is uniformly sufficiently close to spt_T on all resolved rooted triplets, then the method outputs T .

The following result shows that the best possible robustness radius for any method is $R = 1/2$.

Theorem 4.1: No supertree method has robustness radius greater than $1/2$.

Proof: Let $X = \{a, b, c\}$. Let $T = ab|c$ and $W = bc|a$. Suppose that \mathcal{D} consists of n copies of T and m copies of W . One computes that

$$spt_{\mathcal{D}}(ab|c) = n/(n+m) = mspt_X(a, b),$$

$$spt_{\mathcal{D}}(bc|a) = m/(n+m) = mspt_X(b, c),$$

$$spt_{\mathcal{D}}(ac|b) = 0. \text{ Since } spt_T(ab|c) = 1 \text{ we see}$$

$$|spt_{\mathcal{D}}(ab|c) - spt_T(ab|c)| = m/(n+m).$$

Since $spt_T(bc|a) = 0$, it follows

$$|spt_{\mathcal{D}}(bc|a) - spt_T(bc|a)| = m/(n+m).$$

Similarly

$$|spt_{\mathcal{D}}(ac|b) - spt_T(ac|b)| = 0,$$

$$|spt_{\mathcal{D}}(ab|c) - spt_W(ab|c)| = n/(n+m),$$

$$|spt_{\mathcal{D}}(bc|a) - spt_W(bc|a)| = n/(n+m), \text{ and}$$

$$|spt_{\mathcal{D}}(ac|b) - spt_W(ac|b)| = 0.$$

Suppose for some $\alpha > 1/2$ a method yields a tree Y when, for all distinct x, y, z in X ,

$$|spt_{\mathcal{D}}(xy|z) - spt_Y(xy|z)| < \alpha.$$

We could choose $m > n$ with $m/(n+m) < \alpha$. Then since $n < m$ it would also follow that $n/(n+m) < \alpha$. Hence the method would have to output both T and W , which is impossible. ■

The next result is that NTS has optimal robustness radius.

Theorem 4.2: The NTS method has robustness radius $R = 1/2$.

Proof: By Theorem 3.5, we have $R \geq 1/2$. But by Theorem 4.1, $R \leq 1/2$. ■

Let $\mathcal{D} = \{(T_i, X_i) : i \in \Lambda\}$ be a rooted tree X -family. Say \mathcal{D} is *dense in X* if, for all distinct x, y, z in X , there exists $i \in \Lambda$ such that $\{x, y, z\} \subseteq X_i$.

The next result shows that Theorem 3.5 and the definition of robustness radius are moot unless \mathcal{D} is dense in X .

Lemma 4.3: Suppose that T is a binary rooted tree with leaf set X . Assume that there exists $\alpha < 1$ such that, for all distinct x, y, z in X , we have $|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| < \alpha$. Then \mathcal{D} is dense in X .

Proof: Note that, for any distinct x, y, z in X , since T is binary either $xy|z$ or $xz|y$ or $yz|x$ is in T . Suppose $xy|z$ is in T . Then $spt_T(xy|z) = 1$. Hence $spt_{\mathcal{D}}(xy|z) > 1 - \alpha > 0$. It follows that $den(x, y, z) > 0$, so there must exist i such that $\{x, y, z\} \subseteq X_i$. ■

V. ROBUSTNESS RADII OF SOME OTHER SUPERTREE AND CONSENSUS METHODS

In this section we consider some well-known supertree and consensus methods and we show that their robustness radii are disappointing. On the other hand, MRP using triplets has the maximal possible robustness radius.

The method MinCutSupertree is described in [27]. A modification due to Page was described in [22].

Theorem 5.1: For the MinCutSupertree methods (both original and modified) the robustness radius R satisfies $R = 0$.

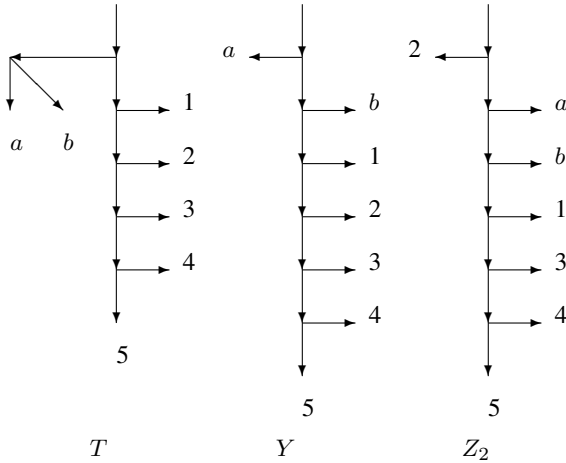


Fig. 2. Computation of the robustness radius R for MinCutSupertree. The example is given for $m = 5$. The input \mathcal{D} consists of y copies of Y , $t = 5y + 14$ copies of T , and one copy each of Z_k for $k = 1, \dots, 5$. Then for all u, v, w , $|spt_{\mathcal{D}}(uv|w) - spt_T(uv|w)| \leq (y + 4)/(6y + 19)$ but MinCutSupertree does not select T . Instead, the tree found by MinCutSupertree is Y . As y approaches ∞ , the example shows that $R \leq 1/6$.

Proof: Given a positive integer m , let $X = \{a, b, 1, 2, \dots, m\}$. See Figure 2 for the case $m = 5$. Let T be the rooted X -tree with the nontrivial clusters $\{a, b\}$, $\{1, 2, \dots, m\}$, $\{2, 3, \dots, m\}$, $\{3, 4, \dots, m\}$, \dots , $\{m-1, m\}$. Let Y be the rooted X -tree with nontrivial clusters $\{b, 1, 2, \dots, m\}$, $\{1, 2, \dots, m\}$, $\{2, 3, \dots, m\}$, \dots , $\{m-1, m\}$. Let \hat{i} indicate in a list that i is missing. For $i = 1, 2, \dots, m$ let Z_i be the rooted X -tree with nontrivial clusters $\{a, b, 1, \dots, \hat{i}, \dots, m\}$, $\{b, 1, \dots, \hat{i}, \dots, m\}$, $\{1, \dots, \hat{i}, \dots, m\}$, $\{2, 3, \dots, \hat{i}, \dots, m\}$, \dots , $\{m-1, m\}$. The input \mathcal{D} consists of t copies of T , y

copies of Y , and 1 copy each of Z_i for $i = 1, \dots, m$. Then $n = t + y + m$ is the number of input trees in \mathcal{D} . Each input tree has leaf set X .

We easily see that, if $y + m \geq 4$, then for all distinct u, v, w

$$|spt_{\mathcal{D}}(uv|w) - spt_T(uv|w)| \leq (y + m - 1)/n.$$

Our notation for MinCutSupertree is as in [27]. The children of X are found by computing the minimum cuts of the graph $X_{\mathcal{D}} = X_{\mathcal{D}}/E_{\mathcal{D}}^{m \times a}$, whose vertices are the members of X . The capacity of edge $\{u, v\}$ is the number of input trees in which u and v are both in a proper cluster. Hence

$\{a, b\}$ has capacity $t + m$.

$\{a, i\}$ has capacity $m - 1$ from all Z_j except Z_i .

$\{b, i\}$ has capacity $y + m - 1$ from Y and all Z_j except Z_i .

$\{i, j\}$ for $i < j$ has capacity $t + y + m - 2$ from T and Y and all Z_k except Z_i and Z_j .

It follows that the cut $\{a, b\}$ has total capacity $(m - 1)(m) + (y + m - 1)(m)$ by cutting all edges $\{a, i\}$ and $\{b, i\}$. The cut $\{a\}$ has total capacity $(t + m) + (m - 1)(m)$ by cutting edge $\{a, b\}$ and all edges $\{a, i\}$. Hence the capacity of cut $\{a\}$ is smaller than the capacity of cut $\{a, b\}$ provided

$$(t + m) + (m - 1)(m) < (m - 1)(m) + (y + m - 1)(m)$$

in which case $\{a, b\}$ is not a minimal cut. Equivalently, this happens when $t < ym + m^2 - 2m$. Set $t = ym + m^2 - 2m - 1$. If this holds and $t > 0$, $y > 0$, $m > 0$, then $\{a, b\}$ is not a minimal cut for $X_{\mathcal{D}}$, whence in the MinCutSupertree the children of X are not $\{a, b\}$ and $\{1, 2, \dots, m\}$. Hence under these conditions, T is not the MinCutSupertree.

On the other hand, under these conditions and assuming $y + m \geq 4$ we have that for all u, v, w

$$\begin{aligned} |spt_{\mathcal{D}}(uv|w) - spt_T(uv|w)| &\leq (y + m - 1)/n \\ &= (y + m - 1)/(ym + m^2 - 2m - 1 + y + m). \end{aligned}$$

For fixed m as y increases without bound, note that the right side has limit $1/(m + 1)$. It follows that, if $r > 1/(m + 1)$ then there exist input data \mathcal{D} for which the binary tree T is not returned by MinCutSupertree yet for all u, v, w , $|spt_{\mathcal{D}}(uv|w) - spt_T(uv|w)| < r$.

Hence the robustness radius R satisfies $R \leq 1/(m + 1)$ for all m , whence $R = 0$. Informally, we see that MinCutSupertree favors tree Y over tree T even though most rooted triplets match those of T .

In the Page version of the algorithm, one notes that every edge in $X_{\mathcal{D}}$ is contradicted since if u and v appear in a proper cluster of some input tree, there is also an input tree containing u and v in which they do not appear in a proper cluster. Hence the removal of contradicted edges leaves no edges at all, and the algorithm continues exactly as in the original MinCutSupertree algorithm. ■

The familiar consensus trees are defined in [28], p. 53, or [17], p. 521.

Theorem 5.2: For each of the majority rule consensus tree M , the strict consensus tree S , and the Adams consensus tree A , the robustness radius R satisfies $R = 0$.

Proof: Let $n \geq 3$ be a positive integer; we show $R \leq 1/n$. See Figure 3 for the case $n = 4$. Let $m = n + 1$, and let $X = \{1, 2, \dots, m\}$. For $i = 2, \dots, m$ let T_i be the rooted tree with leaf set X and nonsingleton clusters $\{1, i\}$, $\{1, 2, i\}$, $\{1, 2, 3, i\}$, \dots , $\{1, 2, \dots, i - 1, i\}$, $\{1, 2, \dots, i + 1\}$, \dots , X .

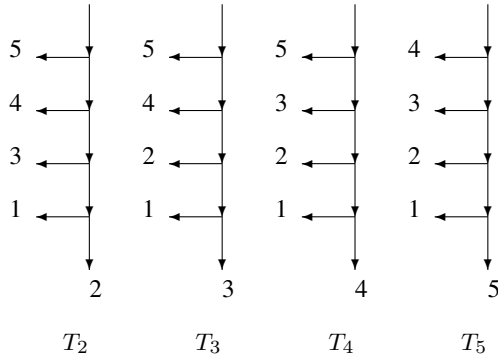


Fig. 3. Computation of the robustness radius R for several consensus methods. The example is given for $n = 4$. The input \mathcal{D} consists of 4 trees. Then for all x, y , and z , $|spt_{\mathcal{D}}(xy|z) - spt_{T_2}(xy|z)| \leq 1/4$ but the consensus methods do not select T_2 . The example shows that $R \leq 1/4$.

Let $\mathcal{D} = \{T_2, T_3, \dots, T_m\}$. If $x < y < z$ then $xy|z$ is in T_i for $i \neq z$. Hence if $x < y < z$ it follows $spt_{\mathcal{D}}(xy|z) = (n - 1)/n$. For other rooted triplets the support is either 0 or $1/n$.

Let $T = T_2$, so T has nonsingleton clusters $\{1, 2\}$, $\{1, 2, 3\}$, \dots , $\{1, 2, \dots, m\}$. If we assume $x < y$ then $xy|z$ is in T iff $x < y < z$. Hence $spt_T(xy|z) = 1$ if $x < y < z$ and $spt_T(xy|z) = 0$ if $x < y$ but $z < y$, so

$$|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| \leq 1/n$$

for all x, y, z . Hence if the method does not yield T , then the robustness radius R must satisfy $R \leq 1/n$.

For the majority rule consensus tree M , if $n \geq 3$, note that the cluster $\{1, 2\}$ arises only once, so $\{1, 2\}$ is not in M . Since $\{1, 2\}$ is a cluster of T , it follows $M \neq T$. Hence $R \leq 1/n$. Since n is arbitrary, it follows $R \leq 0$. Since R must be nonnegative, $R = 0$.

The strict consensus tree S for this example is the star tree since only the singleton clusters and X lie in each T_i . Hence $R \leq 1/n$ and as above $R = 0$.

For the Adams consensus tree A , to find the children of the root X , we form a graph G with vertex set X and with an edge $\{x, y\}$ iff for each i , x and y are together in some proper cluster of T_i . In T_m note that $n = m - 1$ lies only in the clusters X and $\{n\}$. Hence G contains no edge with endpoint n . It follows that $\{1, 2, \dots, n\}$ is not a cluster of A , whence $A \neq T$. Thus $R \leq 1/n$ and as above $R = 0$.

It is perhaps surprising that the majority rule consensus tree has robustness radius 0. The underlying reason seems to be that its calculation deals with the frequency of clusters while the robustness radius deals with the frequency of rooted triplets.

Theorem 5.3: For MRP the robustness radius R satisfies $R \leq 1/100$.

Proof: For each natural number $m > 3$ let $X = \{1, 2, \dots, 2m - 1\}$. See Figure 4 for the case $m = 5$. Consider the rooted tree X -family \mathcal{D} consisting of m trees T_i for $i = 1, \dots, m$, defined as follows: The nontrivial clusters of T_1 are precisely $\{2, 3, \dots, 2m - 1\}$, $\{3, 4, \dots, 2m - 1\}$, \dots , $\{2m - 2, 2m - 1\}$. For $k \neq 1, k \neq m - 1$, the nontrivial clusters of T_k are $\{2, 3, \dots, 2m - 1\}$, $\{3, 4, 5, \dots, 2m - 1\}$, \dots , $\{k - 1, k, k + 1, \dots, 2m - 1\}$, $\{k - 1, k + 1, k + 2, \dots, 2m - 1\}$, $\{k - 1, k + 2, k + 3, \dots, 2m - 1\}$, $\{k - 1, k + 3, k + 4, \dots, 2m - 1\}$, \dots , $\{k - 1, 2m - 1\}$. The nontrivial

clusters of T_{m-1} are $\{2, 3, \dots, 2m - 1\}$, $\{3, 4, \dots, 2m - 1\}$, \dots , $\{m - 2, m - 1, m, m + 1, \dots, 2m - 1\}$, $\{m - 2, m, m + 1, m + 2, \dots, 2m - 1\}$, $\{m - 2, m, m + 1, m + 2, \dots, 2m - 2\}$, $\{m - 2, m, m + 1, m + 2, \dots, 2m - 3\}$, $\{m - 2, m, m + 1, m + 2, \dots, 2m - 4\}$, \dots , $\{m - 2, m\}$.

Let $T = T_1$. It is readily checked that $|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| \leq 1/m$ for all x, y, z . For $m = 3, 4, 5, 6, 7, 8, 20$ it has been verified using PAUP* [31] with branch-and-bound that MRP leads to a unique tree which is, however, different from T . For $m = 30, 50, 90, 100$ the same has likewise been verified using heuristic search. Hence the robustness radius R of MRP satisfies $R \leq 1/100$.

I conjecture that in fact the example in Theorem 5.3 shows that for all $m > 3$, $R \leq 1/m$, whence $R = 0$ for MRP.

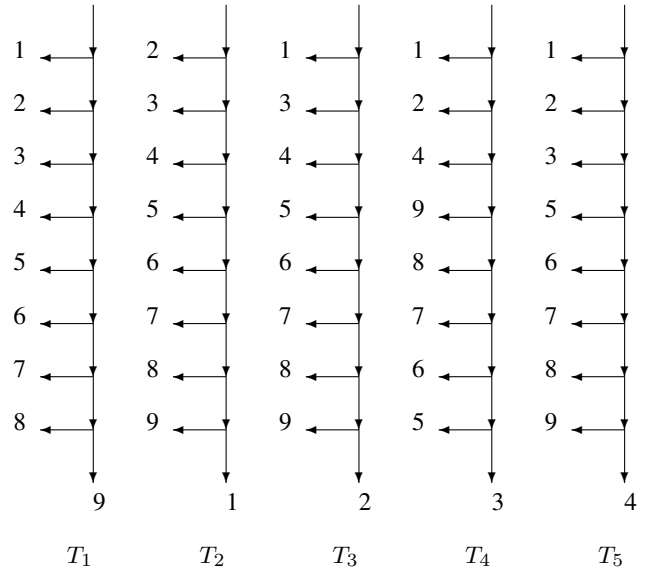


Fig. 4. Computation of the robustness radius R for MRP. The example is given for $m = 5$. The input \mathcal{D} consists of 5 trees. Then for all x, y , and z , $|spt_{\mathcal{D}}(xy|z) - spt_{T_1}(xy|z)| \leq 1/5$ but MRP does not select T_1 . Instead, the tree found by MRP is like T_1 but with 3 and 4 interchanged. The example shows that $R \leq 1/5$.

An alternative to MRP is triplet MRP, in which a different matrix representation M is used from the matrix representation in MRP [24]. Given the input trees \mathcal{D} , for each input tree T_i in \mathcal{D} , for each rooted triplet $ab|c$ in T_i , M will have one column which contains 0 in the rows for the outgroup and for c , 1 in the rows for taxa a and b , and ? in the rows for each other taxon. The method seeks a maximum parsimony tree.

Theorem 5.4: Triplet MRP has robustness radius $R = 1/2$.

Proof: Let \mathcal{D} be a rooted tree X -family and let T be a binary rooted X -tree. Assume that, for all distinct x, y, z , $|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| < 1/2$.

For each rooted triplet $ab|c$, let there be $m_{ab|c}$ copies of $ab|c$ in \mathcal{D} and m_{abc} copies of the star triplet abc in \mathcal{D} .

If $xy|z$ in T , then $spt_T(xy|z) = 1$. Hence $spt_{\mathcal{D}}(xy|z) > 1/2$ whence

$$m_{xy|z} / (m_{xy|z} + m_{xz|y} + m_{yz|x} + m_{xyz}) > 1/2$$

so

$$m_{xy|z} > m_{xz|y} + m_{yz|x} + m_{xyz}.$$

In particular, $m_{xy|z} > m_{xz|y}$ and $m_{xy|z} > m_{yz|x}$ if $xy|z$ is in T .

To see that T is a maximum parsimony tree, we suppose W is a binary X -tree, $W \neq T$, and we show that T requires fewer parsimony steps than W according to the matrix M .

For each rooted triplet $ab|c$ if $ab|c$ is in W , then the corresponding column contributes exactly one step in computing the parsimony since all vertices separated from the outgroup by the most recent common ancestor of a and b may receive 1 and all other vertices may receive 0. If, instead, $ab|c$ is not true in W , then the corresponding column contributes exactly two steps in computing the parsimony by assigning 0 to all vertices except a and b . Thus the parsimony score of W is

$$k(W) = \sum [m_{ab|c} : spt_W(ab|c) = 1] + \sum [2m_{ab|c} : spt_W(ab|c) = 0].$$

Similarly,

$$k(T) = \sum [m_{ab|c} : spt_T(ab|c) = 1] + \sum [2m_{ab|c} : spt_T(ab|c) = 0].$$

I claim that $k(W) > k(T)$ if $W \neq T$. This follows by a comparison of the corresponding terms in the inequality for each 3-subset $\{a, b, c\}$. Since T is binary, we may rename the elements if necessary so as to assume $spt_T(ab|c) = 1$. If $spt_W(ab|c) = 1$ as well, then the contributions to $k(W)$ and $k(T)$ are the same. If $spt_W(ab|c) = 0$, suppose $spt_W(ac|b) = 1$. Then the contribution to $k(W)$ is $m_{ac|b} + 2m_{ab|c} + 2m_{bc|a}$ while the contribution to $k(T)$ is $m_{ab|c} + 2m_{ac|b} + 2m_{bc|a}$. Hence the contribution to $k(W) - k(T)$ is $m_{ab|c} - m_{ac|b} > 0$. ■

The results in this section show that several familiar supertree methods have robustness radii that are far from optimal. For many methods, nevertheless, a simple pre-processing of the data could correct the difficulty and lead to robustness radius $1/2$. Suppose \mathcal{D} is a rooted tree X -family. One could compute (in polynomial time by Theorem 3.2) for each x, y , and z in X the value $spt_{\mathcal{D}}(xy|z)$. For each 3-set $\{x, y, z\}$ select the rooted triplet $xy|z$, $xz|y$, or $yz|x$ with support greater than 0.5 , if such exists, and let $RT(\mathcal{D})$ denote the set of rooted triplets so obtained. It is easy to check whether there exists a rooted tree T whose set of rooted triplets coincides with $RT(\mathcal{D})$, using, for example, the algorithm BUILD [2]. If so, then revise the supertree method to return T ; if not, proceed as usual with the supertree method. In most situations, of course, no such tree T will exist. The resulting combined method will have the optimal robustness radius $1/2$.

VI. MORE PROPERTIES OF NTS

Wilkinson *et al.* [32] propose some “desiderata for liberal supertrees.” In this section we show that NTS satisfies many of these desiderata.

(1) **Shapeless.** The method should not be biased by the shape of input trees. Note that if $\mathcal{D} = \{(T_i, X_i) : i \in \Lambda\}$ is a rooted tree X -family, then the NTS for \mathcal{D} is precisely the same as that for \mathcal{D}' where \mathcal{D}' is the multiset of triplets obtained by replacing each T_i by the set of its rooted triplets and star triplets. Hence we may always assume that the rooted tree X -family merely consists of triplets, and the shape of the trees in \mathcal{D} is not directly relevant.

(2) **Order invariance.** The method should not be influenced by the order in which members of \mathcal{D} are introduced, or the order of leaves in the adjacency matrix of an input tree. This property of NTS is obvious.

(3) **Uniqueness.** The method should give a unique answer, which is obvious for NTS.

(4) **Plenary.** The resulting supertree should contain all the leaves of the input trees. This is obvious for NTS.

(5) **Weightable.** The method should allow different input trees to be weighted. In fact, if $\mathcal{D} = \{(T_i, X_i) : i \in \Lambda\}$ with (T_i, X_i) weighted by $\gamma_i > 0$, then for each distinct a, b, c in X , redefine $spt_{\mathcal{D}}(ab|c)$ as the ratio of

$$\begin{aligned} \text{num}(ab|c) &= \sum \{\gamma_i : i \in \Lambda, ab|c \in T_i\}, \text{ by} \\ \text{den}(a, b, c) &= \sum \{\gamma_i : i \in \Lambda, \{a, b, c\} \subseteq X_i\}. \end{aligned}$$

(6) **Speed.** NTS has polynomial time-complexity, as seen in Theorem 3.2.

(7) **Assessable.** The method should allow a measure of the amount of support of the output supertree. The smallest number α such that for all distinct x, y, z ,

$$|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)| \leq \alpha$$

gives a measure of the overall support for the tree T . A smaller α means a better fit. This aspect will be discussed further in Section 8.

(8) **Pareto.** Relationships in all input trees should appear in the output. For this property, NTS does not quite satisfy the desired property. We nevertheless have the following results:

Theorem 6.1: Suppose that \mathcal{D} is a rooted tree X -family and contains at least one input tree containing all the taxa X . Suppose $spt_{\mathcal{D}}(ab|c) = 1$. Then $ab|c$ is in the NTS.

Proof: In the construction of the NTS, let U be the smallest cluster containing $\{a, b, c\}$. Note that $mspt_U(a, b) = 1$ since $spt_{\mathcal{D}}(ab|c) = 1$. Suppose there are m input trees. I claim that the minimal disconnection threshold $\tau(U)$ satisfies $\tau(U) \leq 1 - 1/m$. To see this, suppose W is an input tree containing all members of U (which exists since there exists an input tree containing all members of X). Then $W|U$ has children A_1, A_2, \dots, A_k for some $k > 1$. Suppose u and v are in distinct A_i 's. Then there is no $w \in U$ such that $uv|w$ in $W|U$. Hence whenever u and v are in distinct A_i 's, it follows that $spt_{\mathcal{D}}(uv|w) \leq 1 - 1/m$ since there is at least one tree W in which $uv|w$ fails. In particular, for $\tau = 1 - 1/m$ we have no edge in A_{τ} between members of different A_i 's. Since $k > 1$, A_{τ} is disconnected. But $mspt_U(a, b) = 1$, whence a and b are in the same component of A_{τ} , whence they are in the same component of $A_{\tau(U)}$ where $\tau(U)$ is the minimal disconnection threshold. Since U is the smallest cluster containing a, b , and c , it follows that c is not in that same component, whence $ab|c$ is in the NTS. ■

Corollary 6.2: Suppose that \mathcal{D} consists only of trees each with the leaf set X . If $spt_{\mathcal{D}}(ab|c) = 1$, then $ab|c$ is in the NTS.

Corollary 6.2 applies, for example, when the NTS is used as a consensus tree for input trees all containing all the members of X , as opposed to only a subset of X .

Theorem 6.1 is not true without the assumption in the first sentence. For example, let $X = \{a, b, c, d\}$. Let \mathcal{D} consist of one copy each of the rooted triplet trees $ab|c$, $bc|d$, $cd|a$, and $ad|b$. Since each 3-set occurs exactly once, we have $mspt_X(a, b) = mspt_X(b, c) = mspt_X(c, d) = mspt_X(a, d) = 1$. Hence $K(X)$ has edges $\{a, b\}$, $\{b, c\}$, $\{c, d\}$, and $\{a, d\}$ each with weight 1, and the NTS is the star tree.

VII. GEOMETRIC INTERPRETATION

The definition of the NTS has a natural geometric interpretation in a certain high-dimensional Euclidean space. In this section we show that NTS in fact solves a geometric optimization problem in this space, called the Weak Closest Tree Problem. Thus NTS

is not merely a plausible heuristic method for finding a supertree; instead, NTS is a method solving a global optimization problem. A stronger version of the problem, however, remains open.

Recall that $RT(X)$ denotes the set of all rooted triplets from the set X . The total number of rooted triplets is

$$m = |RT(X)| = 3 \binom{n}{3} = n(n-1)(n-2)/2.$$

List the members of $RT(X)$ in some arbitrary but fixed order. Let $H_X = [0, 1]^m$ be the m -dimensional *triplet hypercube* in which each coordinate corresponds to a rooted triplet $ab|c$. If $u \in H_X$, we will write its coordinate corresponding to the rooted triplet $ab|c$ as $u_{ab|c}$. A *corner* of H_X is a point $u \in H_X$ such that for each $ab|c$ we have $u_{ab|c}$ is 0 or 1. The l_∞ norm on \mathbb{R}^m is defined by

$$\|u-v\|_\infty = \max\{|u_{ab|c}-v_{ab|c}| : a, b, c \text{ are distinct elements of } X\}$$

and makes \mathbb{R}^m a normed vector space. We will regard H_X as a subspace of \mathbb{R}^m with the l_∞ norm.

H_X contains a kind of rooted-triplet-landscape space. To every rooted tree X -family \mathcal{D} there is a point $spt_{\mathcal{D}} \in H_X$ given by $(spt_{\mathcal{D}})_{ab|c} = spt_{\mathcal{D}}(ab|c)$. To any rooted X -tree T there corresponds $spt_T \in H_X$ given by $(spt_T)_{ab|c} = spt_T(ab|c)$. Since each value of $spt_T(ab|c)$ is either 0 or 1, spt_T is a corner of H_X . If T and T' are distinct rooted X -trees, then they differ on some rooted triplet; i.e., there exists $ab|c$ in $RT(X)$ such that either $ab|c \in T$ and $ab|c \notin T'$, or $ab|c \in T'$ and $ab|c \notin T$. Hence no two distinct trees correspond to the same corner.

Not all corners correspond to trees. For example, if u is a corner point and there exist a, b, c such that $u_{ab|c} = u_{ac|b} = 1$, then there is no tree T such that $u = spt_T$. We see that the elements spt_T are spread out among the corners of H_X .

Theorem 3.5 may be restated as follows:

Theorem 7.1: Let \mathcal{D} be a rooted tree X -family. Suppose that (T, X) is a binary rooted tree. Assume that $\|spt_{\mathcal{D}} - spt_T\|_\infty < 1/2$. Let S denote the NTS. Then $S = T$.

Since spt_T is a corner of H_X , the set

$$B_{1/2}(T) = \{u \in H_X : \|u - spt_T\|_\infty < 1/2\}$$

is a rectangular box of dimension m such that each side has length $1/2$. Theorem 7.1 asserts that if $spt_{\mathcal{D}}$ lies in $B_{1/2}(T)$, then the NTS is T .

A natural problem that arises in this context is:

Closest Tree Problem. Let \mathcal{D} be a rooted tree X -family. Find a rooted tree (T, X) that minimizes $\|spt_{\mathcal{D}} - spt_T\|_\infty$.

By Theorem 7.1, if S is the NTS, then S solves the Closest Tree Problem provided the solution T is binary and satisfies $\|spt_{\mathcal{D}} - spt_T\|_\infty < 1/2$. Examples show, however, that the NTS need not solve the Closest Tree Problem more generally.

Note that $\|spt_{\mathcal{D}} - spt_T\|_\infty = \max\{|spt_{\mathcal{D}}(xy|z) - spt_T(xy|z)|\} = \max\{\max\{spt_{\mathcal{D}}(xy|z) : xy|z \notin T\}, \max\{(1 - spt_{\mathcal{D}}(xy|z)) : xy|z \in T\}\}$.

Define the *weak distance* of T from \mathcal{D} by

$$D_{\mathcal{D}}^w(T) := \max\{spt_{\mathcal{D}}(xy|z) : xy|z \notin T\}.$$

Observe that for every T , $D_{\mathcal{D}}^w(T) \leq \|spt_{\mathcal{D}} - spt_T\|_\infty$.

For each rooted triplet $xy|z$ in T , there are two rooted triplets $xz|y$ and $yz|x$ not in T . Hence $D_{\mathcal{D}}^w(T)$ in some sense deals with

two rooted triplets for every one rooted triplet dealt with by the omitted expression $\max\{(1 - spt_{\mathcal{D}}(xy|z)) : xy|z \in T\}$.

The definition of $D_{\mathcal{D}}^w(T)$ leads to the following related problem:

Weak Closest Tree Problem. Let \mathcal{D} be a rooted tree X -family. Find a rooted tree (T, X) that minimizes $D_{\mathcal{D}}^w(T)$.

We shall see that the NTS S solves the Weak Closest Tree Problem. In the process, we observe in Lemma 7.2 that $D_{\mathcal{D}}^w(S)$ is the largest minimum disconnection threshold that arises in the computation of the NTS.

Lemma 7.2: Let \mathcal{D} be a rooted tree X -family. Let S be the NTS for \mathcal{D} . If $U \in S$ satisfies $|U| \geq 3$, let $\tau(U)$ denote the minimal disconnection threshold for $K(U)$. Let $\bar{\tau} = \max\{\tau(U) : U \in S, |U| \geq 3\}$. Then $D_{\mathcal{D}}^w(S) = \bar{\tau}$.

Proof: Suppose $xy|z$ is not in S . Let U be the smallest cluster of S that contains x, y , and z . Then $K(U)_{\tau(U)}$ places x and y in different components. It follows that $spt_{\mathcal{D}}(xy|z) \leq \tau(U)$. Hence $D_{\mathcal{D}}^w(S) \leq \bar{\tau}$.

Conversely, choose a cluster U of S such that $\tau(U) = \bar{\tau}$. Since $\tau(U)$ is the minimal disconnection threshold, there exists a component A of $K(U)_{\tau(U)}$ such that there exist $a \in A$ and $b \in U - A$ such that $mspt_U(a, b) = \tau(U)$. Hence there exists $c \in U$ such that $\tau(U) = mspt_U(a, b) = spt_{\mathcal{D}}(ab|c)$. Since a and b are in different children of U in S , it follows $ab|c \notin S$. Hence $\bar{\tau} \leq D_{\mathcal{D}}^w(S)$. ■

Theorem 7.3: NTS solves the Weak Closest Tree Problem.

Proof: Let $\tau(U)$ and $\bar{\tau}$ be as in Lemma 7.2. Let the rooted X -tree T minimize $D_{\mathcal{D}}^w(T)$ among all rooted X -trees. Let U be a cluster of S such that $\tau(U)$ is maximal, whence $\bar{\tau} = \tau(U) = D_{\mathcal{D}}^w(S)$ by Lemma 7.2. If $\tau(U) = 0$, then $\bar{\tau} = 0$, so $D_{\mathcal{D}}^w(S) = 0$ and we have $D_{\mathcal{D}}^w(T) \geq D_{\mathcal{D}}^w(S)$, whence S also solves the Weak Closest Tree Problem. Hence we may assume $\tau(U) > 0$, whence U contains at least three points.

Let W denote the smallest cluster of T that contains U . Let B denote a child of W in T such that $B \cap U \neq \emptyset$. Then $U - B$ is nonempty, since otherwise $U \subseteq B$, contradicting the minimality of W .

I claim that there exist $b \in B$, $a \in U - B$, and $c \in U$, all distinct, such that $spt_{\mathcal{D}}(ab|c) \geq \tau(U)$. If not then, for all such a, b, c , $spt_{\mathcal{D}}(ab|c) < \tau(U)$ and there is a number $\epsilon > 0$ such that for all such a, b, c , $spt_{\mathcal{D}}(ab|c) < \tau(U) - \epsilon$. It follows that for all $b \in B$ and $a \in U - B$, $mspt_U(a, b) < \tau(U) - \epsilon$. Hence every edge between B and $U - B$ has weight less than $\tau(U) - \epsilon$, whence $K(U)_{\tau(U) - \epsilon}$ is disconnected. This contradicts the minimality of $\tau(U)$, proving the claim.

Observe that $ab|c \notin T$ since a, b , and c are all in W , but a and b are not in the same child B of W . Hence $D_{\mathcal{D}}^w(T) \geq spt_{\mathcal{D}}(ab|c)$. But then $spt_{\mathcal{D}}(ab|c) \geq \tau(U) = D_{\mathcal{D}}^w(S)$. This proves that S also solves the Weak Closest Tree Problem. ■

There is typically not a unique solution to the Weak Closest Tree Problem.

VIII. MEASURING SUBSETS OF H_X

In Section 7 we defined the triplet hypercube H_X for X and gave a geometric interpretation of the optimizing problem that NTS solves. In this section we measure the volumes of subsets of H_X . The main result is Corollary 8.2, which gives an estimate

of the probability that there exists a tree T such that $\|spt_{\mathcal{D}} - spt_T\|_{\infty} \leq \alpha$.

Recall that $H_X = [0, 1]^m$ is the triplet hypercube for X , where $|X| = n$ and $m = 3\binom{n}{3}$. For each 3-subset $\{a, b, c\}$ of X let $H_{(a,b,c)} = [0, 1]^3$ with coordinates $ab|c$, $ac|b$, and $bc|a$. Then H_X may be identified with the Cartesian product of $H_{(a,b,c)}$ where $\{a, b, c\}$ ranges over all distinct 3-subsets of X :

$$H_X = \prod [H_{(a,b,c)} : \{a, b, c\} \text{ are distinct 3-subsets of } X]$$

By Lemma 3.1 for any \mathcal{D} we have

$$0 \leq spt_{\mathcal{D}}(ab|c) + spt_{\mathcal{D}}(ac|b) + spt_{\mathcal{D}}(bc|a) \leq 1$$

and a similar inequality applies for spt_T .

For each 3-subset $\{a, b, c\}$ of X define

$$BH_{(a,b,c)} = \{u \in H_{(a,b,c)} : u_{ab|c} + u_{ac|b} + u_{bc|a} \leq 1\}.$$

Then define

$$BH_X = \prod BH_{(a,b,c)}.$$

Note $BH_X = \{u \in H_X : \text{for all distinct } a, b, c, u_{ab|c} + u_{ac|b} + u_{bc|a} \leq 1\}$. Only vectors u in BH_X are of biological interest since each vector $spt_{\mathcal{D}}$ and each vector spt_T must lie in BH_X . It is clear that BH_X is a compact convex subspace of H_X .

The 3-dimensional volume of $BH_{(a,b,c)}$ is

$$\int_0^1 \int_0^{1-x} (1-x-y) dy dx = 1/6.$$

Hence the m -dimensional volume of BH_X is $(1/6)^{\binom{n}{3}}$.

Suppose T is a binary rooted X -tree. For any α , $0 \leq \alpha \leq 1$, let

$$BH_{\alpha}(T) = \{u \in BH_X : \|u - spt_T\|_{\infty} \leq \alpha\}$$

denote the box centered at spt_T with radius α . Thus $BH_{\alpha}(T)$ is that portion of $\{u \in H_X : \|u - spt_T\|_{\infty} \leq \alpha\}$ which is biologically relevant. If for example $u \in BH_{0.49}(T)$, then Theorem 7.1 says that the NTS tree S satisfies $S = T$.

Theorem 8.1: Let T be a binary rooted X -tree. The m -dimensional volume of $BH_{\alpha}(T)$ is $\alpha^m (1/6)^{\binom{n}{3}}$.

Proof: For each collection $\{a, b, c\}$ of three distinct members from X let

$$BH_{(a,b,c,T,\alpha)} = \{u \in BH_{(a,b,c)} : |u_{ab|c} - spt_T(ab|c)| \leq \alpha,$$

$$|u_{ac|b} - spt_T(ac|b)| \leq \alpha, |u_{bc|a} - spt_T(bc|a)| \leq \alpha\}.$$

Since T is binary, one of $spt_T(ab|c)$, $spt_T(ac|b)$, and $spt_T(bc|a)$ is equal to 1, and the rest are 0. By symmetry we may assume $spt_T(ab|c) = 1$. Let the coordinates be x, y, z where x corresponds to $ab|c$. Then $BH_{(a,b,c,T,\alpha)}$ may be identified with $\{(x, y, z) \in \mathbb{R}^3 : 1-\alpha \leq x \leq 1, 0 \leq y \leq \alpha, 0 \leq z \leq \alpha, x+y+z \leq 1\}$. Note that when $\alpha = 1$, $BH_{(a,b,c,T,1)} = BH_{(a,b,c)}$ of volume $1/6$. For $0 < \alpha < 1$, the shape is exactly similar but rescaled by a factor α . Since the shape is 3-dimensional, the volume is therefore $\alpha^3/6$. Alternatively, the volume can be computed explicitly as $\int_{1-\alpha}^1 \int_0^{1-x} (1-x-y) dy dx$.

Since $B_{\alpha}(T) = \prod BH_{(a,b,c,T,\alpha)}$ with one factor for each of the $\binom{n}{3}$ sets $\{a, b, c\}$, it follows that its m -dimensional volume is

$$(\alpha^3/6)^{\binom{n}{3}} = \alpha^{3\binom{n}{3}} (1/6)^{\binom{n}{3}}.$$

Recall [28], p. 20, that the number of rooted binary trees with n leaves is $(2n-3)!! = (2n-3)(2n-5)\cdots(5)(3)(1)$.

Corollary 8.2: Let $0 \leq \alpha \leq 1$.

- (1) Let (T, X) be a binary rooted tree. The probability that a random member of BH_X lies in $BH_{\alpha}(T)$ is α^m .
- (2) The probability p that a random member of BH_X lies in $BH_{\alpha}(T)$ for some binary rooted phylogenetic X -tree T satisfies $p \leq (2n-3)!! \alpha^m$.

Proof: If U is a closed subset of \mathbb{R}^m , let $V(U)$ denote the m -dimensional volume of U . For (1), note $V(BH_{\alpha}(T)) = \alpha^m (1/6)^{\binom{n}{3}}$ and $V(BH_X) = (1/6)^{\binom{n}{3}}$, so their ratio gives the desired probability. For (2) let T_i for $i = 1, \dots, (2n-3)!!$ denote the distinct rooted binary phylogenetic X -trees. Then

$$p = V(\cup [BH_{\alpha}(T_i) : i = 1, \dots, (2n-3)!!]) / V(BH_X) \\ \leq \sum V(BH_{\alpha}(T_i)) / V(BH_X) = (2n-3)!! \alpha^m$$

since for each i , $V(BH_{\alpha}(T_i)) / V(BH_X) = \alpha^m$ by (1). ■

Corollary 8.2 has an interesting biological interpretation. Suppose that for the input data \mathcal{D} a binary rooted tree T is identified such that $\|spt_{\mathcal{D}} - spt_T\|_{\infty} = \alpha$. By Corollary 8.2 the probability that there exists such a tree is at most $(2n-3)!! \alpha^m$. Typically m is very large, so α does not need to be very small before this probability is quite low.

For example, suppose X contains 7 taxa and $\alpha = 1/2$. Then $m = 105$. The probability that there is a binary X -tree T such that $\|spt_{\mathcal{D}} - spt_T\|_{\infty} \leq \alpha$ is at most $(2n-3)!! \alpha^m = 2.563 \times 10^{-28}$. If therefore such a tree has been detected, the tree fits the data very well.

IX. DISCUSSION AND A BIOLOGICAL EXAMPLE

This paper deals with how effectively various supertree methods identify an appropriate supertree when the data are dense and sufficiently close to a tree. For supertree methods that are being used to extrapolate information from data that are not dense, the results may not be relevant. Frequently, however, consensus methods are utilized to find a unique supertree that summarizes a collection of trees all with the same leaf set. In this situation, the data are always dense. It is possible that methods with robustness radius $R = 1/2$, such as NTS, may be superior to the frequently used consensus methods which have been shown to have $R = 0$.

Even when the input trees do not all have the same leaf set, it may occur that the data are dense and the principal concern is reconciliation of incompatibilities. For example, Philip, Creevey, and McInerney [23] considered 780 single-gene trees for a set of ten eukaryotes. The numbers of taxa in a given tree ranged widely. They performed a Most Similar Supertree analysis [12] using the software package CLANN [13]. They published and analyzed the resulting tree.

I used the 664 trees that contained humans (out of 780 obtained at the web site <http://bioinf.nuim.ie/supplementary/eukaryotes/>). Each tree was rooted so that the human species was adjacent to the root. The resulting dataset \mathcal{D} was dense. The NTS S was computed, and it was topologically the same binary tree as that published in [23]. The dataset was very appropriate to analysis by NTS, since the problem posed was the reconciliation of many different input trees.

Of interest was that the NTS S satisfied that $\|spt_{\mathcal{D}} - spt_S\|_{\infty} = 0.611111$. By Corollary 8.2(2), the probability, assuming that a

vector u is randomly chosen in BH_X , that a binary tree lies within distance 0.611111 of u is at most $(2n - 3)!! \cdot 0.611111^m$. Here $n = 10$ and $m = 360$, so the probability is at most 3.47×10^{-70} . This low value indicates that the dataset vector $spt_{\mathcal{D}}$ is far from randomly related to S .

There remain several open problems. Theorem 7.3 showed that NTS solved the Weak Closest Tree Problem. In general, however, no solution was given to the Closest Tree Problem. It would be interesting to know the complexity of the Closest Tree Problem.

Lemma 4.3 showed that the results in this paper are essentially moot unless \mathcal{D} is dense in X . Also of considerable interest would be a useful extension of the results to the case where \mathcal{D} is not dense in X .

This paper concerned a way of measuring how well a method reconciles incompatibilities in a dataset. Of great interest would be a way to measure how well a method extrapolates information missing in the data.

ACKNOWLEDGMENTS

Thanks to Mike Steel and Vincent Berry for very helpful suggestions, conversations, and references. Thanks to Vincent Ranwez and the referees for very useful corrections and suggestions. Thanks also to the Isaac Newton Institute in Cambridge, U.K., for support in a wonderful setting while I wrote this paper.

REFERENCES

[1] E.N. Adams III, "N-trees as nestings: complexity, similarity and consensus," *J. Classification* 3, 1986, 299-317.

[2] A.V. Aho, Y. Sagiv, T.G. Szymanski, and J.D. Ullman, "Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions," *SIAM J. Comput* 10 (3), 1981, 405-421.

[3] K. Atteson, "The performance of neighbor-joining methods of phylogenetic reconstruction," *Algorithmica* 25, 1999, 251-278.

[4] J.-P. Barthelemy and A. Guenoche, *Trees and Proximity Representations*, New York: Wiley, 1991.

[5] B.R. Baum, "Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees," *Taxon* 41, 1992, 3-10.

[6] V. Berry and O. Gascuel, "Inferring evolutionary trees with strong combinatorial evidence," *Theoretical Computer Science* 240, 2000, 271-298.

[7] V. Berry, T. Jiang, P. Kearney, M. Li, and T. Wareham, "Quartet cleaning: improved algorithms and simulations," in J. Nesetril, (ed.) *Algorithms - ESA '99*, 7th Annual European Symposium, Prague, Czech Republic, July 16-18, 1999, Proceedings, Lecture Notes in Computer Science 1643, Springer, 1999.

[8] O. Bininda-Emonds and M. J. Sanderson, "Assessment of the accuracy of matrix representation with parsimony analysis supertree construction," *Syst. Biol.* 50(4), 2001, 565-579.

[9] O. Bininda-Emonds, J.L. Gittleman, and M. Steel, "The (super)tree of life: procedures, problems, and prospects," *Ann. Rev. Ecol. Syst.* 33, 2002, 265-289.

[10] O.R.P. Bininda-Emonds, ed., *Phylogenetic Supertrees: Combining Information To Reveal the Tree of Life*, Dordrecht, the Netherlands: Kluwer Academic, 2004.

[11] O.R.P. Bininda-Emonds, M. Cardillo, K.E. Jones, R.D.D. MacPhee, R.M.D. Beck, R. Grenyer, S.A. Price, R.A. Vos, J.L. Gittleman, and A. Purvis, "The delayed rise of present-day mammals," *Nature* 446, 2007, 507-512.

[12] C.J. Creevey, D.A. Fitzpatrick, G.K. Philip, R.J. Kinsella, M.J. O'Connell, M.M. Pentony, S.A. Travers, M. Wilkinson, and J.O. McInerney, "Does a tree-like phylogeny only exist at the tips in the prokaryotes?" *Proc. R. Soc. Lond. B. Biol. Sci.* 271, 2004, 2551-2558.

[13] C.J. Creevey and J.O. McInerney, "CLANN: software for supertree analysis," *Bioinformatics* 21, 2005, 390-392.

[14] J.H. Degnan and N.A. Rosenberg, "Discordance of species trees with their most likely gene trees," *PLoS Genet* 2(5): e68, 2006. DOI: 10.1371/journal.pgen.0020068

[15] P.L. Erdős, M.A. Steel, L.A. Székely, and T.J. Warnow, "Constructing big trees from short sequences," in *Automata, Languages and Programming. 24th International Colloquium, ICALP'97*, Bologna, Italy, July 7 - 11, 1997, (P. Degano, R. Gorrieri, A. Marchetti-Spaccamela, Eds.) Proceedings (LNCS 1256), 1997, 827-837.

[16] P.L. Erdős, M.A. Steel, L.A. Székely, and T.J. Warnow, "A few logs suffice to build (almost) all trees (I)," *Random Structures and Algorithms* 14, 1999, 153-184.

[17] J. Felsenstein, *Inferring Phylogenies*, Sunderland, Mass: Sinauer Associates, Inc., 2004.

[18] L.R. Foulds and R.L. Graham, "The Steiner problem in phylogeny is NP-complete," *Advances in Applied Mathematics*, 3, 1982, 43-49.

[19] M. Gondran, "La structure algébrique des classifications hiérarchiques," *Annls INSEE* 22-23, 1976, 181-190.

[20] M. Gondran and M. Mindux, *Graphs and Algorithms*, (trans. Steven Vajda), Chichester, NY: Wiley, 1984.

[21] T. Jiang, P.E. Kearney, and M. Li, "Orchestrating quartets: Approximation and data corrections," Proceedings of the 39th IEEE Symposium on Foundations of Computer Science, 1998, 416-425.

[22] R.D.M. Page, "Modified mincut supertrees," in R. Guig and D. Gusfield (Eds), *Algorithms in Bioinformatics, Proceedings of the second international workshop, WABI 2002*, Rome, Italy, September 17-21, 2002, Lecture Notes in Computer Science 2452, Springer-Verlag, Berlin, 2003, pp. 537-552.

[23] G.K. Philip, C.J. Creevey, and J.O. McInerney, "The Opisthogonta and the Ecdysozoa may not be clades: Stronger support for the grouping of plant and animal than for animal and fungi and stronger support for the Coelomata than Ecdysozoa," *Mol. Biol. Evol.* 22(5), 2005, 1175-1184.

[24] M.A. Ragan, "Phylogenetic inference based on matrix representation of trees," *Mol. Phylogenet. Evol.* 1, 1992, 53-58.

[25] V. Ranwez, V. Berry, A. Criscuolo, P.-H. Fabre, S. Guillemot, C. Scornavacca, and E. Douzery, "PhySIC: a veto supertree method with desirable properties," *Systematic Biology* 56(5), 2007, 798 -817.

[26] M.J. Sanderson, A. Purvis, and C. Henze, "Phylogenetic supertrees: Assembling the trees of life," *Trends Ecol Evol.* 13, 1998, 105-109.

[27] C. Semple and M. Steel, "A supertree method for rooted trees," *Discrete Applied Mathematics* 105, 2000, 147-158.

[28] C. Semple and M. Steel, *Phylogenetics*, Oxford: Oxford University Press, 2003.

[29] S. Snir and S. Rao, "Using max cut to enhance rooted trees consistency," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 3(4), 2006, 323-333.

[30] M. Steel, S. Böcker, and A.W.M. Dress, "Simple but fundamental limitations on supertree and consensus tree methods," *Systematic Biology* 49(2), 2000, 363-368.

[31] D.L. Swofford, PAUP*. *Phylogenetic Analysis Using Parsimony (*and Other Methods)*, Version 4. Sunderland, Massachusetts: Sinauer Associates, 2002.

[32] M. Wilkinson, J.L. Thorley, D. Pisani, F.-J. Lapointe, and J.O. McInerney, "Some desiderata for liberal supertrees," in O.R.P. Bininda-Emonds, (ed.) *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, Dordrecht, The Netherlands: Kluwer, 2004, pp. 227-246, .



Stephen Willson Stephen J. Willson received his A.B. in Mathematics from Harvard in 1968. In 1973 he received his Ph.D. in Mathematics from the University of Michigan in Ann Arbor. His dissertation was in algebraic topology under the supervision of A.G. Wasserman.

He went to Iowa State University in Ames, Iowa in 1973, where he is currently Professor of Mathematics. His research interests include phylogenetics, fractals, and game theory. His hobbies include classical piano, choral singing, bird-watching, and

kayaking.