A SUPERTREE METHOD FOR ROOTED TREES

CHARLES SEMPLE AND MIKE STEEL

Abstract. The amalgamation of leaf-labelled (phylogenetic) trees on overlapping leaf sets into one (super)tree is a central problem in several areas of classification, particularly evolutionary biology. In this paper, we describe a new technique for amalgamating rooted phylogenetic trees. This appears to be the first such method to provably exhibit particular desirable properties which we list and establish.

1. Introduction

The amalgamation of a collection of leaf-labelled trees (the input trees) into a single output tree is an important task in various areas of classification, particularly evolutionary biology. In general, a method for amalgamating trees on overlapping leaf sets is called a *supertree* method; in the special case where all the input trees have the same leaf set it is called a *consensus tree* method.

Two problems that arise for any supertree approach are (i) finding a reasonable criteria by which to combine the input trees, and (ii) designing a polynomial time algorithm to carry this out.

Regarding problem (i), if the trees all have the same leaf set, then there exist simple and natural consensus tree criteria - these include strict consensus, majority rule consensus, and (for rooted trees) Adams consensus (for the latter, see [1] and [2]). (For a good survey of these and other consensus methods, the reader is referred to [12].) In case the leaf sets of the input trees are different (and usually overlapping), it is shown in [5] that no "reasonable" supertree method exists for when the input trees are unrooted. Consequently, in this paper we will restrict our attention to rooted trees. Here a root may either be some hypothetical ancestor, or it may be a common leaf shared by a set of unrooted input trees.

Regarding problem (ii), the question of whether a collection of unrooted trees on overlapping leaf sets fit together compatibly is already an NP-hard problem [15]. (A set of input trees is *compatible* if there is a parent tree that *displays* each of them as a subtree, as defined below). However, if all the input trees are rooted, then compatibility can be decided via a polynomial-time algorithm [3]. Unfortunately, in applications involving either numerous trees or large trees, incompatibility is

Date: January 5, 2000.
Key words and phrases. Consensus, rooted phylogenetic tree, supertree.
This work was supported by the New Zealand Marsden Fund (UOC-MIS-003).

frequently encountered, and so, even for rooted trees, it is not entirely clear how to simultaneously address both of the problems (i) and (ii) listed above.

In this paper, we describe a new method for constructing rooted supertrees, which has the following desirable properties:

- The method has a polynomial time algorithm.
- The method preserves nestings and binary subtrees that are shared by all of the input trees.
- In case the input trees are compatible, the output tree displays each of the input trees.
- The method satisfies two natural symmetry requirements, as listed in [5]. In particular, (1) the output tree is independent of the order in which the input trees are listed and (2) if we rename all the leaves, and then apply our method to the new set of input trees, the output tree is simply the original output tree, but with the leaves renamed as before.
- The method extends naturally to allow the input trees to be weighted.

As far as we are aware, our method is the only supertree technique that has been shown to have these properties. The approach we take is to modify the algorithm described by Aho et al. (see [3] and [6]) which returns a tree exactly when the input trees are compatible. In brief, if the associated graph is connected, we delete all of the edges in the union of the minimum (-weight) cut sets of a (possibly different but) related graph when the algorithm would otherwise terminate without returning a tree.

The paper is organised as follows. In the next section, we recall some basic terminology. In Section 3, we present our new method, called MinCutSupertree, for constructing rooted supertrees. The desirable properties of this method are then established in Section 4, in which we also compare our method with the Adams consensus for trees on a common leaf set.

2. Preliminaries

In this section, we recall some relevant notation and terminology.

Graphs and cut sets

A graph is *simple* if it has no loops or parallel edges. Throughout this paper, we will denote a simple graph G as a pair (V, E) where E is a subset of $\{\{x, y\} : x, y \in V; x \neq y\}$. Given a subset V' of V, we let $G[V']$ denote the induced subgraph (V', E') of (V, E), where E' is the set of edges of G having both endpoints in V'. Given $E' \subseteq E$, we let $G\backslash E'$ denote the graph obtained from G by deleting all of the edges in E' and we let G/E' denote the graph obtained from G by contracting all of the edges in E'.

Suppose that $w : E \to \mathbb{Q}^+$ is a weight function on the edges of G and let E' be a subset of E. If $G\backslash E'$ is disconnected, then E' is said to be a *cut set* of G.

Moreover, if E' is a cut set of G and minimizes $\sum_{e\in E'} w(e) \in \mathbb{Q}^+$, then E' is a *minimum-weight cut set* of G (with respect to w). We denote this minimum value by $c(G, w) \in \mathbb{Q}^+$ or, more simply, $c(G)$ if no ambiguity can arise. We remark that "$\mathbb{Q}^+$" is chosen here instead of $\mathbb{R}^+$ as this limits the computational complexity of our method and, moreover, is not restrictive for applications.

Rooted phylogenetic trees, clusters, and rooted triples

Let $T = (V, E)$ be a tree. A vertex $v \in V$ is *internal* if the degree of v is greater than one, otherwise v is a *leaf*. An edge $e = \{u, v\} \in E$ is *internal* if both u and v are internal vertices, otherwise we say e is an *external* edge. Let $\mathcal{L}(T)$ denote the set of leaves of T.

If $\mathcal{L}(T) = X$, and T has exactly one distinguished internal vertex, while the remaining internal vertices each have degree at least three, then T is called a *rooted phylogenetic tree (on X)*. Such trees are also referred to in the literature as a *phylogeny*, an *evolutionary tree*, or a *cladogram*. The distinguished vertex of T is called the *root*. Two rooted phylogenetic trees on X, $T = (V, E)$ and $T' = (V', E')$, are considered identical if there exists a bijection $\alpha : V \to V'$ which induces a bijection from E to E' and which fixes X. Thus, except for the root, the labelling of the internal vertices of a rooted phylogenetic tree is unimportant.

Let T be a rooted phylogenetic tree on X. An element of X is a *descendant* of a vertex v of T if the path from this element to the root passes through v. A *cluster* of T is a subset of X that consists of all the elements of X that are the descendants of some particular vertex of T. The set X is always a cluster of T; every other cluster is said to be *proper*.

A rooted phylogenetic tree is *binary* if all internal vertices have degree three except for the root which has degree two. For example, the trees T_1 and T_2 in Figure 1 are both binary. A *rooted triple* is a binary rooted phylogenetic tree with three leaves. The rooted triple with leaves a, b, and c is denoted $ab|c$ if the path from a to b does not intersect the path from c to the root. If T is a rooted phylogenetic tree, then we let $r(T)$ denote the set of rooted triples of T.

Compatibility

Let T be a rooted phylogenetic tree. A rooted phylogenetic tree T' is said to be obtained from T *by contraction* if T' can be obtained from T by contracting a sequence of internal edges.

Let A be a subset of $\mathcal{L}(T)$. Consider the minimal subtree $T(A)$ of T containing A. Let $T|A$ denote the rooted phylogenetic tree on A obtained from $T(A)$ by distinguishing the vertex of $T(A)$ closest to the root of T and suppressing all vertices of degree two (except for the distinguished vertex). We call $T|A$ the *subtree of T induced by A*. For a multiset $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ of rooted phylogenetic trees, we let $\mathcal{T}|A$ denote the multiset $\{T_1|A, T_2|A, \ldots, T_k|A\}$.

A rooted phylogenetic tree T *displays* a rooted phylogenetic tree t if t can be obtained from an induced subtree of T by contraction (or, equivalently, t is an

induced subtree of a contraction of T). This provides a convenient partial order on the set of rooted phylogenetic trees which we denote by $\leq$. In the case above, we write $t \leq T$. We say a collection of rooted phylogenetic trees is *compatible* precisely if there is a phylogenetic tree that displays all of them.

3. The MinCutSupertree Algorithm

In this section, we describe the algorithm MinCutSupertree. Before doing this, however, we define two associated simple graphs, both of which will play an important role in this algorithm.

Suppose that $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ is a multiset of rooted phylogenetic trees (with possibly different leaf sets). Let w be a weight function from $\{T_1, T_2, \ldots, T_k\}$ into $\mathbb{Q}^+$ (by default taken to be the constant function).

(1) Given $S \subseteq \bigcup_{i=1}^{k} \mathcal{L}(T_i)$, let $S_{\mathcal{T}}$ denote the graph $(S, E_{\mathcal{T}})$, where $\{a, b\} \in E_{\mathcal{T}}$ precisely if there exists at least one tree in $\mathcal{T}|S$ for which a and b both appear in the same proper cluster.
(2) The second graph $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$ is obtained from $S_{\mathcal{T}}$ as follows: weight each edge, $\{a, b\} \in E_{\mathcal{T}}$ say, by the sum of the weights of the trees that have a and b in a proper cluster; contract each edge whose weight is $w_{\text{sum}} := \sum_{T \in \mathcal{T}} w(T)$; and, lastly, delete all loops, and replace each parallel class of edges with a single edge whose weight is the sum of the weights of those trees in $\mathcal{T}|S$ that have a proper cluster that contains the endpoints of at least one edge in that parallel class. We denote the set of edges of $S_{\mathcal{T}}$ whose weight is w_{sum} by $E_{\mathcal{T}}^{\max}$.

The second graph is introduced in order to ensure that our method has the desirable properties outlined in Section 1 and proved in Section 4, as detailed by the remark immediately following the statement of Corollary 4.5.

We now present the algorithm for our method. An example, illustrating this algorithm, is given at the end of this section.

Algorithm: MinCutSupertree$(\mathcal{T}, w)$.

Input: A multiset of rooted phylogenetic trees $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$. A weight function $w : \{T_1, T_2, \ldots, T_k\} \to \mathbb{Q}^+$ (by default taken to be the constant function).

Output: A rooted phylogenetic tree on $\bigcup_{i=1}^{k} \mathcal{L}(T_i)$, denoted $\mathcal{M}(\mathcal{T})$.

(1) Initially set $S := \bigcup_{i=1}^{k} \mathcal{L}(T_i)$.
(2) If $|S| \leq 2$, then return the tree with the elements of S as leaves.
(3) Otherwise, if $|S| > 2$, then construct $S_{\mathcal{T}}$.
(4) If $S_{\mathcal{T}}$ is disconnected, then list the vertex sets, denoted $S_1, S_2, \ldots, S_r$ ($r \geq 2$), of the components of this graph.
(5) Otherwise, if $S_{\mathcal{T}}$ is connected, then construct the graph $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$. Construct the set E' of edges of $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$ that lie in at least one minimum-weight cut

set of $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$. For each edge in E', delete the corresponding edge(s) of $E_{\mathcal{T}}$ from $S_{\mathcal{T}}$, and list the vertex sets $S_1, S_2, \ldots, S_r$ ($r \geq 2$) of the resulting components.

(6) For all $j \in \{1, 2, \ldots, r\}$, construct $T_j := \text{MinCutSupertree}(\mathcal{T}|S_j, w_j)$, and w_j is the weight function from $\mathcal{T}|S_j$ into $\mathbb{Q}^+$ defined by $w_j(T_i|S_j) = w(T_i)$ for all $i \in \{1, 2, \ldots, k\}$.

(7) Construct a new tree T by making the roots of the trees $T_1, T_2, \ldots, T_r$ adjacent to a new root ρ.

(8) Output $\mathcal{M}(\mathcal{T}) := T$.

Evidently the algorithm MinCutSupertree satisfies the symmetry properties (1) and (2) of the introduction and, moreover, returns at most one tree. The fact that it returns exactly one tree follows from the next proposition. Of course, the tree returned by MinCutSupertree does depend on the weighting of the input trees.

Proposition 3.1. *Let $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ be a multiset of rooted phylogenetic trees, with a corresponding weight function w. Then* MinCutSupertree *applied to $(\mathcal{T}, w)$ returns a tree.*

Proof. It is clear that MinCutSupertree returns a tree when applied to $\mathcal{T}$ provided that, at each iteration of the algorithm, either $S_{\mathcal{T}}$ is disconnected, or, if this is not the case, then $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$ is not a single vertex. Thus the proposition is proved by showing that if $S_{\mathcal{T}}$ is connected for some $\mathcal{T}$ and for some S where $|S| \geq 3$, then the associated graph $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$ contains at least two vertices. We consider two cases.

First assume that, for some $i \in \{1, 2, \ldots, k\}$, there exists a tree T_i such that $S \not\subseteq \mathcal{L}(T_i)$. Then there is an element of S that is not incident with an edge of $S_{\mathcal{T}}$ of weight w_{sum}. It follows that, in this case, $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$ contains at least two vertices.

For the second case, assume that, for all $i \in \{1, 2, \ldots, k\}$, $S \subseteq \mathcal{L}(T_i)$. Suppose, to the contrary, that $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$ consists of a single vertex. Under our assumption, the subgraph G of $S_{\mathcal{T}}$ consisting of S together with those edges of $S_{\mathcal{T}}$ of weight w_{sum} is a connected graph. Furthermore, G has the property that whenever it contains edges $\{u, v\}$ and $\{u, v'\}$ it must also contains the edge $\{v, v'\}$. But it is easily checked that any connected graph satisfying this last property has an edge between each pair of vertices. It follows that $S_{\mathcal{T}}$ is a clique of size $|S|$ in which every edge has weight w_{sum}. But this is impossible since, for all $i \in \{1, 2, \ldots, k\}$, the root of $T_i|S$ has degree at least 2. This provides the required contradiction, thereby completing the proof of the second case and the proposition. □

We conclude this section by illustrating MinCutSupertree with an example. Let T_1 and T_2 be the rooted phylogenetic trees as shown in Figure 1(a) and suppose that the weight of each tree is 1. Then S is initially $\{a, b, c, d, e\}$, and $S_{\{T_1,T_2\}}$ and $S_{\{T_1,T_2\}}/E_{\{T_1,T_2\}}^{\max}$ are the graphs shown in Figure 1(b). Note that, as $S_{\{T_1,T_2\}}$ is connected, the latter graph needs to be constructed. Now $S_1 = \{a, b\}$, $S_2 = \{c\}$, $S_3 = \{d\}$, and $S_4 = \{e\}$. This completes the first iteration of MinCutSupertree.

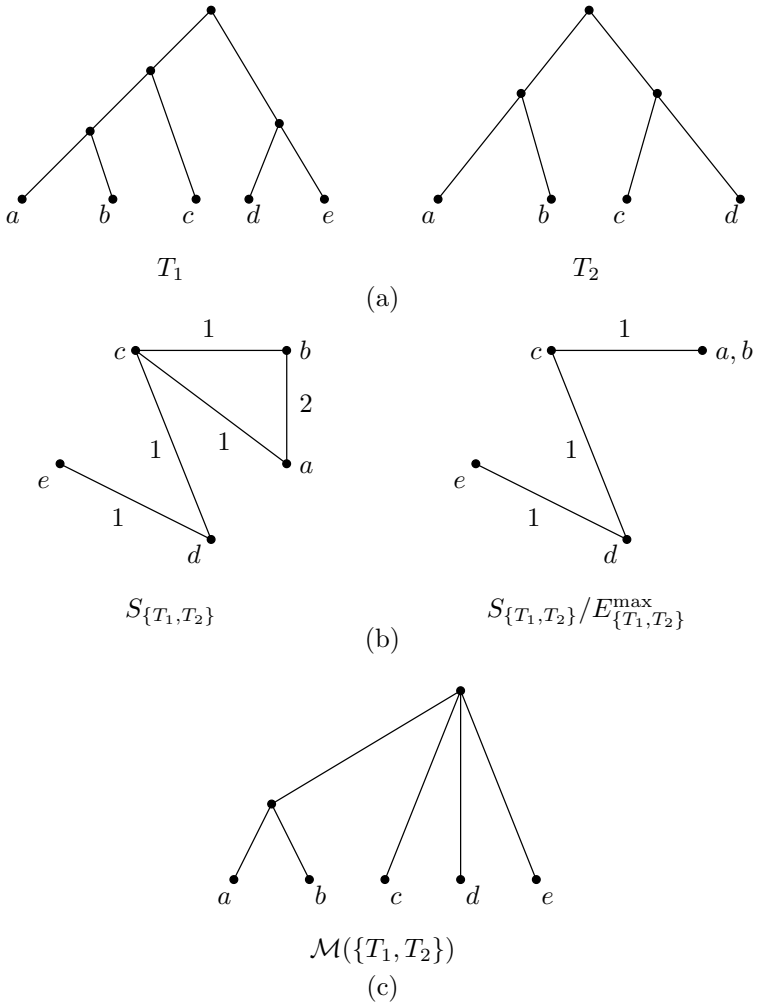

FIGURE 1. An example illustrating MINCUTSUPERTREE.

The algorithm is completed by applying, for all $j \in \{1,2,3,4\}$, MINCUTSUPERTREE to $\{T_1|S_j, T_2|S_j\}$, where both $T_1|S_j$ and $T_2|S_j$ have weight 1, to construct the tree T_j, and then connecting the roots of these trees to a new root to obtain $\mathcal{M}(\{T_1,T_2\})$. This last tree is shown in Figure 1(c).

4. PROPERTIES OF MINCUTSUPERTREE

In this section, we establish the remaining desirable properties of MINCUTSUPERTREE and compare this method with the Adams consensus for trees on the same leaf set.

We first show that the tree returned by MINCUTSUPERTREE can be constructed in polynomial time. From the construction of MINCUTSUPERTREE, it is evident

that this is indeed the case provided that one is able to determine in polynomial time whether an edge of a graph G with weight function $w : E \to \mathbb{Q}^+$ is in the union of all minimum-weight cut sets of G.

Now the quantity $c(G, w)$ (the weight of a minimum-weight cut set of G) can be calculated in polynomial time by standard network-flow techniques (see [8]). This can then be used to determine in polynomial time which edges are in a minimum-weight cut set of G by using the next proposition (and although it is almost certainly not new, we include its short proof for completeness).

Proposition 4.1. *Let $G = (V, E)$ be a graph with a weight function $w : E \to \mathbb{Q}^+$. Let e be an edge of G. Then e is in a minimum-weight cut set of G if and only if $c(G\backslash e) + w(e) = c(G)$.*

Proof. We first establish, for every edge e of G, the following inequality:

$$c(G\backslash e) + w(e) \geq c(G). \tag{1}$$

Let A be the set of edges of a minimum-weight cut set of $G\backslash e$ and suppose, to the contrary, that $c(G\backslash e) + w(e) < c(G)$. Then $\sum_{f\in A\cup\{e\}} w(f) < c(G)$. But $A \cup \{e\}$ is a cut set of G; a contradiction. This establishes inequality (1).

With inequality (1) in hand, suppose that e is in a minimum-weight cut set B of G and suppose, to the contrary, that $c(G\backslash e)+w(e) \neq c(G)$. Then, by inequality (1), $c(G\backslash e) + w(e) > c(G)$. Now B is a cut set of G, so $B - \{e\}$ is a cut set of $G\backslash e$. But $\sum_{f\in B-\{e\}} w(f) = c(G) - w(e)$ and so $\sum_{f\in B-\{e\}} w(f) < c(G\backslash e)$; a contradiction.

To prove the converse, let A be the set of edges of a minimum-weight cut set of $G\backslash e$ and suppose that $c(G\backslash e) + w(e) = c(G)$. Since $A \cup \{e\}$ is a cut set of G and since $\sum_{f\in A\cup\{e\}} w(f) = c(G\backslash e) + w(e) = c(G)$, it follows that $A \cup \{e\}$ is a minimum-weight cut set of G, and so e is in a minimum-weight cut set of G as required. □

The proof of Theorems 4.3 and 4.6 rely on the following result ([6, Theorem 1]).

Lemma 4.2. *Let T and T' be two rooted phylogenetic trees. Then $T \leq T'$ if and only if $r(T) \subseteq r(T')$ and $\mathcal{L}(T) \subseteq \mathcal{L}(T')$.*

Theorem 4.3. *Let $\mathcal{T}$ be a (weighted) multiset of rooted phylogenetic trees, and suppose that $\mathcal{T}$ is compatible. Then $\mathcal{M}(\mathcal{T})$ displays each of the trees in $\mathcal{T}$.*

Proof. Suppose that $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$. Let $r(\mathcal{T}) = \bigcup_{i=1}^{k} r(T_i)$, where we recall that $r(T_i)$ denotes the set of rooted triples of T_i. Then, by comparing the algorithm "OneTree" described in [11] (which returns a single tree if the inputted rooted triples are compatible) with MinCutSupertree, it is easily seen that, in the case $\mathcal{T}$ is compatible, the trees returned by both algorithms are identical when applied to $r(\mathcal{T})$. Therefore, as $r(\mathcal{T})$ is a subset of the set of rooted triples of the tree returned by the former algorithm, $r(\mathcal{T}) \subseteq r(\mathcal{M}(\mathcal{T}))$. It now follows by Lemma 4.2 that $\mathcal{M}(\mathcal{T})$ displays each of the trees in $\mathcal{T}$. □

Before going further, some more preliminaries are required.

Nestings. Let T be a rooted phylogenetic tree on X. Adams [2] defines a relation $<_T$ on the subsets of X as follows. If A and B are subsets of X such that the most recent common ancestor of A is a proper descendant of the most recent common ancestor of B, then $A <_T B$, in which case, we say that A *nests in* B.

Adams consensus $\mathcal{A}(\mathcal{T})$. In [2], Adams showed that the Adams consensus tree (which was first described in [1]) can be characterized via the notion of nesting. In particular, suppose that $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ is a multiset of rooted phylogenetic trees, with $\mathcal{L}(T_i) = X$ for all $i \in \{1, 2, \ldots, k\}$. Then the Adams consensus tree for $\mathcal{T}$, denoted $\mathcal{A}(\mathcal{T})$, is the unique rooted phylogenetic tree on X that satisfies the following two properties:

- **(A1)** If A and B are subsets of X such that $A <_{T_i} B$ for all $i \in \{1, 2, \ldots, k\}$, then $A <_{\mathcal{A}(\mathcal{T})} B$.
- **(A2)** If C and D are clusters of $\mathcal{A}(\mathcal{T})$ such that $C <_{\mathcal{A}(\mathcal{T})} D$, then $C <_{T_i} D$ for all $i \in \{1, 2, \ldots, k\}$.

Note that, in the statement of (A2), neither "C" nor "D" is necessarily a cluster of T_i for all $i \in \{1, 2, \ldots, k\}$. Also note that a polynomial-time algorithm to construct $\mathcal{A}(\mathcal{T})$ has been described elsewhere [10].

For a multiset of rooted phylogenetic trees on overlapping leaf sets, the next theorem shows that the analogue of (A1) holds for the tree returned by MinCut-Supertree when applied to such a multiset of trees.

Theorem 4.4. *Let $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ be a (weighted) multiset of rooted phylogenetic trees. Suppose that A and B are subsets of $\bigcap_{i=1}^{k} \mathcal{L}(T_i)$ such that $A <_{T_i} B$ for all $i \in \{1, 2, \ldots, k\}$. Then $A <_{\mathcal{M}(\mathcal{T})} B$*

Proof. Referring to the algorithm MinCutSupertree, it suffices to show that if $B \subseteq S$, then all the elements of A are identified as a single vertex of $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$.

Since $A <_{T_i} B$ for all $i \in \{1, 2, \ldots, k\}$, and since $B \subseteq S$, it follows that $A <_{T_i|S} B$ for all $i \in \{1, 2, \ldots, k\}$. Thus, for each T_i, there exists an element, s_i say, of S such that, for all distinct a_1 and a_2 of A, $a_1a_2|s_i$ is a rooted triple of $T_i|S$. Hence $S_{\mathcal{T}}[A]$ is a clique of size $|A|$ with each edge having weight w_{sum}, and therefore the elements of A are identified as a single vertex in $S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$. This completes the proof of Theorem 4.4. □

An immediate consequence of Theorem 4.4 is Corollary 4.5.

Corollary 4.5. *Let $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ be a (weighted) multiset of rooted phylogenetic trees. Suppose that a, b, and c are elements of $\bigcap_{i=1}^{k} \mathcal{L}(T_i)$ such that $ab|c$ is a rooted triple of T_i for all $i \in \{1, 2, \ldots, k\}$. Then $ab|c$ is a rooted triple of $\mathcal{M}(\mathcal{T})$.*

Remark. The reason for constructing the graph "$S_{\mathcal{T}}/E_{\mathcal{T}}^{\max}$" in MinCutSupertree is that if we were to delete all of the edges in the union of all the minimum-weight cut sets of $S_{\mathcal{T}}$ at each iteration, then we would have no guarantee that the output tree displays all of the nestings and, in particular, all of the rooted triples shared by all of the input trees. An example of this situation is provided by choosing the following two trees as our input trees: let T and T' be two rooted phylogenetic trees on $\{a, b, c, d, e, f\}$ so that the maximal proper clusters of T are $\{a, b, c\}$ and $\{d, e, f\}$, and the maximal proper clusters of T' are $\{a, b, e\}$ and $\{c, d, f\}$.

Corollary 4.6 establishes the final desirable property of MinCutSupertree.

Corollary 4.6. *Let $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ be a (weighted) multiset of rooted phylogenetic trees, and let T be a rooted phylogenetic tree. Suppose that $\mathcal{L}(T)$ is a subset of $\bigcap_{i=1}^{k} \mathcal{L}(T_i)$ such that, for all $i \in \{1, 2, \ldots, k\}$, $T = T_i|\mathcal{L}(T)$. Then $\mathcal{M}(\mathcal{T})$ displays T. Furthermore, if T is binary, then $T = \mathcal{M}(\mathcal{T})|\mathcal{L}(T)$.*

Proof. Since T is a subtree of T_i for all $i \in \{1, 2, \ldots, k\}$, it follows by Corollary 4.5 that $r(T)$ is a subset of $r(\mathcal{M}(\mathcal{T}))$. Therefore, by Lemma 4.2, $\mathcal{M}(\mathcal{T})$ displays T. Now if T is binary, every 3–element subset of $\mathcal{L}(T)$ induces a rooted triple of T, and therefore, in this case, T must be a subtree of $\mathcal{M}(\mathcal{T})$. □

We end this section with a detailed look at the relationship between the Adams consensus tree $\mathcal{A}(\mathcal{T})$ for a multiset $\mathcal{T}$ of rooted phylogenetic trees having the same leaf set, and the tree $\mathcal{M}(\mathcal{T})$ returned by MinCutSupertree when applied to $\mathcal{T}$. The first point to note is that, like $\mathcal{A}(\mathcal{T})$, the tree $\mathcal{M}(\mathcal{T})$ preserves the nestings shared by all of the trees in $\mathcal{T}$. However, $\mathcal{M}(\mathcal{T})$ is not necessarily equal to $\mathcal{A}(\mathcal{T})$. In fact, under $\leq$, the two trees may not even be comparable. To see this, consider the example illustrated in Figure 2. Nevertheless, there is still a strong connection between $\mathcal{A}(\mathcal{T})$ and $\mathcal{M}(\mathcal{T})$. This connection is established in Theorem 4.7 and Corollary 4.8.

Theorem 4.7. *Let $\mathcal{T}$ be a multiset of rooted phylogenetic trees having the same leaf set X. Let A and B be subsets of X. If $A <_{\mathcal{A}(\mathcal{T})} B$, then $A <_{\mathcal{M}(\mathcal{T})} B'$ for every cluster B' of $\mathcal{A}(\mathcal{T})$ that contains B.*

Proof. Suppose that $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$, and let A' and B' be clusters of $\mathcal{A}(\mathcal{T})$ that extend A and B, respectively, such that A' is minimal with respect to containing A. Note that this proviso means that A' is a proper subset of B, and therefore a proper subset of B'. Now $A' <_{\mathcal{A}(\mathcal{T})} B'$ and so, as A' and B' are both clusters of $\mathcal{A}(\mathcal{T})$, it follows by (A2) that $A' <_{T_i} B'$ for all $i \in \{1, 2, \ldots, k\}$. Therefore, as $A \subseteq A' \subset B'$, $A <_{T_i} B'$ for all $i \in \{1, 2, \ldots, k\}$. Thus, by Theorem 4.4, $A <_{\mathcal{M}(\mathcal{T})} B'$ as required. □

Figure 3 shows that $\mathcal{A}(\mathcal{T})$ and $\mathcal{M}(\mathcal{T})$ may be comparable under $\leq$ for a multiset of rooted phylogenetic trees on the same leaf set. In particular, for the example illustrated in Figure 3, we have $\mathcal{A}(\{T_6, T_7\}) \leq \mathcal{M}(\{T_6, T_7\})$. In fact, Figures 2 and 3 illustrate the only possibilities that can occur when $\mathcal{A}(\mathcal{T})$ and $\mathcal{M}(\mathcal{T})$ are compared with respect to $\leq$.

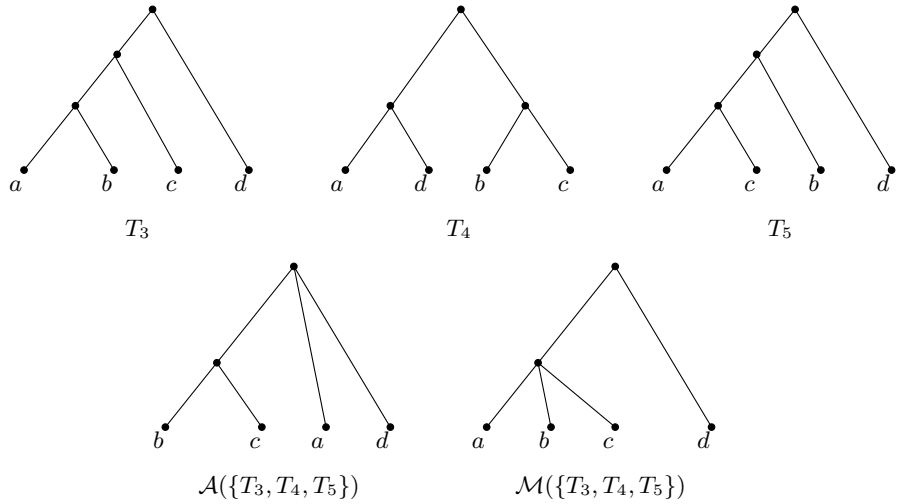

FIGURE 2. T_3, T_4, T_5, $\mathcal{A}(\{T_3, T_4, T_5\})$, and $\mathcal{M}(\{T_3, T_4, T_5\})$.

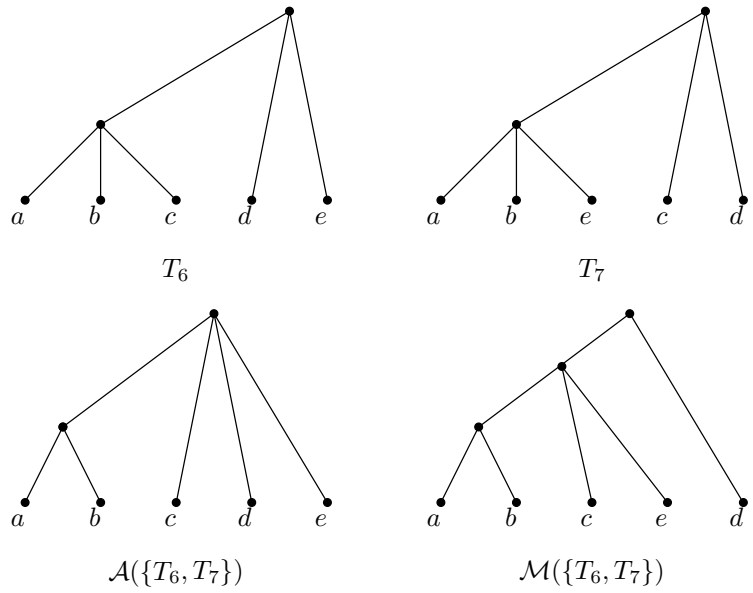

FIGURE 3. T_6, T_7, $\mathcal{A}(\{T_6, T_7\})$, and $\mathcal{M}(\{T_6, T_7\})$.

Corollary 4.8. *Let $\mathcal{T}$ be a multiset of rooted phylogenetic trees on the same leaf set. Then exactly one of the following holds:*

(i) $\mathcal{A}(\mathcal{T}) \leq \mathcal{M}(\mathcal{T})$*; or*
(ii) $\mathcal{A}(\mathcal{T})$ *is not comparable to* $\mathcal{M}(\mathcal{T})$ *under* $\leq$.

Proof. Suppose, to the contrary, that, for some multiset $\mathcal{T}$ of rooted phylogenetic trees, each having leaf set X, $\mathcal{M}(\mathcal{T}) \leq \mathcal{A}(\mathcal{T})$ but $\mathcal{M}(\mathcal{T}) \neq \mathcal{A}(\mathcal{T})$. Then, as $\mathcal{M}(\mathcal{T})$

and $\mathcal{A}(\mathcal{T})$ both have leaf set X, it follows that $\mathcal{M}(\mathcal{T})$ can be obtained from $\mathcal{A}(\mathcal{T})$ by contracting at least one internal edge. Let u and v denote the end vertices of such an edge so that the path from u to the root of $\mathcal{A}(\mathcal{T})$ passes through v. Let U and V denote the maximal clusters of $\mathcal{A}(\mathcal{T})$ whose most recent common ancestor is u and v, respectively. Then, as U is a proper subset of V, $U <_{\mathcal{A}(\mathcal{T})} V$ and so, by Theorem 4.7, $U <_{\mathcal{M}(\mathcal{T})} V$. But $\mathcal{M}(\mathcal{T})$ can be obtained from $\mathcal{A}(\mathcal{T})$ by contracting internal edges one of which is $\{u, v\}$, so U does not nest in V in $\mathcal{M}(\mathcal{T})$. This contradiction completes the proof of the corollary. □

Remark. Because MinCutSupertree uses an optimisation principle, rather than the set theoretic operations that are used in the construction of the Adams consensus tree [1, 2, 10], it is not surprising that the two methods can disagree in the setting where they both apply. Informally, one can regard the Adams tree as the tree that provides a conservative estimate of nestings shared by the input trees; by contrast MinCutSupertree finds a recursively optimal modification of the algorithm described by Aho et al. [3] in the more general setting where the input trees have unequal leaf sets. It would be interesting to see if our approach could be further modified so that, when applied to trees having a common leaf set, the Adams tree was equal to (or $\leq$) the corresponding output tree.

Acknowledgements

We thank Sebastian Böcker and Joe Thorley for reading an earlier draft and providing helpful comments. We also thank the referee for some useful comments.

References

[1] E. Adams III, Consensus techniques and the comparison of taxonomic trees, *Syst. Zool.* **21** (1971), 390–397.
[2] E. Adams III, *N*–trees as nestings: complexity, similarity and consensus, *J. Classif.* **3** (1986), 299–317.
[3] A. V. Aho, S. Yehoshua, 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.
[4] J. P. Barthélemy, F. R. McMorris, and R. C. Powers, Dictactorial consensus functions on *n*–trees, *Math. Biosci.* **25**, 59–64.
[5] S. Böcker, A. W. M. Dress, and M. Steel, Simple but fundamental limitations on supertree and consensus tree methods, *Syst. Biol.*, in press.
[6] D. Bryant and M. Steel, Extension operations on leaf-labelled trees, *Adv. Appl. Math.* **16** (1995), 425–453.
[7] M. Constantinescu and D. Sankoff, An efficient algorithm for supertrees, *J. Classif.* **12** (1995), 101–112.
[8] R. E. Gomory and T. C. Hu, Multiterminal network flows, *SIAM J. Appl. Math.* **9** (1961), 551–570.
[9] A. G. Gordon, Consensus supertrees: the synthesis of rooted trees containing overlapping sets of labelled leaves, *J. Classif.* **3** (1986), 335–348.
[10] F. R. McMorris, D. B. Meronk, and D. A. Neumann, A view of some consensus methods for trees, in *Numerical Taxonomy* (ed. J. Felsenstein) NATO ASI Series **G1** (1983), 122–126.
[11] M. P. Ng and N. C. Wormald, Reconstruction of rooted trees from subtrees, *Discr. Appl. Math.* **69** (1996), 19–31.

[12] R. D. M. Page and E. C. Holmes, *Molecular evolution: a phylogenetic approach*, Blackwell Science (1998).
[13] A. Purvis, A composite estimate of primate phylogeny, *Phil. Trans. R. Soc. Lond. B* **348** (1995), 405–421.
[14] M. J. Sanderson, A. Purvis, and C. Henze, Phylogenetic supertrees: assembling the trees of life, *Trends Ecol. Evol.* **13(3)** (1998), 105–109.
[15] M. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, *J. Classif.* **9(1)** (1992), 91–116.

Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand

E-mail address: c.semple@math.canterbury.ac.nz, m.steel@math.canterbury.ac.nz