ELSEVIER

Discrete Applied Mathematics 71 (1996) 311–335

DISCRETE APPLIED MATHEMATICS

The asymmetric median tree – A new model for building consensus trees

Cynthia Phillips [a,1], Tandy J. Warnow [b,*]

[a] *Sandia National Labs, Albuquerque, NM, USA*

[b] *Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA*

Abstract

Inferring the consensus of a set of different evolutionary trees for a given species set is a well-studied problem, for which several different models have been proposed. In this paper, we propose a new optimization problem for consensus tree construction, which we call the *asymmetric median tree*, (*AMT*). Our main theoretical result is the equivalence between the asymmetric median tree problem on k trees and the maximum independent set (MIS) problem on k-colored graphs. Although the problem is NP-*hard* for three or more trees, we have polynomial-time algorithms to construct the AMT for two trees and an approximation algorithm for three or more trees. We define a measure of phylogenetic resolution and show that our algorithms (both exact and approximate) produce consensus trees that on every input are at least as resolved as the standard models in use (strict consensus, majority tree, Nelson tree). Finally, we show that the AMT combines desirable features of many of the standard consensus tree models in use.

1. The consensus tree problem

A fundamental problem in biology (and a number of other fields) is the inference of evolutionary trees for data sets. An evolutionary tree for a set S of taxa is a tree (sometimes required to be rooted, but not necessarily) with the elements of S as leaves. Often the practicing biologist is confronted with a set, called a *profile*, of different possible evolutionary trees. Different hypotheses for the evolutionary history can arise in many ways, one of which is that the taxa can be described through different types of data (e.g. morphological vs. biomolecular), each requiring perhaps a different method of analysis, and perhaps each suggesting a different evolutionary history (such as can happen when gene trees and species trees do not match precisely [30, 31]). Finally, even when the data are all of the same type and are all

* Corresponding author. E-mail: tandy@central.cis.upenn.edu. Research partly supported by an NSF National Young Investigator Award under contract CCR-9457800, by an NSF grant in Linguistics under contract SBR-9512092, by the Institute for Research in Cognitive Science at the University of Pennsylvania, and by generous financial support from Paul Angello.

[1] This work was performed under US Department of Energy under contract DE-AC04-76AL85000.

PII S0166-218X(96)00071-6

derived from one tree, thus permitting an analysis based upon one data set rather than several, it may not be the case that a single tree will be obtained from that one input. One of the reason for this to happen is that it may be necessary (for example, if the optimization surface is relatively flat) to consider many near-optimal trees in cases where the optimal tree is not obviously significantly superior to the near-optimal trees. Another reason is when there is reason to believe that the true, or model, tree may only obtain a near-optimal score, rather than a globally optimal score (as has been shown for a variety of optimization criteria for simulated data [16, 19, 20]). Finally, optimization problems in this area are for the most part NP-hard, so that finding optimal trees can be quite difficult and sometimes even finding near-optimal trees can be difficult. Thus, handling multiple hypotheses of evolution is a necessity in evolutionary tree construction methodology. Because in the end the objective is, if possible, a single evolutionary tree, the problem of inferring a consensus tree arises.

Several methods have been proposed for inferring consensus trees. One approach has been to eliminate taxa in order to infer a better consensus; methods along such lines include computing the *maximum agreement subtree* [13, 14, 24, 25, 33]. Most approaches presume that the consensus tree must include all the taxa; examples for such consensus trees include the *strict consensus* [3, 7], the *majority tree* [34] and its close relative the *median tree* [4], the *Nelson consensus* [27], the *Adams consensus* [1, 2], the *compatibility tree* [18, 35], and the recently introduced *local consensus tree* [23]; most of these methods have been made available through publically distributed software. Probably, the most popular of these methods are the strict consensus and majority tree; however, on many data sets both the strict consensus and the majority tree can be rather unresolved, and hence may not indicate many decisions about the evolutionary history of the taxa.

Motivated by a desire to produce resolved consensus trees, this paper presents a new consensus tree, which we call an *asymmetric median tree*. We propose an optimization criterion and define the asymmetric median trees of a profile of trees to be those trees optimizing that criterion. Since our objective in defining this new consensus is to obtain a tree which maintains as much evolutionary information as possible contained in the input set of trees, we propose a measure of degree of resolution which quantifies the evolutionary information in a phylogenetic tree. The results in this paper can be summarized as follows:

- Algorithms:
  - We present a polynomial-time algorithm to find an asymmetric median tree for a profile of two trees.
  - We present a polynomial-delay algorithm to enumerate all asymmetric median trees for a profile of two trees.
  - We present a polynomial-time algorithm to determine if a degree-d asymmetric median tree exists for an arbitrary profile of trees.
  - We present a polynomial-time algorithm to *approximate* the asymmetric median tree for an arbitrary profile of trees.

- *Hardness results (see section 9 for comments on computing in practice)*:
  - For three or more arbitrary trees, and for an unbounded number of binary trees, finding an asymmetric median tree is NP-hard.
  - Approximating the AMT of an unbounded number of arbitrary trees is hard.
- Comparison to other methods:
  - We propose a measure of how informative an evolutionary tree is, which we call the *degree of resolution*, and we show that for any profile of trees, both the exact and approximate solutions for the asymmetric median tree are at least as informative (according to the definition we will propose) as the Nelson consensus, the majority tree, the strict consensus tree, and any median tree.
  - We show that when the compatibility tree exists, our methods (both exact and approximate) will return it as the asymmetric median tree.

Our results therefore provide methods for efficiently inferring from an arbitrary profile of evolutionary trees a consensus tree that contains at least as much information, and potentially significantly more information, than the most popular methods used today for consensus tree construction.

2. Preliminaries

2.1. Different types of biological data

Recent methodological work for inferring evolutionary history has examined the problem of combining different types of evolutionary information (morphological, biomolecular, paleontological, etc.), where it may be necessary to do separate analyses of different types of data, and then combine the evolutionary trees afterwards into one single tree representing all the information from the different data sets. This is the *consensus tree problem.* Some of these types of data may yield only partially resolved trees; for example, the vertebrate–invertebrate distinction is best represented by a rooted tree with one internal edge separating vertebrates from invertebrates. Many of the existing methods produce consensus trees that are unresolved or poorly resolved. In this paper we propose a new model for inferring consensus trees which is able to handle such cases without losing information from the input set of trees. We call this consensus tree the *Asymmetric median tree.* This consensus tree is appropriate for a larger range of data types, and will perform at least as well (in terms of resolution) as the current popular methods on any data set.

Our major technical contribution is the observation given in Theorem 2 that the Asymmetric Median Tree problem (AMT) for k trees is equivalent to the maximum independent set problem on k-colored graphs. Based upon this, we solve the asymmetric median tree problem in polynomial time for two trees, but show it is NP-*hard* to solve for three or more trees. We also show that we can approximate the value of the asymmetric median tree of k trees to at least a factor of $2/k$, but when k is part of the

input the problem is hard to approximate. We also give a polynomial time algorithm for the case where the degree of the AMT is bounded.

2.2. Basic definitions

Let S be a finite set of species. We generalize the notion of an evolutionary tree (in which the species label the leaves) by defining an *S-labelled tree*. In this case, the labels at the internal nodes can be disjoint subsets of species. This generalization is made only so that we can prove an equivalence between the AMT problem and a graph-theoretic formulation, and makes no difference to the discussion otherwise. A *profile* is a set $\mathscr{T} = \{T_1, T_2, \ldots, T_k\}$ of S-labelled trees. A *consensus function* is a map $\phi : \mathscr{P}_S \rightarrow \mathscr{T}_S$, where $\mathscr{P}_S$ is the set of profiles of S-labelled trees, and $\mathscr{T}_S$ is the set of S-labelled trees.

Definition 1. A *character* on a set S is a function $c : S \rightarrow \mathscr{Z}$, for $\mathscr{Z}$ the set of integers.

Characters are the basis of most tree construction methods and can be derived from morphological, biomolecular, or other types of data. When a tree is S-labelled, we can define the tree by a set of binary (two-state) characters, called the character encoding as follows.

Definition 2. Given an S-labelled tree T, each edge $e \in E(T)$ defines a bipartition of the species set S. We represent this partition as a character $c_e : S \rightarrow \{0, 1\}$ so that species on the same side of e are assigned the same state.

The tree can then be represented by the set of characters derived from its edges.

Definition 3. The set $C(T) = \{c_e : e \in E(T)\}$ is called the *character encoding* of T.

Given two S-labelled trees T and T', if $C(T) \subset C(T')$, we say that T' *refines* T; in this case T' makes more decisions about the evolutionary history of S than T does.

2.3. Previous consensus models based upon character encodings

Some previous methods for inferring consensus are defined in terms of the character encodings of the trees in the profile; this common characterization allows us to examine them in relation to each other. For the following discussion, assume that $\mathscr{T} = \{T_1, T_2, \ldots, T_k\}$ is a profile of trees, each leaf-labelled by S, with $|S| = n$.

The compatibility tree: In the classical *Tree Compatibility Problem* (also called the *Cladistic Character Compatibility Problem*) [12, 18, 35], we wish to find T such that $C(T) = \cup_i C(T_i)$. When such a tree exists, the profile $\mathscr{T}$ is said to be "compatible," and T is called the *compatibility tree*. Determining whether the compatibility tree exists and constructing it when it does can be done in $O(nk)$ time[18, 35].

The strict consensus: The *strict consensus tree* [3, 7] contains only the common information; that is, the strict consensus tree T satisfies $C(T) = \cap_i C(T_i)$. The strict consensus tree always exists, and can be constructed in $O(nk)$ time as well.

The majority tree: The *majority tree* [34] contains exactly characters that appear in more than half the input trees. That is, $C(T) = \{\alpha : |\{i : \alpha \in C(T_i)\}| > k/2\}$. The majority tree is unique and always exists, and can be constructed in $O(n^2k)$ time.

The median tree: A median tree [4] minimizes the function $f_{\text{med}}(T, \mathcal{T}) = \sum_i |C(T) \triangle C(T_i)|$, where $\triangle$ denotes the symmetric difference. The majority tree is a median tree, so that there is always at least one median tree and it can be constructed in $O(n^2k)$ time. When k is even, the median tree can also contain characters appearing in *exactly* half the input trees.

The Nelson Consensus: The Nelson tree [27] is more complicated to define than the previous consensus trees. Given a profile of trees and character $c : S \to \{0, 1\}$, we weight c by the number of trees in the profile whose encodings include c. Thus, $w(c) = |\{i : c \in C(T_i)|\}$. Characters not contained in any profile's encoding have weight 0. Given a profile $P = T_1, T_2, \ldots, T_k$ of trees, we define the *Nelson basis* of P to be the set $NB(P)$ of trees T such that $Nelson(T) \equiv w(T) - |C(T)|$ is maximum over all trees on S. Note that characters c which are not in $\cup C(T_i)$ will not be in any tree in the Nelson Basis, because the removal of such characters (by contracting the edge defining that character) results in a tree of higher value. Thus, all trees in the Nelson Basis satisfy $C(T) \subset \cup C(T_i)$. The Nelson tree is then defined to be the strict consensus of the trees in the Nelson Basis i.e. the unique tree T_N such that $C(T_N) = \cap_{T \in NB(P)} C(T)$.

Constructing the Nelson Tree can be seen as having the following steps (though it need not be implemented in this way).

Step 1: The set of characters from the encodings of the different trees is obtained. Each character is assigned a weight equal to the number of times it has appeared in the profile. A graph is constructed from this set of characters in which the nodes are the distinct characters, and two nodes are connected by an edge if their associated characters are incompatible (this can be determined in $O(n)$ time per pair of characters). Note that each independent set I in the graph defines a tree T_I by $C(T_I) = I$.

Step 2: Given an independent set I in the graph, let $W(I) = \sum_{v \in I} w(v)$. For every independent set I maximizing $W(I) - |I|$, include T_I in the profile $NB(P)$ (note, this is the Nelson basis).

Step 3: Return the strict consensus of $NB(P)$; this is the Nelson tree.

Computing the Nelson tree in this way can obviously be computationally expensive, because computing the Nelson Basis is at least as hard as computing the maximum independent set. The question of how to compute the Nelson tree efficiently is beyond the scope of this paper.

Example: Fig. 1 illustrates the various character-based consensus methods just defined applied to a three-tree example. The example input trees are taken from [29], which is in turn a copy from [28]. Edges in the trees correspond to characters and are labelled

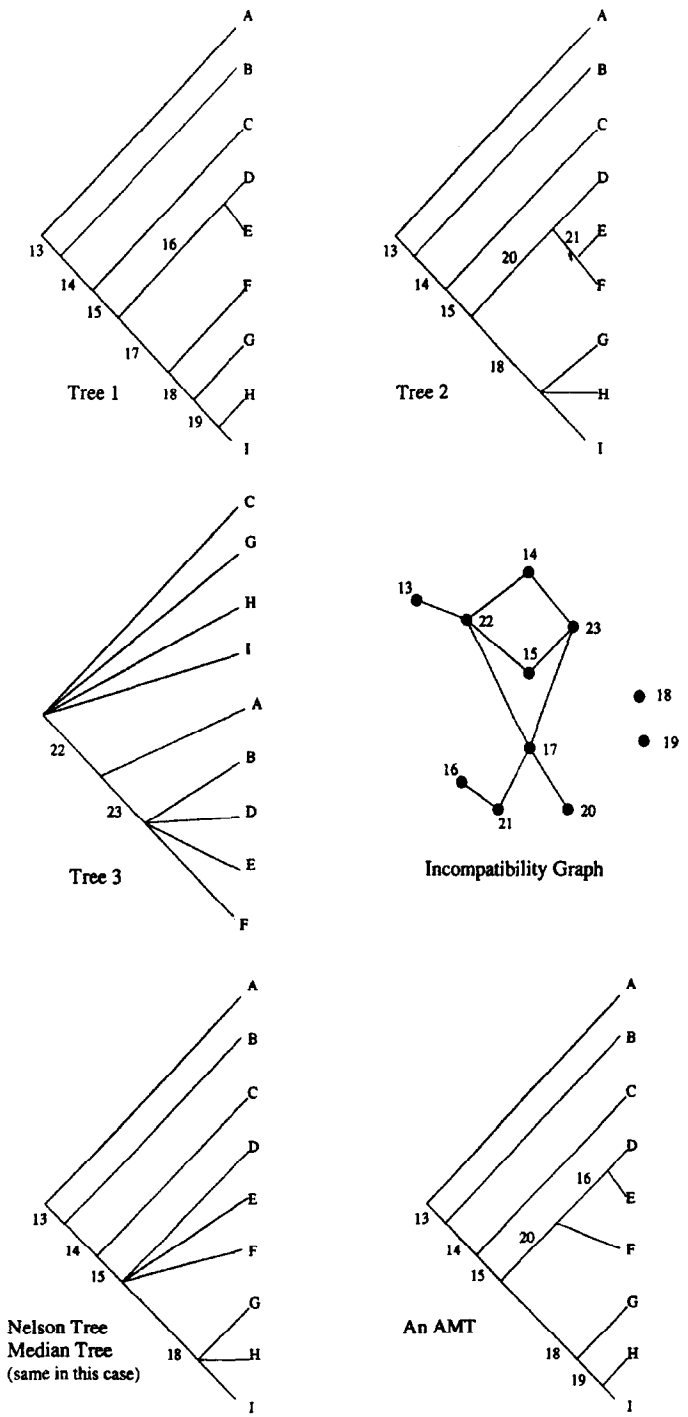

Fig. 1. An example from [29, 28]. Trees 1, 2, and 3 are the input trees (the profile). The compatibility tree does not exist and the strict consensus tree is the star graph over the taxa. In this case, the Nelson tree is the same as the median and majority tree, and all AMTs are refinements of this tree. Note that Page [29] and Nelson and Platnick [28] erroneously included edge 19 in the Nelson tree.

as they were in [29, 28]. The "incompatibility graph" represents the incompatibility between pairs of characters, by representing the characters from the three trees as vertices, and indicating incompatibility between characters by edges between the corresponding vertices. Because no character appears in all three trees, the strict consensus tree is the star graph. Because there exist character incompatibilities (edges in the incompatibility graph), the compatibility tree does not exist.

The median tree, which is the same as the majority tree for an odd number of input trees, contains all characters that appear in at least half of the trees (i.e. two or more trees). There are four characters which appear in two trees: 13, 14, 15, and 18.

In this case, the Nelson Tree is equal to the median tree. Note that this is a *correction* to the tree given in [29, 28]. These papers add the character 19 to the Nelson tree. The following observation shows why character 19 should not be included.

Observation 1. *Any character appearing in only one tree of the input profile cannot be in the Nelson tree*

Proof. Consider a profile P where character c is in exactly one tree in P. For character c to be in the Nelson tree, it must be in every tree in the Nelson basis. Therefore, let T be a tree in the Nelson basis of P that contains character c and has $Nelson(T) = w(T) - |C(T)|$. Let T' be tree T with character c contracted. That is, $C(T') = C(T) - \{c\}$. Then $Nelson(T') = (w(T) - 1) - (|C(T)| - 1) = Nelson(T)$. Thus, tree T' has the same score and is also in the Nelson basis. Since character c is not contained in tree T', it cannot be in the strict consensus of the Nelson basis and is therefore not in the Nelson Tree. □

In this example, character 19 appears only in Tree 1 and therefore cannot be in the consensus tree. The four characters that appear in at least two trees are the only ones to be considered for inclusion in the Nelson tree. Since they are compatible, there is only one tree with highest Nelson score: the one that contains all four characters.

Discussion of the other character-based models: Of these models the median tree is perhaps the most used in the biological community since it provides more evolutionary information than the strict consensus tree, is efficiently computable, and always exists; but the Nelson Tree is also popular. The compatibility tree seems desirable, but is rarely found since if any two characters are incompatible, the compatibility tree will not exist.

2.4. The Asymmetric Median Tree Problem

The consensus tree we will propose originates from an examination of the limitations of the median tree, and is motivated by the desirability of the compatibility tree.

Consensus trees are often used to resolve discrepancies in trees produced by different methods. We assume that all input trees are equally reliable when they do make an

evolutionary decision, or at least that they meet some user-specified "reliability threshold". We will show that the popular consensus method, the median tree, frequently has low resolution because it can contain few internal edges (even when the trees in the profile are compatible). We therefore examine the optimization function, f_{med}, upon which the median tree is based.

If we expand f_{med}, we find that $f_{med}(T,\mathscr{T}) = \sum_i [|C(T) - C(T_i)| + |C(T_i) - C(T)|]$. This explicitly penalizes for characters in T which are not in at least half of the trees in $\mathscr{T}$. This seemed an unreasonable requirement for the consensus tree, especially given that information from each tree, whenever possible, should be used. When trees are based upon different types of information, and thus can be small, this is especially important. Thus, we reconsider the optimization criterion upon which the median tree is based.

Definition 4. Let $w(c)$ (the *weight* of character c) be the number of trees in the profile $\mathscr{T}$ containing c; i.e. $w(c) = |\{i : c \in C(T_i)\}|$. Let $C^*(\mathscr{T}) = \cap_i C(T_i)$. The value of a consensus tree T with respect to $\mathscr{T}$ is defined to be

$$val(T,\mathscr{T}) = \begin{cases} -\infty & \text{if } C^*(\mathscr{T}) - C(T) \neq \emptyset, \\ -\infty & \text{if } C(T) - \cup_i C(T_i) \neq \emptyset, \\ \displaystyle\sum_{[c \in C(T) - C^*(\mathscr{T})]} w(c) & \text{otherwise.} \end{cases}$$

Characters which are in $C(T)$ but not in any of the trees in the profile are forbidden. Allowing them would not effect $\sum w(c)$, but inclusion of such characters would be misleading as it would imply evolution not supported by any tree in the profile. Characters which appear more often in the profile are of greater weight than those which appear rarely. However, characters which appear in *every* tree are treated as a special case. These characters can *always* be added to a tree T such that $C(T) \subseteq \cup_i C(T_i)$. Had we defined $val(T,\mathscr{T}) = \sum_{[c \in C(T)]} w(c)$, as might seem natural, these characters would always be included. Indeed the first line of the definition requires they be present, but we do not count them in the calculation of $val(T,\mathscr{T})$. This is a technicality added to rule out trivial "approximation algorithms". We will motivate this choice more fully in Section 5.

We can now define the asymmetric median tree.

Definition 5. Let $\mathscr{T} = \{T_1, T_2, \ldots, T_k\}$ be a profile of S-labelled trees. Any tree T containing only characters in $\cup_i C(T_i)$ and minimizing $f_{a.med}(T,\mathscr{T}) = \sum_i |C(T_i) - C(T)|$ (equivalently, maximizing $\text{val}(T,\mathscr{T})$) is an *Asymmetric Median Tree (AMT)*, denoted $T_{\text{a.med}}$.

The Asymmetric Median Tree (**AMT**) Problem is:
Input: A profile $\mathscr{T}$ of S-labelled trees.
Output: S-labelled tree T such that $val(T,\mathscr{T})$ is maximized.

Note then the following observation.

Observation 2. *Let T and T' be S-labelled trees with $C(T') \subseteq \cup_{[T_i \in \mathcal{T}]} C(T_i)$. If T' refines T (i.e. $C(T) \subset C(T')$), then $f_{a.\text{med}}(T', \mathcal{T}) \leqslant f_{a.\text{med}}(T, \mathcal{T})$.*

For an example, consider the profile of trees in Fig. 1. There are three AMTs for this profile. Any maximum-value character set for these three trees contains the four characters from the median tree (and Nelson tree) which are each contained in two input trees. These characters are compatible with three sets of three singleton characters (characters that appear in only one tree): $\{16, 17, 19\}, \{16, 19, 20\}$, and $\{19, 20, 21\}$. Thus, Tree 1 is in fact an AMT.

In some cases, it may be desirable to select the tree from the profile (input) that maximizes $val(T, \mathcal{T})$; the corresponding approach has been suggested by Penny and Hendy for constructing median trees. We call this the Restricted AMT problem. The Restricted AMT problem is easily solved in polynomial time, for any size profile and any number of taxa.

3. Characterizing the Asymmetric Median Tree

Recall that an asymmetric median tree of a profile $\mathcal{T}$ is any S-labelled tree minimizing $f_{a.\text{med}}$ or, equivalently, maximizing the value with respect to $\mathcal{T}$. Every profile of S-labelled trees has an AMT, and possibly several. In this section we explore the structure of the AMT and show that the AMT problem on k trees is polynomially equivalent to finding maximum independent sets on k-colored graphs.

We first review the theory of binary character compatibility and apply it to the AMT problem.

Definition 6. A set C of binary characters on a set S is *compatible* if there exists an S-labelled tree T such that $C \subseteq C(T)$.

Lemma 1 (Estabrook and McMorris[12]). *A set C of binary characters is compatible if and only if there exists a tree T with $C(T) = C$.*

Lemma 2 (Estabrook and McMorris[12]). *A set C of binary characters is compatible if and only if each pair of characters in C is compatible.*

The following lemma appears in [18] as part of the folklore in this area.

Lemma 3. *Let S be a set of species, C a set of binary characters defined on S and assume $\exists s \in S$ such that $c(s) = 0$ for all $c \in C$. Then C is compatible if and only if for every pair α, β of characters in C, $\phi(\alpha) \cap \phi(\beta) = \emptyset$ or $\phi(\alpha) \subseteq \phi(\beta)$ or $\phi(\beta) \subseteq \phi(\alpha)$.*

The following definitions will be used throughout the paper. Let $c : S \to \{0,1\}$. We let $\phi(c) = c^{-1}(1) = \{s \in S : c(s) = 1\}$. By specifying S and $\phi(c)$ we uniquely define the character c. Let S be a set of species defined by binary characters C. For each $s \in S$, define the binary character α_s by $\alpha_s(s) = 1$ and $\alpha_s(s') = 0$ for all $s' \in S - \{s\}$. We let $C_S = \{\alpha_s : s \in S\}$. Note that α_s may not belong to C, so this definition defines a (possibly) new character. The characters in C_S are part of the encoding of every *evolutionary* tree, though not necessarily part of every S-labelled tree. We will call C_S *trivial* characters because they can be added to any tree on species set S and they add no evolutionary information. We will use this observation in the future, so we state it now:

Observation 3. *Let $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ be a profile of S-labelled trees and let T be any S-labelled tree such that $C(T) \subseteq \cup_i C(T_i)$. Then $C(T) \cup C_S$ is a compatible set of characters. Equivalently, let $C_0 \subseteq \cup_i C(T_i)$ be any subset of compatible characters. Then $C_0 \cup C_S$ is a compatible set of characters.*

The AMT problem on S-labelled trees is not harder than the AMT problem on evolutionary trees, as the following lemma shows. Note that even for S-labelled trees, the size of the input for k trees on n species is $\Theta(nk)$, from the leaf labels alone.

Lemma 4. *If AMT on k evolutionary trees can be solved in $O(f(n,k))$ time where $n = |S|$, then AMT on k S-labelled trees can be solved in $O(f(n,k) + nk)$ time.*

Proof. To compute the AMT of k S-labelled trees, $T_1, T_2, \ldots, T_k$, let T_i' be the evolutionary tree defined by $C(T_i') = C(T_i) \cup C_S$. Let T' be the AMT of this new profile $T_1', T_2', \ldots, T_k'$. The AMT T of $T_1, T_2, \ldots, T_k$ is then defined by $C(T) = C(T') \cap (\cup_i C(T_i))$. □

Corollary 1. *The AMT problem on evolutionary trees is polynomially equivalent to the AMT problem on S-labelled trees.*

As a consequence we do not need to distinguish between S-labelled trees and evolutionary trees, and can simply refer to either as "trees" without loss of generality.

3.1. The incompatibility graph

We begin with some basic definitions in graph theory that will be relevant to our discussion. Let $G = (V,E)$ be a graph. An *independent set* is a subset $V_0 \subseteq V$ such that for all $v, w \in V_0, (v,w) \notin E$. Let $v \in V$. Then we say v is *isolated* if for all $w \in V$, $(v,w) \notin E$. Let $G = (V,E)$ be a graph. A *k-coloring* is a function $c : V \to \{1,2,\ldots,k\}$ such that for all $(v,w) \in E$, $c(v) \neq c(w)$. We say that G is *k-colored* if we are given G and the k-coloring c. We say that G is *k-partite* if a k-coloring of G exists. We now define the incompatibility graph. Let $\mathcal{T} = T_1, T_2, \ldots, T_k$ be S-labelled trees. The *incompatibility graph* of $\mathcal{T}$ is the k-partite graph $G(\mathcal{T}) = (V_1, V_2, \ldots, V_k, E)$ with

$V_i = \{v_e : e \in E(T_i)\}$ and $E = \{(v_e, v_{e'}) : c_e$ and $c_{e'}$ are incompatible$\}$. Note that $G(\mathcal{T})$ is given with the k-partition (i.e. it is k-colored).

We will now describe a representation of an arbitrary k-colored graph G as the incompatibility graph of k trees. We begin by defining a set of binary characters we will use throughout the paper.

Definition 7. Let $G = (V,E)$ be an arbitrary graph. We define a set **F(G)** of $|V|$ binary characters on a species set S as follows. Let $S = V \cup E$, (so $|S| = n+m$, where $n = |V|$ and $m = |E|$). The character c_v associated to node $v \in V$ is defined by $c_v(s) = 1$ if and only if $s \in \{v\} \cup \{(v,w) \in E\}$ (i.e. $\phi(c_v) = \{v\} \cup \{(v,w) \in E\}$).

Lemma 5. *Let $G = (V,E)$ be a k-partite graph with parts $V_1, V_2, \ldots, V_k$, and let $S = V \cup E \cup \{r\}$. Then there exists a set $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ of S-labelled trees, such that $G = G(\mathcal{T})$.*

Proof. Let $F(G)$ be the set of binary characters defined above, and let v and w be arbitrary vertices in G. If $(v,w) \in E$ then $\phi(c_v) \cap \phi(c_w) \neq \emptyset$, $\phi(c_v) \not\subseteq \phi(c_w)$ and $\phi(c_w) \not\subseteq \phi(c_v)$. Hence, by Lemma 3, c_v and c_w are incompatible. Conversely, if $(v,w) \notin E$ then $\phi(c_v) \cap \phi(c_w) = \emptyset$ so that c_v and c_w are compatible. Thus $(v,w) \in E$ if and only if c_v and c_w are incompatible.

Now consider v, w drawn from the same part of G. Then c_v and c_w are compatible since $(v,w) \notin E$. Thus, by Lemma 2, the set $\{c_v : v \in V_i\}$ is a compatible set of binary characters and thus, by Lemma 1, there exists a tree T_i such that $C(T_i) = \{c_v : v \in V_i\}$. Hence, G is the incompatibility graph of $T_1, T_2, \ldots, T_k$ under the mapping $V_i \to C(T_i)$. □

Thus, given a graph G and a partition of G into k independent sets (i.e. a proper k-coloring), this theorem gives a canonical representation of G as the incompatibility graph of k S-labelled trees, where $S = V \cup E \cup \{r\}$.

3.2. The character encoding of $T_{a.\text{med}}$

Recall that $C^*(\mathcal{T}) = \cap_i C(T_i)$.

Definition 8. Let $V^*(\mathcal{T})$ be the nodes of $G(\mathcal{T})$ corresponding to the character set $C^*(\mathcal{T})$.

Observation 4. *$V^*(\mathcal{T})$ is a set of isolated vertices in $G(\mathcal{T})$.*

Theorem 1. *Let $\mathcal{T}$ be a profile of S-labelled trees. Then the AMT on $\mathcal{T}$ has value v if and only if $G(\mathcal{T}) - V^*(\mathcal{T})$ has an independent set of size v.*

Proof. Suppose $G(\mathcal{T}) - V^*(\mathcal{T})$ has an independent set V_0 of size v. Let C_0 be the character set associated to V_0. Thus, $C_0 \subseteq \cup_i C(T_i)$, and since V_0 is independent, C_0

is a set of pairwise compatible characters. By Lemma 2, C_0 is a compatible set of characters, and by the same reasoning $C_1 = C_0 \cup C^*(\mathscr{T})$ is also a compatible set of characters. Hence, by Lemma 1 there exists an S-labelled tree T such that $C_1 = C(T)$. Note that $val(T, \mathscr{T}) = \sum_{c \in C_1 - C^*(\mathscr{T})} w(c) = \sum_{c \in C_0} w(c) \geq |V_0|$. Equality is achieved if each character $c \in C_0$ is included $w(c)$ times in V_0.

For the converse, suppose that $val(T, \mathscr{T}) = k$ where T is an asymmetric median tree for $\mathscr{T}$. Let $C_0 = C(T) - C^*(\mathscr{T})$, and let V_0 be the vertices associated to C_0; since C_0 is a compatible set of characters, V_0 is an independent set of vertices. Then $k = val(T, \mathscr{T}) = \sum_{c \in C(T) - C^*(\mathscr{T})} w(c) = \sum_{c \in C_0} w(c) = |V_0|$, so that $G(\mathscr{T}) - V^*(\mathscr{T})$ has an independent set of size k. □

Corollary 2. *If T is an AMT, then $C(T)$ is a maximum independent set in $G(\mathscr{T})$.*

We conclude with a fundamental observation.

Theorem 2. *The AMT problem for k trees is polynomially equivalent to the Independent Set Problem on k-colored graphs, and the class of k-partite graphs equals the class of incompatibility graphs of k S-labelled trees.*

4. Computing the asymmetric median tree of two S-labelled trees

By Theorem 1, if we can compute the maximum independent set of a k-partite graph, we can compute the asymmetric median tree of k S-labelled trees. Since the maximum independent set of a bipartite graph can be computed in polynomial time ([17], the algorithm is folklore), we are able to compute the AMT of two trees in polynomial time. In this section we will describe an algorithm for doing this, with indications of where it might be possible to speed up the algorithm.

4.1. The Algorithm

Step 1: Compute $G(T_1, T_2)$

Step 2: Compute the maximum independent set I of vertices in $G(T_1, T_2)$. Let C_0 be the characters associated with I, where $C_0 \subseteq C(T_1) \cup C(T_2)$.

Step 3: Compute T satisfying $C(T) = C_0$, and return T.

4.1.1. An easy but not so efficient implementation

Step 1: To compute $G(T_1, T_2)$, we apply Lemma 3 to each pair of characters $\alpha \in C(T_1), \beta \in C(T_2)$. This will require $O(n)$ work for each pair of characters, and hence $O(n^3)$ time overall.

Step 2: We use the folklore algorithm which solves the maximum independent set problem in bipartite graphs in $O(n^{2.5})$, where n is the total number of vertices.

Step 3: We use the $O(vn)$ algorithm of [18] to construct the tree T satisfying $C(T) = C_0$, where $v = |C_0|$ and $n = |S|$. In our case, $v \leqslant 2n$ so that Step 3 will require no more than $O(n^2)$ time.

Thus, this implementation has $O(n^3)$ time.

4.1.2. A faster implementation

By Lemma 4, we need only describe the algorithm for two evolutionary trees. We transform T_1 and T_2 into evolutionary trees by defining T_i' such that $C(T_i') = C(T_i) \cup C_S$ for $i = 1,2$. These trees exist and can be constructed in $O(n)$ time. The profile $\{T_1', T_2'\}$ is the input to the AMT algorithm. Let $\hat{C} = \{\alpha_s : \alpha_s \notin C(T_i), i = 1,2\}$. Then $C(T) - \hat{C}$ is the character encoding of an AMT of the profile $\{T_1, T_2\}$.

Because we assume T_1 and T_2 are evolutionary trees, $n \leqslant |V(T_i)| \leqslant 2n$, where $|S| = n$ for $i = 1,2$. Let $n_1 = |V(T_1)|$ and $n_2 = |V(T_2)|$. We demonstrate here that we can speed up Steps 1 and 3 of the algorithm of Section 4.1.1. Step 1 can be computed in time $O(n_1 n_2) = O(n^2)$ and Step 3 can be computed in $O(n)$ time, so that overall we will require $O(n_1 n_2 + f(n_1 + n_2))$ time, where $f(z)$ is the cost of computing the maximum independent set of a bipartite graph on z vertices. The bound of the best algorithm known for this problem is $O(n^{2.5})$[21].

Constructing the incompatibility graph. We can compute the incompatibility graph $G(T_1, T_2)$(Step 1) in time $O(n_1 n_2)$ as follows. Root each of T_1 and T_2 at a particular node, s_1. Let $\phi(e)$ indicate the set of species (leaves) in the subtree below edge e. For each edge $e \in T_1$, we will compute $T_2(e)$, a copy of tree T_2 with each edge $e' \in T_2$ labelled "+" if e is compatible with e' and "−" otherwise. Recall that edges e and e' are incompatible if and only if $\phi(e)$ and $\phi(e')$ properly intersect. Edges labelled with "−" are the neighbors of node e in the incompatibility graph. To assist in the computation of incompatibility, each edge $e' \in T_2(e)$ is also given a *count* $c(e')$ equal to $|\phi(e) \cap \phi(e')|$. In our data structure for tree $T_2(e)$, we store the leaves in an array. Therefore, each leaf in $T_2(e)$ can be identified by an index. When we compute tree $T_2(e)$, we also compute the set $L(e)$ of indices of the leaves in T_2 corresponding to the set $\phi(e)$.

Let $e = (v_1, v_2) \in E(T_1)$ and suppose node v_2 is further from the root in tree T_1 than node v_1. We say that edges of the form (v_2, x) are *children* of edge e. We process each edge e in T_1 after processing all children of edge e.

If e is an edge to a leaf in tree T_1 ($|\phi(e)| = 1$), then we compute $T_2(e)$ as follows. Every edge in $T_2(e)$ is labelled "+" since a set of size one cannot properly intersect any other set. We do not need the counts in this case to assist in the determination of compatibility. The set $L(e)$ is the index of the single leaf corresponding to $\phi(e)$ which is found by searching the leaf array.

Suppose that edge e has children $e_1, e_2, \ldots, e_k$ and that we have computed $L(e_i)$ for $i = 1, \ldots, k$. We now wish to compute the labels for $T_2(e)$. We begin with an unlabelled copy of T_2 and mark all the edges into the leaves with "+" and counts

of 0. We then go through each $L(e_i)$ in order and change the counts of the the edges into these leaves to 1. We then make a pass over all the leaves in T_2 to compute $L(e)$.

Now consider an edge $e' \in T_2$ with children $e'_1, e'_2, \ldots, e'_l$. Consider the labels and counts on the edges e'_i. We have $c(e') = \sum_{i=1}^{l} c(e'_i)$. If $c(e') = |\phi(e)|$, then $\phi(e) \subseteq \phi(e')$ and edges $e \in T_1$ and $e' \in T_2$ are compatible (labelled "+"). Since every edge in the subtree underneath edge e' has been computed and only these edges can be incompatible, we can terminate the algorithm when this condition is detected. If $c(e') = 0$, then the edges are disjoint, and therefore also compatible. If $0 < c(e') < |\phi(e)|$, all e'_i are labelled "+" and $c(e'_i) > 0$ for all $i = 1, \ldots, l$, then $\phi(e') \subset \phi(e)$ and the edges are compatible. Otherwise, when $0 < c(e') < |\phi(e)|$ and either some child of e' is incompatible (marked "-") or some child e'_i has $c(e'_i) = 0$, then the subsets intersect properly and edges e and e' are incompatible.

Computing the $L(e)$ costs $O(n_1 n_2)$ overall. Each of the n_1 sets is of size $O(n_2)$. It is created once in time $O(n_2)$ and used once by the parent of e (read linearly). We compute compatibility of each of the $O(n_1 n_2)$ pairings of an edge in T_1 with an edge in T_2 (each pair corresponding to a possible edge in the incompatibility graph). Each edge is computed in time $O(d(e))$, where $d(e)$ is the number of children of e (the degree of the endpoint farthest from the root). We have that $\sum_{e \in T_2} d(e) = O(n_2)$, however, so each edge is computed in constant amortized time.

Theorem 3. *The Incompatibility Graph of two S-labelled trees can be constructed in $O(n^2)$ time, where $|S| = n$.*

It then follows:

Corollary 3. *The Incompatibility Graph of k S-labelled trees can be constructed in $O(k^2 n^2)$ time, where $|S| = n$.*

We compute the tree containing the characters of the maximum independent set (Step 3 of the algorithm in Section 4.1.1) as follows. Having computed the maximum independent set V_0 of size n in the bipartite graph, we can find in $O(n)$ time the character set C_0 associated with V_0, and the edge set E_0 associated with C_0. Thus, $C_0 = \{c_e : e \in E_0\}$. We contract each edge $e \in E(T_1) \cup E(T_2) - E_0$, and obtain two trees T'_1 and T'_2. This takes only $O(n)$ time. We then compute the compatibility tree T^* of T'_1 and T'_2 in $O(n)$ time [18, 35]. Since $C(T^*) = C_0$, T^* is an AMT. We have proven:

Theorem 4. *The AMT of two evolutionary trees of size n_1 and n_2 can be computed in $O(n_1 n_2 + f(n_1 + n_2))$ time, where $f(z)$ is the time needed to compute the maximum independent set of a bipartite graph with z nodes. Hence, the AMT of two S-labelled trees can be computed in $O(n^{2.5})$ time, where $|S| = n$.*

4.2. Enumeration

Consider the problem of enumerating all the AMTs of a profile $P = \{T_1, T_2\}$ of S-labelled trees, and let $|S| = n$. This is equivalent to enumerating the maximum independent sets in the bipartite graph $G(P)$. Let us assume the maximum independent set size for $G(P)$ is s. Let the nodes of $G(P)$ be $v_1, v_2, \ldots, v_t$, for $t \leqslant 4n-2$ (since T_1 and T_2, even if fully resolved, have at most $2n-1$ edges). Let $\Gamma(v)$ denote the set of neighbors of v in $G(P)$, $G' = G(P) - \Gamma(v_1) - \{v_1\}$, and $G'' = G - \{v_1\}$. Then let $Indep(X,k)$ denote the set of independent sets of size k in the bipartite graph X. It is easy to see that $Indep(G(P),s) = \{X \cup \{v_1\} : X \in Indep(G', s-1)\} \cup Indep(G'', s)$.

4.2.1. The enumeration algorithm

This indicates a natural enumeration algorithm. We build a search tree. Each node is labelled by a pair (G,i), where G is a vertex induced subgraph of $G(P)$ and i is an integer. The root is labelled $(G(P),s)$, and its two children are labelled $(G', s-1)$ and (G'', s), respectively. The two children of a node (G,i) are computed as follows: let v be the first vertex (in the ordering $v_1, v_2, \ldots, v_t$) which is in G. Let $G' = G - \{v\} - \Gamma(v)$ and let $G'' = G - \{v\}$. The left child is labelled $(G', i-1)$ and the right child is labelled (G'', i).

We construct the search tree as follows. When we visit a node labelled (G,i) we run the $O(n^{2.5})$ maximum independent set algorithm of [21] to determine if there is an independent set of size at least i in G. If the answer is *yes*, then we continue to search from this node (now called a *yes*-node), and otherwise we back up the tree. By construction, if (G,i) is a yes-node, then at least one of its children is a yes-node, and the no-nodes indicate deadends. Each path from the root to a yes-leaf indicates a unique maximum independent set in $G(P)$.

Theorem 5. *The set $\mathcal{A}$ of all AMTs of a profile P of two S-labelled trees can be computed in $O(n^{3.5}p)$ time where $p = |\mathcal{A}|$.*

Proof. Both s, the size of the maximum independent set of $G(P)$ and t, the number of nodes in $G(P)$ are $O(n)$. The number of internal nodes of the search tree is bounded by $O(np)$ since descending from a node (G,i) to one of its children decreases either the the size of the first component (i.e. the number of vertices in the subgraph of $G(P)$) or the size of the second component (i.e. the size of the independent set). The leaves in this search tree either indicate maximum independent sets or dead-ends; since the number of internal nodes is bounded by $O(np)$, the total number of nodes is also bounded by $O(np)$ since each internal node has two children. Since visiting a node costs at most $O(n^{2.5})$ (it entails one maximum independent set calculation) the algorithm has running time $O(n^{3.5}p)$. □

Constructing the search tree depth-first yields a polynomial-delay listing algorithm.

5. Hardness results

We will show in this section that computing an AMT of three or more trees is NP-*hard* and that the value of the AMT of k trees is hard to approximate if k is not bounded. We will also show that the problem remains hard for k unbounded even if we constrain all the input trees to be fully resolved evolutionary trees.

We begin by proving that finding a maximum independent set (MIS) of a tripartite graph is NP-complete. Recall that MIS of bipartite graphs is known to be in P.

Lemma 6. *Maximum independent set on* 3*-colored graphs is* NP-hard.

Proof. Let I be an instance of 1-in-3 3-SAT with n variables and m clauses. We now define a 3-colored graph G_I such that I has a 1-in-3 satisfying assignment if and only if G_I has an independent set of size $3n + m$.

For each variable X, make vertices $X_0(v), X_1(v), X_2(v), \overline{X}_0(v)\overline{X}_1(v), \overline{X}_2(v)$. For each clause $c = (X, \overline{Y}, Z)$ (these literals may be arbitrarily complemented or not), make vertices $X_0(c), \overline{Y}_1(c)$ and $Z_2(c)$. The nodes labelled with (v) are called *variable* nodes, and the nodes with (c) are called *clause* nodes. The edges are as follows. For each variable X, add edges $\{(X_i(v), \overline{X}_j(v)) : j \neq i\}$. If $(X, \overline{Y}, Z)$ is a clause, then add the triangle on $X_0(c), \overline{Y}_1(c)$, and $Z_2(c)$ along with edges $(X_0(c), \overline{X}_1(v))$, $(X_0(c), \overline{Y}_1(v))$, $(X_0(c), Z_1(v))$, $(\overline{Y}_1(c), Y_2(v))$, $(\overline{Y}_1(c), X_2(v))$, $(\overline{Y}_1(c), Z_2(v))$, $(Z_2(c), \overline{Z}_0(v))$, $(Z_2(c), X_0(v))$, and $(Z_2(c), \overline{Y}_0(v))$. The number in the subscript of each vertex indicates the color class. Since no edges are introduced between edges with the same subscript, the graph is 3-colored.

Let I be an independent set in G. Then I has at most one node associated with each clause (that is, at most one of the nodes $X_0(c), \overline{Y}_1(c), Z_2(c)$ is selected), and at most three nodes associated with each variable. To obtain $3n + m$ nodes in I, we must use exactly one of $X_0(c), \overline{Y}_1(c), Z_2(c)$ for each clause $(X, \overline{Y}, Z)$, and exactly 3 of the variable nodes for each variable. The only way to get an independent set of size three from the six variable nodes associated with X, is to choose $X_i(v)$ for $i = 0, 1, 2$ or to choose $\overline{X}_i(v)$ for $i = 0, 1, 2$. This defines the truth assignment. Note that if we choose $X_i(c)$ we cannot choose $\overline{X}_{i+1}(v), \overline{Y}_{i+1}(v)$ or $Z_{i+1}(v)$ (where the addition is taken modulo 3). Thus if node $X_0(c) \in I$, we have that $X_1(v) \in I, Y_1(v) \in I$, and $\overline{Z}_1(v) \in I$. Therefore, clause c is satisfied by exactly one literal. Since this is true of all clauses, we have that an independent set of size $3n + m$ provides a truth assignment satisfying instance I. The converse holds trivially. Given a solution to I, the set of vertices corresponding to the truth assignment and satisfying literal in each clause yields an independent set of the required size. □

Corollary 4. *The AMT problem is* NP-hard *when the profile contains three trees.*

Proof. By Corollary 1, the AMT problem on k S-labelled trees is polynomially equivalent to the AMT problem on k evolutionary trees. We will show that Independent

Set on tripartite graphs reduces to the AMT problem on three S-labelled trees. Let $(G = (V,E),k)$ be an input to the independent set problem and let G be tripartite; without loss of generality, we can assume G has no isolated vertices. We will define a profile $\mathcal{T}$ of three evolutionary trees such that G has an independent set of size k if and only if the value of the AMT of $\mathcal{T}$ is at least k.

By Lemma 5, there exists a profile $\mathcal{T} = \{T_1, T_2, T_3\}$ of S-labelled trees such that $G(\mathcal{T}) = G$. By Observation 4, $V^*(\mathcal{T})$ is a set of isolated vertices in $G(\mathcal{T}) = G$, but since G has no isolated vertices (by construction), $V^*(\mathcal{T}) = \emptyset$. Hence $C^*(\mathcal{T}) = \emptyset$. Note then that $G(\mathcal{T}) - V^*(\mathcal{T}) = G$, and that by Theorem 1 the value of the asymmetric median tree of $\mathcal{T}$ is the size of the maximum independent set in G. □

We will show in Section 6 that we can approximate the value of the AMT of k S-labelled trees when k is bounded. We now consider the problem of approximating the value of the AMT of k trees, when k is part of the input.

The value of an AMT of a profile $\mathcal{T}$, $val(T,\mathcal{T})$ is defined to count only characters which appear in some input tree, but not in all. In some sense, it measures difficult consensus decisions. If we had counted characters which appeared in every tree, then there would be a trival $\frac{1}{2}$-approximation algorithm for every profile of evolutionary trees. That is, there would be an algorithm that in polynomial time produces a tree whose value is at least $\frac{1}{2}$ the value of an AMT for that profile. This is because all evolutionary trees on species set S contain the trivial character set C_S. The total weight of characters in C_S for a profile of k trees is nk. However, an evolutionary tree on n species can have at most $2n-2$ edges. Therefore, the maximum value of an AMT is $(2n-2)k$ and the star graph containing only the set of trivial characters has more than half the maximum value.

By giving no credit for the strict consensus characters, approximation becomes harder.

Theorem 6. *The value of an AMT of a profile P containing k trees cannot be approximated to within a factor of $k^{1/4-o(1)}$ unless QNP = co-QR.*

Proof. We reduce maximum independent set to AMT. Suppose we are given a graph G. We define the set of binary characers $F(G)$ given in Definition 7. We form a profile $\mathcal{T}$ of trees for the AMT problem by creating a tree for each character. Thus, each tree in the profile has a single notrivial edge. We can add the trivial characters C_S to form evolutionary trees.

We now show that G contains an independent set of size t if and only if $\mathcal{T}$ has an asymmetric median tree of value t. Suppose G contains an independent set $\{v_1,\ldots,v_t\}$. From the argument in Lemma 5, the characters defined on these nodes are all compatible. Each has weight 1 (ie. appears in one tree of the profile) and therefore they define a tree T such that $val(T,\mathcal{T}) = t$.

Suppose conversely that the profile $\mathcal{T}$ has an AMT T with value t. Since no characer appears in more than one tree in $\mathcal{T}$, there are t characters in the AMT. These characters

represent a set of t nodes in G which are pairwise independent and therefore form an independent set of size t.

Since this is a linear reduction (the value of the AMT calculation is exactly the value of the independent set), the AMT problem is as hard to approximate as maximum independent set. Bellare and Sudan [5] have proved that the maximum clique (and therefore maximum independent set, since it is just clique on the complemented graph) on a graph with n nodes cannot be approximated to within a factor of $n^{1/4-o(1)}$ unless QNP = co-QR. Here Q stands for quasi-polynomial time (a function $T(n)$ is quasi-polynomial if there is a constant c such that $T(n) \leqslant n^{\log^c n}$). Thus, QNP is the quasi-polynomial version of NP and co-QR is the set of problems whose complement is in the quasi-polynomial version of R, where R, "random polynomial time" is a complexity class computationally closely related to P; see [22] for more details. □

6. Approximation algorithms

Since the AMT of k trees is hard to compute when $k \geqslant 3$, we consider the question of finding near-optimal trees. In this section we present polynomial-time algorithms for finding approximations to the asymmetric median tree which will always produce a tree at least as informative as the median tree. This guarantees that the output to the approximation algorithm will always be at least as good as the median tree (at least in terms of the evolutionary information content).

We present two algorithms for approximating the asymmetric median tree. The first approximates the value of the asymmetric median tree within a factor of $2/k$. We also give a second algorithm that is better when the value of the AMT is large.

Assume we are given a profile $\mathscr{T} = \{T_1, T_2, \ldots, T_k\}$ and let $T_{a.\text{med}}$ be an AMT for $\mathscr{T}$. Let $V_{\text{opt}} = \text{val}(T_{a.\text{med}}, \mathscr{T})$. By Theorem 1, $V_{\text{opt}} = MIS(G')$, where $G' = G(\mathscr{T}) - V^*(\mathscr{T})$.

6.1. Algorithm 1

For every pair T_i, T_j, compute the asymmetric median tree of T_i, T_j, and call it AMT_{ij}. Let T be the tree maximizing $val(AMT_{ij}, \mathscr{T})$ for any AMT_{ij}. Then T satisfies $val(T, \mathscr{T}) \geqslant 2V_{\text{opt}}/k$. To see this, let M be an arbitrary maximum independent set on graph G'. Let $M_1, \ldots, M_k$ be the partition induced on M by the profile (ie. M_1 are the nodes from the first tree, M_2 from the second tree, and so on). Let M_i and M_j be such that $|M_i| \geqslant |M_j| \geqslant |M_p|$ for all $p \neq i, j$. Then $|M_i| + |M_j| \geqslant (2/k)|M|$. $M_i + M_j$ is a candidate set for the MIS of trees i and j, but it may not be maximum. Therefore, $val(AMT_{ij}, \mathscr{T}) \geqslant |M_i| + |M_j|$, the result follows.

This algorithm has running time $O(k^2 n^{2.5})$, since it computes the asymmetric median tree for every pair of trees (of which there are $O(k^2)$).

6.2. Algorithm 2

Compute a maximum matching M in G'. Let I be the unmatched nodes, and C_I the characters associated to I. Return the tree T satisfying $C(T) = C_I$.

The running time of Algorithm 2 is $O((kn)^{2.5})$, since maximum matching can be done in an arbitrary graph $G = (V,E)$ in $O(|V|^{2.5})$ time [21].

This second approximation algorithm yields a tree that can take characters from many different input trees, whereas the first will produce a tree with characters from at most two trees. This distinction will in general make the second algorithm more desirable. However, the value of the resulting AMT is better only when there is substantial sharing of characters among the trees.

Theorem 7. *The guaranteed performance of Algorithm* 2 *is better than the guaranteed performance of Algorithm* 1 *when* $V_{\text{opt}} \geq Nk/(2k-2)$.

Proof. Let $N = |V(G(\mathscr{T}))| = \sum_i |E(T_i)|$ be the number of nodes in the k-partite incompatibility graph. Let M^* be the size of the minimum vertex cover in this graph. Then we have $V_{\text{opt}} = N - M^*$ and we can find an independent set of size at least $N - 2M^*$. We have that $(N - 2M^*)/(N - M^*) \geq 2/k$ only when $V_{\text{opt}} \geq Nk/(2k-2)$. □

6.3. Practical considerations

If desired, the output of the approximation algorithms described above can compared to the median tree, and the more informative of the two outputs returned as the approximation to the asymmetric median tree. This will ensure that the output to the approximation algorithm will always be at least as informative as the median tree.

7. Comparison to other models

In an earlier section, we examined a profile of three trees which demonstrated that the AMT of the profile was distinct from the median, Nelson, and strict consensus trees. In this section we will extend this comparative study. We define a natural measure of the degree of resolution of an S-labelled tree, and show that the asymmetric median tree is at least as resolved with respect to this measure as the strict consensus tree, the median tree, and the Nelson tree, on any profile. We will also show that when the compatibility tree exists, it is an AMT.

7.1. Degree of resolution

We are interested in quantifying the resolution provided by a consensus tree. We note that traditionally evolutionary trees are rooted, and an internal node with three or more children indicates not that a three-way speciation event occurred (which is

unlikely), but rather the inability to determine exactly what happened at that point in time. Thus, the most resolved evolutionary tree (in some sense the most informative) is binary, while the least resolved evolutionary tree is the star. The best quantification of resolution in a rooted evolutionary tree (or cladistic character) would be the number of resolved triples; a natural approximation of this is the number of edges in the tree.

Definition 9. The degree of resolution $Res(T)$ of an S-labelled tree $T = (V, E)$ is $|E|$.

We now compare the resolution of the four character-based consensus trees we have considered thus far: *strict consensus trees, median trees, AMTs,* and *compatibility trees.* The compatibility tree T_{comp}, when it exists, represents the sum of all the phylogenetic information available from the profile since $C(T_{\text{comp}}) = \cup_i C(T_i)$; that is, the trees are *compatible*[18]. Although compatibility is uncommon in practice, it is reasonable to desire that when the trees are compatible the output of a consensus method should indeed be the compatibility tree.

Theorem 8. *Suppose* $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ *is a profile of S-labelled trees. Let* T_{sc} *be the strict consensus tree,* T_{med} *be any median tree,* T_{nels} *the Nelson tree, and let* $T_{a.\text{med}}$ *be any asymmetric median tree. Then* $Res(T_{\text{sc}}) \leqslant Res(T_{\text{med}}) \leqslant Res(T_{a.\text{med}})$ *and* $Res(T_{\text{nels}}) \leqslant Res(T_{a.\text{med}})$. *Furthermore, if the compatibility tree* T_{comp} *exists, then* $T_{\text{comp}} = T_{a.\text{med}}$, *so that* $Res(T_{\text{comp}}) = Res(T_{a.\text{med}})$.

Proof. Because T_{sc} is refined by T_{med}, we know that $Res(T_{\text{sc}}) \leqslant Res(T_{\text{med}})$. The compatibility tree T_{comp} is characterized by $C(T_{\text{comp}}) = \cup_i C(T_i)$ and thus $f_{a.\text{med}}(T_{\text{comp}}, \mathcal{T}) = \sum_i |C(T_i) - C(T_{\text{comp}})| = 0$ (optimal), so that T_{comp} is an AMT for $\mathcal{T}$.

We now show that $Res(T_{\text{med}}) \leqslant Res(T_{a.\text{med}})$ for an arbitrary median tree T_{med} and an arbitrary AMT $T_{a.\text{med}}$.

Let $w(c)$ denote the weight of character c. That is, $w(c) = |\{i : c \in C(T_i)\}|$. For $C \subseteq \mathscr{C} = \cup_i C(T_i)$, we set $W(C) = \sum_{c \in C} w(c)$. By [4], we know that $C_1(\mathcal{T}) \subseteq C(T_{\text{med}}) \subseteq C_0(\mathcal{T})$, where $C_1 = \{c \in \mathscr{C} : w(c) > k/2\}$ and $C_0 = \{c \in \mathscr{C} : w(c) \geqslant k/2\}$. We let $C(T_{\text{med}}) \cap C(T_{a.\text{med}}) = X$, $C(T_{\text{med}}) - C(T_{a.\text{med}}) = Y$, and $C(T_{a.\text{med}}) - C(T_{\text{med}}) = Z$. Therefore

- $Res(T_{\text{med}}) = |X| + |Y|$ and $val(T_{\text{med}}) = W(X) + W(Y)$,
- $Res(T_{a.\text{med}}) = |X| + |Z|$ and $val(T_{a.\text{med}}) = W(X) + W(Z)$,
- $W(Y) \geqslant (k/2)|Y|$, since $Y \subseteq C_0(\mathcal{T})$, and
- $W(Z) \leqslant (k/2)|Z|$, since $Z \cap C_1(\mathcal{T}) = \emptyset$.

Since $T_{a.\text{med}}$ is an AMT it follows that $val(T_{a.\text{med}}) \geqslant val(T_{\text{med}})$, so that $W(Z) \geqslant W(Y)$.

Combining the above we get $(k/2)|Z| \geqslant W(Z) \geqslant W(Y) \geqslant (k/2)|Y|$, so that $|Z| \geqslant |Y|$. It follows that $Res(T_{a.\text{med}}) \geqslant Res(T_{\text{med}})$.

We now consider the Nelson consensus, T_{nels}. Let T be one of the trees in the Nelson basis. Clearly $Res(T_{\text{nels}}) \leqslant Res(T)$ since $C(T_{\text{nels}})$ only contains those characters that are common to all the trees in the Nelson basis. Thus, it will suffice for us to show that $Res(T) \leqslant Res(T_{a.\text{med}})$.

By the definition of the AMT, $w(c(T_{a.\text{med}})) \geqslant w(c(T))$. However, by the definition of the Nelson basis, $w(c(T)) - |C(T)| \geqslant w(c(T_{a.\text{med}})) - |C(T_{a.\text{med}})|$. Hence $Res(T_{a.\text{med}}) \geqslant Res(T)$. □

7.2. Worst-case profiles for the median and Nelson consensus trees

In this section we consider cases where the median and Nelson trees are far less resolved than the AMT. Thus, these constitute bad cases when one's goal is resolution.

Let T be an evolutionary (fully resolved) tree with $C(T) = C_S \cup \{e_1, \ldots, e_j\}$. Let T_i be the evolutionary tree defined by $C(T_i) = C_S \cup \{e_i\}$. For the profile $\{T_1, \ldots, T_j\}$, the median, strict, and nelson consensus trees are identically equal to the star, but the (fully resolved) compatibility tree nevertheless exists and hence equals the asymmetric median tree.

Another bad case for the median tree is as follows. From the same tree T, consider a profile of trees $P = \{T_1, T_2, \ldots, T_k\}$ constructed so that each character e_i appears in $k/2 - 1$ trees, each tree T_i is a fully-resolved evolutionary tree, and no character $c \in \cup_i C(T_i)$ has $w(c) \geqslant k/2$. P is a profile of compatible trees. Again the median tree T_m of P has $C(T_m) = C_S$ (i.e. the star-graph) equal to the strict consensus tree, while the AMT is the fully resolved compatibility tree. The median tree includes characters only if they appear in a majority of the trees of a profile, which drops a large amount of potentially useful information. In the introduction, we commented upon a general phenomenon where low resolution of some input trees forces the median tree to also have low resolution. This is because when some of the trees in the profile have few edges (characters), it is harder to have characters included in a majority of the trees. This example indicates that there can be great discrepancy in resolution between the AMT and the median tree even when the input trees are fully resolved.

8. Special cases of the AMT problem

Given that the general AMT problem is hard, we ask what special cases are tractable. We consider two cases: where we require that the output tree have bounded degree, and where we require that the input trees have bounded degree.

8.1. Constraining the degree of the consensus tree

In this section we show that if we add the further restriction that the AMT have bounded degree d, then the AMT problem for k trees on n species can be solved in time $O(ndk^d)$. In time $O(nD^2k^D)$ we can determine the minimum D such that an AMT of degree D exists and construct such an AMT.

More precisely, we consider the following problem: The *Degree Bounded Asymmetric Median Tree Problem* (DBAMT):

Input: Profile P of S-labelled trees, $k, d \in Z^*$.

Question: Does there exist an S-labelled tree T with maximum degree d such that $f_{a.\mathrm{med}}(T,P) \leqslant k$?

This problem can be solved for fixed degree bound d by direct application of an algorithm by David Bryant to solve a related character compatibility problem [6]. Bryant gives an $O(ndk^d)$-time algorithm to solve the following problem: given a weighted set C of k binary characters defined on leaf set S, with $|S| = n$ and a degree bound d, find a maximum-weight tree T (if one exists) of maximum degree d such that $C(T) \subseteq C$. The weight of the tree T is the sum of the weights of the characters in the tree. He uses this algorithm in an $O(nD^2k^D)$-time algorithm to find the minimum degree D such that a degree-D tree exists and to construct the minimum-weight tree with that degree. The degree-bounded AMT problem can be converted to an instance of Bryant's problem by letting $C = \cup_i C(T_i)$, where $P = \{T_1, \ldots, T_k\}$ and letting the weight of the characters be $w(c)$ as defined in Section 2.4. Bryant has also observed independently that this algorithm can be used to infer consensus trees from profiles, though he did not observe the relationship between the consensus obtained and the median tree.

8.2. *Profiles of fully resolved trees*

In this section we consider the problem when all the input trees are fully resolved evolutionary trees. This is a possible outcome in practice, since some methods of tree construction automatically return binary (i.e. fully resolved) trees (for example, the nearest-neighbor joining method of Saitou and Nei [32]). Although the exact and approximation algorithms we have presented earlier can be applied to profiles of fully resolved trees, the NP-hardness results may not be applicable. Thus, what we wish to determine is whether inferring the AMT of fully resolved trees is potentially easier than inferring the AMT for general profiles. Unfortunately, the answer is not helpful.

We state the *Asymmetric Median of Binary Trees Problem* (AMBT):

Input: Profile P of binary S-labelled trees, $k \in Z^*$.

Question: Does there exist an S-labelled tree T such that $f_{a.\mathrm{med}}(T,P) \leqslant k$?

There is a variation of the median tree problem called the *Binary Median Tree Problem* (BMT), in which both the input and the output are constrained to be binary trees; this was shown NP-*complete* in [26]:

Input: Profile P of binary S-labelled trees, $k \in Z^+$.

Question: Does there exist a binary S-labelled tree T such that $f_{\mathrm{med}}(T,P) \leqslant k$?

Theorem 9. *The Asymmetric Median of Binary Trees (AMBT) Problem is* NP-Complete.

Proof. Clearly, AMBT is in NP. We now show it is NP-*hard*. By [26], the Binary Median Tree problem is NP-*hard*. Let (P,k) be an instance of the binary median tree problem. We will now show that (P,k) is a yes-instance to the BMT problem if and only if $(P, \lfloor k/2 \rfloor)$ is a yes-instance to the AMBT problem. Suppose (P,k) is a

yes-instance to the BMT problem. Hence every tree in P is binary, and there exists T, a binary tree, with $f_{\text{med}}(T,P) \leqslant k$. Let T' be tree T after removing any characters that are not in $\cup_i C(T_i)$. In general such characters may be necessary to produce a fully refined tree. Thus, $C(T') = \cup_i C(T_i) \cap C(T)$.

Since T is a binary tree and every tree T_i in P is binary, they all have the same number of edges (characters). Thus, for all T_i, $|C(T) - C(T_i)| = |C(T_i) - C(T)| = |C(T)| - |C(T) \cap C(T_i)|$, and $|C(T) \triangle C(T_i)| = 2|C(T_i) - C(T)|$. Since tree T' differs from T only in characters that appear in no tree, we have $2|C(T_i) - C(T)| = 2|C(T_i) - C(T')|$. Summing over the individual contributions to the median-tree cost and asymmetric-median-tree cost, we have $f_{\text{med}}(T,P) = 2f_{a.\text{med}}(T',P)$. Hence $f_{a.\text{med}}(T',P) \leqslant \lfloor k/2 \rfloor$ and $(P, \lfloor k/2 \rfloor)$ is a yes-instance to the AMBT problem.

Conversely, suppose $(P, \lfloor k/2 \rfloor)$ is a yes-instance to the AMBT problem. Hence, there exists a T' (not necessarily binary) such that $f_{a.\text{med}}(T',P) \leqslant \lfloor k/2 \rfloor$. Let T be any binary resolution of T'. By the same argument as above, we have $f_{\text{med}}(T,P) = 2f_{a.\text{med}}(T',P)$. Hence $f_{\text{med}}(T',P) \leqslant k$, and so (P,k) is a yes-instance to the BMT problem. Hence, the AMBT problem is also NP-*hard*. □

9. Summary

We have proposed an optimization criterion for evalating consensus trees, and define a new consensus tree, which we call the asymmetric median tree, which optimizes this criterion. We show that the asymmetric median tree always exists (though it may not be unique), and have presented polynomial-time algorithms for inferring the asymmetric median tree of two trees, or approximating the asymmetric median tree of three or more trees. We also consider the case where the output is restricted to be of bounded degree d, and present a polynomial-time algorithm for finding the best such tree. We have defined a measure of evolutionary information content, which we call the degree of resolution, and we have compared other popular models for consensus tree construction to the asymmetric median tree. We showed that for any profile of trees, any asymmetric median tree is always at least as informative as the median tree, Nelson Tree, and strict consensus tree. We also showed that the approximation to the asymmetric median tree is at least as informative as the median tree or the strict consensus tree. Furthermore, both the approximation and the exact algorithms will return the compatibility tree, when it exists.

We show that computing the asymmetric median tree of three or more trees is NP-hard, but in practice, it may not be difficult to solve. A closely related problem to the constructing the asymmetric median tree is the *binary character compatibility criterion* problem. In this case, the input is a set of binary characters and the objective is the largest set of compatible characters. Joe Felsenstein [15] has reported that branch-and-bound algorithms solve this problem efficiently on real and on simulated data; it is possible that branch-and-bound may also be an efficient means in practice for solving this problem.

We have shown AMTs can be more useful than median trees when the input does not exclusively consist of fully resolved trees or even if it does but the trees do not display majority consensus. As biologists continue to combine information of different types, the importance of consensus tree methods such as these which can handle small trees without losing information will become increasingly apparent.

We leave open certain problems, perhaps the most relevant of which are the following:

- For what value of k is the problem of finding the AMT of k fully resolved evolutionary trees NP-*hard*?
- How hard is it to approximate the AMT of k fully resolved evolutionary trees?
- Is there a simple characterization of the incompatibility graph of a profile of fully resolved evolutionary trees?
- Can the value of the AMT of k S-labelled trees be approximated better than we have achieved in this paper?

Acknowledgements

We thank the two anonymous referees for helpful comments.

References

[1] E. Adams III, Consensus techniques and the comparison of taxonomic trees, Syst. Zoology 21 (1972) 390–397.

[2] E. Adams III, N-trees as nestings : complexity, similarity, and consensus, J. Classification 3 (1986) 299–317.

[3] J. Barthélemy and F. Janowitz, A formal theory of consensus, SIAM J. Discrete Math. 3 (1991) 305–322.

[4] J. Barthélemy and F. McMorris, The median procedure for n-Trees, J. Classification 3: (1986) 329–334.

[5] M. Bellare and M. Sudan, Improved non-approximability results, Proceedings of the 26th Annual ACM Symposium on Theory of Computing, Montreal, (ACM, New York) 184–193.

[6] D. Bryant, Hunting for binary trees in binary character sets: efficient algorithms for extraction, enumeration, and optimization, Research Report no. 124, Department of Mathematics and Statistics, Canterbury University, Christchurch, New Zealand, (April 1995).

[7] W.H.E. Day, Optimal algorithms for comparing trees with labelled leaves, J. Classification 2 (1985) 7–28.

[8] W.H.E. Day and D. Sankoff, Computational complexity of inferring phylogenies by compatibility, Syst. Zoology 35 (1986) 224–229.

[9] G.F. Estabrook, C.S. Johnson, Jr. and F.R. McMorris, An idealized concept of the true cladistic character, Math. Biosci. 23 (1975) 263–272.

[10] G.F. Estabrook, C.S. Johnson, Jr. and F.R. McMorris, A mathematical foundation for the analysis of cladistic character compatibility, Math. Biosci. 29 (1976) 181–187.

[11] G.F. Estabrook, C.S. Johnson, Jr. and F.R. McMorris, An algebraic analysis of cladistic characters, Discrete Math. 16 (1976) 141–147.

[12] G.F. Estabrook and F.R. McMorris, When is one estimate of evolutionary relationships a refinement of another? J. Math. Biosci. 10 (1980) 327–373.

[13] M. Farach, T. Przytycka and M. Thorup, On the agreement of many trees, Inform. Process. Lett., to appear.

[14] M. Farach and M. Thorup, Optimal evolutionary tree comparisons by sparse dynamic programming, Proceedings of the 35th annual IEEE Foundations of Computer Science (1994) 770–779; SIAM J. on Computing, to appear.
[15] J. Felsenstein, personal communication.
[16] J. Felsenstein, Cases in which parsimony or compatibility methods will be positively misleading, Syst. Zoology 27 (1978) 401–410.
[17] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of *NP*-Completeness (W.H. Freeman and Company, New York, 1979).
[18] D. Gusfield, Efficient algorithms for inferring evolutionary trees, Networks 21 (1991) 19–28.
[19] M. Hendy, The relationship between simple evolutionary tree models and observable sequence data, Syst. Zoology 38 (1989) 310–321.
[20] M. Hendy and D. Penny, A framework for the quantative study of evolutionary trees, Syst. Zoology 38 (1989) 297–309.
[21] J. Hopcroft and R.M. Karp, An $O(n^{2.5})$ algorithm for maximum matching in bipartite graphs, SIAM J. Comput. (1975) 225–231.
[22] D.S. Johnson, A catalog of complexity classes, in: Algorithms and Complexity, Handbook of Theoretical Computer Science, Vol. A (Elsevier, Amsterdam, 1990) 67–161.
[23] S. Kannan, S. Yooseph and T. Warnow, Computing the local consensus of trees, Proceedings of the 1995 ACM/SIAM Symposium on Discrete Algorithms (SODA) (1995).
[24] M. Kao, Tree contractions and evolutionary trees, submitted.
[25] D. Keselman and A. Amir, Maximum agreement subtree in a set of evolutionary trees – metrics and efficient algorithms, Proceedings of the 35th Annual IEEE Foundations of Computer Science (1994) 758–769.
[26] F.R. McMorris and M. Steel, The complexity of the median procedure for binary trees, Proceedings of the 4th Conference of the International Federation of Classification Societies, Paris, 1993, Studies in Classification, Data Analysis, and Knowledge Organization (Springer, Berlin, 1993), to be published.
[27] G. Nelson, Cladistic analysis and synthesis: Principles and definitions, with a historical note on Adanson's Familles des Plantes (1763–1764), Syst. Zoology 28 (1979) 1–21.
[28] G. Nelson and N.I. Platnick, Systematics and Biogeography: Cladistics and Vicariance (Columbia Univ. Press, New York, 1981).
[29] R.D.M. Page, Tracks and trees in the antipodes: a reply to humphries and seberg, Syst. Zoology 39 (1990) 288–299.
[30] R.D.M. Page, Genes, Organisms and areas: The problem of multiple lineages, Syst. Bio. 42 (1993) 77–84.
[31] R.D.M. Page, Reconciled trees and Cladistic analysis of historical associations between genes, organisms, and areas, manuscript (1993).
[32] N. Saitou and M. Nei, The neighbor-joining method: a new method for reconstructing evolutionary trees, Mol. Biol. Evol. 4 (1987) 406–425.
[33] M.A. Steel and T.J. Warnow, Kaikoura tree theorems: computing the maximum agreement subtree, Inform. Process. Lett. 48 (1993) 72–82.
[34] H.T. Wareham, An efficient algorithm for computing MI consensus trees, Honors Dissertation, Department of Computer Science, Memorial University of Newfoundland, St. John's, Newfoundland (1985).
[35] T.J. Warnow, Tree compatibility and inferring evolutionary history, J. Algorithms 16 (1991) 388–407.