Pattern Identification in Biogeography

Ganeshkumar Ganapathy, Barbara Goodson, Robert Jansen, Hai-son Le, Vijaya Ramachandran, and Tandy Warnow

Abstract—Identifying common patterns among area cladograms that arise in historical biogeography is an important tool for biogeographical inference. We develop the first rigorous formalization of these pattern-identification problems. We develop metrics to compare area cladograms. We define the *maximum agreement area cladogram (MAAC)* and we develop efficient algorithms for finding the MAAC of two area cladograms, while showing that it is NP-hard to find the MAAC of several binary area cladograms. We also describe a linear-time algorithm to identify if two area cladograms are identical.

Index Terms—Biogeography, area cladograms, distance metrics, maximum agreement area cladogram, maximum agreement subset.

1 Introduction

Biogeography is the study of the geographic distribution of organisms [4], [6]. Biogeographers seek to understand ecological processes (e.g., climatic stability and effect of area) that influence the distribution of living organisms over short periods of time and to uncover events occurring in the distant past (e.g., continental drift, glaciation, and evolution) which have resulted in the geographic distribution observed today. Historical biogeography is the study of the geographic distribution of organisms in the light of their evolutionary history. One of the main tools of historical biogeographic inference is the comparison of phylogenetic trees of different groups of organisms that share their geographic distributions, in order to detect common patterns. However, until very recently, comparisons have largely been made visually. In this paper, we formalize the comparison and pattern identification problems, we develop efficient algorithms to detect common patterns, we prove an NP-hardness result, and we develop distance metrics so as to compare two patterns. Such pattern-identification problems arise in the context of *indirect* historical biogeographic inference. In the following section, we provide a brief introduction to historical biogeography and to direct and indirect historical biogeographic inference so as to place our work in context.

Historical Biogeography. One of the ways of understanding the geographic distribution of species is by studying the evolutionary history of the species, and this forms the basis for the discipline of historical biogeography [3], [6], [9], [19].

- *G. Ganapathy, H-s. Le, V. Ramachandran, and T. Warnow are with the Department of Computer Sciences, The University of Texas at Austin, Taylor Hall 2.124, 1 University Station C0500, Austin, TX 78712-0233. E-mail: {gsgk, haison, vlr, tandy}@cs.utexas.edu.*
- *B. Goodson and R. Jansen are with the Section of Integrative Biology, The University of Texas at Austin, Biological Laboratories 311, Austin, TX 78712. E-mail: {bgoodson, jansen}@mail.utexas.edu.*

Manuscript received 16 Feb. 2006; accepted 21 Apr. 2006; published online 31 Oct. 2006.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBBSI-0020-0206.

The evolutionary relationships are typically represented as branching tree structures called *phylogenetic trees* or, simply, phylogenies. Historical biogeographers generally assume that the current geographic distribution of organisms is a result of the following past events: 1) an event that splits an area into two or more distinct parts, known as geographic vicariance, 2) extinction of species in an area, and 3) dispersal of organisms from one area to another. Historical biogeographical inference, then, aims to reconstruct these past events, and usually takes one of the following two forms:

- **Direct Inference.** In direct inference methods, a branching history of areas, called an *area cladogram*, is inferred from the phylogeny of organisms living in the areas. Brooks parsimony analysis (BPA) [3], Assumptions 0, 1, and 2 [31], and Page's reconciliation maps [24] are examples of this approach.
- **Indirect Inference.** Here, phylogenies of different groups of organisms which share their geographic distributions are compared. Common patterns observed in the different phylogenies are taken to be evidence of common past geological or climatic events that influenced the geographic distribution of species [19], [21].

The contributions of our paper are toward indirect historical biogeographic inference. We formalize the notion of comparison of phylogenies of codistributed groups of organisms and develop algorithms and metrics in order to compare such phylogenies. However, in order to place our work in context, we will review previous inference methods as well.

1.1 Direct Inference

We now look at the direct inference methods in more detail. Brooks parsimony analysis and Assumptions 0, 1, and 2 take as input a phylogeny of the organisms whose geographic distribution is to be understood and the geographic distribution of the organisms. Fig. 1 depicts two hypothetical phylogenies and geographic distributions. The output of these methods is a *branching history of the areas*

 Published by the IEEE CS, CI, and EMB Societies & the ACM

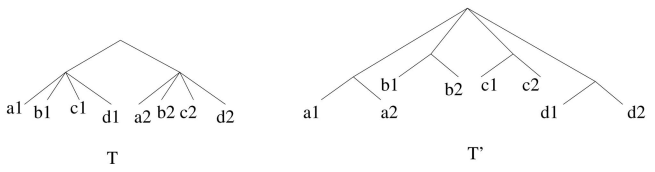

Fig. 1. Two hypothetical phylogenies on eight taxa on four islands (a, b, c, d) with two ecological zones each (1 and 2).

that support the organisms. As a first step, these methods construct a *general area cladogram* by replacing the taxon label of the leaf with the label of the area in which the taxon is found; see Fig. 2. Note that some taxa may occur in more than one area (called *widespread taxa*) and there may be many taxa endemic to one area (called *redundant taxa*). This in turn translates to many leaves with the same area label or leaves with more than one area label in the general area cladogram. Hence, the area cladograms constructed as above do not, and cannot, represent a history of the areas. Consequently, the direct inference methods further process the general area cladograms to produce a branching history of areas where each leaf is labeled with a unique area, called *resolved area cladograms*.

Brooks Parsimony Analysis produces a resolved area cladogram as follows: Each node, including the leaves, in the general area cladogram is given a number. In a general area cladogram with n leaves, there will be a total of $2n - 1$ nodes. Each area is then represented as a binary string of length $2n - 1$. The ith bit of the string for an area p is 1 if node i is an ancestor of any occurrence of the area p in the general cladogram. This process is illustrated in Fig. 3. The set of binary strings representing areas is then subjected to a *maximum parsimony analysis* to produce a resolved area cladogram. Thus, Brooks parsimony analysis reconstructs a purely vicariance-based history of the areas.

Page's Reconciliation Maps combine the branching histories of two associated entities into one summary of historical association between the two entities. The associated entities can be hosts and parasites, organisms and genes, or, as in our case, areas and organisms that live in them. The need for reconciliation arises when the branching histories of the associated entities are incongruent. The simplest hypothesis about the coevolution of two associated entities (such as areas and organisms) is that a vicariance event in one entity corresponds to a speciation event in the other entity; incongruence arises when this hypothesis is violated. Reconciliation maps explain incongruence in terms of vicariance-independent speciation of organisms and extinct or uncollected species lineages; see Fig. 4 (from

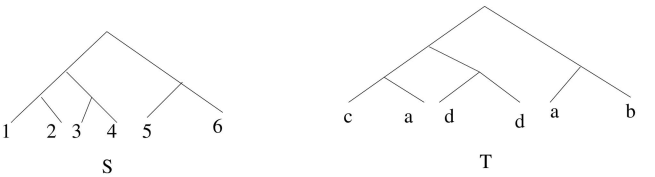

Fig. 2. A phylogeny S and its associated area cladogram T, assuming taxon 1 appears in area c, 2 appears in area a, 3 appears in area d, 4 appears in area d, 5 appears in area a, and 6 appears in area b.

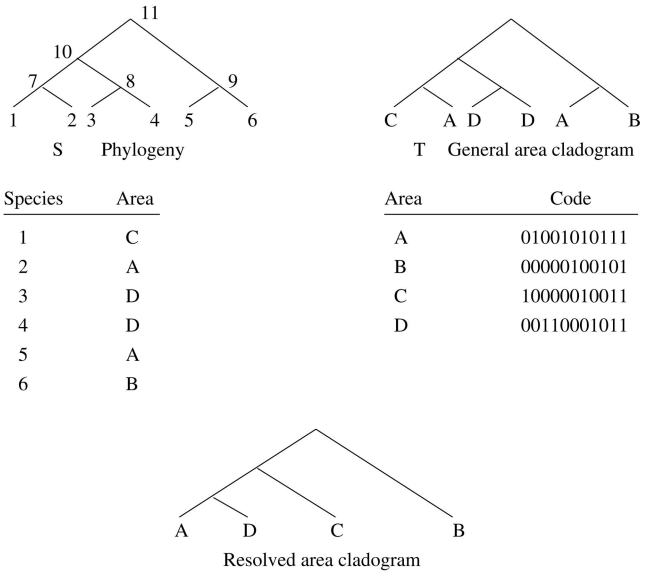

Fig. 3. Brooks parsimony analysis: A general area cladogram T is derived from the phylogeny S and the geographic distribution. The areas are then coded as binary strings and subjected to a maximum parsimony analysis. The resolved area cladogram is a most parsimonious tree for the set of strings that encode the four areas.

[24]) for an illustration of this. Reconciliation maps thus invoke vicariance and extinction in order to understand the geographic distribution of species.

Assumptions 0, 1, **and** 2 are again methods that produce resolved area cladograms from general area cladograms. In $A0$, vicariance is the only a priori hypothesis used to explain the geographic distribution of species. In $A1$, vicariance and extinction are the a priori hypotheses and, in $A2$, vicariance, extinction, and dispersal are the a priori hypotheses. However, how exactly these assumptions must be applied is contentious [30].

1.2 Indirect Inference

The fundamental idea behind indirect inference is that *a consistent pattern* observed in the phylogenies of species from different genera in the same geographic area will imply stronger evidence for the particular hypotheses suggested by the pattern. As an example of this approach, consider a group of islands, each containing multiple ecological zones (for example, each island can contain coastal and mountain ecological zones). Suppose our goal is to understand the observed geographic distribution of species on the islands. One hypothesis about the distribution, called *interisland colonization*, is that species dispersed from each ecological zone in each island to similar zones in other islands and then differentiated. Another hypothesis, called *adaptive radiation* is that dispersal *between* islands happened first, followed by dispersal to the different ecological zones and differentiation into many species [20]. The crucial idea is that we might be able to infer which of the above two hypotheses is responsible for the observed distribution: Interisland colonization is suggested by taxa on different islands but the same ecological zone forming a monophyletic group (i.e., a "clade" or rooted subtree), and adaptive radiation is suggested if species on the same island in different ecological zones form a

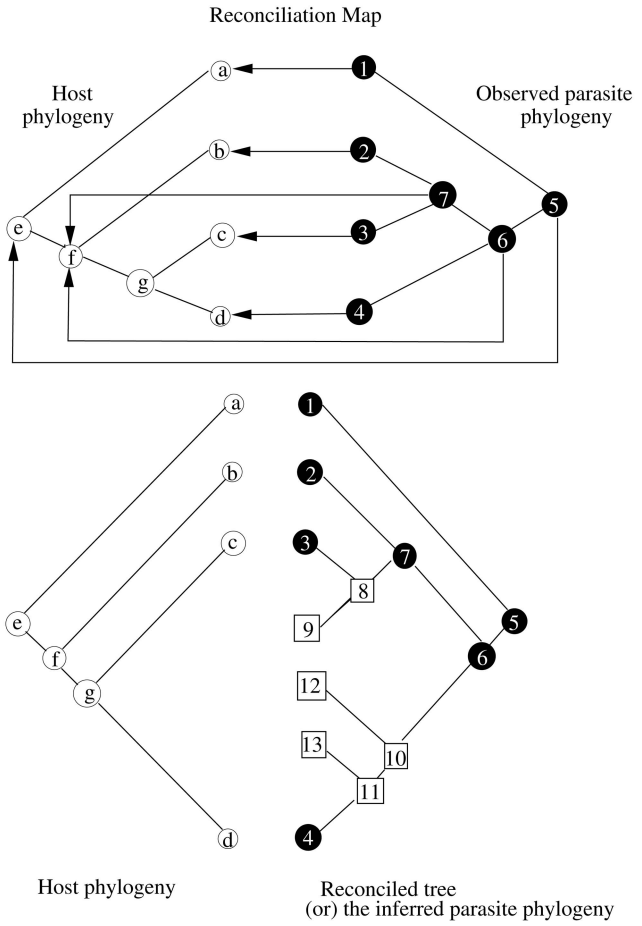

Fig. 4. Reconciliation Maps: A host phylogeny, H, and a parasite phylogeny, P are incongruent. The parasites 1, 2, 3, and 4 live in hosts a, b, c, and d, respectively. The nodes 6 and 7 of the parasite phylogeny are both mapped to node f of the host phylogeny. This represents a speciation event of the parasite inside the host species f without an accompanying host speciation. The ancestors of the two resulting parasitic lineages are labeled 7 and 10 in the reconciled tree; both the parasite lineages then follow the host lineage f's speciation pattern, leading to a parasite phylogeny shown as the reconciled tree. The dissimilarity between this inferred parasite phylogeny and the observed parasite phylogeny on top is explained by postulating the extinct or unsampled parasite lineages 9, 12, and 13.

monophyletic group. For example, in Fig. 1, T suggests dispersal, while T' suggests adaptive radiation.

In practice, common patterns are identified between general area cladograms derived from the phylogenies of the different groups of codistributed species. Until very recently, such common patterns have been identified by visual observation [19], [18], [7], Recently, Lapointe and Rissler in [21] identified common patterns among area cladograms by applying the *maximum agreement subtree (MAST)* method originally developed for phylogenies [15]. A maximum agreement subtree between two rooted trees is obtained by deleting a minimum number of leaves from either tree so that, on the remaining set of leaves, the trees are identical (i.e., isomorphic). However, before the the application of the MAST algorithm, the authors of [21] obtain area cladograms using a distant-based approach, as follows: 1) First, pairwise distances between areas are computed, where the distance between two areas $A1$ and $A2$ is the average distance between a species in $A1$ and a species in $A2$. The distance between species is the distance between sequence that represents the species. 2) Then, a *neighbor-joining* [27] tree for the areas is computed based on the calculated distances between the areas. The problem with this approach is the calculated distance between the areas does not capture all the evolutionary history of the species in the areas. Further, neighbor-joining is not the best method for obtaining phylogenies; most realistic phylogenies are computed using either maximum parsimony or maximum likelihood [29], [14].

In this paper, we show that the Steel-Warnow algorithm for MAST from [28] can in fact be applied without modification to the problem of identifying the largest common patterns among area cladograms, called the Maximum Agreement Area Cladogram (MAAC) problem. However, in general, care must be exercised before adapting any MAST algorithm for the MAAC problem.

Comparing Area Cladograms. Apart from identifying common patterns among area cladograms, it is of interest to quantify the difference between an observed area cladogram and a hypothesized area cladogram. Earlier work on comparing area cladograms has included pruning the cladograms until the two cladograms agree on the remaining leaves (see [26]) and using similarity metrics such as the *bipartition* metric (also called the *component* metric or the *character encoding* metric in the literature [23]) and the *triplets* metric between rooted area cladograms [23]. However, all these methods only apply to resolved area cladograms. In this paper, we develop distance metrics to compare general area cladograms.

1.3 Our Contributions

Our contributions are two-fold: We develop both metrics and algorithmic results for comparing area cladograms. More specifically,

- We show that the equivalence between the edge-contract-and-refine metric ("RF-distance") and the bipartition metric ("character-encoding" metric) that holds for phylogenies *does not hold* for area cladograms. More specifically, we show that the bipartition metric, when extended to area cladograms, is not a metric. For the edge-contract-and-refine edit distance between two area cladograms we present a simple, but worst-case exponential-time algorithm. This edit distance can compare only area cladograms that are on the same number of leaves and when each area labels the same number of leaves in both area cladograms (Section 3).
- We define another metric, the *MAAC* distance metric, for comparing two rooted area cladograms, which is based on the size of the largest common pruned subtree between the two area cladograms. The MAAC distance metric can compare two arbitrary trees that are not necessarily on the same number of leaves, which is particularly useful when comparing area cladograms (Section 3).
- We present two polynomial-time algorithms for computing a MAAC of two rooted area cladograms. The first algorithm is a standard dynamic

programming algorithm that runs in $O(n^{2.5}\log n)$ time, where n is the maximum number of leaves in either cladogram. We then present a "sparse" version of this basic dynamic program that achieves a running time of $O(n^2)$ when the number of leaves with any given label is not too large. We also describe a linear-time algorithm to decide if two area cladograms are identical (Section 4).

- We study the problem of computing the MAAC of k area cladograms, and we show that computing MAAC for k area cladograms is NP-hard even if all trees are binary (Section 5).

2 Phylogenies: Distance Metrics and Agreement Subsets

In this section, we define some basic concepts: the formal notion of a phylogenetic tree, distance metrics between phylogenetic trees, and the maximum agreement subset problem for phylogenetic trees.

Character Encoding of Phylogenies. Tests for equality between phylogenies are based on the notion of the *character encoding* of phylogenies. Another notion crucial to the study of phylogenies is that of a *bipartition*: Removing an edge e from a leaf-labeled tree T induces a bipartition π_e on its set of leaves.

Definition 1 (Character Encoding of a Phylogeny). *The* character encoding *of a phylogeny T is the set $C(T) = \{\pi_e : e \in E(T)\}$, which represents the set of bipartitions induced by the edges of T.*

Theorem 1 (Character-Encoding Metric [5]). *Let T and T' be two phylogenies on the same set of taxa. Then, $|C(T)\triangle C(T')| = |(C(T) - C(T')) \cup (C(T') - C(T))|$ defines a distance metric.*

By Theorem 1, two phylogenies, T and T', are isomorphic (with the isomorphism preserving the leaf labels) if and only if $|C(T)\triangle C(T')| = 0$.

A *contraction* operation applied on an edge in a tree collapses that edge and identifies its two end points; a *refinement* operation reverses a contraction and, when applied at an unresolved node (i.e., an internal node with degree greater than three), expands that unresolved node into two nodes connected by an edge.

Definition 2 (Robinson-Foulds (RF) Distance). *The* Robinson-Foulds distance *between two phylogenies T_1 and T_2 is defined as the number of edge-contractions and refinements necessary to transform T_1 into T_2 (or vice versa) and is denoted $RF(T_1, T_2)$. Thus, it is also the "edge-contract-and-refine" distance.*

The RF distance naturally defines a metric since it is an edit distance.

Theorem 2 [25]. *Let T_1 and T_2 be two phylogenies, each on the same set of taxa. Then, $RF(T_1, T_2) = |C(T_1)\triangle C(T_2)|$.*

Finally, we define the maximum agreement subtree problem for phylogenies. The analogue of this problem for area cladograms is crucial to addressing the problems outlined in Section 1. We begin by defining what we mean by a *restriction* of an unrooted tree T to a subset L' of its leaf set L: We delete from T all the leaves in $L - L'$, and we then suppress all nodes of degree two by contracting an edge incident with each such node. If T is rooted, the second step is equivalent to suppressing all internal nodes with only one child. The resulting tree is given by the notation $T|L'$.

Definition 3 (Maximum Agreement Subset (MAST)). *Let $\{T_1, T_2, \ldots, T_k\}$ be a set of phylogenetic trees, each on a set L of leaves. A* maximum agreement subset *(MAST) of trees T_1 through T_k is a subset $L' \subseteq L$ of maximum cardinality such that the restrictions of the trees $T_1, \ldots, T_k$ to the set L' are all isomorphic, with the isomorphisms preserving leaf labels.*

The maximum agreement subset problem was introduced in [15] and has been studied thoroughly since then. The rooted and unrooted versions of MAST are polynomially related since the unrooted MAST problem can be solved by solving a polynomial number of rooted MAST problems. Computing a MAST is NP-hard for three or more trees [2]. An $O(n^{2+o(1)})$ time algorithm for the case of two trees on n leaves is given in [13]. For two rooted binary trees, the best known algorithm takes $O(n\log^3 n)$ time [11], [10]; for two rooted trees which may not be binary, the best known algorithm takes $O(n^{1.5}c^{\sqrt{\log n}})$ time, where c is a constant [13]. For computing a MAST of k rooted trees, an $O(kn^3 + n^d)$ algorithm (with d the maximum degree of a node in any tree) was presented in [10].

3 Distance Measures between Area Cladograms

In this section, we develop distance metrics for the set of area cladograms. We first show that the character encoding distance between two different area cladograms can be zero and, hence, the character-encoding "distance" is not a metric on area cladograms and, in particular, cannot be used as a test of isomorphism. While the character-encoding metric for phylogenies does not extend to area cladograms, the contract-and-refine edit distance still defines a metric since it is an edit distance. We present an algorithm to compute the edge contract-and-refine edit distance between area cladograms. This algorithm is efficient if there are only a few occurrences of widespread taxa, but it is exponential-time in general. For phylogenies, this edit distance (which is called the *Robinson-Foulds* distance) can be computed efficiently since it equals the character-encoding distance.

In the Section 3.3, we define the notion of a *Maximum Agreement Area Cladogram (MAAC)* of a collection of area cladograms, which is roughly a largest pruned subtree of all trees in the collection (see Fig. 6). We propose the MAAC distance metric for comparing area cladograms, and we argue that this is a more appropriate metric for area cladograms than the contract-and-refine edit distance. In the rest of the paper, we present algorithms and an NP-hardness result for computing MAAC.

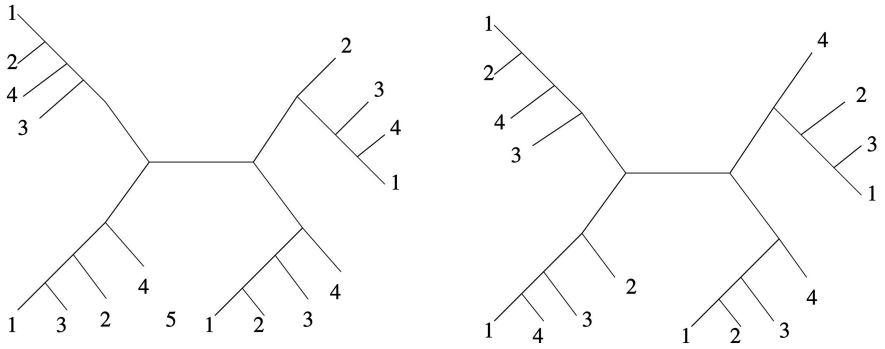

Fig. 5. Two different binary area cladograms that induce the same multiset of partitions.

3.1 The Character Encoding Cannot Distinguish between Area Cladograms

We first formally define an area cladogram:

Definition 4. *An area cladogram is a rooted or unrooted tree whose leaves are labeled with areas. Thus, an area cladogram T is a triplet (t, A, M), where t is its unlabeled topology, A is the set of labels, and M is an onto map from the set of leaves of t to A.*

The map M can map many leaves to the same label and may also map single leaves to many labels. It will not always be necessary in this paper to explicitly refer to the triplet (t, A, M) of an area cladogram T. The triplet will be left out of the notation where unnecessary.

We now define the *extended character encoding* of an area cladogram.

Definition 5. *Let T be an area cladogram. The multiset $\{\pi_e : e \in E(T)\}$ is called the* extended character encoding *of Tw and will be denoted by $C(T)$. Here, π_e denotes the bipartition of the multiset of leaf labels induced by the edge e.*

Contrary to our experience with phylogenetic trees, where the mapping between leaves and labels is one-one, it is possible for two area cladograms T_1 and T_2 to satisfy $C(T_1) = C(T_2)$ and yet not be isomorphic. We exhibit such a pair of trees in Fig. 5.

3.2 The Edge-Contract-and-Refine Distance Metric for Area Cladograms

Though the character-encoding distance fails to extend to area cladograms, the RF distance, being an edit distance, can be extended to unrooted area cladograms to provide a distance metric.

Definition 6 (Robinson-Foulds Distance between Unrooted Area Cladograms). *The* Robinson-Foulds distance *between two unrooted area cladograms T_1 and T_2 is defined to be the number of contractions and refinements necessary to transform T_1 to T_2 (or, equivalently, T_2 to T_1).*

Handling Widespread Taxa. Taxa endemic (resident) to more than one area would result in cladograms with leaves labeled by many areas. Our definition of the Robinson-Foulds distance applies to such cladograms as well: If a leaf is labeled with a set of areas, we can consider that set of areas to be the unique label for that leaf. Thus, throughout the rest of this section, we will assume that each area cladogram leaf has just one area label.

Notation. We let n_1 and n_2 be the number of leaves in trees T_1 and T_2, respectively, and we let $n = max\{n_1, n_2\}$. We let $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ be the set of areas with which the leaves of T_1 and T_2 are labeled, and we let $\pi_{i,j}$, $j = 1, 2$, be the number of leaves in tree T_j which are labeled with a_i; hence, $\sum_{i=1}^{k} \pi_{i,j} = n_j$ for $j = 1, 2$. Our analysis is parameterized on the numbers $\pi_{i,j}$. (This notation will also be used in Section 4.2.)

Note that, if $\pi_{i,1} \neq \pi_{i,2}$ for some i, then there is no sequence of contractions and refinements that can transform T_1 into T_2; in such cases, we define $RF(T_1, T_2) = \infty$. So, throughout the rest of *this* section, we will assume that a given pair of cladograms T_j, T_k will have $\pi_{i,j} = \pi_{i,k}$ for all i and, hence, $n_j = n_k$. We therefore will set n to denote the number of leaves in each of the cladograms and π_i to be the number of leaves labeled with area a_i.

As shown in Section 3.1, the RF distance may not be equal to the extended-character-encoding distance for area cladograms (see Definition 5). However, we can relate the RF distance between two area cladograms to the RF distance between two associated phylogenies, as we now show. We begin with some definitions.

Definition 7 (Full Differentiation of an Area Cladogram). *Let $T = (t, A, M)$ be an unrooted area cladogram, with the unrooted topology t, set of labels A, and the map M from the leaves of t to A. Then, a* full differentiation *of T is a leaf-labeled tree $T^* = (t, A^*, M^*)$ such that M^* is one-one.*

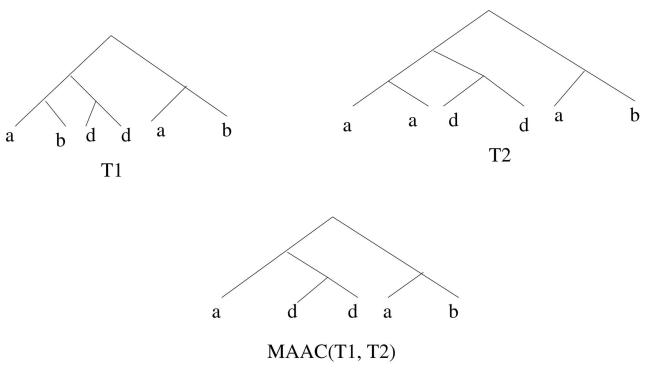

Fig. 6. Two area cladograms, T1 and T2, and their MAAC.

In other words, T^* has the same topology as T, but has its leaves labeled uniquely. Therefore, $A \neq A^*$ is possible.

Definition 8 (Consistent Full Differentiations). *Let $T_1 = (t_1, A, M_1)$ and $T_2 = (t_2, A, M_2)$ be two unrooted area cladograms with the same set A of leaf labels and let $T_1^* = (t_1, A^*, M_1^*)$ and $T_2^* = (t_2, A^*, M_2^*)$ be full differentiations of T_1 and T_2, respectively. T_1^* and T_2^* are* consistent full differentiations *if, for each label $l \in A$, the set of labels assigned to leaves in T_1^* that were labeled l in T_1 is identical to the set of labels assigned to leaves in T_2^* that were labeled l in T_2. Mathematically, this is: $\forall l \in A, \{M_1^*(x) : M_1(x) = l\} = \{M_2^*(x) : M_2(x) = l\}$.*

Theorem 3. *Let T_1 and T_2 be two unrooted area cladograms. Then, $RF(T_1, T_2) = \min\{RF(T_1^*, T_2^*) : T_1^*$ and T_2^* are mutually consistent full differentiations of T_1 and T_2, respectively$\}$.*

Proof. Let S_1 and S_2 be two mutually consistent full differentiations of T_1 and T_2 such that $RF(S_1, S_2)$ is minimum. We will show that $RF(T_1, T_2) = RF(S_1, S_2)$.

We first show that $RF(T_1, T_2) \leq RF(S_1, S_2)$ by induction on the RF distance between S_1 and S_2. We begin with a simple observation: If S_1 and S_2 are isomorphic, then T_1 and T_2 are also isomorphic. To see this, let g_i be the isomorphism from T_i to S_i, for $i = 1, 2$, and let f be the isomorphism between S_1 and S_2. We define f' from T_1 to T_2 implicitly by $g_2(f'(u)) = f(g_1(u))$. This mapping f' is an isomorphism since S_1 and S_2 are consistent.

We now continue with our proof. Suppose $RF(S_1, S_2) = 1$ and assume without loss of generality that S_2 is obtained by contracting an edge (u, v) in S_1 to a single vertex w in S_2. Then, there is a mapping f between the vertices of S_1 and S_2 such that $f(u) = f(v) = w$ and, for any pair of vertices x and y in S_1 such that $\{x, y\} \neq \{u, v\}$, x and y are adjacent if and only if $f(x)$ and $f(y)$ are adjacent. The mapping f also preserves leaf labels. Hence, an analogous mapping f' can be defined between the vertices of T_1 and T_2 that preserves leaf-labels (this is possible because S_1 and S_2 are consistent). Hence, $RF(T_1, T_2) \leq 1$. Suppose that $RF(S_1, S_2) = k + 1$. Then, there is a phylogeny S_3 such that $RF(S_1, S_3) = 1$ and $RF(S_3, S_2) = k$. Assume that it takes a contraction to convert S_1 to S_3 (the claim can be proven in a very similar manner when it takes a refinement). Then, it can be shown that there is an area cladogram T_3 such that S_3 is a full differentiation of T_3 consistent with S_1. Since S_1 and S_3 are consistent and S_1 and S_2 are consistent, S_2 and S_3 are consistent as full differentiations of T_2 and T_3. Hence, we can conclude by induction that $RF(T_1, T_3) \leq 1$ and $RF(T_3, T_2) \leq k$. Hence, we have $RF(T_1, T_2) \leq k + 1$.

It can be shown similarly that there exist consistent full differentiations X_1 and X_2 of T_1 and T_2 such that $RF(X_1, X_2) \leq RF(T_1, T_2)$. It follows that $RF(S_1, S_2) \leq RF(T_1, T_2)$ since we assumed that the S_1 and S_2 minimize the RF distance between two consistent full differentiations of T_1 and T_2. Hence, we have $RF(T_1, T_2) = RF(S_1, S_2)$ and this completes our proof. □

Note that the RF distance between two cladograms T_1 and T_2 is at most the RF distance between *any* consistent full differentiations of T_1 and T_2. Hence, this provides a linear time method for obtaining an upper bound on the RF distance between two area cladograms T_1 and T_2: We first compute two mutually consistent full differentiations and then compute their RF distance. We can compute two mutually consistent full differentiations of two area cladograms in linear time and, since the second step also can be performed in linear time [8], this is a linear time algorithm. Similarly, by Theorem 3, we can compute the RF distance between two area cladograms, T_1 and T_2, by computing the RF distance between all the possible consistent full differentiations of T_1 and T_2 and choosing the minimum.

Theorem 4. *Let T_1 and T_2 be two unrooted area cladograms on n leaves on the same set of areas. For each area a_i appearing at the leaves of T_1 and T_2, let both trees have π_i leaves labeled with area a_i. Then, the RF distance between T_1 and T_2 can be calculated in $\Theta(n\Pi_{i=1}^{k}(\pi_i)!)$ time.*

Proof. The number of different consistent full differentiations of A_1 and A_2 is $\Pi_{i=1}^{k}(\pi_i)!$. Each such differentiation can be obtained in $O(n)$ time. Computing the RF distance between two consistent full differentiations takes $\Theta(n)$ time [8]. □

3.3 The MAAC Distance Metric between Area Cladograms

In this section, we define the problem of computing the largest common pruned subtree of two rooted area cladograms and describe a distance metric based on the size of a largest common pruned subtree. We call a largest common pruned subtree a *Maximum Agreement Area Cladogram (MAAC)*; thus, the MAAC is analogous to the maximum agreement subtree (MAST) of two phylogenies.

Let T be an area cladogram on a set L of leaves. The *restriction* of T to a set of leaves L' is the cladogram obtained by deleting leaves in the set $L - L'$ from T and then suppressing internal nodes of degree two (except the root, if there is one).

Definition 9 (Maximum Agreement Area Cladogram (MAAC) and MAAC distance). *Let $\{T_1, T_2, \ldots, T_k\}$ be a set of rooted area cladograms, with L_i the leaf set of tree T_i, for $i = 1, 2, \ldots, k$. Let $\lambda_1 \subseteq L_1$ through $\lambda_k \subseteq L_k$ be sets of leaves of maximum cardinality such that the respective restrictions of the trees $T_1, \ldots, T_k$ to the sets $\lambda_1 \ldots \lambda_k$ are all isomorphic, with the isomorphisms preserving leaf labels. A restriction of any tree T_i to such a subset of leaves λ_i is a* maximum agreement area cladogram *(MAAC) for the cladograms T_1 through T_k. The size of the MAAC is defined to be the number of leaves in the maximum agreement area cladogram and is denoted by $size_{maac}(T_1, T_2, \ldots, T_k)$.*

The MAAC distance *between two trees T_1 and T_2 is $d_M(T_1, T_2) = max(n_1, n_2) - size_{maac}(T_1, T_2)$, where n_1 and n_2 are the number of leaves in T_1 and T_2, respectively.*

The MAAC distance can be viewed as a generalization of the maximum agreement subtree metric for phylogenies [17], which, for two phylogenies on the same set of n labeled leaves, was defined as $n - size_{mast}$, where $size_{mast}$ is the size of a maximum agreement subset of the two phylogenies.

Handling Widespread Taxa. For comparing cladograms using maximum agreement area cladograms, leaves labeled by more than one area can be treated thus: Each leaf labeled by a group of areas can be split into many separate leaves

(all having the same parent), each of which is labeled by a single unique area from the group of areas.

We now show that the MAAC distance defines a metric on the set of area cladograms.

Theorem 5. *The MAAC distance d_M is a metric on the set of all area cladograms.*

Proof. We begin with a simple observation about the MAAC distance. Let T_1 and T_2 be area cladograms. It is clear that $d_M(T_1, T_2) = 0$ if and only if T_1 and T_2 are isomorphic and that $d_M(T_1, T_2) = d_M(T_1, T_2)$. Hence, all we need to do is to prove that d_M satisfies the triangle inequality.

So, let T_1, T_2, and T_3 be three area cladograms with n_1, n_2, and n_3 leaves, respectively. We have to show that $d_M(T_1, T_2) + d_M(T_2, T_3) \geq d_M(T_1, T_3)$. We begin by defining some notation.

Let M_{ij} be the set of leaves in a MAAC of T_i and T_j and $m_{ij} = |M_{ij}|$. We also let $n_{ij} = max\{n_i, n_j\}$. Let $d_{ij} = d_M(T_i, T_j)$ (i.e., d_{ij} is the MAAC distance between T_i and T_j) so that $d_{ij} = n_{ij} - m_{ij}$. Let $m_{123} = |M_{12} \cap M_{23} \cap M_{13}|$ and let $m'_{ij} = m_{ij} - m_{123}$ so that $m_{ij} = m'_{ij} + m_{123}$.

We have:

$$\begin{aligned} d_{12} + d_{23} &= \max(n_1, n_2) - m_{12} + \max(n_2, n_3) - m_{23} \\ &= \underline{\max(n_1, n_2) + \max(n_2, n_3)} \\ &\quad - (\underline{m'_{12} + m_{123} + m'_{23}} + m_{123}) \\ &\geq \max(n_1, n_2, n_3) + n_2 - (n_2 + m_{123}) \\ &\geq \max(n_1, n_2, n_3) - m_{123} \\ &\geq \max(n_1, n_3) - m_{13} = d_{13}. \end{aligned}$$

□

Note that twice the MAAC distance between two cladograms is an upper bound on the number of insertions and deletions of leaves necessary to transform one of the cladograms to the other.

In Sections 4.1 and 4.2, we present polynomial-time algorithms for computing a maximum agreement area cladogram for two area cladograms. However, in Section 5, we show that finding the MAAC of several area cladograms is NP-hard, even if all area cladograms have bounded degrees.

An important feature of the MAAC definition is that *we do not require that all the trees in the given set contain the same number of leaves or that they be labeled with the same set of areas or even that they be consistent*. Thus, the MAAC distance metric is a more versatile metric for area cladograms than the Robinson-Foulds distance. Further, as we show in the next section, the MAAC of two trees can be computed in polynomial time, in contrast to the result in Theorem 4 for the RF distance.

4 Algorithms for the Maximum Agreement Area Cladogram Problem

In this section, we present several algorithms for the MAAC of two area cladograms. In Section 4.1, we present a basic dynamic programming algorithm which is based on an algorithm for the MAST problem given in [28]. In Section 4.2, we present a refined version of this algorithm that is more efficient when the number of leaves with any given label is not too large. For the problem of determining if two area cladograms are isomorphic, we present a linear-time algorithm in Section 4.3. Finally, to complement these algorithmic results, in the next section we show that the problem of computing the MAAC of k trees is NP-hard, even if all trees are binary.

4.1 Basic Dynamic Programming Algorithm for MAAC

In this section, we describe an algorithm for computing a MAAC of two given rooted area cladograms. This is a dynamic programming algorithm and is an adaptation to MAAC of the first polynomial-time algorithm for the phylogenetic rooted MAST algorithm presented by Steel and Warnow [28]. We will first describe the recursive structure of MAAC solutions which makes the problem amenable to dynamic programming. We will then present the MAAC algorithm in pseudocode and analyze its running time.

The Basic Recursion in MAAC. In our description, we let $MAAC(T, T')$ denote a maximum agreement cladogram of the leaves of T and T'. We describe the algorithm for the case where T and T' are binary; extending this to the case where T and T' are not binary is straightforward.

Let T and T' be two given binary rooted area cladograms. Let v be a node in T and denote by T_v the subtree of T rooted at v. Similarly, denote by T'_w the subtree of T' rooted at a node w in T'. Let v_1 and v_2 be the two children of node v and let w_1 and w_2 be the two children of w. The dynamic programming algorithm for MAAC operates by computing $MAAC(T_v, T'_w)$ for all pairs of nodes (v, w) in $V(T) \times V(T')$ "bottom-up." We now show how to reduce computing $MAAC(T_v, T'_w)$ to computing a small number of smaller MAAC computations, $MAAC(S, S')$, where S and S' are subtrees of T_v and T'_w, respectively, with at least one of them being a proper subtree.

To begin with, $MAAC(T_v, T'_w)$ is easy to compute when either v or w is a leaf. Therefore, in the following discussion, we assume neither v nor w is a leaf.

Let T^* be a MAAC of T_v and T'_w. Then, there exist homeomorphisms mapping T^* to a rooted subtree of T_v and to a rooted subtree of T'_w. In fact, because T and T' may contain more than one leaf with the same label, T^* might be homeomorphically mapped to more than one rooted subtree of T_v and T'_w; however, this cannot happen if there is only one leaf with any given label.

Let p be the (not necessarily proper) *farthest* descendant of v such that the root of T^* is mapped to p. Similarly, let q be the farthest descendant of w in T' such that the root of T^* is mapped to w. Then, $MAAC(T_v, T'_w)$ is, in fact, equal to $MAAC(T_p, T'_q)$.

The vertex p may actually be v or it might be a descendant of v. Similarly, q may be w or some descendant of w. Based on the location of p and q, we have the following cases:

1. *Vertex p is a proper descendent of v.* In this case, T_p is a proper subtree of T_v, and $MAAC(T_v, T'_w)$ equals $MAAC(T_p, T'_w)$. Since p is a proper descendant of v,

$MAAC(T_p, T'_w)$ either equals $MAAC(T_{v_1}, T'_w)$ or $MAAC(T_{v_2}, T'_w)$.

2. *Vertex q is a proper descendent of w.* In this case, $MAAC(T_v, T'_w)$ equals $MAAC(T_v, T'_q)$. Since q is a proper descendant of w, $MAAC(T_v, T'_q)$ either equals $MAAC(T_v, T'_{w_1})$ or $MAAC(T_v, T'_{w_2})$.
3. *Vertex p equals v and vertex q equals w.* Let T_1^* and T_2^* be the subtrees of the root of the MAAC T^*. Then, T_1^* is homeomorphic to a subtree of T_{v_1} (or to a subtree of T_{v_2}; there is no loss of generality in assuming that it is homeomorphic to a subtree of T_{v_1}). Similarly, T_2^* is homeomorphic to a subtree of T_{v_2}. It cannot be homeomorphic to a subtree of T_{v_1} since then T^* would be homeomorphic to a subtree of T_{v_1}, contradicting the assumption that there is no proper descendent p of v such that root of T^* is mapped to p. Arguing similarly, we can conclude that T_1^* and T_2^* are homeomorphic to subtrees of T'_{w_1} and T'_{w_2}, respectively. Now, since T^* is a MAAC, we can conclude that T_1^* is a MAAC of T_{v_1} and T'_{w_1} and that T_2^* is a MAAC of T_{v_2} and T'_{w_2}. So, in this case, we have reduced computing $MAAC(T_v, T'_w)$ to computing $MAAC(T_{v_1}, T'_{w_1})$ and $MAAC(T_{v_2}, T'_{w_2})$ and then taking their union.

The above discussion suggests a straightforward dynamic programming algorithm: We do not know which of the above three cases is true, but we do know that one of them is true. Hence, we solve the subproblems corresponding to all three cases and then choose the largest solution. Note that the algorithm described above is the same as the MAST algorithm from [28], but the reason it is correct for MAAC is somewhat different from the reason it is correct for MAST.

We now describe this MAAC algorithm in pseudocode, but, before we do so, we introduce some notation.

Notation For a node v in T_1 or T_2, let $c(v)$ denote the set of children of v and let $A(v)$ denote the set of all labels of leaves that descend from v. For each pair of nodes $v \in T_1$ and $w \in T_2$, we let $G_{v,w}$ be a weighted complete bipartite graph with bipartition $(c(v), c(w))$, where the weight of the edge $(x, y) \in G_{v,w}$ is the number of leaves in $MAAC(T_x, T_y)$. We denote by $MWBM(G_{v,w})$ the maximum weighted bipartite matching of $G_{v,w}$. We let $V(T)$ be the set of all nodes of the tree T. In the pseudocode, the subroutine DIAG corresponds to the first two cases in our discussion of the MAST dynamic program and the subroutine MATCH corresponds to the third case.

```
MATCH (v, w)
1 Construct G_{v,w}
2 Construct E_0 = MWBM(G_{v,w})
3 Let E_0 = {(v_1, w_1), (v_2, w_2), ..., (v_k, w_k)}
4 Construct tree M with root s such that MAAC(T_{v_i}, T_{w_i}) is
  the ith child of s
5 return M
```

```
DIAG(v, w)
1 t_1 ← largest MAAC(T_v, T_x) such that x ∈ c(w)
2 t_2 ← largest MAAC(T_y, T_w) such that y ∈ c(v)
3 return the larger of t_1 and t_2
```

```
ALGORITHM MAAC (T_1, T_2)
1 Let O be an ordering of V(T_1) × V(T_2)
2   such that if (v_1, w_1) is before (v_2, w_2),
3   then v_1 is not an ancestor of v_2 and w_1 is not an ancestor
    of w_2.
4 for (v, w) in increasing order of O
5   do if v or w is a leaf
6     then MAAC(T_v, T_w) ← a node with label in
      S = A(v) ∩ A(w) if S ≠ ∅; else ∅
7     else MAAC(T_v, T_w) ← larger of MATCH(v, w) and
      DIAG(v, w)
8 return MAAC(T_{r_1}, T_{r_2}); r_1 is the root of T_1 and r_2 is the
  root of T_2.
```

The Running Time of the MAAC Algorithm. The running time of the above algorithm is $O(n^2)$ for binary trees as well as for trees of bounded degree d since there are $O(n^2)$ calls to MATCH and each call runs in $O(d)$ time. If the maximum degree of both trees is unbounded, the $O(n^{2.5} \log n)$ algorithm from [16] can be used to compute the maximum weighted matching (MWBM) in the bipartite graph. Thus, a straightforward bound on the running time is $O(n^{4.5} \log n)$.

A careful analysis of the algorithm reveals that the running time is, in fact, $O(n^{2.5} \log n)$. To obtain this bound, we use the following more precise bound on the running time of the MWBM algorithm in [16]: If the two sets of vertices in the bipartition of the bipartite graph have p and q vertices, then the running time of the algorithm in [16] is $O(p \cdot q \cdot \sqrt{(p+q)} \log(p+q))$. The MAAC algorithm performs the MWBM computation on graphs $G_{u,v}$, for each pair u, v with $u \in V(T_1)$ and $v \in V(T_2)$. Let n_1 and n_2 be the number of leaves in T_1 and T_2, respectively, with $n = max(n_1, n_2)$. Also, let d_u be the degree of node u in either tree. If we let $T(n_1, n_2)$ denote the running time of the MAAC algorithm, we have:

$$\begin{aligned} T(n_1,\ n_2) &\le c \sum_{u \in T_1} \sum_{v \in T_2} (d_u d_v \sqrt{(d_u + d_v)} \log(d_u + d_v)) \\ &\le c\sqrt{(n_1+n_2)} \log(n_1+n_2) \sum_{u \in T_1} d_u \sum_{v \in T_2} d_v \\ &\le c\ n_1 n_2 \sqrt{(n_1+n_2)} \log(n_1+n_2) \\ &\le c n^{2.5} \log n. \end{aligned}$$

4.2 Sparse Dynamic Program for MAAC

The MAAC algorithm given in Section 4.1 spends most of its time computing maximum weighted bipartite matchings in complete bipartite graphs, where the weight of each edge in the bipartite graph represents the size of a MAAC between some pair of rooted subtrees. For the MAST problem, a faster version of this algorithm is presented in [12] by Farach-Colton and Thorup. The speedup is achieved by eliminating many edges in many of the bipartite graphs constructed by the algorithm. In particular, observe that if two subtrees do not share any leaf label, the size of their MAAC is zero. Thus, if there are only a few leaves with any given label, it is highly likely that many edge weights are zero. Further, it turns out that several edges can be deleted in many of the bipartite graphs without affecting the

optimum solution. Farach-Colton and Thorup's sparse dynamic programming algorithm for MAST incorporates these features into the Steel-Warnow algorithm, thereby achieving a running time of $O(n^2)$ [12].

In this section, we adapt the Farach-Colton-Thorup MAST algorithm [12] to MAAC. We show that as long as the number of leaves with any given label is $O(n^{1/2-\epsilon})$, the algorithm runs in $O(n^2)$ time, which matches the bound in [12] for MAST, where there is only one leaf with a given label in each tree. The worst-case running time of our algorithm, however, is $O(n^{2.5}\log n)$, matching that of the straightforward dynamic programming algorithm given in the previous section.

Key Lemmas. The following discussion uses notation from the straightforward MAAC algorithm given in the previous section: For a given node v in a rooted tree T, we let $A(v)$ be the set of all labels of leaves that descend from v, $p(v)$ be the parent of v, and $c(v)$ be the set of children of v. For a given rooted tree T, we let $V(T)$ be the set of all nodes of a tree.

For each internal node v of T_1 and T_2, among all children of v, we choose the child having the greatest number of descendent leaves to be the "heavy" child and all the remaining children are "light" children. If there are many nodes that have the same maximum number of descendants, we designate one of them as the heavy child arbitrarily. A node is "heavy" if it is the heavy child of its parent and, otherwise, it is "light."

For vertices $u_1 \in T_1$, $u_2 \in T_2$, consider the weighted bipartite graph G_{u_1,u_2} constructed by the basic MAAC algorithm (see Section 4.1). In G_{u_1,u_2}, we will let h_1 and h_2 denote the heavy child of u_1 in T_1 and the heavy child of u_2 in T_2, respectively. An edge in G_{u_1,u_2} will be called "heavy-heavy" if it is between h_1 and h_2; similarly, we will refer to "heavy-light" and "light-light" edges.

We will denote by $\mathcal{M}$ the set of all bipartite graphs encountered throughout the course of the algorithm. Also, from now on, we will assume that we have modified all the bipartite graphs in $\mathcal{M}$ to get rid of all zero-weight edges.

We first bound the total number of light-light edges across all the bipartite graphs in $\mathcal{M}$ in Lemmas 1 and 2. In Lemma 3, we show how to delete most of the heavy-light edges in each bipartite graph in $\mathcal{M}$ without affecting the value of the MWBM solution. Thus, we create, for each bipartite graph G in $\mathcal{M}$, a bipartite graph G' with fewer edges. We will call the set of all such reduced bipartite graphs $\mathcal{M}'$. All MWBM computations are performed only on these reduced bipartite graphs in $\mathcal{M}'$. Finally, in Lemma 4, we bound the total number of edges across all bipartite graphs in $\mathcal{M}'$ and this helps us bound the total running time of the algorithm.

As defined in Section 3.2, we let $\pi_{i,j}$ be the number of leaves labeled with area a_i in tree T_j.

Lemma 1. *Each leaf in trees T_1 and T_2 has $O(\log n)$ ancestors that are light nodes.*

Proof. Consider a leaf l in T_1. Let r be the root of T_1. Suppose that l has more than $\log_2(n)$ ancestors that are light nodes. It is easy to see that if a node v is the light child of $p(v)$, then $|L(p(v))| \geq 2|L(v)|$. Thus, if a node has more than $\log_2(n)$ ancestors that are light, we would have $|L(r)| > 2^{\log_2(n)}$ or $|L(r)| > n$, which is a contradiction. Therefore, l has at most $\log_2(n)$ light ancestors. The same argument holds for a leaf l in T_2. □

Lemma 2. *Across all bipartite graphs in $\mathcal{M}$, the total number of light-light edges is $O\big((\sum_{a_i \in \mathcal{A}} \pi_{i,1}\pi_{i,2})\log^2 n\big)$.*

Proof. The weight of an edge (x, y) is nonzero if and only if the two sets of descendent leaves $L(x)$ and $L(y)$ intersect. Consider a label $a_i \in \mathcal{A}$ and let S_1 and S_2, respectively, be the sets of leaves of T_1 and T_2 which are labeled with a_i. Note that $|S_j| = \pi_{i,j}, j = 1, 2$. A light ancestor of a pair of leaves, one in S_1 and one in S_2, accounts for one light-light edge across all graphs in $\mathcal{M}$. By Lemma 1, there are $O(\pi_{i,j}\log n)$ ancestors of elements of S_j, $j = 1, 2$, that are light. Therefore, there are at most $O(\pi_{i,1}\pi_{i,2}\log^2 n)$ light-light edges produced by elements of S_1 and S_2. Summing the quantity over all labels a_i, we get the desired upper bound on the number of light-light edges. □

Lemma 3. *For each bipartite graph $G = (V, E)$ in $\mathcal{M}$ with α light-light edges, we can reduce the number of edges in G to get $G' = (V, E')$ such that $MWBM(G) = MWBM(G')$ and $|E'| \leq 3\alpha + 3$.*

Proof. Let V_1 and V_2 be the two parts of V and let h_1 and h_2 be the heavy nodes in V_1 and V_2, respectively. Let E^* be the set of light-light edges, with $|E^*| = \alpha$. We partition the sets $V_1 \setminus \{h_1\}$ and $V_2 \setminus \{h2\}$ into two disjoint subsets as follows: $V_1 \setminus \{h_1\} = V_1^\alpha \cup V_1^\beta$ and $V_2 \setminus \{h_2\} = V_2^\alpha \cup V_2^\beta$ such that $v \in V_j^\alpha$ iff there exists no edge $e \in E^*$ such that v is in e. Among all edges connecting h_1 and an element of V_2^α, we can delete all except the heaviest edge since no maximum matching can contain them. The same reasoning applies for h_2 and an element of V_1^α. So, we can construct a new graph G' with the same set of vertices: The new set of edges E' contains α light-light edges: one possible edge between h_1 and h_2, two possible edges between h_1 and V_2^α, and h_2 and V_1^α, and at most $|V_1^\beta| + |V_2^\beta| \leq 2\alpha$ edges between h_1 and V_2^β, and h_2 and V_1^β. Therefore, $|E'| \leq 3\alpha + 3$ or the graph G' contains at most $3\alpha + 3$ edges. □

We now present SP-MAAC, our sparse dynamic programming algorithm for the MAAC problem. The differences between this algorithm and the earlier MAAC algorithm are italicized.

SP-MATCH(v, w)
1 Construct $G_{v,w}$
2 *Remove all zero-weight edges from G*
3 *For each heavy child, remove all edges incident to it except for the heaviest one.*
4 Construct $E_0 = \text{MWBM}(G_{v,w})$
5 Let $E_0 = \{(v_1, w_1), (v_2, w_2), \ldots, (v_k, w_k)\}$
6 Construct tree M with root s such that SP-MAAC(T_{v_i}, T_{w_i}) is the ith child of s.
7 **return** M

DIAG(v, w)
1 $t_1 \leftarrow$ largest SP-MAAC(T_v, T_x) such that $x \in c(w)$

2 $t_2 \leftarrow$ largest SP-MAAC(T_y, T_w) such that $y \in c(v)$
3 **return** the larger of t_1 and t_2

ALGORITHM SP-MAAC(T_1, T_2)
1 *Choose a heavy child for each internal node of T_1 and T_2*
2 Let $\mathcal{O}$ be an ordering of $V(T_1) \times V(T_2)$
3 such that if (v_1, w_1) is before (v_2, w_2),
4 then v_1 is not an ancestor of v_2 and w_1 is not an ancestor of w_2.
5 **for** (v, w) in increasing order of $\mathcal{O}$
6 **do if** v or w is a leaf
7 **then** SP-MAAC$(T_v, T_w) \leftarrow$ a node with label in $S = A(v) \cap A(w)$ if $S \neq \emptyset$; else $\emptyset$
8 **else** SP-MAAC$(T_v, T_w) \leftarrow$ larger of SP-MATCH(v, w) and DIAG(v, w)
9 **return** SP-MAAC(T_{r_1}, T_{r_2}); r_1 is the root of T_1 and r_2 is the root of T_2.

Lemma 4. *Across all graphs in $\mathcal{M}'$, the total number of edges is $O(\min\{(\sum_{a_i \in \mathcal{A}} \pi_{i,1}\pi_{i,2}) \log^2 n, n^2\})$.*

Proof. By Lemmas 2 and 3, the total number of edges across all graphs in $\mathcal{M}'$ is $O((\sum_{a_i \in \mathcal{A}} \pi_{i,1}\pi_{i,2}) \log^2 n)$. This summation can be shown to be $\Omega(n^2 \log^2 n)$ in the worst case. However, the total number of edges is at most the number of edges in all complete bipartite graphs, which is

$$\sum_{x \in V(T_1)} \sum_{y \in V(T_2)} |c(x)| \cdot |c(y)| = O(n_1 n_2) = O(n^2).$$

Hence, the total number of edges across all graphs in $\mathcal{M}'$ is $O(\min\{(\sum_{a_i \in \mathcal{A}} \pi_{i,1}\pi_{i,2}) \log^2 n, n^2\})$. □

As presented, the algorithm uses $O(n^2)$ time to remove all zero-weight edges from the bipartite graphs. However, it is not difficult to maintain $|V(T_2)|$ queues of nonzero edges incident to each internal node of T_2 and update this queue as we compute the MAAC in the ordering of $\mathcal{O}$.

Running-time Analysis.

Theorem 6. *Algorithm SP-MAAC computes the MAAC in*

$$O\left((\sqrt{n}\log n)\min\left\{\left(\sum_{a_i \in \mathcal{A}} \pi_{i,1}\pi_{i,2}\right)\log^2 n, n^2\right\} + n^2\right).$$

Proof. Let the total running time of SP-MAAC be $time_{\text{SP-MAAC}}$. The algorithm spends a total of $O(n)$ time in Step 1 choosing a heavy child for each node and it spends a total of $O(n^2)$ time computing the ordering $\mathcal{O}$. Each call to DIAG(v, w) takes $O(|c(v)| + |c(w)|)$ time. Over all the calls to DIAG, the total running time is therefore $O(n^2)$. Let the time spent in all calls to SP-MATCH be $time_{\text{SP-MATCH}}$. Therefore,

$$time_{\text{SP-MAAC}} = O(n^2) + time_{\text{SP-MATCH}}.$$

We now show how to bound $time_{\text{SP-MATCH}}$. We let $time_{\text{MWBM}}$ be the running time of a single call to procedure MWBM in SP-MATCH. The MWBM computation is performed using the Gabow-Tarjan algorithm [16]. This algorithm runs in $O(\sqrt{|V|}|E| \log |V|)$ on a graph $G = (V, E)$. We have:

$$\begin{aligned} time_{\text{SP-MATCH}} &= \sum_{G=(V,E)\in\mathcal{M}'} time_{\text{MWBM}}(G) \\ &= \sum_{G=(V,E)\in\mathcal{M}'} O\left(\sqrt{|V|}|E| \log |V|\right) \\ &= \sum_{G=(V,E)\in\mathcal{M}'} O\left(\sqrt{n}|E| \log n\right) \\ &= O\left(\sqrt{n}\log n \sum_{G=(V,E)\in\mathcal{M}'} |E|\right) \\ &= O\left((\sqrt{n}\log n)\min\left\{\left(\sum_{a_i \in \mathcal{A}} \pi_{i,1}\pi_{i,2}\right)\log^2 n, n^2\right\}\right) \end{aligned}$$

from Lemma 4.

Hence,

$$time_{\text{SP-MAAC}} = O\left((\sqrt{n}\log n)\min\left\{\left(\sum_{a_i \in \mathcal{A}} \pi_{i,1}\pi_{i,2}\right)\log^2 n, n^2\right\} + n^2\right).$$

□

Finally, we note the following two bounds for the running time of the algorithm:

- It is not difficult to see that if every $\pi_{i,j}$ is $O(n^{1/2-\epsilon})$, then $time_{\text{SP-MAAC}} = O(n^2)$. Thus, as long as no leaf label occurs a huge number of times, the algorithm is as efficient as the Farach-Colton-Thorup MAST algorithm, where it is assumed that every leaf label occurs exactly once.
- In the worst case, the running time $time_{\text{SP-MAAC}}$ remains $O(n^{2.5} \log n)$, matching the time bound of the basic MAAC algorithm given in the previous section.

4.3 Testing Isomorphism between Two Rooted Area Cladograms

The MAAC distance metric between area cladograms gives us a polynomial-time algorithm for testing isomorphism: We apply the maximum agreement area cladogram algorithm from the previous section to compute the MAAC distance between the two area cladograms, and we conclude that the two cladograms are isomorphic if and only if the distance is zero. However, we can do better: We present a fast algorithm for testing isomorphism between area cladograms without computing the MAAC distance between the cladograms. The algorithm is adapted from the algorithm for testing rooted tree isomorphism from [1].

The input to the algorithm consists of two rooted area cladograms, T_1 and T_2, on n leaves (if the number of leaves is different, then clearly they are not isomorphic). We assume that the leaves are labeled with integers from 1 through n, not all distinct. The algorithm is based on assigning an integer $index(u)$ to each node u in the tree. When the node u is a leaf, the index is just its label. The algorithm is as follows:

1. Compute the *height*, the maximum distance between the root and a leaf, of the two trees. If the heights are not the same, then the trees are not isomorphic; otherwise, let the height be h.
2. Based on the height, assign level numbers to the nodes of the trees. The level number of a node at a distance of d from the root is set to be $h - d$.
3. For each leaf u at level 0, set $index[u]$ to be the leaf-label.
4. For each level i, in order, we compute $index[v]$ for each node v at level i as follows:
   - We define an ordered list of the indices of the children of the node v, sorted in ascending order. If v is a leaf, then its tuple consists of just its label. Let L_i be the list of tuples of nodes at level i in T_1. Let L'_i be the corresponding list for T_2. Now, lexicographically sort L_i and L'_i to obtain S_i and S'_i, respectively.
   - Set $index[v]$ to be the *rank* of v's tuple in the sorted list S_i. The ranks start from 1 and all identical tuples receive the same rank. Indices for vertices in T_2 are assigned similarly.
   - If S_i and S'_i are not identical, then declare T_1 and T_2 to be nonisomorphic and quit.
5. If the roots of T_1 and T_2 are assigned the same index, then the trees are isomorphic; otherwise, not.

Proof of Correctness. We first show that if the algorithm declares two trees to be isomorphic, then they are indeed so. The proof is by induction on the number of levels in the trees. Suppose there is only one level, then the trees have only one leaf each. If the algorithm declares the trees to be isomorphic, then the leaves have the same label and, hence, they are indeed isomorphic. Inductively, assume that the algorithm correctly tests the isomorphism of trees that have up to k levels. Suppose T_1 and T_2 have $k+1$ levels each. If the algorithm declares T_1 and T_2 to be isomorphic, then the tuples assigned to the roots of T_1 and T_2 are identical, which means that, for each node of T_1 at level k, there is a node at level k in T_2 that is assigned the same index and vice versa. From the induction hypothesis, subtrees at level k that are assigned identical indices are isomorphic. Hence, for each subtree of T_1 at level k, there is a level k subtree isomorphic to it in T_2 and vice versa. This implies that T_1 and T_2 are isomorphic themselves. It can be proved similarly by induction that if T_1 and T_2 are isomorphic, the algorithm declares them so. This completes our proof. □

Running Time. The running time of the above algorithm for testing isomorphism is $O(n)$, where n is the number of leaves in the input trees (see [1]).

5 MAAC for k Trees

In this section, we study the complexity of computing the MAAC of many area-labeled trees.

In [2], Amir and Keselman show that computing the MAST of just three trees with unbounded degrees is NP-hard by a reduction from three-dimensional matching. Since the MAAC problem is a less restricted version of the MAST problem, computing the MAAC of three or more unbounded degree cladograms is also NP-hard. However, polynomial time algorithms for computing MAST of k trees with bounded degrees were first presented in [2] and then, later, in [10]. We now establish that such a result is not possible for MAAC unless $\mathrm{P} = \mathrm{NP}$; more specifically, we show that computing the MAAC of a set of k *binary* trees is NP-hard. In view of this result, it would appear that natural generalizations to MAAC of the approaches used to compute the MAST of k trees with maximum degree d (in [2] and [10]) would run in time exponential in both k and d.

5.1 NP-Completeness of k-Tree MAAC

The NP-completeness proof will use a reduction from VERTEX-COVER and is adapted from the NP-completeness proof of the Longest Common Subsequence (LCS) problem for k sequences presented in [22]. We will use the following description of the decision version of VERTEX-COVER.

— VERTEX-COVER
— Input: Graph $G = (V, E)$ and an integer k
— Question: Is there a subset $S \subseteq V$ of at most k vertices such that, for every edge $e = (x, y) \in E$, $\{x, y\} \cap S \neq \emptyset$?

We will reduce VERTEX-COVER to the following decision version of the problem of computing the MAAC of many binary area-labeled trees:

— BIN-MAAC
— Input: set $\mathcal{T}$ of binary area-labeled trees, and an integer k.
— Question: Is $|\mathrm{MAAC}(\mathcal{T})| \geq k$?

Theorem 7. *BIN-MAAC is NP-complete.*

Proof. BIN-MAAC is in NP since a naive algorithm can simply guess a subset of leaves of each tree in $\mathcal{T}$ and check if all induced trees are isomorphic in polynomial time. Hence, it will suffice to show that VERTEX-COVER reduces to BIN-MAAC.

Consider an instance $(G = (V, E), k)$ of VERTEX-COVER. Let $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{e_1, e_2, \ldots, e_m\}$. We will construct a set $\mathcal{T} = \{T_0, T_1, \ldots, T_m\}$ of $(m+1)$ binary area-labeled trees such that G has a vertex cover of size k if and only if $|\mathrm{MAAC}(\mathcal{T})| \geq n - k$.

The set of areas with which the leaves of tree are labeled is $\mathcal{A} = \{v_1, v_2, \ldots, v_n\}$.

The tree T_0 is a binary tree on leaf set $v_1, v_2, \ldots, v_n$, with no nontrivial left subtrees. Thus, T_0 is a rooted "caterpillar" tree defined by the ordering on its leaves, which we will assume is given by $v_1, v_2, \ldots, v_n$. We use the notation $T_0 \setminus X$ to denote the tree obtained by deleting the leaves in X from the tree T_0 and suppressing nodes with only one child.

Now, consider an edge $e = (v_x, v_y)$ in the graph G, with $x < y$. We will define the rooted tree T_e as follows: T_e is obtained by "concatenating" the trees $T_0 \setminus \{v_x\}$ and $T_0 \setminus \{v_y\}$, where by "concatenation" we mean replacing the deepest leaf of the first tree by a branching node whose children are the second tree and the old leaf. Note, therefore, that, for each $i \neq x, y$, node v_i appears twice in the tree T_e, but that v_x and v_y each appear once.

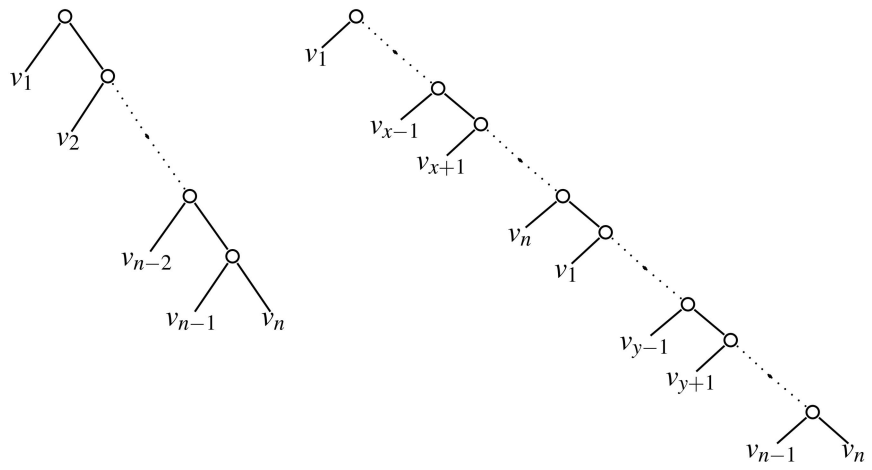

Fig. 7. Tree T_0 (left) and T_e (right) corresponding to $e = (v_x, v_y)$ with $x < y$.

Furthermore, v_x appears "below" v_y (in the sense that the parent of v_y is an ancestor of v_x). Fig. 7 illustrates this construction.

Now, we show that G has a vertex cover of size k if and only if $|\mathrm{MAAC}(\mathcal{T})| \geq n - k$:

- (only if:) Suppose G has a vertex cover S of size k. Let $T^* = T_0 \setminus S$. It is clear that $|L(T^*)| = n - k$. Therefore, it is enough to show that T^* is an agreement subtree of the set of trees $\{T_e : e \in E(G)\} \cup \{T_0\}$. Obviously, T^* is a subtree of T_0. Now, consider an edge $e = (v_x, v_y) \in E(G)$, with $x < y$. The tree T_e is the concatenation of $T_0 \setminus \{v_x\}$ and $T_0 \setminus \{v_y\}$ in which the ancestor of v_y lies above that of v_x. Thus, the top half of T_e contains all the vertices (in order) except for v_x, and the bottom half of T_e contains all the vertices except for v_y. Since S is a vertex cover, at least one of v_x and v_y is in S; without loss of generality, suppose $v_x \in S$. Then, T^* is a subtree of the top half of T_e. (If $v_y \in S$, we would deduce that T^* is a subtree of the bottom half of T_e.)
- (if:) If $|MAAC(\mathcal{T})| \geq n - k$, then let $S = V \setminus L(\mathrm{MAAC}(\mathcal{T}))$. Because $\mathrm{MAAC}(\mathcal{T})$ is a subtree of T_0, every v_i labels only one leaf. That implies $|S| = |V| - |L(\mathrm{MAAC}(\mathcal{T}))| \leq k$. It is sufficient to show that S is a vertex cover of G in order for G to have a vertex cover of size at most k. Consider any edge $e = (v_x, v_y)$, $x < y$, and assume that neither endpoint is in S; hence, both are labels of leaves of $\mathrm{MAAC}(\mathcal{T})$. In the trees T_0 and T_e, there is only one instance of label v_x and one instance of label v_y. Because of the structure of T_0, the parent of the leaf labeled v_x is above the leaf labeled v_y. However, in T_e, the leaf labeled v_x is strictly below the parent of the leaf labeled v_y. This contradicts the assumption that both v_x and v_y are in $L(\mathrm{MAAC}(\mathcal{T}))$. Therefore, for every edge e, S contains at least one of its vertices and S is a vertex cover of G. □

ACKNOWLEDGMENTS

The research of Geneshkumar Ganapathy was supported by US National Science Foundation (NSF) grants 0331453 and 0121680, Hai-son Le by an Undergraduate Research Opportunity Program (UROP) grant from the Computer Sciences Department at the University of Texas at Austin, Vijaya Ramachandran by NSF grant CCF-0514876, Tandy Warnow by NSF grants 0331453, 0312830, and 0121680, Barbara Goodson by NSF IGERT training grant 0114387, and Robert Jansen by NSF grant DEB 0120709.

REFERENCES

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.
[2] A. Amir and D. Keselman, "Maximum Agreement Subtrees in a Set of Evolutionary Trees: Metrics and Efficient Algorithms," *SIAM J. Computing,* vol. 26, no. 6, pp. 1656-1669, 1997.
[3] D.R. Brooks, "Hennig's Parasitological Method: A Proposed Solution," *Systematic Zoology,* vol. 30, pp. 229-249, 1981.
[4] J.H. Brown and M.V. Lomolino, *Biogeography,* second ed. Sinauer Assoc., 1998.
[5] P. Buneman, "The Recovery of Trees from Measures of Dissimilarity," *Math. in the Archaelogical and Historical Sciences,* pp. 387-395, 1971.
[6] J.V. Crisci, L. Katinas, and P. Posadas, *Historical Biogeography: An Introduction.* Harvard Univ. Press, 2003.
[7] M. Crisp, H.P. Linder, and P. Weston, "Cladistic Biogeography of Plants in Australia and New Guinea: Congruent Pattern Reveals Two Endemic Tropical Tracts," *Systematic Biology,* vol. 44, no. 4, pp. 457-473, 1995.
[8] W.H.E. Day, "Optimal Algorithms for Comparing Trees with Labeled Leaves," *J. Classification,* vol. 2, pp. 7-28, 1985.
[9] B.C. Emerson and P. Oromi, "Diversification of the Forest Beetle Genus Tarphius on the Canary Islands, and the Evolutionary Origins of Island Endemics," *Evolution,* vol. 59, no. 3, pp. 586-598, 2005.
[10] M. Farach-Colton, T.M. Przytycka, and M. Thorup, "On the Agreement of Many Trees," *Information Processing Letters,* vol. 55, pp. 297-301, 1995.
[11] M. Farach-Colton, T.M. Przytycka, and M. Thorup, "The Maximum Agreement Subtree Problem for Binary Trees," *Proc. Second Ann. European Symp. Algorithms,* pp. 381-393, 1995.
[12] M. Farach-Colton and M. Thorup, "Fast Comparison of Evolutionary Trees," *Proc. Fifth Ann. ACM-SIAM Symp. Discrete Algorithms,* pp. 481-488, 1994.
[13] M. Farach-Colton and M. Thorup, "Sparse Dynamic Programming for Evolutionary Tree Comparison," *Proc. 35th Ann. Symp. Foundations of Computer Science,* pp. 770-779, 1994.

[14] J. Felsenstein, *Inferring Phylogenies.* Sinauer Assoc. Inc., 2003.
[15] C.R. Finden and A.D. Gordon, "Obtaining Common Pruned Trees," *J. Classification,* vol. 2, pp. 255-276, 1985.
[16] H. Gabow and R.R. Tarjan, "Faster Scaling Algorithms for Network Problems," *SIAM J. Computing,* vol. 18, no. 5, pp. 1013-1036, 1989.
[17] W.D. Goddard, E. Kubicka, and G. Kubicki, "The Agreement Metric for Labeled Binary Trees," *Math. Biosciences,* vol. 123, pp. 215-226, 1994.
[18] A.P. Jackson, "Cophylogeny of the Ficus Microcosm," *Biological Rev.,* vol. 79, no. 4, pp. 751-768, 2004.
[19] A.P. Jackson, "Phylogeny and Biogeography of the Malagasy and Australasian Rainbowfishes (Teleostei: Melanotaenioidei): Gondwanan Vicariance and Evolution in Freshwater," *Molecular Phylogenetics and Evolution,* vol. 33, no. 3, pp. 719-734, 2004.
[20] C. Juan, B.C. Emerson, P. Oromi, and G.M. Hewitt, "Colonization and Diversification: Towards a Phylogeographic Synthesis for the Canary Islands," *Trends in Ecology and Evolution,* vol. 15, no. 3, pp. 104-109, 2000.
[21] F. Lapointe and L. Rissler, "Congruence, Consensus and Comparative Phylogeography of Codistributed Species in California," *The Am. Naturalist,* vol. 166, no. 2, pp. 290-299, 2005.
[22] D. Maier, "The Complexity of Some Problems on Subsequences and Supersequences," *J. ACM,* vol. 25, no. 2, pp. 322-336, 1978.
[23] R.D.M. Page, "Quantitative Cladistic Biogeograpy: Constructing and Comparing Area Cladograms," *Systematic Zoology,* vol. 37, pp. 254-270, 1988.
[24] R.D.M. Page, "Maps between Trees and Cladistic Analysis of Historical Associations among Genes," *Systematic Biology,* vol. 43, no. 1, pp. 58-77, 1994.
[25] D.F. Robinson and L.R. Foulds, "Comparison of Phylogenetic Trees," *Math. Biosciences,* vol. 53, pp. 131-147, 1981.
[26] D.E. Rosen, "Vicariant Patterns and Historical Explanation in Biogeograpy," *Systematic Zoology,* vol. 27, pp. 159-188, 1978.
[27] N. Satou and M. Nei, "The Neighbor Joining Method: A New Method for Reconstructing Phylogenetic Trees," *Molecular Biology and Evolution,* vol. 4, pp. 406-425, 1987.
[28] M. Steel and T. Warnow, "Kaikoura Tree Theorems: Computing the Maximum Agreement Subtree," *Information Processing Letters,* vol. 48, pp. 77-82, 1993.
[29] D. Swofford, G.J. Olson, P.J. Waddell, and D.M. Hillis, "Phylogenetic Inference," *Molecular Systematics,* second ed., pp. 407-425, Sinauer Assoc., 1996.
[30] M.G.P. van Veller, M. Zandee, and D.J. Kornet, "Two Requirements for Obtaining Common Patterns Under Different Assumptions in Vicariance Biogeography," *Cladistics,* vol. 15, pp. 393-405, 1999.

Ganeshkumar Ganapathy received the bachelor's degree in computer science from the Birla Instituteof Technology and Science (BITS), Pilani, in 1999. He is currently completing the PhD degree in computer sciences at the University of Texas at Austin, under the supervision of Vijaya Ramachandran and Tandy Warnow. Beginning in Fall 2006, he will be a postdoctoral fellow at the National Evolutionary Synthesis Center at Durham, North Carolina. His research interests are in computational phylogeny and biogeography.

Barbara Goodson received the master's degree in biology from Tennessee Tech University in 2000. She is currently completing the PhD degree in botany under the supervision of Dr. Robert K. Jansen at the University of Texas-Austin. Her research interests focus on molecular systematics and biogeography of Descurainia (Brassicaceae), and colonization and adaptive radiation in Macaronesian island endemics.

Robert Jansen received the PhD degree in botany from Ohio State University under the direction of Tod F. Stuessy and did postdoctoral research with Jeffrey D. Palmer at the University of Michigan. He holds the position of Blake Centennial Professor in the Section of Integrative Biology at the University of Texas at Austin. His research uses molecular data and phylogenetic methods to examine questions in chloroplast genome evolution, evolution and systematics of the flowering plant families Asteraceae and Campanulaceae, and the origin and evolution of oceanic island floras. He is a member of three graduate programs at the University of Texas, including plant biology, ecology, evolution, and behavior, and cellular and molecular biology. He currently serves as chair of the Section of Integrative Biology.

Hai-son Le attended the University of Houston for one year. Currently, he is at the University of Texas at Austin pursuing a bachelor's degree in computer sciences. His research interests are in algorithm design and theoretical computer sciences. He is a student member of the ACM and a member of the Turing Scholars Honor program.

Vijaya Ramachandran received the PhD degree in electrical engineering and computer science from Princeton University in 1983. She was an assistant professor at the University of Illinois at Urbana-Champaign 1983-1988 and an associate professor 1989-1995 and full professor 1995-present at the University of Texas at Austin. Her research interests are in algorithm design and theoretical computer science. She is currently on the editorial boards of the *ACM Transactions on Algorithms*, *Journal of the ACM*, *SIAM Journal on Computing*, and *SIAM Journal on Discrete Mathematics*.

Tandy Warnow received the PhD degree in mathematics from the University of California at Berkeley under the direction of Gene Lawler and did postdoctoral training with Simon Tavare and Michael Waterman at the University of Southern California. She is a professor of computer sciences at the University of Texas at Austin. Her research combines mathematics, computer science, and statistics to develop improved models and algorithms for reconstructing complex and large-scale evolutionary histories in both biology and historical linguistics. She received the US National Science Foundation (NSF) Young Investigator Award in 1994 and the David and Lucile Packard Foundation Award in Science and Engineering in 1996. She is a member of five graduate programs at the University of Texas, including computer science; ecology, evolution, and behavior; molecular and cellular biology; mathematics; and computational and applied mathematics. She is also one of the co-PI's for the multidisciplinary CIPRES (Cyber-Infrastructure for Phylogenetic Research) Project, currently funded by the NSF under their Information Technology Program.