Comparing and Aggregating Partially Resolved Trees[†]

Mukul S. Bansal[‡ §] Jianrong Dong[§] David Fernández-Baca[§¶]

Abstract

Partially-resolved — that is, non-binary — trees arise frequently in the analysis of species evolution. Non-binary nodes, also called multifurcations, must be treated carefully, since they can be interpreted as reflecting either lack of information or actual evolutionary history. While several distance measures exist for comparing trees, none of them deal explicitly with this dichotomy. Here we introduce two kinds of distance measures between rooted and unrooted partially-resolved phylogenetic trees over the same set of species; the measures address multifurcations directly. For rooted trees, the measures are based on the topologies the input trees induce on triplets; that is, on three-element subsets of the set of species. For unrooted trees, the measures are based on quartets (four-element subsets). The first class of measures are parametric distances, where there is a parameter that weighs the difference between an unresolved triplet/quartet topology and a resolved one. The second class of measures are based on Hausdorff distance, where each tree is viewed as a set of all possible ways in which the tree can be refined to eliminate unresolved nodes. We give efficient algorithms for computing parametric distances and give conditions under which Hausdorff distances can be approximated in polynomial time. Additionally, we (i) derive the expected value of the parametric distance between two random trees, (ii) characterize the conditions under which parametric distances are near-metrics or metrics, (iii) study the computational and algorithmic properties of consensus tree methods based on the measures, and (iv) analyze the interrelationships among Hausdorff and parametric distances.

Keywords. Aggregation, computational biology, consensus, Hausdorff distance, phylogenetic trees, quartet distance, triplet distance.

1 Introduction

Evolutionary trees, also known as phylogenetic trees or phylogenies, represent the evolutionary history of sets of species. Such trees have uniquely labeled leaves, corresponding to the species,

[†]An extended abstract of this paper was presented at the 8th Latin American Symposium on Theoretical Informatics, Búzios, Brazil, April 2008.

[‡]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Email: mukul@csail.mit.edu.

[§]Department of Computer Science, Iowa State University, Ames, IA 50011, USA. Email: {jrdong, fernande}@iastate.edu.

[¶]Corresponding author.

and unlabeled internal nodes, representing hypothetical ancestors. The trees can be either rooted, if the evolutionary origin is known, or unrooted, otherwise.

This paper addresses two related questions:

(1) How does one measure how close two evolutionary trees for the same set of species are to each other?

(2) How does one combine or *aggregate* the phylogenetic information from conflicting trees over the same set of species into a single *consensus tree*?

Among the motivations for the first question is the growth of phylogenetic databases, such as TreeBase [35], with the attendant need for sophisticated querying mechanisms and for means to assess the quality of answers to queries. The second question arises from the fact that phylogenetic analyses — e.g., by parsimony or by maximum likelihood [26] — typically produce multiple evolutionary trees (often in the thousands) for the same set of species. Another motivation arises from the *supertree* problem, which generalizes the consensus tree problem to the case where the input trees may not all have to share the same species[1] [8, 21].

Question (1) can be approached by defining appropriate *distance measures* between phylogenies. These distance measures can be used to cast question (2) as a *median* problem, where the objective is to find a consensus tree whose total distance to the given trees is minimized.

Here we define, analyze the properties of, and give algorithms for two new kinds of distance measures between phylogenies over the same set of species. For rooted trees, our measures are based on the topologies the input trees induce on *triplets*; that is, on three-element subsets of the set of species. For unrooted trees, the measures are based on *quartets* (four-element subsets). Our approach is motivated by the observation that triplet and quartet topologies are the basic building blocks of rooted and unrooted trees, in the sense that they are the smallest topological units that completely identify a phylogenetic tree [41]. Triplet and quartet-based distances thus provide a robust and fine-grained measure of the differences and similarities between trees[2]. In contrast with traditional quartet and triplet distances, our two classes of distance measures deal cleanly with the presence of *unresolved* nodes, also known as *polytomies*. For rooted trees polytomies are nodes with more than two children; for unrooted trees, they are nodes of degree greater than three. Polytomies cannot simply be ignored, since they arise naturally in phylogenetic analyses. Furthermore, they must be treated with care: A node may be unresolved because it truly must be so or because there is not enough evidence to break it up into resolved nodes — that is, the polytomies are either "hard" or "soft" [33].

Next, we give an overview of our results, contrasting them with previous work. We then discuss the relationship between our distance measures and recent work on aggregating partial rankings. Unless explicitly stated otherwise, all results mentioned in the rest of the paper deal with trees over the same leaf set.

[1]The distinction between consensus trees and supertrees — the first requiring complete overlap between species sets and the second only partial overlap — is maintained throughout the paper.

[2]Biologically-inspired arguments in favor of triplet-based measures can be found in [15].

Distance measures for partially-resolved phylogenies. In Section 3, we introduce two classes of distance measures. The first is *parametric distance*: Given a triplet (quartet) X, we compare the topologies that each of the two input trees induces on X. If they are identical, the contribution of X to the distance is zero. If both topologies are fully resolved but different, then the contribution is one. Otherwise, the topology is resolved in one of the trees, but not the other. In this case, X contributes p to the distance, where p is a real number between 0 and 1. Parameter p allows one to make a smooth transition between hard and soft views of polytomy. At one extreme, if $p = 1$, an unresolved topology is viewed as different from a fully resolved one. At the other, when $p = 0$, unresolved topologies are viewed as identical to resolved ones. Intermediate values of p allow one to adjust for the degree of certainty one has about a polytomy. Traditional quartet and triplet distances are essentially parametric quartet and triplet distances with parameter $p = 1$.

The second kind of measures proposed here are based on viewing each tree as a set of all possible fully resolved trees that can be obtained from it by refining its unresolved nodes. The distance between two trees is defined as the Hausdorff distance between the corresponding sets[3], where the distance between trees in the sets is the triplet or quartet distance, as appropriate.

Naturally, several other measures for comparing trees have been proposed. While they do not take the degree of resolution of the trees into account, we mention a few of the more important ones and contrast them with triplet- and quartet-based measures.

A popular class of distances are those based on symmetric distance between sets of *clusters* (that is, on sets of species that descend from the same internal node in a rooted tree) or of *splits* (bipartitions of the set of species induced by the removal of an edge in an unrooted tree); the latter is the well-known Robinson-Foulds distance [38]. It is not hard to show that two rooted (unrooted) trees can share many triplet (quartet) topologies but not share a single cluster (split). Cluster- and split-based measures are also coarser than triplet and quartet distances.

Another way to measure the distance between two trees is by counting the number of *branch-swapping* operations — e.g., nearest-neighbor interchange or subtree pruning and regrafting operations [26] — needed to convert one of the trees into the other [3]. However, the associated measures can be hard to compute, and they fail to distinguish between operations that affect many species and those that affect only a few.

An alternative to distance measures are *similarity* measures, of which a notable example is the size of the *maximum agreement subtree* (MAST) [27]. This quantity can be computed efficiently [25, 30]. The range of values that the MAST measure can assume is significantly smaller than that of triplet-based distance; i.e., $\Theta(n)$ versus $\Theta(n^3)$. On the other hand, the two measures appear to be, in some sense, orthogonal[4]: Moving just one leaf from one place to another in a tree can change a large number of triplets without significantly affecting the MAST. Conversely, changing a small proportion of triplets can move a non-negligible number of leaves, leading to a big change in the MAST. While there is a connection between the MAST and the set of rooted triplets common to the input trees (see, e.g., Bryant [11, Chapter 6] and Lee et al. [32]), elucidating the precise relationship between triplet distance and MAST size is, to our knowledge, an open question.

[3]Informally, two sets A and B are at Hausdorff distance τ of each other if each element of A is within distance τ of B and vice-versa. For a formal definition, see Section 3.

[4]We thank one of the reviewers for pointing this out to us.

Properties of the distance measures. In Section 4 we derive exact and asymptotic bounds on expected values of parametric triplet distance and parametric quartet distance. In Section 5, we determine the values of p for which parametric triplet and quartet distances are metrics, *near-metrics* (in the sense of [23]), or non-metrics. We then analyze the relationship between parametric and Hausdorff distances (Section 6), showing that, under a certain density assumption, Hausdorff distance is within a constant factor of parametric distance. That is, the measures are *equivalent* in the sense of [23].

We investigate the properties of median consensus trees relative to parametric distance. In Section 5, we show that there are values of p for which parametric distance is a metric, but the median consensus tree relative to parametric distance may not be fully resolved even if all the input trees are. However, beyond a threshold, the median tree is guaranteed to be fully resolved if the input trees are fully resolved. It has been noted [37] that the NP-completeness of the *maximum triplet compatibility problem*[5] [11] directly implies the NP-hardness of several triplet-based supertree methods, including those based on parametric distance. We conjecture that the consensus version of the problem is also NP-hard. Nevertheless, we argue that the results of Section 5 imply that there is a simple constant-factor approximation algorithm for finding a median tree relative to parametric distance for every $p > 0$.

There is an extensive literature on consensus methods for phylogenetic trees. A non-exhaustive list of methods based on splits or clusters includes strict consensus trees [34], majority-rule trees [5], and the Adams consensus [1]. In *local consensus* methods, the goal is to find a consensus tree that satisfies a given set of constraints on the topology of each triplet [29]. For more thorough surveys of consensus methods, their properties and interrelationships, see [12, 39].

The fact that consensus methods tend to produce unresolved trees, with an attendant loss of information, has been observed before. An alternative approach is to provide multiple consensus trees, instead of a single one. The idea can be developed in different ways. Stockham et al. [47] propose clustering the input trees using some distance measure into groups, each of which is represented by a single consensus tree, in such a way as to minimize some measure of information loss. Bonnard et al. [9] propose a "multipolar" consensus method, which identifies the minimum set of trees (the "poles") that display all input tree splits with support above some threshold. Yet another kind of approach are the "reduced consensus" methods of Wilkinson [50] in which multiple well-supported consensus trees on different subsets of the species are determined. Our distance measures can be used within such alternative consensus frameworks.

There are a number of papers dealing with the problem of, given a set of quartet trees or triplet trees, finding a large compatible subsets or a small subset whose removal leaves a compatible set [7, 13, 42, 43]. Supertree methods based on such approaches have recently received some attention; e.g., see [36, 37, 48]. These methods do not deal explicitly with partially-resolved trees. In contrast, Scornavacca et al. [40] have developed a triplet-based supertree method that handles missing species and partial resolution. Their approach is based on PhySIC [36], a "veto" method, which builds supertrees displaying only triplet information that is not in conflict with any input tree

[5]The input to this problem consists of a set of trees, each of which has three leaves; the leaf sets of these trees may not be identical. The question is to find the largest subset of these triplet trees such that all of the trees are consistent with a single tree T whose leaf set is the union of the leaves of the input triplet trees.

or combination of input trees. The technique presented in [40] finds a subset of the species and a veto supertree for this subset in such a way as to maximize the *cladistic information content* [49, 16] of the supertree.

Algorithmic results. In Section 7, we give efficient algorithms to compute the parametric distance between two trees. For unrooted trees, we rely on existing algorithms for non-parametric distance. Let T_1 and T_2 be two partially-resolved unrooted trees on n nodes. For $i \in \{1, 2\}$, let d_i be the maximum degree of a node in T_i and let $d = \max\{d_1, d_2\}$. The best known algorithms to compute the quartet distance between T_1 and T_2 are the one by Christiansen et al. [14], which runs in time $O(n^2 \min\{d_1, d_2\})$, and the one by Stissing et al. [46], which runs in $O(d^9 n \log n)$ time. In Section 7.5, we discuss how these algorithms can be easily adapted to compute the parametric quartet distance within the same time bounds. It is important to note that the presence of unresolved nodes seems to complicate distance computation. Indeed, the quartet distance between a pair of *fully-resolved* unrooted trees can be obtained in $O(n \log n)$ time [10].

We present a novel $O(n^2)$-time algorithm for computing the parametric triplet distance between two partially resolved rooted treesrooted trees. To our knowledge, there was no previous algorithm for computing triplet distance (parametric or not) other than by enumerating all $\Theta(n^3)$ triplets. Critchlow et al. [18] gave a $O(n^2)$ algorithm for computing the triplet distance between two *fully-resolved* rooted trees. We remark that there is a well-known bijection between rooted and unrooted trees (see Section 4), suggesting that the above-mentioned algorithms for parametric quartet distance could perhaps be used to compute parametric triplet distance. However, even under moderate bounds on the maximum vertex degree, the worst-case times of these algorithms are asymptotically larger than our $O(n^2)$ bound.

Relationship to rank aggregation. The consensus problem on trees exhibits parallels with the *rank aggregation problem*, a problem with a rich history and which has recently found applications to Internet search [2, 6, 17, 20, 31, 22, 23]. Here, we are given a collection of rankings (that is, permutations) of n objects, and the goal is to find a ranking of minimum total distance to the input rankings. A distance between rankings of particular interest is *Kendall's tau*, defined as the number of pairwise disagreements between the two rankings. Like triplet and quartet distances, Kendall's tau is based on elementary ordering relationships. Dwork et al. [22] showed that rank aggregation under Kendall's tau is NP-complete even for four lists.

A permutation is the analog of a fully resolved tree, since every pairwise relationship between elements is given. The analog to a partially-resolved tree is a *partial ranking*, in which the elements are grouped into an ordered list of *buckets*, such that elements in different buckets have known ordering relationships, but elements within a bucket are not ranked [23]. Our definitions of parametric distance and Hausdorff distance are inspired by Fagin et al.'s *Kendall tau with parameter* p and their Hausdorff version of Kendall's tau, respectively [23]. We note, however, that aggregating partial rankings seems computationally easier than the consensus problem on trees. For example, while the Hausdorff version of Kendall's tau has a simple and easily-computable expression [17, 23], it is unclear whether the Hausdorff triplet or quartet distances are polynomially-computable for trees.

2 Preliminaries

Phylogenies. By and large, we follow standard terminology (i.e., similar to [11] and [41]). We write $[N]$ to denote the set $\{1, 2, \ldots, N\}$, where N is a positive integer.

Let T be a rooted or unrooted tree. We write $\mathcal{V}(T)$, $\mathcal{E}(T)$, and $\mathcal{L}(T)$ to denote, respectively, the node set, edge set, and leaf set of T. A *taxon* (plural *taxa*) is some basic unit of classification; e.g., a species. Let S be a set of taxa. A *phylogenetic tree* or *phylogeny* for S is a tree T such that $\mathcal{L}(T) = S$. Furthermore, if T is rooted, we require that each internal node have at least two children; if T is unrooted, every internal node is required to have degree at least three. We write $RP(n)$ to denote the set of all rooted phylogenetic trees over $S = [n]$ and $P(n)$ to denote the set of all unrooted phylogenetic trees over $S = [n]$.

An internal node in a *rooted* phylogeny is *resolved* if it has exactly two children; otherwise it is *unresolved*. Similarly, an internal node in an *unrooted* phylogeny is *resolved* if it has degree three, and *unresolved* otherwise. Unresolved nodes in rooted and unrooted trees are also referred to as *polytomies* or *multifurcations*. A phylogeny (rooted or unrooted) is *fully resolved* if all its internal nodes are resolved. A *fan* is a completely unresolved phylogeny; i.e., it contains a single internal node, to which all leaves are connected (if the phylogeny is rooted, this internal node is the root).

A *contraction* of a phylogeny T is obtained by deleting an internal edge and identifying its endpoints. A phylogeny T_2 is a *refinement* of phylogeny T_1, denoted $T_1 \preceq T_2$, if and only if T_1 can be obtained from T_2 through 0 or more contractions. Tree T_2 is a *full refinement* of T_1 if $T_1 \preceq T_2$ and T_2 is fully resolved. We write $\mathcal{F}(T)$ to denote the set of all full refinements of T.

Let X be a subset of $\mathcal{L}(T)$ and let $T[X]$ denote the minimal subtree of T having X as its leaf set. The *restriction* of T to X, denoted $T|X$, is the phylogeny for X defined as follows. If T is unrooted, then $T|X$ is the tree obtained from $T[X]$ by suppressing all degree-two nodes. If T is rooted, $T|X$ is obtained from $T[X]$ by suppressing all degree-two nodes except for the root.

A *triplet* is a three-element subset of S. A *triplet tree* is a rooted phylogeny whose leaf set is a triplet. The triplet tree with leaf set $\{a, b, c\}$ is denoted by $a|bc$ if the path from b to c does not intersect the path from a to the root. A *quartet* is a four-element subset of S and a *quartet tree* is an unrooted phylogeny whose leaf set is a quartet. The quartet tree with leaf set $\{a, b, c, d\}$ is denoted by $ab|cd$ if the path from a to b does not intersect the path from c to d. A triplet (quartet) X is said to be *resolved* in a phylogenetic tree T over S if $T|X$ is fully resolved; otherwise, X is *unresolved*. An unresolved triplet (quartet) tree is often called a *fan*.

Finally, we introduce notation for certain useful subtrees of a tree T. Suppose T is rooted and v is a node in T. Then, $T(v)$ denotes the subtree of T rooted at v. Suppose T is unrooted and $\{u, v\}$ is an edge in T. Removal of edge $\{u, v\}$ splits the tree T into two subtrees. We denote the subtree that contains node u by $T(u, v)$, and the subtree that contains v by $T(v, u)$.

Distance measures, metrics, and near-metrics. A *distance measure* on a set D is a binary function d on D satisfying the following three conditions: (i) $d(x, y) \geq 0$ for all $x, y \in D$; (ii) $d(x, y) = d(y, x)$ for all $x, y \in D$; and (iii) $d(x, y) = 0$ if and only if $x = y$. Function d is a *metric* if, in addition to being a distance measure, it satisfies the triangle inequality; i.e., $d(x, z) \leq d(x, y) + d(y, z)$ for all $x, y, z \in D$. Distance measure d is a *near-metric* if there is a

constant $c > 0$, independent of the size of D, such that d satisfies the *relaxed polygonal inequality*: $d(x,z) \leq c(d(x,x_1) + d(x_1,x_2) + \cdots + d(x_{n-1},z))$ for all $n > 1$ and $x, z, x_1, \ldots, x_{n-1} \in D$ [23]. Two distance measures d and d' with domain D are *equivalent* if there are constants $c_1, c_2 > 0$ such that $c_1 d'(x,y) \leq d(x,y) \leq c_2 d'(x,y)$ for every pair $x, y \in D$ [23].

3 Distance measures for phylogenies

Here we define the distance measures for rooted and unrooted trees to be studied in the rest of the paper. We use essentially the same notation for the rooted tree measures as for the unrooted tree measures. We do so because the concepts for each case are close analogs of those for the other, the key difference being the use of triplets in one setting (rooted trees) and of quartets in the other (unrooted trees). It will be easy to distinguish between the two settings by simply specifying the context in which the measures are being applied.

Let T_1 and T_2 be any two rooted (respectively, unrooted) phylogenies over taxon set $[n]$. Define the following five sets of triplets (quartets) over $[n]$.

$\mathcal{S}(T_1, T_2)$**:** The set of all triplets (quartets) X such that $T_1|X$ and $T_2|X$ are fully resolved, and $T_1|X = T_2|X$.

$\mathcal{D}(T_1, T_2)$**:** The set of all triplets (quartets) X such that $T_1|X$ and $T_2|X$ are fully resolved, and $T_1|X \neq T_2|X$.

$\mathcal{R}_1(T_1, T_2)$**:** The set of all triplets (quartets) X such that $T_1|X$ is fully resolved, but $T_2|X$ is not.

$\mathcal{R}_2(T_1, T_2)$**:** The set of all triplets (quartets) X such that $T_2|X$ is fully resolved, but $T_1|X$ is not.

$\mathcal{U}(T_1, T_2)$**:** The set of all triplets (quartets) X such that $T_1|X$ and $T_2|X$ are unresolved.

Let p be a real number in the interval $[0, 1]$. The *parametric triplet (quartet) distance between* T_1 *and* T_2 is defined as[6]

$$d^{(p)}(T_1, T_2) = |\mathcal{D}(T_1, T_2)| + p\left(|\mathcal{R}_1(T_1, T_2)| + |\mathcal{R}_2(T_1, T_2)|\right). \tag{1}$$

When the domain of $d^{(p)}$ is restricted to fully resolved trees, and thus $\mathcal{R}_1(T_1, T_2) = \mathcal{R}_2(T_1, T_2) = \mathcal{U}(T_1, T_2) = \emptyset$, we refer to it simply as the *triplet (quartet) distance*.

Parameter p allows one to make a smooth transition from soft to hard views of polytomy: When $p = 0$, resolved triplets (quartets) are treated as equal to unresolved ones, while when $p = 1$, they are treated as being completely different. Choosing intermediate values of p allows one to adjust for the amount of evidence required to resolve a polytomy[7].

[6] Note that the sets $\mathcal{S}(T_1, T_2)$ and $\mathcal{U}(T_1, T_2)$ are not used in the definition of $d^{(p)}$, but are needed for other purposes.

[7] We note that parametric triplet/quartet distance is a *profile-based metric*, in the sense of [23]. However, the use of the word "profile" in [23] is quite different from our use of the term.

An alternative distance measure (inspired by References [23, 17]), is the *Hausdorff distance*, defined as follows. Let d be a metric over fully resolved trees. Metric d is extended to partially resolved trees as follows.

$$d_{\text{Haus}}(T_1, T_2) = \max \left\{ \max_{t_1 \in \mathcal{F}(T_1)} \min_{t_2 \in \mathcal{F}(T_2)} d(t_1, t_2), \max_{t_2 \in \mathcal{F}(T_2)} \min_{t_1 \in \mathcal{F}(T_1)} d(t_1, t_2) \right\} \tag{2}$$

When d is the triplet (quartet) distance, d_{Haus} is called the *Hausdorff triplet (quartet) distance*.

Definition (2) requires some explanation. The quantity $\min_{t_2 \in \mathcal{F}(T_2)} d(t_1, t_2)$ is the distance between t_1 and the set of full refinements of T_2. Hence,

$$\max_{t_1 \in \mathcal{F}(T_1)} \min_{t_2 \in \mathcal{F}(T_2)} d(t_1, t_2)$$

is the maximum distance between a full refinement of T_1 and the set of full refinements of T_2. Similarly,

$$\max_{t_2 \in \mathcal{F}(T_2)} \min_{t_1 \in \mathcal{F}(T_1)} d(t_1, t_2)$$

is the maximum distance between a full refinement of T_2 and the set of full refinements of T_1. Therefore, T_1 and T_2 are at Hausdorff distance r of each other if every full refinement of T_1 is within distance r of a full refinement of T_2 and vice-versa.

Aggregating phylogenies. Let k be a positive integer and S be a set of taxa. A *profile of length* k (or simply a *profile*, when k is understood from the context) is a mapping $\mathcal{P}$ that assigns to each $i \in [k]$ a phylogenetic tree $\mathcal{P}(i)$ over S. We refer to these trees as *input trees*. A *consensus rule* is a function that maps a profile $\mathcal{P}$ to some phylogenetic tree T over S called a *consensus tree*.

Let d be a distance measure whose domain is the set of phylogenies over S. We extend d to define a distance measure from profiles to phylogenies as $d(T, \mathcal{P}) = \sum_{i=1}^{k} d(T, \mathcal{P}(i))$. A consensus rule is a *median rule* for d if for every profile $\mathcal{P}$ it returns a phylogeny T^* of minimum distance to $\mathcal{P}$; such a T^* is called a *median*. The problem of finding a median for a profile with respect to a distance measure d is referred to as the *median problem* (relative d), or as the *aggregation problem*.

4 Expected parametric triplet and quartet distances

We now consider the expected value of parametric triplet and quartet distances. Let $u(n)$ and $r(n)$ denote the probabilities that a given quartet is, respectively, unresolved or resolved in an unrooted phylogeny chosen uniformly at random from $P(n)$; thus, $u(n) = 1 - r(n)$. The following are the two main results of this section.

Theorem 4.1. *Let T_1 and T_2 be two unrooted phylogenies chosen uniformly at random with replacement from $P(n)$. Then,*

$$E(d^{(p)}(T_1, T_2)) = \binom{n}{4} \cdot \left(\frac{2}{3} \cdot r(n)^2 + 2 \cdot p \cdot r(n) \cdot u(n) \right). \tag{3}$$

Theorem 4.2. *Let T_1 and T_2 be two rooted phylogenies chosen uniformly at random with replacement from $RP(n)$. Then,*

$$E(d^{(p)}(T_1, T_2)) = \binom{n}{3} \cdot \left(\frac{2}{3} \cdot r(n+1)^2 + 2 \cdot p \cdot r(n+1) \cdot u(n+1) \right). \tag{4}$$

(Note that, in Theorem 4.2, the quantities $r(n+1)$ and $u(n+1)$ refer to *unrooted* trees on $n+1$ leaves, while the theorem itself refers to *rooted* trees on n leaves.)

It is known [45, 44] that

$$u(n) \sim \sqrt{\frac{\pi(2\ln 2 - 1)}{4n}}. \tag{5}$$

Together with Theorems 4.1 and 4.2, this implies that $E(d^{(p)}(T_1, T_2))$ is asymptotically $\frac{2}{3} \cdot \binom{n}{4}$ for unrooted trees and $\frac{2}{3} \cdot \binom{n}{3}$ for rooted trees.

The proof of Theorem 4.1 follows directly from the work of Day [19]; hence, it is omitted (however, we should note that the proof is similar to that of Lemma 4.1 below). Theorem 4.2 extends the result of Critchlow et al. [18] to unresolved trees, and the remainder of this section is devoted to its proof.

We need some notation. Let $u'(n)$ and $r'(n)$ denote the probabilities that a given triplet is, respectively, unresolved or resolved in an rooted phylogeny chosen at random from $RP(n)$.

Lemma 4.1. *Let T_1 and T_2 be two rooted phylogenies chosen uniformly at random with replacement from $RP(n)$. Then,*

$$E(d^{(p)}(T_1, T_2)) = \binom{n}{3} \cdot \left(\frac{2}{3} \cdot r'(n)^2 + 2 \cdot p \cdot r'(n) \cdot u'(n) \right). \tag{6}$$

Proof. By the definition of $d^{(p)}$ and the linearity of expectation, it suffices to establish the equalities below.

$$E(\mathcal{D}(T_1, T_2)) = \binom{n}{3} \cdot \frac{2}{3} \cdot r'(n)^2 \tag{7}$$

$$E(\mathcal{R}_1(T_1, T_2)) = E(\mathcal{R}_2(T_1, T_2)) = \binom{n}{3} \cdot r'(n) \cdot u'(n)) \tag{8}$$

Equation (7) follows directly from [18] (Equation (1)); however, for the sake of completeness, we prove its correctness. Consider a triplet X. The probability that X is resolved in T_1 (or T_2) is $r'(n)$. Thus, the probability that X is resolved in both T_1 and T_2 is $r'(n)^2$. There are exactly three different ways in which any given triplet can be resolved. Hence, if α is resolved in both T_1 and T_2, the probability that it is resolved differently in both trees is $\frac{2}{3}$. Thus, the probability of a pre-given triplet being resolved in both T_1 and T_2, but with different types in each, is $\frac{2}{3}r'(n)^2$. By the linearity of expectation and since the total number of triplets from $\mathcal{L}(T_1)$ (and $\mathcal{L}(T_2)$) is $\binom{n}{3}$, $E(\mathcal{D}(T_1, T_2)) = \binom{n}{3} \cdot \frac{2}{3} r'(n)^2$.

To establish Equation (8), we only need to study $E(\mathcal{R}_1(T_1, T_2))$; the expression for $E(\mathcal{R}_2(T_1, T_2))$ follows by symmetry. Consider a triplet X. The probability that X is unresolved in T_1 is $u'(n)$ and the probability that X is resolved in T_2 is $r'(n)$. The expression for $E(\mathcal{R}_1(T_1, T_2))$ now follows by linearity of expectation. □

Let us define the function ADD-LEAF : $RP(n) \to P(n+1)$ as follows. Given a rooted tree $T \in RP(n)$, ADD-LEAF(T) is the unrooted tree constructed from T by (1) adding a leaf node labeled $n+1$ to T by adjoining it to the root node of T and (2) unrooting the resulting tree. The next two lemmas are well known (for proofs, see [45, 26] and [41, p. 20], respectively).

Lemma 4.2. *For all $n \geq 1$, $|RP(n)| = |P(n+1)|$.*

Lemma 4.3. *Function* ADD-LEAF *is a bijection from the set $RP(n)$ to the set $P(n+1)$.*

For any triplet X over $[n]$, we define two functions $g_X \colon RP(n) \to \{0,1\}$ and $f_X \colon P(n+1) \to \{0,1\}$ as follows:

$$g_X(T) = \begin{cases} 1 & \text{if triplet } X \text{ is resolved in tree } T \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$f_X(T) = \begin{cases} 1 & \text{if quartet } X \cup \{n+1\} \text{ is resolved in tree } T \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

We have the following result.

Lemma 4.4. *Let X be any triplet over $[n]$. Consider a tree $T \in RP(n)$, and let $T' =$* ADD-LEAF(T). *Then, $f_X(T') = g_X(T)$.*

Proof. Follows from the observation that triplet X is resolved in T if and only if quartet $X \cup \{n+1\}$ is resolved in T'. □

Lemma 4.5. *For all $n \geq 1$, $r'(n) = r(n+1)$ and $u'(n) = u(n+1)$.*

Proof. Let X be any triplet over $[n]$. By definition, $r(n+1)$ is the probability of any given quartet being resolved in a random unrooted tree in $P(n)$. In particular, $r(n+1)$ is the probability that quartet $X \cup \{n+1\}$ is resolved in a random unrooted tree. Now,

$$\begin{aligned} r(n+1) &= \sum_{T \in P(n+1)} \frac{f_X(T)}{|P(n+1)|} \\ &= \sum_{T \in P(n+1)} \frac{f_X(T)}{|RP(n)|} \\ &= \sum_{T' \in RP(n)} \frac{g_X(T')}{|RP(n)|} \\ &= r'(n), \end{aligned}$$

where the first and last equalities follow from the definitions of $r(n+1)$ and $r'(n)$, respectively, the second equality follows from Lemma 4.2, and the third follows from Lemma 4.3 and Lemma 4.4.

Since $u'(n) = 1 - r'(n)$ and $u(n+1) = 1 - r(n+1)$, it follows that $u'(n) = u(n+1)$. □

Proof of Theorem 4.2. Simply substitute the expressions for $r'(n)$ and $u'(n)$ given in Lemma 4.5 into the expression for $E(d^{(p)}(T_1, T_2))$ given in Lemma 4.1. □

Critchlow et al. [18] and Steel and Penny [44] derive expressions for the variance of the triplet and quartet distances between two fully resolved trees. It follows from their analysis that, in the case of parametric distances, the variance is $O(p^2n^5)$ and $O(p^2n^7)$, respectively, for triplets and quartets.

5 Properties of parametric distance

In what follows, unless mentioned explicitly, whenever we refer to parametric distance, we mean both its triplet and quartet varieties. We begin with a useful observation.

Proposition 5.1. *For every p, q such that $p, q \in (0, 1]$, $d^{(p)}$ and $d^{(q)}$ are equivalent.*

Proof. Let T_1 and T_2 be two rooted (unrooted) trees. Let M be the number of triplets (quartets) resolved differently in T_1 and let N be the number of triplets (quartets) resolved only in one of T_1 and T_2. Then, $d^{(p)}(T_1, T_2) = M + pN$, and $d^{(q)}(T_1, T_2) = M + qN$. Without loss of generality, let $p \geq q$. Now, if $c_1 = q/p$, then we have $c_1 d^{(q)}(T_1, T_2) = qM/p + q^2N/p \leq M + pN = d^{(p)}(T_1, T_2)$. Similarly, if $c_2 = p/q$, then we have $c_2 d^{(q)}(T_1, T_2) = pM/q + pN \geq M + pN = d^{(p)}(T_1, T_2)$. Thus, $c_1 d^{(q)}(T_1, T_2) \leq d^{(p)}(T_1, T_2) \leq c_2 d^{(q)}(T_1, T_2)$, and, consequently, $d^{(p)}$ and $d^{(q)}$ are equivalent. □

The next result precisely characterizes the ranges of p for which $d^{(p)}$ is a metric or near-metric:

Theorem 5.1.

(i) For $p = 0$, $d^{(p)}$ is not a distance measure.

(ii) For $p \in (0, 1/2)$, $d^{(p)}$ is a distance measure and a near-metric; however, $d^{(p)}$ is not a metric.

(iii) For $p \in [1/2, 1]$, $d^{(p)}$ is a metric.

Proof. Our proof is analogous to the proof of the corresponding result for partial rankings given by Fagin et al. [23]. For the sake of completeness, we prove this result formally. For concreteness, we state our arguments in terms of rooted trees and triplets. The extension to unrooted trees and quartets is direct.

To prove (i), consider the three triplet trees, $t_1 = ab|c$, $t_2 = abc$ (i.e., a completely unresolved tree), and $t_3 = ac|b$. Note that $d^{(0)}(t_1, t_2) = 0$, even though $t_1 \neq t_2$. Thus $d^{(0)}$ is not a distance measure. Observe also that $d^{(0)}$ violates the triangle inequality, since $d^{(0)}(t_1, t_2) + d^{(0)}(t_2, t_3) = 2p = 0 < 1 = d^{(0)}(t_1, t_3)$.

To prove (ii), we begin by showing that $d^{(p)}$ is not a metric for $p \in (0, 1/2)$. Consider the same three triplet trees t_1, t_2, and t_3 used in the proof of part (i). Observe that $d^{(p)}(t_1, t_2) = d^{(p)}(t_2, t_3) = p$, and $d^{(p)}(t_1, t_3) = 1$. Thus, $d^{(p)}(t_1, t_3) = 1 > 2p = d^{(p)}(t_1, t_2) + d^{(p)}(t_2, t_3)$, violating the triangle inequality.

On the other hand, it is straightforward to verify that for any $p \in (0, 1/2)$ — as well, indeed, as for any $p \in [1/2, 1]$ — and any trees T_1 and T_2, we have $d^{(p)}(T_1, T_2) \geq 0$, $d^{(p)}(T_1, T_2) = d^{(p)}(T_2, T_1)$, and $d^{(p)}(T_1, T_2) = 0$ if and only if $T_1 = T_2$. Thus, $d^{(p)}$ is a distance measure for $p \in (0, 1/2)$.

To finish the proof of part (ii), observe that Proposition 5.1 implies that, for every $p \in (0, 1/2)$, $d^{(p)}$ is equivalent to $d^{(1/2)}$, which, as we prove in part (iii), is a metric. Fagin et al. [24] have shown that a distance measure is a near metric if and only if it is equivalent to a metric. Therefore, $d^{(p)}$ is a near metric for every $p \in (0, 1/2)$.

We now prove (iii). As mentioned in the proof of part (ii), $d^{(p)}$ is a distance measure for $p \in [1/2, 1]$. To complete the proof, we show that the triangle inequality holds; i.e., $d^{(p)}(T_1, T_3) \leq d^{(p)}(T_1, T_2) + d^{(p)}(T_2, T_3)$ for any three trees T_1, T_2, T_3. Note that for any $i, j \in \{1, 2, 3\}$, we can express $d^{(p)}(T_i, T_j)$ as

$$d^{(p)}(T_i, T_j) = \sum_{\{a,b,c\} \subseteq [n]} d^{(p)}(T_i|\{a, b, c\}, T_j|\{a, b, c\}).$$

That is, the distance between T_i and T_j can be expressed as the sum of parametric distances between all possible triplet trees induced by T_i and T_j. For any $\{a, b, c\} \subseteq [n]$, and each $i \in \{1, 2, 3\}$, let $t_i = T_i|\{a, b, c\}$. It now suffices to show that $d^{(p)}(t_1, t_3) \leq d^{(p)}(t_1, t_2) + d^{(p)}(t_2, t_3)$. If $t_1 = t_3$, then $d^{(p)}(t_1, t_3) = 0 \leq d^{(p)}(t_1, t_2) + d^{(p)}(t_2, t_3)$, since distances are nonegative. If $t_1 \neq t_3$, then $d^{(p)}(t_1, t_3) \leq 1$, while $d^{(p)}(t_1, t_2) + d^{(p)}(t_2, t_3) \geq 2p$. Thus, $d^{(p)}(t_1, t_3) \leq d^{(p)}(t_1, t_2) + d^{(p)}(t_2, t_3)$ if $p \in [1/2, 1]$. □

The following corollary is analogous to an observation regarding aggregation of partial rankings made in [23]).

Corollary 5.1. *For every $p \in (0, 1]$, there is a constant-factor approximation algorithm for finding the median tree of a profile $\mathcal{P}$ relative to parametric triplet (quartet) distance. This algorithm is 2-approximate for $p \in [1/2, 1]$.*

Proof. Our approximation algorithm simply returns tree $T = \mathcal{P}(\ell)$, where

$$\ell = \arg\min_i d^{(p)}(\mathcal{P}(i), \mathcal{P}).$$

Let T^* be a median tree for $\mathcal{P}$. Consider first the case where $p \in [1/2, 1]$. Then, by Theorem 5.1(iii), $d^{(p)}$ is a metric, and, by a standard argument we have that $d^{(p)}(T, \mathcal{P}) \leq 2d^{(p)}(T^*, \mathcal{P})$ (for an example of such a proof, see, e.g., [28, p. 351]). That is, the algorithm is 2-approximate. Now, consider the case where $p \in (0, 1/2]$. Then, by Theorem 5.1(ii), $d^{(p)}$ is a near-metric. This, along with the fact that our algorithm is 2-approximate for $p \in [1/2, 1]$, implies that the same algorithm gives a constant factor approximation for $p \in (0, 1/2)$ □

The next result establishes a threshold for p beyond which a collection of fully resolved trees give enough evidence to produce a fully resolved tree, despite the disagreements among them.

Theorem 5.2. *Let $\mathcal{P}$ be a profile of length k, such that for all $i \in [k]$, tree $\mathcal{P}(i)$ is fully resolved. Then, if $p \geq 2/3$, there exists median tree T for $\mathcal{P}$ relative to $d^{(p)}$ such that T is fully resolved.*

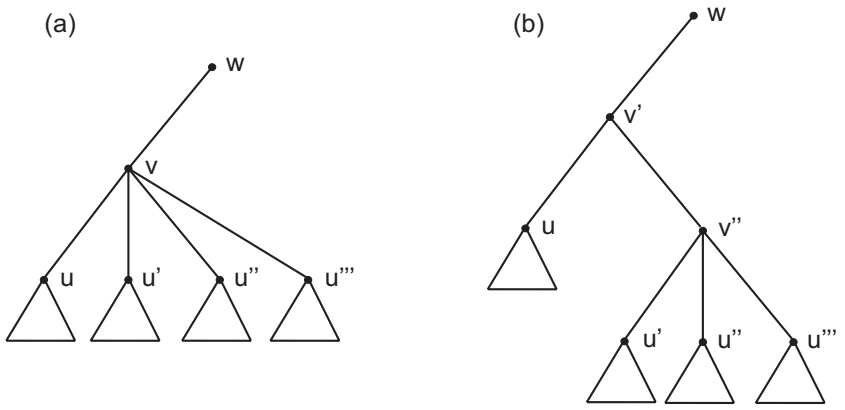

Figure 1: **PULL-OUT.** (a) Original tree. Node w is the parent of v; w may have other neighbors, which are not shown. (b) PULL-OUT(T, u).

It is interesting to compare Theorem 5.2 with analogous results for partial rankings. Consider the variation of Kendall's tau for partial rankings in which a pair of items that is ordered in one ranking but in the same bucket in the other contributes p to the distance, where $p \in [0, 1]$. This distance measure is a metric when $p \geq 1/2$ [23]. Furthermore, if $p \geq 1/2$ the median ranking relative to this distance (that is, the one that minimizes the total distance to the input rankings) is a full ranking if the input consists of full rankings [6]. In contrast, Proposition 5.1 and Theorem 5.2 show that, in the range $p \in [1/2, 2/3)$, parametric triplet or quartet distance are metrics, but the median tree is not guaranteed to be fully resolved even if the input trees are. The intuitive reason is that for rankings there are only two possible outcomes for a comparison between two elements, but there are three ways in which a triplet or quartet may be resolved. This opens up a potentially useful range of values for p wherein parametric triplet/quartet distance is a metric, but where one can adjust for the degree of evidence (or confidence) needed to resolve a node.

Our proof of Theorem 5.2 relies on two lemmas, which make use of the two procedures below.

PULL-OUT(T, u)**:** The arguments are a rooted phylogenetic tree T and a non-root node u in T, whose parent, denoted by v, has 3 or more children. The procedure returns a new tree T' obtained from T as follows. Split v into two nodes v' and v'' such that the parent of v' equals the parent of v, the children of v' are u and v'', and the children of v'' are all the children of v except for u. See Figure 1.

PULL-2-OUT(T, u_1, u_2)**:** The arguments are an unrooted phylogenetic tree T and two nodes u_1, u_2 sharing the same neighbor v whose degree is at least four in T. The procedure returns a new tree T' obtained from T as follows. Split v into two nodes v' and v'' such that the neighbors of v' are v'', u_1, and u_2, the neighbors of v'' are v' and the neighbors of v except for u_1 and u_2. See Figure 2.

In what follows, we write T_i to denote $\mathcal{P}(i)$, the i-th tree in profile $\mathcal{P}$, for $i \in [k]$. We need to introduce separate but analogous concepts for rooted and unrooted trees.

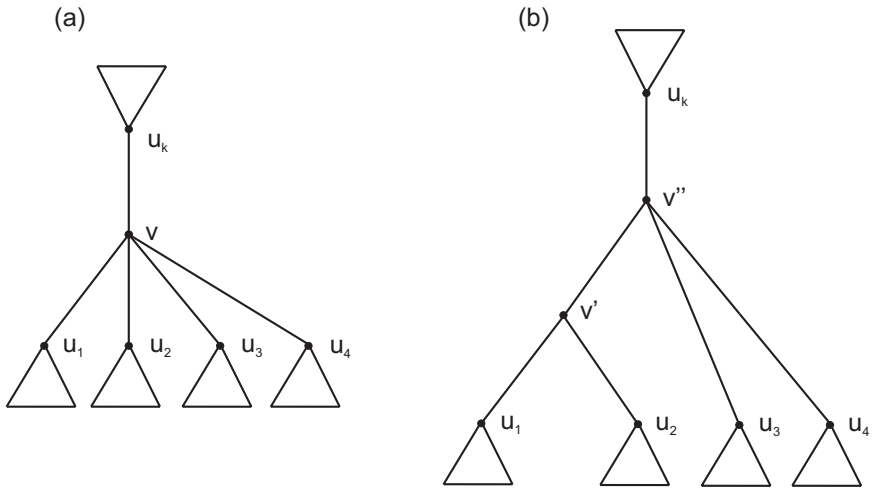

Figure 2: **PULL-2-OUT.** (a) Original tree. (b) PULL-2-OUT(T, u_1, u_2).

Suppose T is a rooted phylogenetic tree and let v be any node in T with at least 3 children, denoted $u_1, u_2, \ldots, u_d$. For $q \in [d]$, let $T^{(q)} = \text{PULL-OUT}(T, u_q)$ and let L_q denote the set of triplets X such that $T|X$ is not fully resolved but $T^{(q)}|X$ is fully resolved. Define the following two quantities.

$$f_q = \sum_{X \in L_q} |\{i \in [k] : T_i|X \text{ agrees with } T^{(q)}|X\}| \tag{11}$$

$$a_q = \sum_{X \in L_q} |\{i \in [k] : T_i|X \text{ disagrees with } T^{(q)}|X\}|. \tag{12}$$

Informally, f_q and a_q are the number of *votes* cast by the trees in profile $\mathcal{P}$ for and against the way the triplets in L_q are resolved in $T^{(q)}$. Indeed, note that, by assumption, every tree in profile $\mathcal{P}$ is fully resolved. Thus, for each triplet $X = \{x, y, z\}$ and every $i \in [k]$, $T_i|X$ must agree with exactly one of $x|yz$, $y|xz$, or $z|xy$. Thus, there are k votes associated with each triplet X, some for, some against.

Now suppose T is an unrooted phylogenetic tree. Let v be any node in phylogeny T and let $u_1, u_2, \ldots, u_d$ be the neighbors of v. For $q, r \in [d]$, let $T^{(qr)} = \text{PULL-2-OUT}(T, u_q, u_r)$ and let L_{qr} denote the set of quartets X such that $T|X$ is not fully resolved but $T^{(qr)}|X$ is fully resolved. Define the following two quantities.

$$f_{qr} = \sum_{X \in L_{qr}} |\{i \in [k] : T_i|X \text{ agrees with } T^{(qr)}|X\}| \tag{13}$$

$$a_{qr} = \sum_{X \in L_{qr}} |\{i \in [k] : T_i|X \text{ disagrees with } T^{(qr)}|X\}|. \tag{14}$$

We have the following result.

Lemma 5.1. *For the rooted case, there exists an index $q \in [d]$ such that $f_q \geq a_q/2$. For the unrooted case, there exists two indices $q, r \in [d]$ such that $f_{qr} \geq a_{qr}/2$.*

Proof. For the rooted case, let $L = \bigcup_{q=1}^{d} L_q$. Thus, L consists of those triplets that are unresolved in T, but resolved in $T^{(q)}$, for some $q \in [d]$. Equivalently, L consists of those triplets whose elements are leaves from three different subtrees of v.

Let $X = \{x, y, z\}$ be a triplet in L. Assume that $x \in \mathcal{L}(T(u_q))$, $y \in \mathcal{L}(T(u_r))$, and $z \in \mathcal{L}(T(u_s))$, where q, r, s must be distinct indices in $[d]$. Then, X is in L_q, L_r, and L_s.

Consider any $i \in [k]$. By assumption, $T_i|X$ is a fully resolved triplet tree. Assume without loss of generality that $T_i|X = x|yz$. Then, $T^{(q)}|X$ agrees with $T_i|X$, so $T_i|X$ contributes $+1$ to f_q. On the other hand, both $T^{(r)}|X$ and $T^{(s)}|X$ disagree with $T_i|X$, so $T_i|X$ contributes $+1$ to a_r and $+1$ to a_s. Furthermore, for any $t \notin \{q, r, s\}$, $T_i|X$ contributes nothing to f_t or a_t, since the triplet tree $T^{(t)}|X$ is not fully resolved. Therefore, we have the following equalities.

$$\sum_{q=1}^{d} a_q = 2k \cdot |L| \tag{15}$$

$$\sum_{q=1}^{d} f_q = k \cdot |L| \tag{16}$$

Now suppose that for all $q \in [d]$, $f_q < a_q/2$. This yields the following contradiction:

$$k \cdot |L| = \sum_{q=1}^{d} f_q < \frac{1}{2} \sum_{q=1}^{d} a_q = k \cdot |L|.$$

Here, the first equality follows from Equation (16) and the last equality follows from Equation (15). Thus, there must be some $q \in [d]$ such that $f_q \geq a_q/2$.

Similarly, for the unrooted case, let $L = \bigcup_{q,r \in [d], q \neq r} L_{qr}$. Thus, L consists of those quartets that are unresolved in T, but resolved in $T^{(qr)}$, for some $q, r \in [d]$, $q \neq r$. Equivalently, L consists of those quartets whose elements are leaves from four different neighboring subtrees of v.

Let $X = \{w, x, y, z\}$ be a quartet in L. Assume that $w \in \mathcal{L}(T(u_q, v))$, $x \in \mathcal{L}(T(u_r, v))$, $y \in \mathcal{L}(T(u_s, v))$, and $z \in \mathcal{L}(T(u_t, v))$, where q, r, s, t must be distinct indices in $[d]$. Then, X is in L_q, L_r, L_s, and L_t.

Consider any $i \in [k]$. By assumption, $T_i|X$ is a fully resolved quartet tree. Assume, without loss of generality, that $T_i|X = wx|yz$. Then, $T^{(qr)}|X$ and $T^{(st)}|X$ agree with $T_i|X$, so $T_i|X$ contributes $+1$ to f_{qr} and f_{st}, respectively. This double contribution is due to the symmetry of quartets. On the other hand, $T^{(qs)}|X$, $T^{(qt)}|X$, $T^{(rs)}|X$, and $T^{(rt)}|X$ disagree with $T_i|X$, so $T_i|X$ contributes $+1$ to a_{qs}, a_{qt}, a_{rs}, and a_{rt}, respectively. Furthermore, if at least one of $t_1, t_2 \notin \{q, r, s, t\}$, then $T_i|X$ contributes nothing to $f_{t_1 t_2}$ or $a_{t_1 t_2}$, since the quartet tree $T^{(t_1 t_2)}|X$ is not fully resolved.

Therefore, similar to the rooted case, we have the following equalities.

$$\sum_{\substack{q,r\in[d]\\ q\neq r}} a_{qr} = 4k \cdot |L| \tag{17}$$

$$\sum_{\substack{q,r\in[d]\\ q\neq r}} f_{qr} = 2k \cdot |L| \tag{18}$$

Now suppose that for all $q, r \in [d]$, $q \neq r$, $f_{qr} < a_{qr}/2$. This yields the following contradiction:

$$2k \cdot |L| = \sum_{\substack{q,r\in[d]\\ q\neq r}} f_{qr} < \frac{1}{2} \sum_{\substack{q,r\in[d]\\ q\neq r}} a_{qr} = 2k \cdot |L|.$$

Here, the first equality follows from Equation (18) and the last equality follows from Equation (17). Thus, there must be some $q, r \in [d]$, $q \neq r$, such that $f_{q,r} \geq a_{qr}/2$. □

Lemma 5.2. *Let $\mathcal{P}$ be a profile for $[k]$ over S consisting entirely of fully-resolved rooted trees or fully resolved unrooted trees. Let T be a phylogeny for S; T is rooted or unrooted according to whether $\mathcal{P}$ consists of rooted or unrooted trees. Suppose T contains an unresolved node v, and suppose $p \geq 2/3$. Then, the following holds.*

(i) If T is rooted, v has a child u such that $d^{(p)}(\widehat{T}, \mathcal{P}) \leq d^{(p)}(T, \mathcal{P})$, where $\widehat{T} = \textsc{Pull-Out}(T, u)$.

(ii) If T is unrooted, v has two neighbors u_q and u_r such that $d^{(p)}(\widehat{T}, \mathcal{P}) \leq d^{(p)}(T, \mathcal{P})$, where $\widehat{T} = \textsc{Pull-2-Out}(T, u_q, u_r)$.

Proof. We will show that in the rooted case, for all $q \in [d]$,

$$d^{(p)}(T^{(q)}, \mathcal{P}) = d^{(p)}(T, \mathcal{P}) - p \cdot f_q + (1-p) \cdot a_q. \tag{19}$$

And, similarly, in the unrooted case, for all $q, r \in [d]$,

$$d^{(p)}(T^{(qr)}, \mathcal{P}) = d^{(p)}(T, \mathcal{P}) - p \cdot f_{qr} + (1-p) \cdot a_{qr}. \tag{20}$$

To verify this, consider any triplet or quartet $X \in L_q$. For every j such that $T^{(q)}|X$ or $T^{(qr)}|X$ is identical to $T_j|X$, the net change in the distance from $\mathcal{P}$ is $-p$, since, for this X, T_j contributes p to the distance to T, but contributes 0 to the distance to $T^{(q)}$ or $T^{(qr)}$. For every j such that $T^{(q)}|X$ or $T^{(qr)}|X$ is different from $T_j|X$, the net change in the distance from $\mathcal{P}$ is $1-p$, since, for this X, T_j contributes p to the distance to T, but contributes $+1$ to the distance to $T^{(q)}$ or $T^{(qr)}$.

Now, for the rooted case, choose a $q^* \in [d]$ such that $f_{q^*} \geq a_{q^*}/2$; for the unrooted case, choose two indices $q^*, r^* \in [d]$, $q^* \neq r^*$, such that $f_{q^*r^*} \geq a_{q^*r^*}/2$. The existence of such a q^* (or q^* and r^*) is guaranteed by Lemma 5.1. Then, Equation (19) and $p \geq 2/3$ imply that $d^{(p)}(T^{(q^*)}, \mathcal{P}) \leq d^{(p)}(T, \mathcal{P})$. Similarly, Equation (20) and $p \geq 2/3$ imply that $d^{(p)}(T^{(q^*r^*)}, \mathcal{P}) \leq d^{(p)}(T, \mathcal{P})$. □

Proof of Theorem 5.2. If $\mathcal{P}$ consists of only fully-resolved trees, then any phylogeny T can be transformed into a fully-resolved tree T' such that $d^{(p)}(T', \mathcal{P}) \leq d^{(p)}(T, \mathcal{P})$ by doing the following. First, let $T' = T$. Next, while T' contains an unresolved node, perform the following three steps:

1. Pick any unresolved node v in T'.
2. If T is rooted, find a child u of v such that $d^{(p)}(\widehat{T}, \mathcal{P}) \leq d^{(p)}(T, \mathcal{P})$, where $\widehat{T} = \textsc{Pull-Out}(T, u)$. If T is unrooted, find two neighbors u_q, u_r of v such that $d^{(p)}(\widehat{T}, \mathcal{P}) \leq d^{(p)}(T, \mathcal{P})$, where $\widehat{T} = \textsc{Pull-2-Out}(T, u_q, u_r)$.
3. Replace T' by $\widehat{T}$.

Note that the existence of a node u such as the one required in Step 2 is guaranteed by Lemma 5.2. Thus, for $p \geq 2/3$, there always exists a fully-resolved median tree relative to $d^{(p)}$. □

The proof of Theorem 5.2 implies that if $p > 2/3$ and the input trees are fully resolved, the median tree relative to $d^{(p)}$ *must* be fully resolved. On the other hand, it is easy to show that when $p \in [1/2, 2/3)$, there are profiles of fully resolved trees whose median tree is only partially resolved.

6 Relationships among the metrics

We do not know whether the Hausdorff triplet or Hausdorff quartet distances are computable in polynomial time. Indeed, we suspect that, unlike their counterparts for partial rankings, this may not be possible. On the positive side, we show here that, in a broad range of cases, it is possible to obtain an approximation to the Hausdorff distance by exploiting its connection with parametric distance. As in the previous section, our results apply to both triplet and quartet distances. Our first result, which is proved later in this section, is as follows.

Lemma 6.1. *For every two phylogenies T_1 and T_2 over the same set of taxa,*

$$d_{\text{Haus}}(T_1, T_2) \geq |\mathcal{D}(T_1, T_2)| + \frac{2}{3} \cdot \max\{|\mathcal{R}_1(T_1, T_2)|, |\mathcal{R}_2(T_1, T_2)|\}.$$

An upper bound on d_{Haus} is obtained by assuming that T_1 and T_2 are refined so that the triplets (quartets) in $\mathcal{R}_1(T_1, T_2)$, $\mathcal{R}_2(T_1, T_2)$, and $\mathcal{U}(T_1, T_2)$ are resolved differently in each refinement. This gives us the following result, which we state without proof.

Lemma 6.2. *For every two phylogenies T_1 and T_2 over the same set of taxa,*

$$d_{\text{Haus}}(T_1, T_2) \leq |\mathcal{D}(T_1, T_2)| + |\mathcal{R}_1(T_1, T_2)| + |\mathcal{R}_2(T_1, T_2)| + |\mathcal{U}(T_1, T_2)|.$$

It is instructive to compare Lemmas 6.1 and 6.2 with the situation for partial rankings. The Hausdorff version of Kendall's tau is obtained by viewing each partial ranking as the set of all possible full rankings that can be obtained by refining it (that is, ordering elements within buckets). The distance is then the Hausdorff distance between the two sets, where the distance between two

elements is the Kendall tau score. Critchlow [17] has given exact bounds on this distance measure, which allow it to be computed efficiently and to establish an equivalence with the parametric version of Kendall's tau defined in Section 5 [23]. To be precise, let L_1 and L_2 be two partial rankings. Re-using notation, let $\mathcal{D}(L_1, L_2)$ be the set of all pairs that are ordered differently in L_1 and L_2, $\mathcal{R}_1(L_1, L_2)$ be the set of pairs that are ordered in L_1 but in the same bucket in L_2, and $\mathcal{R}_2(L_1, L_2)$ be the set of pairs that are ordered in L_2 but in the same bucket in L_1. Then, it can be shown that $d_{\text{Haus}}(L_1, L_2) = |\mathcal{D}(L_1, L_2)| + \max\{|\mathcal{R}_1(L_1, L_2)|, |\mathcal{R}_2(L_1, L_2)|\}$ (see [17, 23]).

It seems unlikely that a similar simple expression can be obtained for Hausdorff triplet or quartet distance. There are at least two reasons for this. Let L_1 and L_2 be partial rankings. Then, it is possible to resolve L_1 so that it disagrees with L_2 in any pair in $\mathcal{R}_2(L_1, L_2)$. Similarly, there is a way to resolve L_2 so that it disagrees with L_1 in any pair in $\mathcal{R}_1(L_1, L_2)$. We have been unable to establish an analog of this property for trees; hence, the $\frac{2}{3}$ factor in Lemma 6.1. The second reason is due to the properties of the set $\mathcal{U}(L_1, L_2)$. It can be shown that one can refine rankings L_1 and L_2 in such a way that pairs of elements that are unresolved in both rankings are resolved the same way in the refinements. This seems impossible to do, in general, for trees and leads to the presence of $|\mathcal{U}(T_1, T_2)|$ in Lemma 6.2.

The above observations prevent us from establishing equivalence between d_{Haus} and $d^{(p)}$, although they do not disprove equivalence either. In any event, the next result shows that when the number of triplets (quartets) that are unresolved in both trees is suitably small, equivalence *does* hold.

Theorem 6.1. *Let β be a positive real number. Then, for every $p \in (0, 1]$, Hausdorff distance and parametric distance are equivalent when restricted to pairs of trees (T_1, T_2) such that $|\mathcal{U}(T_1, T_2)| \leq \beta(|\mathcal{D}(T_1, T_2)| + |\mathcal{R}_1(T_1, T_2)| + |\mathcal{R}_2(T_1, T_2)|)$.*

Proof. By Proposition 5.1, it suffices to show that d_{Haus} is equivalent to $d^{(2/3)}$. Lemma 6.1 shows that $d^{(2/3)}(T_1, T_2) \leq d_{\text{Haus}}(T_1, T_2)$. Thus, we only need to show that, under our assumption about $|\mathcal{U}(T_1, T_2)|$, there is some c such that $d_{\text{Haus}}(T_1, T_2) \leq c \cdot d^{(2/3)}(T_1, T_2)$. The reader can verify that the result follows by choosing $c = 3(1 + \beta)$ and invoking Lemma 6.2. □

The remainder of this section is devoted to the proof of Lemma 6.1. The argument proceeds in two steps. First, we show that T_1 can be refined so that it disagrees with T_2 in at least two thirds of the triplets (quartets) in $\mathcal{R}_2(T_1, T_2)$. Next, we show the existence of an analogous refinement of T_2. Note that the triplets (quartets) in $\mathcal{D}(T_1, T_2)$ are resolved differently in any refinements of T_1 and T_2. This gives lower bounds for both arguments in the outer $\max$ of the definition of $d_{\text{Haus}}(T_1, T_2)$ (Equation 2) and yields the lemma.

Let v be a node in T_1. If T_1 is rooted, then, as in Section 5, let $u_1, \ldots, u_d$ denote the children of v in T_1 and $T_1^{(q)}$ denote PULL-OUT(T, u_q). Define $\mathcal{M}_q(v)$ to be the set of all triplets $X \in \mathcal{R}_2(T_1, T_2)$ such that (i) the lca of X in T_1 is v and (ii) $T_1|X$ is unresolved but $T_1^{(q)}|X$ is fully resolved. Let $\mathcal{M}(v) = \bigcup_{q=1}^{d} \mathcal{M}_q(v)$. Thus, $\mathcal{M}(v)$ is the set of triplets associated with v that are resolved in T_2 but not in T_1.

If T_1 is unrooted, $u_1, \ldots, u_d$ denote the neighbors of v in T_1 and $T_1^{(qr)}$ denotes PULL-2-OUT(T_1, u_{qr}), where PULL-2-OUT is the function defined in Section 5. Define $\mathcal{M}_{qr}(v)$ to be the set of all quartets $X \in \mathcal{R}_2(T_1, T_2)$ such that (i) $T_1|X$ is a fan, (ii) the paths between any two distinct

pairs of taxa in X meet at v, and (iii) $T_1|X$ is unresolved but $T_1^{(qr)}|X$ is fully resolved. Let $\mathcal{M}(v) = \bigcup_{q,r\in[d],q\neq r} \mathcal{M}_{qr}(v)$. Thus, $\mathcal{M}(v)$ is the set of quartets associated with v that are resolved in T_2 but not in T_1.

Define the following two sets for the rooted case.

$$F_q = \{X \in \mathcal{M}_q(v) : T_2|X \text{ agrees with } T_1^{(q)}|X\} \quad (21)$$

$$A_q = \{X \in \mathcal{M}_q(v) : T_2|X \text{ disagrees with } T_1^{(q)}|X\}. \quad (22)$$

Define the following two sets for the unrooted case.

$$F_{qr} = \{X \in \mathcal{M}_{qr}(v) : T_2|X \text{ agrees with } T_1^{(qr)}|X\} \quad (23)$$

$$A_{qr} = \{X \in \mathcal{M}_{qr}(v) : T_2|X \text{ disagrees with } T_1^{(qr)}|X\}. \quad (24)$$

The next result is, in a sense, a counterpart to Lemma 5.1.

Lemma 6.3. *For the rooted case, there exists an index $q \in [d]$ such that $|A_q| \geq 2|F_q|$. For the unrooted case, there exist two indices $q, r \in [d]$, $q \neq r$, such that $|A_{qr}| \geq 2|F_{qr}|$.*

Proof. We start with the rooted case. Consider any triplet $X = \{x, y, z\}$ in $\mathcal{M}(v)$. Assume that $x \in \mathcal{L}(T_1(u_q))$, $y \in \mathcal{L}(T_1(u_r))$, and $z \in \mathcal{L}(T_1(u_s))$, where q, r, s must be distinct indices in $[d]$. Thus, X is in $\mathcal{M}_q(v)$, $\mathcal{M}_r(v)$, and $\mathcal{M}_s(v)$.

By definition of $\mathcal{M}(v)$, $T_2|X$ is a fully resolved triplet tree. Assume that $T_2|X = x|yz$. Then, $T_1^{(q)}|X$ agrees with $T_2|X$, so X contributes exactly one element to F_q. On the other hand, both $T_1^{(r)}|X$ and $T_1^{(s)}|X$ disagree with $T_2|X$, so X contributes exactly one element to A_r and one element to A_s. Furthermore, for any $t \notin \{q, r, s\}$, X contributes nothing to F_t or A_t, since the triplet tree $T_1^{(t)}|X$ is not fully resolved. Therefore, we have that

$$\sum_{q=1}^{d} |A_q| = 2 \cdot |\mathcal{M}(v)| \qquad \text{and} \qquad \sum_{q=1}^{d} |F_q| = |\mathcal{M}(v)|. \quad (25)$$

Assume that for all $q \in [d]$, $|F_q| > |A_q|/2$. This and (25) imply that

$$|\mathcal{M}(v)| = \sum_{q=1}^{d} |F_q| > \frac{1}{2} \sum_{q=1}^{d} |A_q| = |\mathcal{M}(v)|,$$

a contradiction.

We now consider the unrooted case. Consider any quartet $X = \{w, x, y, z\}$ in $\mathcal{M}(v)$. Assume that $w \in \mathcal{L}(T_1(u_q, v))$, $x \in \mathcal{L}(T_1(u_r, v))$, $y \in \mathcal{L}(T_1(u_s, v))$, and $z \in \mathcal{L}(T_1(u_t, v))$, where q, r, s, t must be distinct indices in $[d]$. Thus, X is in $\mathcal{M}_{qr}(v)$, $\mathcal{M}_{qs}(v)$, $\mathcal{M}_{qt}(v)$, $\mathcal{M}_{rs}(v)$, $\mathcal{M}_{rt}(v)$ and $\mathcal{M}_{st}(v)$.

By definition of $\mathcal{M}(v)$, $T_2|X$ is a fully resolved quartet tree. Assume that $T_2|X = wx|yz$. Then, $T_1^{(qr)}|X$ and $T_1^{(st)}|X$ agree with $T_2|X$, so X contributes exactly one element to F_{qr} and F_{st}. On the other hand, $T_1^{(qs)}|X$, $T_1^{(qt)}|X$, $T_1^{(rs)}|X$ and $T_1^{(rt)}|X$ disagree with $T_2|X$, so X contributes

exactly one element to A_{qs}, A_{qt}, A_{rs} and A_{rt}, respectively. Furthermore, for any j_1 and $j_2 \notin \{q, r, s, t\}$, X contributes nothing to $F_{j_1 j_2}$ or $A_{j_1 j_2}$, since the quartet tree $T_1^{(j_1 j_2)}|X$ is not fully resolved. Therefore, we have that

$$\sum_{\substack{q,r\in[d]\\ q\neq r}} |A_{qr}| = 4 \cdot |\mathcal{M}(v)| \qquad \text{and} \qquad \sum_{\substack{q,r\in[d]\\ q\neq r}} |F_{qr}| = 2 \cdot |\mathcal{M}(v)|. \tag{26}$$

Assume that for all $q, r \in [d]$, $|F_{qr}| > |A_{qr}|/2$. This and (26) imply that

$$2 \cdot |\mathcal{M}(v)| = \sum_{\substack{q,r\in[d]\\ q\neq r}} |F_{qr}| > \frac{1}{2} \sum_{\substack{q,r\in[d]\\ q\neq r}} |A_{qr}| = 2 \cdot |\mathcal{M}(v)|,$$

a contradiction. □

Proof of Lemma 6.1. Define the following functions. For any two phylogenies T_1, T_2 over S, let

$$d_{H1}(T_1, T_2) = \max_{t_1 \in \mathcal{F}(T_1)} \min_{t_2 \in \mathcal{F}(T_2)} d(t_1, t_2), \tag{27}$$

$$d_{H2}(T_1, T_2) = \max_{t_2 \in \mathcal{F}(T_2)} \min_{t_1 \in \mathcal{F}(T_1)} d(t_1, t_2). \tag{28}$$

We show that

$$d_{H1}(T_1, T_2) \geq |\mathcal{D}(T_1, T_2)| + \frac{2}{3} \cdot |\mathcal{R}_2(T_1, T_2)| \tag{29}$$

$$d_{H2}(T_1, T_2) \geq |\mathcal{D}(T_1, T_2)| + \frac{2}{3} \cdot |\mathcal{R}_1(T_1, T_2)|. \tag{30}$$

Since $d_{\text{Haus}}(T_1, T_2) = \max\{d_{H1}(T_1, T_2), d_{H2}(T_1, T_2)\}$, this proves Lemma 6.1.

By symmetry, it suffices to prove Inequality (29). Our argument relies on two observations. First, note that if T_1' is a refinement of T_1 (but possibly not a full refinement), then, $d_{H1}(T_1, T_2) \geq d_{H1}(T_1', T_2)$. This holds because $\mathcal{F}(T_1') \subseteq \mathcal{F}(T_1)$. Second, for any two phylogenies T_1 and T_2, $d_{H1}(T_1, T_2) \geq |\mathcal{D}(T_1, T_2)|$. This holds because for any $t_1 \in \mathcal{F}(T_1)$, $t_2 \in \mathcal{F}(T_2)$, we have that $\mathcal{D}(T_1, T_2) \subseteq \mathcal{D}(t_1, t_2)$, and (by definition) $d(t_1, t_2) = |\mathcal{D}(t_1, t_2)|$.

By the preceding observations, if we prove that it is possible to construct a refinement T_1' of T_1 such that $|\mathcal{D}(T_1', T_2)| \geq |\mathcal{D}(T_1, T_2)| + \frac{2}{3}|\mathcal{R}_2(T_1, T_2)|$, then Inequality (29) follows. The idea is to find a refinement T_1' of T_1 such that for at least two-thirds of the triplets or quartets $X \in \mathcal{R}_2(T_1, T_2)$, we have that $T_1'|X \neq T_2|X$. To obtain the desired refinement of T_1, we initially set $T_1' = T_1$ and then perform the following steps while they apply:

1. Pick an unresolved node v in T_1' such that $\mathcal{M}'(v) \neq \emptyset$, where $\mathcal{M}'(v)$ is the set of triplets (quartets) associated with v that are resolved in T_2 but not in T_1'. In the rooted case, let $u_1, \ldots, u_d$ be the children of v; in the unrooted case, let $u_1, \ldots, u_d$ be the neighbors of v.

2. For rooted trees, find a $q \in [d]$ such that $|A_q| \geq 2|F_q|$ (such a q exists by Lemma 6.3). For unrooted trees, find $q, r \in [d]$ such that $|A_{qr}| \geq 2|F_{qr}|$ (such q, r exist by Lemma 6.3).

3. In the rooted case, set $T_1' = \text{PULL-OUT}(T_1', u_q)$; in the unrooted case, set $T_1' = \text{PULL-2-OUT}(T_1', u_q, u_r)$.

When this algorithm terminates, $\mathcal{M}'(v) = \emptyset$ for every $v \in \mathcal{V}(T_1')$. Thus, $\mathcal{R}_2(T_1', T_2) = \emptyset$. Furthermore, the choice of q (or q_1 and q_2) in step (2) guarantees that $|\mathcal{D}(T_1', T_2)| \geq |\mathcal{D}(T_1, T_2)| + \frac{2}{3} \cdot |\mathcal{R}_2(T_1, T_2)|$. □

7 Computing parametric distance

In this section we discuss the problem of efficiently computing parametric triplet and quartet distances. Efficient algorithms exist for computing traditional quartet distances between-partially resolved unrooted trees (e.g., [14], [46]), and these can be readily used for computing parametric quartet distances as well. However, no such efficient algorithms exist for computing triplet distances. Consequently, we only briefly discuss the problem of computing parametric quartet distance (Section 7.5), and devote the bulk of this section to the problem of efficiently computing parametric triplet distance. In particular, we show that the parametric triplet distance (PTD), $d^{(p)}$, between two phylogenetic trees T_1 and T_2 over the same set of n taxa can be computed in $O(n^2)$ time.

Before we outline our PTD algorithm, we need some notation. Let T be a rooted phylogenetic tree. Then, $R(T)$ denotes the set of all triplets that are resolved in T and $U(T)$ denotes the set of all triplets that are unresolved in T.

The next proposition is easily proved.

Proposition 7.1. *For any two phylogenies T_1, T_2 over the same set of taxa,*

(i) $|\mathcal{R}_1(T_1, T_2)| + |\mathcal{U}(T_1, T_2)| = |U(T_2)|$

(ii) $|\mathcal{R}_2(T_1, T_2)| + |\mathcal{U}(T_1, T_2)| = |U(T_1)|$,

(iii) $|\mathcal{S}(T_1, T_2)| + |\mathcal{D}(T_1, T_2)| + |\mathcal{R}_1(T_1, T_2)| = |R(T_1)|$.

By Prop. 7.1 and Eqn. (1), the parametric distance between T_1 and T_2 can be expressed as

$$d^{(p)}(T_1, T_2) = |R(T_1)| - |\mathcal{S}(T_1, T_2)| + p \cdot (|U(T_1)| - |U(T_2)|) + (2p - 1) \cdot |\mathcal{R}_1(T_1, T_2)|. \quad (31)$$

Our PTD algorithm proceeds as follows. After an initial $O(n^2)$ preprocessing step (Section 7.1), the algorithm computes $|R(T_1)|$, $|U(T_1)|$ and $|U(T_2)|$ using a $O(n)$-time procedure (Section 7.2). Next, it computes $|\mathcal{S}(T_1, T_2)|$ and $|\mathcal{R}_1(T_1, T_2)|$. As described in Sections 7.3 and 7.4, this takes $O(n^2)$ time. Then, it uses these values to compute $d^{(p)}(T_1, T_2)$, in $O(1)$ time, via Equation (31). To summarize, we have the following result.

Theorem 7.1. *The parametric triplet distance $d^{(p)}(T_1, T_2)$ for two rooted phylogenetic trees T_1 and T_2 over the same set of n taxa can be computed in $O(n^2)$ time.*

In the rest of this section we use the following notation. We write $rt(T)$ to denote the root node of a tree T. Let v be a node in T. Then, $pa(v)$ denotes the parent of v in T and $Ch(v)$ is the set of children of v. We write $\overline{T(v)}$ to denote the tree obtained by deleting $T(v)$ from T, as well as the edge from v to its parent, if such an edge exists.

7.1 The preprocessing step

The purpose of the preprocessing step is to calculate and store the following four quantities for every pair (u, v), where $u \in \mathcal{V}(T_1)$ and $v \in \mathcal{V}(T_2)$: $|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|$, $|\mathcal{L}(T_1(u)) \cap \mathcal{L}(\overline{T_2(v)})|$, $|\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(T_2(v))|$, and $|\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(\overline{T_2(v)})|$. These values are stored in a table so that any value can be accessed in $O(1)$ time by subsequent steps of the PTD algorithm.

Lemma 7.1. *The values $|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|$, $|\mathcal{L}(T_1(u)) \cap \mathcal{L}(\overline{T_2(v)})|$, $|\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(T_2(v))|$, and $|\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(\overline{T_2(v)})|$ can be collectively computed for every pair of nodes (u, v), where $u \in \mathcal{V}(T_1)$ and $v \in \mathcal{V}(T_2)$, in $O(n^2)$ time.*

Proof. We first observe that for each $u \in \mathcal{V}(T_1)$, the value $|\mathcal{L}(T_1(u))|$ can be computed in $O(n)$ time by a simple post order traversal of T_1. The same holds for tree T_2.

Consider the value $|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|$. We consider three cases.

1. If u and v are both leaf nodes then computing $|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|$ is trivial.

2. If u is a leaf node, but v is not a leaf node, then
$$|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))| = \sum_{x \in Ch(v)} |\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(x))|.$$

3. If u is not a leaf node, then
$$|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))| = \sum_{x \in Ch(u)} |\mathcal{L}(T_1(x)) \cap \mathcal{L}(T_2(v))|.$$

We compute the value $|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|$, for every pair (u, v), using an interleaved post order traversal of T_1 and T_2. This traversal works as follows: For each node u in a post order traversal of T_1, we consider each node v in a post order traversal of T_2. This ensures that when the intersection sizes for a pair of nodes is computed, the set intersection sizes for all pairs of their children have already been computed. The total time complexity for computing the required values in this way can be bounded as follows. For a pair of nodes u and v from T_1 and T_2 respectively, the value $|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|$ can be computed in $O(|\mathit{Ch}(u)| + |\mathit{Ch}(v)|)$ time and all the remaining three set intersection values in $O(1)$ time. Summing this over all possible pairs of edges, we get a total time of $O(\sum_{u \in \mathcal{V}(T_1)} \sum_{v \in \mathcal{V}(T_2)} |\mathit{Ch}(u)| + |\mathit{Ch}(v)|)$, which is $O(n^2)$.

Once the value $|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|$ has been computed for every pair (u, v), the remaining quantities we seek can be computed using the following relations.

$$|\mathcal{L}(T_1(u)) \cap \mathcal{L}(\overline{T_2(v)})| = |\mathcal{L}(T_1(u))| - |\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|,$$
$$|\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(T_2(v))| = |\mathcal{L}(T_2(v))| - |\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|, \qquad \text{and}$$
$$|\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(\overline{T_2(v)})| = n - (|\mathcal{L}(T_1(u))| + |\mathcal{L}(T_2(v))| - |\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|).$$

Thus, each of these values can be computed in $O(1)$ time, for a total of $O(n^2)$. □

We store these $O(n^2)$ values in an array indexed by u and v, for each $u \in \mathcal{V}(T_1)$ and $v \in \mathcal{V}(T_2)$. This enables constant time insertion and look-up of any stored value, when the two relevant nodes are given.

7.2 Computing $|R(T_1)|$, $|U(T_1)|$, and $|U(T_2)|$

Here we prove the following result.

Lemma 7.2. *Given a rooted phylogenetic tree T over n leaves, the values $|R(T)|$ and $|U(T)|$ can be computed in $O(n)$ time.*

Thus, $|R(T_1)|$, $|U(T_1)|$ and $|U(T_2)|$ can all be computed in $O(n)$ time.

To prove Lemma 7.2, we need some terminology and an auxiliary result. Let $e = (v, pa(v))$ be any internal edge in T. Consider any two leaves x, y from $\mathcal{L}(T(v))$, and any leaf z from $\mathcal{L}(\overline{T(v)})$. Then, the triplet $\{x, y, z\}$ must appear resolved as $xy|z$ in T; we say that the triplet tree $xy|z$ is *induced* by the edge $(v, pa(v))$. Note that the same resolved triplet tree may be induced by multiple edges in T. We say that the triplet tree $xy|z$ is *strictly induced* by the edge $\{v, pa(v)\}$ if $xy|z$ is induced by $(v, pa(v))$ and, additionally, $x \in \mathcal{L}(T(v_1))$ and $y \in \mathcal{L}(T(v_2))$ for some $v_1, v_2 \in Ch(v)$ such that $v_1 \neq v_2$. See Figure 3 for an example.

Lemma 7.3. *Given a tree T and a triplet X, if $T|X$ is fully resolved then $T|X$ is strictly induced by exactly one edge in T.*

Proof. Let $X = \{a, b, c\}$. Without loss of generality, assume that $T|X = ab|c$. If v denotes the lca of a and b in T, the edge $\{v, pa(v)\}$ must induce $ab|c$. Moreover, v must be the only node in T for which there exist nodes $v_1, v_2 \in Ch(v)$ such that $a \in \mathcal{L}(T(v_1))$ and $b \in \mathcal{L}(T(v_2))$. Thus, there is exactly one edge in T that strictly induces $T|X$. □

Proof of Lemma 7.2. Since $|R(T)| + |U(T)| = \binom{n}{3}$, given $|R(T)|$, the value $|U(T)|$ can be computed in $O(1)$ additional time. Thus, we only need to show that the value of $|R(T)|$ can be computed in $O(n)$ time.

The first step is to traverse the tree T in post order to compute the values $\alpha_v = |\mathcal{L}(T(v))|$ and $\beta_v = n - \alpha_v$ at each node $v \in \mathcal{V}(T)$. This takes $O(n)$ time.

For any $v \in \mathcal{V}(T) \setminus \{rt(T)\}$, let $\phi(v)$ denote the number of triplets that are strictly induced by the edge $\{v, pa(v)\}$ in tree T. Observe that any triplet that is strictly induced by an edge in T must be fully resolved in T. Thus, Lemma 7.3 implies that the sum of $\phi(v)$ over all internal nodes $v \in \mathcal{V}(T) \setminus \{rt(T)\}$ yields the value $|R(T)|$. We now show how to compute the value of $\phi(v)$.

Let $X = \{a, b, c\}$ be a triplet that is counted in $\phi(v)$. And, without loss of generality, let $T_1|X = ab|c$. It can be verified that X must satisfy the following two conditions: (i) $a, b \in \mathcal{L}(T(v))$ and $c \in \mathcal{L}(\overline{T(v)})$, and (ii) there does not exist any $x \in Ch(v)$ such that $a, b \in \mathcal{L}(T(x))$. The number of triplets that satisfy condition (i) is $\binom{\alpha_v}{2} \cdot \beta_v$, and the number of triplets that satisfy condition (i), but not condition (ii) is exactly $\sum_{x \in Ch(v)} \binom{\alpha_x}{2} \cdot \beta_v$. Thus, $\phi(v) = \gamma_v - \sum_{x \in Ch(v)} \binom{\alpha_x}{2} \cdot \beta_v$.

Computing $\phi(v)$ requires $O(|Ch(v)|)$ time; hence, the time complexity for computing $|R(T)|$ is $O(\sum_{v \in \mathcal{V}(T)} |Ch(v)|)$, which is $O(n)$. □

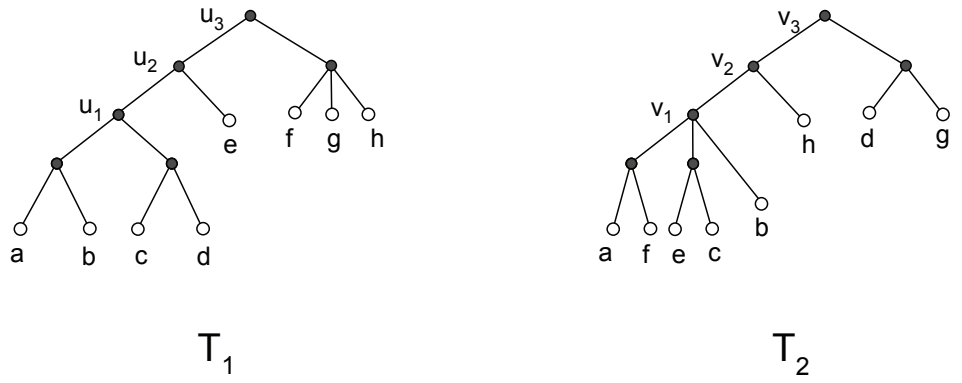

Figure 3: **Counting shared triplet trees.** Consider the triplet tree $X = ac|g$. In T_1, X is induced by the edges $\{u_1, u_2\}$ and $\{u_2, u_3\}$, and strictly induced by the edge $\{u_1, u_2\}$. The triplet tree X also exists in tree T_2, where it is strictly induced by the edge $\{v_1, v_2\}$. Thus, X will be counted in the term $s(u_1, v_1)$. Additionally, the term $s(u_1, v_1)$ will also count triplet trees $ac|h$, $bc|g$, and $bc|h$. Thus, $s(u_1, v_1) = 4$ in this example.

7.3 Computing $|\mathcal{S}(T_1, T_2)|$

We now describe an $O(n^2)$ time algorithm to compute the size of the set $\mathcal{S}(T_1, T_2)$ of shared triplets; that is, triplets that are fully and identically resolved in T_1 and T_2.

For any $u \in \mathcal{V}(T_1) \setminus (rt(T_1) \cup \mathcal{L}(T_1))$ and $v \in \mathcal{V}(T_2) \setminus (rt(T_2) \cup \mathcal{L}(T_2))$, let $s(u, v)$ denote the number of identical triplet trees strictly induced by edge $\{u, pa(u)\}$ in T_1 and edge $\{v, pa(v)\}$ in T_2. This is illustrated in Figure 3. We have the following result.

Lemma 7.4. *Given T_1 and T_2, we have,*

$$|\mathcal{S}(T_1, T_2)| = \sum_{\substack{u \in \mathcal{V}(T_1) \setminus (rt(T_1) \cup \mathcal{L}(T_1)), \\ v \in \mathcal{V}(T_2) \setminus (rt(T_2) \cup \mathcal{L}(T_2))}} s(u, v). \tag{32}$$

Proof. Consider any triplet $X \in \mathcal{S}(T_1, T_2)$. Since $T_1|X$ is fully resolved and $T_1|X = T_2|X$ then, by Lemma 7.3, there exists exactly one node $u \in \mathcal{V}(T_1) \setminus rt(T_1)$ and one node $v \in \mathcal{V}(T_2) \setminus rt(T_2)$ such that the edge $\{u, pa(u)\}$ strictly induces $T_1|X$ in T_1, and edge $\{v, pa(v)\}$ strictly induces $T_2|X$ in T_2. Additionally, neither u nor v can be leaf nodes in T_1 and T_2 respectively. Thus, X would be counted exactly once in the right-hand side of Equation (32) in the value $s(u, v)$. Moreover, by the definition of $s(u, v)$, any triplet tree that is counted on the right-hand side of Equation (32) algorithm must belong to the set $\mathcal{S}(T_1, T_2)$. The Lemma follows. □

The following lemma shows how to compute the value of $s(u, v)$ using the values computed in the preprocessing step.

Lemma 7.5. *Given any $u \in \mathcal{V}(T_1) \setminus (rt(T_1) \cup \mathcal{L}(T_1))$ and $v \in \mathcal{V}(T_2) \setminus (rt(T_2) \cup \mathcal{L}(T_2))$, $s(u, v)$ can be computed in $O(|\,Ch(u)| \cdot |\,Ch(v)|)$ time.*

procedure $\mathcal{S}(T_1, T_2)$
1: **for** each internal node $u \in \mathcal{V}(T_1) \setminus rt(T_1)$ **do**
2: **for** each internal node $v \in \mathcal{V}(T_2) \setminus rt(T_2)$ **do**
3: Compute $s(u, v)$.
4: **return** the sum of all computed $s(\cdot, \cdot)$.

Figure 4: Computing $|\mathcal{S}(T_1, T_2)|$

Proof. We will show that $s(u, v) = n_1(u, v) - n_2(u, v) - n_3(u, v) + n_4(u, v)$, where

$$n_1(u,v) = \binom{|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))|}{2} \cdot |\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(\overline{T_2(v)})|,$$

$$n_2(u,v) = \sum_{x \in Ch(u)} \binom{|\mathcal{L}(T_1(x)) \cap \mathcal{L}(T_2(v))|}{2} \cdot |\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(\overline{T_2(v)})|,$$

$$n_3(u,v) = \sum_{x \in Ch(v)} \binom{|\mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(x))|}{2} \cdot |\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(\overline{T_2(v)})|, \quad \text{and}$$

$$n_4(u,v) = \sum_{x \in Ch(u)} \sum_{y \in Ch(v)} \binom{|\mathcal{L}(T_1(x)) \cap \mathcal{L}(T_2(y))|}{2} \cdot |\mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(\overline{T_2(v)})|.$$

Consider any triplet tree, $ab|c$, counted in $s(u, v)$. It can be verified that $ab|c$ must satisfy the following three conditions: (i) $a, b \in \mathcal{L}(T_1(u)) \cap \mathcal{L}(T_2(v))$ and $c \in \mathcal{L}(\overline{T_1(u)}) \cap \mathcal{L}(\overline{T_2(v)})$, (ii) there does not exist any $x \in Ch(u)$ such that $a, b \in \mathcal{L}(T_1(x))$, and (iii) there does not exist any $x \in Ch(v)$ such that $a, b \in \mathcal{L}(T_2(x))$. Moreover, observe that any triplet tree $ab|c$ that satisfies these three conditions is counted in $s(u, v)$. Therefore, $s(u, v)$ is exactly the number of triplets trees that satisfy all three conditions (i), (ii) and (iii).

The number of triplet trees that satisfy condition (i) is given by $n_1(u, v)$. Some of the triplet trees that satisfy condition (i) may not satisfy conditions (ii) or (iii); these must not be counted in $s(u, v)$. The value $n_2(u, v)$ is exactly the number of triplet trees that satisfy condition (i) but not condition (ii). Similarly, $n_3(u, v)$ is exactly the number of triplet trees that satisfy condition (i) but not (iii). Thus, the second and third terms must be subtracted from the first term. However, there may be triplet trees that satisfy condition (i) but neither (ii) nor (iii), and, consequently, get subtracted in both the second and third terms. In order to adjust for these, the value $n_4(u, v)$ counts exactly those triplet trees that satisfy condition (i) but not (ii) and (iii). □

A summary of our algorithm to compute $|\mathcal{S}(T_1, T_2)|$ appears in Figure 4.

Lemma 7.6. *Given two rooted phylogenetic trees T_1 and T_2 on the same n leaves, the value $|\mathcal{S}(T_1, T_2)|$ can be computed in $O(n^2)$ time.*

Proof. By Lemma 7.4, the algorithm of Figure 4 computes the value $|\mathcal{S}(T_1, T_2)|$ correctly. We now analyze its complexity. The running time of the algorithm is dominated by the complexity of computing the value $s(u, v)$ for each pair of internal nodes $u \in \mathcal{V}(T_1)$ and $v \in \mathcal{V}(T_2)$. According

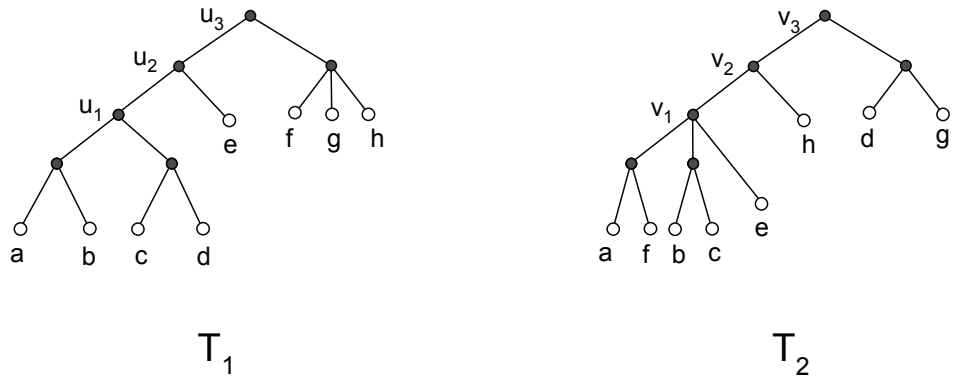

Figure 5: **Counting triplets that are resolved in T_1 and unresolved in T_2.** Consider the triplet $X = \{a, c, e\}$. In tree T_1, $T_1|X$ is strictly induced by the edge $\{u_1, u_2\}$. In tree T_2, X is associated with the node v_1. Thus, X will be counted in the term $r_1(u_1, v_1)$. In this example, the term $r_1(u_1, v_1)$ will not count any other triplets and thus $r_1(u_1, v_1) = 1$.

to Lemma 7.5, the value $s(u, v)$ can be computed in $O(|\,\mathit{Ch}(u)| \cdot |\,\mathit{Ch}(v)|)$ time. Thus, the total time complexity of the algorithm is $O(\sum_{u \in \mathcal{V}(T_1)} \sum_{v \in \mathcal{V}(T_2)} |\,\mathit{Ch}(u)| \cdot |\,\mathit{Ch}(v)|)$, which is $O(n^2)$. □

7.4 Computing $|\mathcal{R}_1(T_1, T_2)|$

Next, we describe an $O(n^2)$-time algorithm that computes the cardinality of the set $\mathcal{R}_1(T_1, T_2)$ of triplets that are resolved only in tree T_1. First, we need a definition. Let X be a triplet that is unresolved in T_2. Let v be the least common ancestor (lca) of X in T_2. We say that X is *associated* with v. Observe that node v must be internal and unresolved. Note also that X is associated with exactly one node in T_2.

For any $u \in \mathcal{V}(T_1) \setminus (\mathit{rt}(T_1) \cup \mathcal{L}(T_1))$ and $v \in \mathcal{V}(T_2) \setminus \mathcal{L}(T_1))$, let $r_1(u, v)$ denote the number of triplets X such that $T_1|X$ is strictly induced by edge $\{u, \mathit{pa}(u)\}$ in T_1, and X is associated with the node v in T_2. See Figure 5 for an example.

The triplets counted in $r_1(u, v)$ must be resolved in T_1 but unresolved in T_2. Our algorithm computes the value $|\mathcal{R}_1(T_1, T_2)|$ by computing, for each $u \in \mathcal{V}(T_1) \setminus (\mathit{rt}(T_1) \cup \mathcal{L}(T_1))$ and $v \in \mathcal{V}(T_2) \setminus \mathcal{L}(T_2)$, the value $r_1(u, v)$. We claim that the sum of all the computed $r_1(u, v)$'s yields the value $|\mathcal{R}_1(T_1, T_2)|$.

Lemma 7.7. *Given T_1 and T_2, we have,*

$$|\mathcal{R}(T_1, T_2)| = \sum_{\substack{u \in \mathcal{V}(T_1) \setminus (\mathit{rt}(T_1) \cup \mathcal{L}(T_1)), \\ v \in \mathcal{V}(T_2) \setminus \mathcal{L}(T_2)}} r_1(u, v). \tag{33}$$

Proof. Consider any triplet $X \in \mathcal{R}_1(T_1, T_2)$. By Lemma 7.3, there exists exactly one node $u \in \mathcal{V}(T_1) \setminus \mathit{rt}(T_1)$ such that the edge $\{u, \mathit{pa}(u)\}$ strictly induces $T_1|X$ in T_1. Also observe that there must be exactly one unresolved node $v \in \mathcal{V}(T_2)$ with which X is associated. Additionally, neither

u nor v can be leaf nodes in T_1 and T_2 respectively. Thus, X would be counted exactly once in the right-hand side of Equation (33); in the value $r_1(u, v)$. Moreover, by the definition of $r_1(u, v)$, any triplet that is counted in the right-hand side of Equation (33) must belong to the set $\mathcal{R}_1(T_1, T_2)$. The lemma follows. □

Given a path $u_1, u_2, \ldots, u_k$, where $k \geq 2$, in tree T_1 such that u_k is an internal node and u_1 is an ancestor of u_k, let $\gamma(u_1, u_k, v)$ denote the number of triplets X such that $T_1|X$ is induced by every edge $\{u_{i-1}, u_i\}$, for $2 \leq i \leq k$, in T_1 and X is associated with node v in T_2.

The following lemma shows how the value of $r_1(u, v)$ can be computed by first computing certain $\gamma(\cdot, \cdot, \cdot)$ values.

Lemma 7.8. *For any $u \in \mathcal{V}(T_1) \setminus (rt(T_1) \cup \mathcal{L}(T_1))$ and $v \in \mathcal{V}(T_2) \setminus \mathcal{L}(T_2))$,*

$$r_1(u, v) = \gamma(pa(u), u, v) - \sum_{x \in Ch(u)} \gamma(pa(u), x, v).$$

Proof. Let $X = \{a, b, c\}$ be a triplet that is counted in $r_1(u, v)$. And, without loss of generality, let $T_1|X = ab|c$. It can be verified that X must satisfy the following three conditions: (i) X must be associated with v in T_2, (ii) $a, b \in \mathcal{L}(T_1(u))$ and $c \in \mathcal{L}(\overline{T_1(u)})$, and (iii) there must not exist any $x \in Ch(u)$ such that $a, b \in \mathcal{L}(T_1(x))$. Moreover, observe that if there exists a triplet $X = \{a, b, c\}$ that satisfies these three conditions, then X will be counted in $r_1(u, v)$; these three conditions are thus necessary and sufficient.

Now observe that $\gamma(pa(u), u, v)$ counts exactly those triplets that satisfy conditions (i) and (ii), while $\sum_{x \in Ch(u)} \gamma(pa(u), x, v)$ counts exactly those triplets that satisfy conditions (i) and (ii), but not condition (iii). The lemma follows immediately. □

To compute the value of $\gamma(\cdot, \cdot, \cdot)$ efficiently we use the following lemma.

Lemma 7.9. *Consider a path $u_1, u_2, \ldots, u_k$, where $k \geq 2$, in tree T_1 such that u_k is an internal node and u_1 is an ancestor of u_k. And let $v \in \mathcal{V}(T_2)$ be an internal unresolved node. Then,*

$$\gamma(u_1, u_k, v) = n_1(u_1, u_k, v) - n_2(u_1, u_k, v) - n_3(u_1, u_k, v) - n_4(u_1, u_k, v),$$

where

$$n_1(u_1, u_k, v) = \binom{|\mathcal{L}(T_2(v)) \cap \mathcal{L}(T_1(u_k))|}{2} \cdot |\mathcal{L}(T_2(v)) \cap \mathcal{L}(\overline{T_1(u_2)})|,$$

$$n_2(u_1, u_k, v) = \sum_{x \in Ch(v)} \binom{|\mathcal{L}(T_2(x)) \cap \mathcal{L}(T_1(u_k))|}{2} \cdot |\mathcal{L}(T_2(x)) \cap \mathcal{L}(\overline{T_1(u_2)})|,$$

$$n_3(u_1, u_k, v) = \sum_{x \in Ch(v)} \binom{|\mathcal{L}(T_1(u_k)) \cap \mathcal{L}(T_2(x))|}{2} \cdot \left(|\mathcal{L}(T_2(v)) \cap \mathcal{L}(\overline{T_1(u_2)})| - |\mathcal{L}(T_2(x)) \cap \mathcal{L}(\overline{T_1(u_2)})|\right),$$

and

$$n_4(u_1, u_k, v) = \sum_{x \in Ch(v)} |\mathcal{L}(T_2(x)) \cap \mathcal{L}(T_1(u_k))| \cdot |\mathcal{L}(T_2(x)) \cap \mathcal{L}(\overline{T_1(u_2)})| \cdot \big(|\mathcal{L}(T_2(v)) \cap \mathcal{L}(T_1(u_k))| - |\mathcal{L}(T_2(x)) \cap \mathcal{L}(T_1(u_k))|\big).$$

procedure $\mathcal{R}_1(T_1, T_2)$

1: **for** each internal node $u \in \mathcal{V}(T_1) \setminus \{rt(T_1)\}$ **do**
2: **for** each internal unresolved node $v \in \mathcal{V}(T_2)$ **do**
3: Compute $r_1(u, v)$.
4: **return** the sum of all computed $r_1(\cdot, \cdot)$.

Figure 6: Computing $|\mathcal{R}_1(T_1, T_2)|$

Proof. Consider those triplets X for which $T_1|X$ is induced by every edge (u_{i-1}, u_i), for $2 \leq i \leq k$, in T_1, and $T_2|X$ is a subtree of $T_2(v)$. Let us call these triplets *relevant*. Any relevant triplet must have all three leaves from $\mathcal{L}(T_2(v))$, two leaves from $\mathcal{L}(T_1(u_k))$, and the third leaf from $\mathcal{L}(\overline{T_1(u_2)})$. Also note that any triplet that satisfies these three conditions must be relevant. The number of triplets that satisfy these conditions is exactly $n_1(u_1, u_k, v)$.

Any relevant triplet X must belong to one of the following four categories:

1. *The lca of X in T_2 is not node v* : This implies that, in addition to being a relevant triplet, all three leaves of X must belong to the same subtree of T_2 rooted at a child of v. The number of such triplets is $n_2(u_1, u_k, v)$.

2. *The lca of X in T_2 is node v, X is resolved in T_2 and $T_1|X = T_2|X$* : A relevant triplet X satisfies this criterion if and only if there exists a child $x \in Ch(v)$, such that the two leaves of this triplet that belong to $\mathcal{L}(T_1(u_k))$ in tree T_1 also occur in $\mathcal{L}(T_2(x))$, and, the third leaf (which occurs in $\mathcal{L}(\overline{T_1(u_2)})|$ in T_1) occurs in $\mathcal{L}(T_2(y))$ where $y \in Ch(v) \setminus \{x\}$. The number of such X is equal to $n_3(u_1, u_k, v)$.

3. *The lca of X in T_2 is node v, X is resolved in T_2, but $T_1|X \neq T_2|X$* : A relevant triplet X satisfies this criterion if and only if there exists a child $x \in Ch(v)$, such that a pair of the leaves of X that occur in $\mathcal{L}(T_1(u_k))$ and $\mathcal{L}(\overline{T_1(u_2)})$ respectively in tree T_1 occur in $\mathcal{L}(T_2(x))$ in tree T_2, and, the third leaf (which occurs in $\mathcal{L}(T_2(x))$ in T_1) occurs in $\mathcal{L}(T_2(y))$ where $y \in Ch(v) \setminus \{x\}$. The number of such X is given by $n_4(u_1, u_k, v)$.

4. *The lca of X in T_2 is node v, and X is unresolved in T_2* : By definition, the number of relevant triplets that satisfy this criterion is exactly $\gamma(u_1, u_k, v)$.

We have shown that $n_2(u_1, u_k, v)$, $n_3(u_1, u_k, v)$, and $n_4(u_1, u_k, v)$ are exactly the number of relevant triplets belonging to categories 1, 2, and 3 respectively. The lemma follows. □

Lemma 7.10. *Given two phylogenetic trees T_1 and T_2 on the same n leaves, the value $|\mathcal{R}_1(T_1, T_2)|$ can be computed in $O(n^2)$ time.*

Proof. Our algorithm for computing $|\mathcal{R}_1(T_1, T_2)|$ appears in Figure 6. The correctness of the algorithm follows from Lemma 7.7. We now analyze its complexity. For any given candidate nodes u, v, Lemma 7.9 shows how to compute $\gamma(\cdot, \cdot, v)$ in $O(|\,Ch(v)|)$ time, and consequently, by Lemma 7.8, the value $r_1(u, v)$ can be computed in $O(|\,Ch(u)| \cdot |\,Ch(v)|)$ time. Thus, the total time complexity of the algorithm is $O(\sum_{u \in \mathcal{V}(T_1)} \sum_{v \in \mathcal{V}(T_2)} |\,Ch(u)| \cdot |\,Ch(v)|)$, which is $O(n^2)$. □

7.5 Computing parametric quartet distance

Existing algorithms for computing traditional quartet distances between partially-resolved unrooted trees (e.g., [14], [46]) can be easily used for computing parametric quartet distances as well. Given two partially resolved unrooted trees T_1 and T_2 on n nodes, let d_i, for $i \in \{1, 2\}$, be the maximum degree of a node in T_i and let $d = \max\{d_1, d_2\}$. Observe that Proposition 7.1 and, thus, Equation (31) hold even when the unit of distance is quartets instead of triplets. Similarly, the values $|R(T_1)|$, $|U(T_1)|$, and $|U(T_2)|$ can be computed in $O(n)$ time for unrooted trees as well. Christiansen et al. [14] show how to compute the values $|\mathcal{S}(T_1, T_2)|$ and $|\mathcal{D}(T_1, T_2)|$ within $O(n^2 \min\{d_1, d_2\})$ time, which, in light of Proposition 7.1(iii) and Equation (31), immediately yields an $O(n^2 \min\{d_1, d_2\})$-time algorithm for computing the parametric quartet distance. Likewise, Stissing et al. [46] show how to compute $|\mathcal{S}(T_1, T_2)|$ and $|\mathcal{U}(T_1, T_2)|$ in $O(d^9 n \log n)$ time, which, in light of Proposition 7.1(i) and Equation (31), yields an $O(d^9 n \log n)$-time algorithm for parametric quartet distance. It can also be shown that, when $p \geq 1/2$, a 2-approximate value of the parametric quartet distance can be computed in $O(n^2)$ time [4].

8 Discussion

We have defined and analyzed distance measures for rooted and unrooted phylogenies that account for partially-resolved nodes. A number of problems remain. While our focus here is on partially-resolved trees, it would nevertheless be interesting to know if there is a $O(n \log n)$ algorithm for triplet distance between fully-resolved rooted trees. More directly relevant to the subject of this paper is the question of determining whether there exists a polynomial-time algorithm for computing the median tree with respect to parametric triplet and quartet distances. We conjecture that this problem is NP-hard. Another natural question is whether or not the Hausdorff triplet (quartet) distance between two partially-resolved trees can be computed in polynomial time. While we suspect that the problem is NP hard, we can, under the density assumption mentioned earlier, partially circumvent the issue by using the equivalence of Hausdorff distance and parametric distance to get an approximation algorithm for the former. Also, many (if not most) applications require the comparison of trees that do not have the same set of taxa. It would be useful to investigate whether any of our distance measures can be extended to this setting.

Finally, existing triplet and quartet measures have been criticized for being too sensitive to the location of unresolved nodes. For the case of rooted trees, unresolved nodes close to the root correspond to many more triplets that those close to leaves, thus, perhaps, granting some nodes more weight than they deserve in the distance computation. Parametric and Hausdorff triplet and quartet distance measures also exhibit such a tendency. An interesting problem is to devise weighing schemes that compensate for this bias.

Acknowledgements

We thank the reviewers for their numerous valuable comments on previous versions of this manuscript. The authors were supported in part by National Science Foundation grants DEB-0334832, DEB-

0829674, and CCF-106029.

References

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

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of Computing*, pages 684–693, New York, NY, USA, 2005. ACM Press.

[3] B. Allen and M. Steel. Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics*, 5:1–13, 2001.

[4] M. S. Bansal, J. Dong, and D. Fernández-Baca. Comparing and aggregating partially resolved trees. arXiv:0906.5089v1.

[5] J. P. Barthélemy and F. R. McMorris. The median procedure for n-trees. *Journal of Classification*, 3:329–334, 1986.

[6] J. J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6, 1989.

[7] V. Berry, T. Jiang, P. E. Kearney, M. Li, and H. T. Wareham. Quartet cleaning: Improved algorithms and simulations. In *Proceedings of the 7th Annual European Symposium on Algorithms*, volume 1643 of *LNCS*, pages 313–324. Springer, 1999.

[8] O. R. P. Bininda-Emonds, editor. *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*, volume 4 of *Series on Computational Biology*. Springer, Berlin, 2004.

[9] C. Bonnard, V. Berry, and N. Lartillot. Multipolar consensus for phylogenetic trees. *Syst. Biol.*, 55(5):837–843, 2006.

[10] G. S. Brodal, R. Fagerberg, and C. N. S. Pedersen. Computing the quartet distance in time $O(n \log n)$. *Algorithmica*, 38(2):377–395, 2003.

[11] D. Bryant. *Building trees, hunting for trees, and comparing trees: Theory and methods in phylogenetic analysis*. PhD thesis, Department of Mathematics, University of Canterbury, New Zealand, 1997.

[12] D. Bryant. A classification of consensus methods for phylogenetics. In M. Janowitz, F.-J. Lapointe, F. McMorris, B. B. Mirkin, and F. Roberts, editors, *Bioconsensus*, volume 61 of *Discrete Mathematics and Theoretical Computer Science*, pages 163–185. American Mathematical Society, Providence, RI, 2003.

[13] J. Byrka, S. Guillemot, and J. Jansson. New results on optimizing rooted triplets consistency. *Discrete Applied Mathematics*, 158(11):1136–1147, June 2010.

[14] C. Christiansen, T. Mailund, C. N. Pedersen, M. Randers, and M. S. Stissing. Fast calculation of the quartet distance between trees of arbitrary degrees. *Algorithms for Molecular Biology*, 1(16), 2006.

[15] J. A. Cotton, C. S. Slater, and M. Wilkinson. Discriminating supported and unsupported relationships in supertrees using triplets. *Systematic Biology*, 55(2):345–350, April 2006.

[16] J. A. Cotton and M. Wilkinson. Quantifying the potential utility of phylogenetic characters. *Taxon*, 57(1):1–6, 2008.

[17] D. E. Critchlow. *Metric Methods for Analyzing Partially Ranked Data*, volume 34 of *Lecture Notes in Statist.* Springer-Verlag, Berlin, 1980.

[18] D. E. Critchlow, D. K. Pearl, and C. Qian. The triples distance for rooted bifurcating phylogenetic trees. *Systematic Biology*, 45(3):323–334, 1996.

[19] W. H. E. Day. Analysis of quartet dissimilarity measures between undirected phylogenetic trees. *Systematic Zoology*, 35(3):325–333, Sep. 1986.

[20] P. Diaconis and R. Graham. Spearman's footrule as a measure of disarray. *J. of the Royal Statistical Society, Series B*, 39(2):262–268, 1977.

[21] A. C. Driskell, C. Ané, J. G. Burleigh, M. M. McMahon, B. C. O'Meara, and M. J. Sanderson. Prospects for building the tree of life from large sequence databases. *Science*, 306(5699):1172–1174, November 2004.

[22] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Tenth International World Wide Web Conference*, Hong Kong, May 2001.

[23] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing partial rankings. *SIAM J. Discrete Math.*, 20(3):628–648, 2006.

[24] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. *SIAM J. Discrete Math.*, 2003.

[25] M. Farach and M. Thorup. Optimal evolutionary tree comparison by sparse dynamic programming. In *Proc. 35th Annual Symposium on Foundations of Computer Science*, pages 770–779, Piscataway, NJ, 1994. IEEE Computer Society Press.

[26] J. Felsenstein. *Inferring Phylogenies*. Sinauer Assoc., Sunderland, Mass, 2003.

[27] C. R. Finden and A. D. Gordon. Obtaining common pruned trees. *J. Classification*, 2(1):225–276, 1985.

[28] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge–New York–Melbourne, 1997.

[29] S. Kannan, T. Warnow, and S. Yooseph. Computing the local consensus of trees. *SIAM J. Comput.*, 27(6):1695–1724, December 1998.

[30] M.-Y. Kao, T.-W. Lam, W.-K. Sung, and H.-F. Ting. An even faster and more unifying algorithm for comparing trees via unbalanced bipartite matchings. *Journal of Algorithms*, 40(2):212–233, 2001.

[31] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88:577–591, 1959.

[32] C.-M. Lee, L.-J. Hung, M.-S. Chang, C.-B. Shen, and C.-Y. Tang. An improved algorithm for the maximum agreement subtree problem. *Information Processing Letters*, 94(5):211–216, June 2005.

[33] W. P. Maddison. Reconstructing character evolution on polytomous cladograms. *Cladistics*, 5:365–377, 1989.

[34] F. R. McMorris, D. B. Meronk, and D. A. Neumann. A view of some consensus methods for trees. In J. Felsenstein, editor, *Numerical Taxonomy*, pages 122–125. Springer-Verlag, 1983.

[35] W. Piel, M. Sanderson, M. Donoghue, and M. Walsh. Treebase. http://www.treebase.org. Last accessed 2 February 2007.

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

[37] V. Ranwez, A. Criscuolo, and E. J. Douzery. Supertriplets: a triplet-based supertree approach to phylogenomics. *Bioinfomatics*, 26(ISMB 2010):i115—i123, 2010.

[38] D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.

[39] C. Scornavacca. *Supertree methods for phylogenomics*. PhD thesis, University of Montpellier II, Montpellier, France, December 2009.

[40] C. Scornavacca, V. Berry, E. J. P. Douzery, and V. Ranwez. PhySIC IST: cleaning source trees to infer more informative supertrees. *BMC Bioinformatics*, 9:413, 2008.

[41] C. Semple and M. Steel. *Phylogenetics*. Oxford Lecture Series in Mathematics. Oxford University Press, Oxford, 2003.

[42] S. Snir and S. Rao. Using max cut to enhance rooted trees consistency. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 3(4):323–333, 2006.

[43] S. Snir and S. Rao. Quartets maxcut: A divide and conquer quartets algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, 7(4):704–718, 2010.

[44] M. Steel and D. Penny. Distributions of tree comparison metrics — some new results. *Systematic Biology*, 42(2):126–141, 1993.

[45] M. A. Steel. *Distributions on bicoloured evolutionary trees*. PhD thesis, Massey University, 1989.

[46] M. Stissing, C. N. S. Pedersen, T. Mailund, G. S. Brodal, and R. Fagerberg. Computing the quartet distance between evolutionary trees of bounded degree. In D. Sankoff, L. Wang, and F. Chin, editors, *APBC*, volume 5 of *Advances in Bioinformatics and Computational Biology*, pages 101–110. Imperial College Press, 2007.

[47] C. Stockham, L.-S. Wang, and T. Warnow. Statistically based postprocessing of phylogenetic analysis by clustering. In *ISMB*, pages 285–293, 2002.

[48] M. S. Swenson, R. Suri, C. R. Linder, and T. Warnow. An experimental study of quartets maxcut and other supertree methods. *Algorithms for Molecular Biology*, 6:7, 2011.

[49] J. L. Thorley, M. Wilkinson, and M. A. Charleston. The information content of consensus trees. In A. Rizzi, A. Vichi, and H. H. Bock, editors, *Advances in Data Science and Classification. Studies in Classification, Data Analysis, and Knowledge Organization*, pages 91–98. Springer, Berlin, 1998.

[50] M. Wilkinson. Majority-rule reduced consensus trees and their use in bootstrapping. *Mol. Biol. Evol.*, 13(3):437–444, 1996.