Journal of Discrete Algorithms 25 (2014) 66–78

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Computing the rooted triplet distance between galled trees by counting triangles ☆

Jesper Jansson [a,*,1], Andrzej Lingas [b,2]

[a] Laboratory of Mathematical Bioinformatics, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011, Japan
[b] Department of Computer Science, Lund University, 22100 Lund, Sweden

ARTICLE INFO

Article history:
Available online 11 October 2013

Keywords:
Graph algorithm
Rooted triplet distance
Phylogenetic network comparison
Triangle counting
Matrix multiplication

ABSTRACT

We consider a generalization of the rooted triplet distance between two phylogenetic trees to two phylogenetic networks. We show that if each of the two given phylogenetic networks is a so-called galled tree with n leaves then the rooted triplet distance can be computed in $o(n^{2.687})$ time. Our upper bound is obtained by reducing the problem of computing the rooted triplet distance between two galled trees to that of counting monochromatic and almost-monochromatic triangles in an undirected, edge-colored graph. To count different types of colored triangles in a graph efficiently, we extend an existing technique based on matrix multiplication and obtain several new algorithmic results that may be of independent interest: (i) the number of triangles in a connected, undirected, uncolored graph with m edges can be computed in $o(m^{1.408})$ time; (ii) if G is a connected, undirected, edge-colored graph with n vertices and C is a subset of the set of edge colors then the number of monochromatic triangles of G with colors in C can be computed in $o(n^{2.687})$ time; and (iii) if G is a connected, undirected, edge-colored graph with n vertices and R is a binary relation on the colors that is computable in $O(1)$ time then the number of R-chromatic triangles in G can be computed in $o(n^{2.687})$ time.

1. Introduction

Phylogenetic trees and their generalization to non-treelike structures, phylogenetic networks, are commonly used by scientists to describe evolutionary relationships [10,14,18,19,22]. In certain applications, it is necessary to compare two given phylogenetic trees and measure their (dis-)similarity, for example when evaluating methods for phylogenetic reconstruction [17] or querying phylogenetic databases [2]. Various ways of measuring the dissimilarity of two phylogenetic trees have been proposed and analyzed in the literature; see [2] and the references therein. One such measure is the *rooted triplet distance* [2,3,8,9], which counts the number of substructures (more precisely, subtrees induced by three leaves) that differ between the two trees. Intuitively, if the two trees are "similar" and share a lot of branching structure then this number will be small.

Formally, the rooted triplet distance is defined as follows. A *rooted phylogenetic tree* is an unordered, rooted tree in which every internal node has at least two children and all leaves are distinctly labeled. A rooted phylogenetic tree with three leaves is called a *rooted triplet*. A rooted triplet leaf-labeled by $\{a, b, c\}$ with exactly one internal node is called a *rooted fan*

☆ A preliminary version of this article appeared in Proceedings of the 23rd Annual Symposium on Combinatorial Pattern Matching (CPM 2012), vol. 7354 of Lecture Notes in Computer Science, pp. 385–398, Springer-Verlag, Berlin, Heidelberg, 2012.

* Corresponding author.
E-mail addresses: jj@kuicr.kyoto-u.ac.jp (J. Jansson), Andrzej.Lingas@cs.lth.se (A. Lingas).

[1] Funded by The Hakubi Project and KAKENHI grant number 23700011.

[2] Research supported in part by VR grant 621-2008-4649.

1570-8667/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jda.2013.10.002

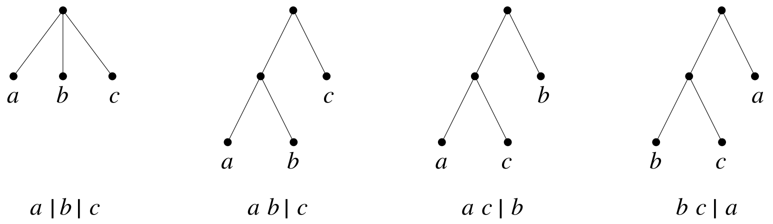

Fig. 1. The rooted fan triplet $a|b|c$ and the three rooted binary triplets $ab|c$, $ac|b$, and $bc|a$.

triplet and is denoted by $a|b|c$, and a rooted triplet leaf-labeled by $\{a, b, c\}$ with exactly two internal nodes is called a *rooted binary triplet*; in the latter case, there are three possibilities, denoted by $ab|c$, $ac|b$, and $bc|a$, corresponding to the three possible topologies. See Fig. 1 for an illustration. A rooted triplet t is said to be *consistent with* a rooted phylogenetic tree T if t is an embedded subtree of T, i.e., $a|b|c$ is consistent with T if $lca^T(a,b) = lca^T(a,c) = lca^T(b,c)$, and $ab|c$ is consistent with T if $lca^T(a,b)$ is a proper descendant of $lca^T(a,c) = lca^T(b,c)$, where for any two leaf labels x and y, $lca^T(x,y)$ denotes the lowest common ancestor in T of the leaves labeled by x and y. Now, given two rooted phylogenetic trees T_1, T_2 with the same set L of leaf labels, the *rooted triplet distance* $d_{rt}(T_1, T_2)$ is the number of rooted triplets over L that are consistent with exactly one of T_1 and T_2.

The rooted triplet distance was introduced by Dobson [9] in 1975. The naive algorithm for computing $d_{rt}(T_1, T_2)$ between two phylogenetic trees T_1 and T_2 with a leaf label set of cardinality n runs in $O(n^3)$ time: Just preprocess T_1 and T_2 in $O(n)$ time so that lowest common ancestor queries can be answered in $O(1)$ time by the method in [13], and then check each of the $O(n^3)$ possible rooted triplets for consistency with T_1 and T_2 in $O(1)$ time. Critchlow et al. [8] provided a more efficient algorithm for computing the rooted triplet distance between two *binary* phylogenetic trees with $O(n^2)$ running time, and Bansal et al. [2] extended the $O(n^2)$-time upper bound to two phylogenetic trees of *arbitrary* degrees. The current record is held by Brodal et al. [3], who achieved a running time of $O(n \log n)$ for two phylogenetic trees of arbitrary degrees.

Due to the recently increasing popularity of the phylogenetic *network* model and its potential impact on evolutionary biology in the near future (see the two textbooks [14,18]), it is compelling to consider generalizations of the rooted triplet distance to the network case. As observed by Gambette and Huber [11], d_{rt} can be canonically extended by replacing the two trees T_1 and T_2 in the definition above by two networks. However, for phylogenetic networks, it seems much harder to improve on the naive $O(n^3)$-time algorithm and to derive a subcubic upper bound on the running time. Therefore, one would like to know if any important special classes of phylogenetic networks such as the *galled trees* [12,14] admit fast algorithms. Galled trees are structurally restricted phylogenetic networks in which all underlying cycles are vertex-disjoint; for a formal definition, refer to Section 3.3 below. They constitute one of the simplest classes of phylogenetic networks and are useful in certain settings where reticulation events do occur but are known to be rare [12]. (See, e.g., Figure 9.22 in [14] for an example of a galled tree for a set of strains of *Fusarium graminearum*.) As a consequence, a number of algorithms for building galled trees from different kinds of data have been published [6,12,14–16].

In this article, we focus on the rooted triplet distance and describe how to compute it efficiently when the two input networks are galled trees. Several other measures of the dissimilarity between two phylogenetic networks, including *the Robinson–Foulds distance*, *the tripartitions distance*, *the μ-distance*, *the nodal distance*, and *the split nodal distance*, were investigated for the special case of galled trees by Cardona et al. in [5]. (See [5] for the definitions of these measures and many references to the literature.)

1.1. New results

Our main contribution is an $o(n^{2.687})$-time algorithm for computing the rooted triplet distance between two galled trees with n leaves each (Theorem 4). This breaks the natural $O(n^3)$-time barrier for any kind of non-tree phylogenetic networks for the first time. The precise running time of our algorithm is $O(n^{(3+\omega)/2})$, where ω denotes the exponent in the running time of the fastest existing method for matrix multiplication. It is currently known that $\omega < 2.373$ [25].

Theorem 4 is obtained in part by a reduction to the problem of counting monochromatic and "almost-monochromatic" triangles in an undirected graph with colored edges. To solve the latter problem quickly, we strengthen a technique based on matrix multiplication used in [1] and [24] for *detecting* if a graph contains a triangle to also *count* the number of triangles in the graph. More exactly, we show that:

- The number of triangles in a connected, undirected, uncolored graph with m edges can be computed in $O(m^{\frac{2\omega}{\omega+1}}) \leqslant o(m^{1.408})$ time (Theorem 1).
- If G is a connected, undirected, edge-colored graph with n vertices and C is a subset of the set of edge colors then the number of monochromatic triangles of G with colors in C can be computed in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time (Theorem 2).

We also need to relax the concept of a monochromatic triangle to what we call an R-chromatic triangle (see Section 2 for the definition), and obtain:

- If G is a connected, undirected, edge-colored graph with n vertices and R is a binary relation on the colors that is computable in $O(1)$ time then the number of R-chromatic triangles in G can be computed in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time (Theorem 3).

The new results on counting triangles in a graph may be of general interest and could be useful in other applications unrelated to the main problem studied here.

1.2. Organization of the article

The article is organized as follows. Section 2 develops techniques for counting different types of colored triangles in an edge-colored graph. Section 3 defines phylogenetic networks, the rooted triplet distance, and galled trees, and proves some structural properties of galled trees. Next, in Section 4, we employ the triangle counting techniques from Section 2 to obtain our main algorithm. Finally, Section 5 mentions some open problems.

2. Counting monochromatic and almost-monochromatic triangles in a graph

A *triangle* in an undirected graph is a cycle of length 3. Alon et al. [1] showed how to determine if a connected, undirected graph with m edges contains a triangle, and if so, how to find a triangle in $O(m^{\frac{2\omega}{\omega+1}}) \leqslant o(m^{1.408})$ time (Theorem 3.5 in [1]). They also showed how to *count* the number of triangles in an undirected graph with n vertices in $O(n^{\omega}) \leqslant o(n^{2.373})$ time (Theorem 6.3 in [1]). We first improve their technique to count the number of triangles more efficiently in case the input graph is sparse ($m \ll n^2$):

Theorem 1. *Let G be a connected, undirected graph with m edges. The number of triangles in G can be computed in $O(m^{\frac{2\omega}{\omega+1}}) \leqslant o(m^{1.408})$ time.*

Proof. The proof is a minor modification of the proof of Theorem 3.5 in [1]. Define $t = m^{\frac{\omega-1}{\omega+1}}$. The method differentiates between two types of triangles, depending on t.

First, count the number of triangles in G whose three vertices all have degree at least t in G. To do this, take the subgraph of G induced by all vertices of degree $\geqslant t$, and apply the triangle counting method from Theorem 6.3 in [1] which runs in $O(|V|^{\omega})$ time for any graph with $|V|$ vertices. Let $N_{\triangle}$ be the computed number of triangles in the subgraph. Since the number of vertices with degree $\geqslant t$ is $O(\frac{m}{t})$, the aforementioned method takes $O(\frac{m^{\omega}}{t^{\omega}}) = O(m^{\omega-\frac{(\omega-1)\omega}{\omega+1}}) = O(m^{\frac{2\omega}{\omega+1}})$ time.

Secondly, count the triangles in G with at least one vertex of degree strictly less than t. For this purpose, let F be the set of edges in G with at least one endpoint of degree $< t$. Enumerate the edges in F in any arbitrary order and let e_i denote the ith edge in this ordering. For $i = 1, \ldots, |F|$, perform the following operations:

- Pick an endpoint v of edge e_i in F with degree less than t. For each edge e incident to e_i at v, check if e_i and e induce a triangle in G that does not include any edge $e_j \in F$ with $j < i$; if yes then increase $N_{\triangle}$ by one.

The above steps can be implemented in $O(t)$ time, so counting the remaining triangles takes $O(mt) = O(m^{1+\frac{\omega-1}{\omega+1}}) = O(m^{\frac{2\omega}{\omega+1}})$ time.

Finally, return $N_{\triangle}$. □

Observe that Theorem 1 is faster than Theorem 6.3 in [1] when $m = o(n^{\frac{\omega+1}{2}})$.

We can similarly refine the part of Theorem 1.8 in [24] which states that a monochromatic triangle in a connected, undirected, edge-colored graph with n vertices can be found (if one exists) in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time. We obtain:

Theorem 2. *Let G be a connected, undirected, edge-colored graph with n vertices and let C be a subset of the set of edge colors. The number of monochromatic triangles of G with colors in C can be computed in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time.*

Proof. For each color $i \in C$, let E_i be the set of edges in G colored by i. As in [24], say that i is *heavily used* if $|E_i| \geqslant n^{(\omega+1)/2}$. For each heavily used color, we count the number of monochromatic triangles by directly applying the triangle counting method from Theorem 6.3 in [1] to the subgraph induced by edges colored with i in $O(n^{\omega})$ time. This takes $O(n^{\omega}) \cdot O(n^2/n^{(\omega+1)/2}) = O(n^{\omega+2-(\omega+1)/2}) = O(n^{(3+\omega)/2})$ time in total.

To count the remaining monochromatic triangles, for each non-heavily used color $i \in C$, we apply the method of Theorem 1 above to the subgraph induced by the edges in E_i. This takes $O(|E_i|^{2\omega/(\omega+1)})$ time. As in the proof of Theorem 1.8 in [24], the total time taken by all the non-heavily used colors is maximized if $|E_i| = \Theta(n^{(\omega+1)/2})$ holds for each of them and there are $\Theta(n^{2-(\omega+1)/2})$ non-heavily used colors. Since $O((n^{(\omega+1)/2})^{\frac{2\omega}{\omega+1}}) \cdot O(n^{2-(\omega+1)/2}) = O(n^{\omega}) \cdot O(n^{(3-\omega)/2}) = O(n^{(3+\omega)/2})$, this shows that the total time to count all remaining monochromatic triangles is also $O(n^{(3+\omega)/2})$. □

Next, we consider a kind of relaxation of the concept of a monochromatic triangle to an "almost-monochromatic triangle" in an undirected, edge-colored graph G. Let R be a binary relation on the edge colors. A triangle in G with two edges of the same color i and the third one of color k such that iRk holds is called an *R-chromatic triangle* (e.g., if R stands for $<$ then k is simply required to be larger than i). We need to extend Theorem 2 to count R-chromatic triangles. We begin with a lemma whose role is analogous to Theorem 6.3 in [1].

Lemma 1. *Let G be a connected, undirected, edge-colored graph with n vertices and let R be a binary relation on the edge colors of G computable in constant time. For any edge color i, the number of R-chromatic triangles in G with at least two edges of color i can be computed in $O(n^{\omega})$ time.*

Proof. Let G_i be the subgraph of G induced by the edges with color i. Build the adjacency matrix A_i of G_i and compute the square $C_i = (A_i)^2$ in $O(n^{\omega})$ time. For each entry $C_i[k,l]$ with $k < l$, check if $\{k,l\}$ is an edge of G whose color j is in the relation R with the color i, i.e., if iRj holds. If so, increase the count of triangles by the value of $C_i[k,l]$; in case $\{k,l\}$ is an edge whose color is also i and iRi holds, increase the count of triangles by $C_i[k,l]/3$ only. □

Theorem 2 can now be generalized to R-chromatic triangles:

Theorem 3. *Let G be a connected, undirected, edge-colored graph with n vertices, and let R be a binary relation on the edge colors of G computable in constant time. The number of R-chromatic triangles in G can be computed in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time.*

Proof. For every edge color i, let E_i denote the set of edges of G having color i and let G_i be the subgraph of G induced by E_i.

Proceed as in the proof of Theorem 2. First, for each heavily used color i, i.e., satisfying $|E_i| \geqslant n^{(\omega+1)/2}$, apply Lemma 1 to count the number of R-chromatic triangles in G with at least two edges of color i. This takes $O(n^{\omega+2-(\omega+1)/2}) = O(n^{(3+\omega)/2})$ time in total.

Then, every remaining R-chromatic triangle in G has at least two edges colored with a non-heavily used color. We now explain how to count these triangles, too. Construct all the subgraphs G_i of G using $O(n^2)$ time in total. For each non-heavily used color i, run the method of Theorem 1 with the following modifications that do not affect the asymptotic time complexity:

1. In the first step, instead of counting the number in triangles in G_i whose three vertices all have degree at least t in G_i, where $t = |E_i|^{\frac{\omega-1}{\omega+1}}$, apply Lemma 1 to count the number of R-chromatic triangles in G with at least two edges of color i and whose three vertices all have degree at least t in G_i.
2. In the second step, when scanning the edges e of G_i having at least one vertex v of degree less than t, for every edge e' of G incident to e at v, check if e and e' induce an R-chromatic triangle in G that was not counted before and if so, increase the count by one. This will count the R-chromatic triangles in G with at least two edges of color i and at least one vertex of degree strictly less than t.

Thus, each non-heavily used color i requires $O(|E_i|^{2\omega/(\omega+1)})$ time. By the same argument as in the proofs of Theorem 1.8 in [24] and Theorem 2 above, we infer that the total time needed to count the remaining R-chromatic triangles is $O(n^{(3+\omega)/2})$. □

3. Definitions and properties of phylogenetic networks

3.1. Basic definitions

A (rooted) *phylogenetic network* U is a directed acyclic graph with a single root vertex and a set of distinctly labeled leaves, and no vertices having both indegree 1 and outdegree 1. Throughout the paper, we refer to the leaves in a phylogenetic network by their leaf labels. Also, we use the standard convention of drawing a phylogenetic network with the root at the top and all edges oriented downwards. A vertex u is an *ancestor* of a vertex v (or, equivalently, v is a *descendant* of u) in U if and only if there is a directed path from u to v in U. In particular, u is an ancestor and descendant of itself. If the path from u to v has non-zero length then v is a *proper descendant* of u. Next, a vertex w is a *common ancestor* of vertices u and v in U if and only if w is an ancestor of both u and v in U. Furthermore, w is a *junction common ancestor* (jca) of u and v in U if and only if there are two directed paths from w to u and from w to v, respectively, which are vertex-disjoint but for the start vertex w. Finally, w is a *lowest common ancestor* (lca) of u and v in U if and only if: (1) w is a common ancestor of u and v; and (2) w has no proper descendant that is a common ancestor of u and v.

The definitions imply that any two vertices u and v in a phylogenetic network U have at least one lca in U and at least one jca in U. In general, two vertices u and v in a phylogenetic network U may have more than one lca; for example, if w_1 and w_2 are two different parents of u and v in U then both of w_1 and w_2 are lca's of u and v. (Lemma 2 below shows that this is not possible for the restricted case of galled trees.) Also, if w is an lca of u and v in U then w is a jca of u

and v in U. However, a jca of u and v in U is not necessarily an lca of u and v in U. As an example, in Fig. 2, vertices w and z are two different jca's of a and c, w is an lca of a and c, and z is not an lca of a and c.

Rooted triplet consistency in a phylogenetic network U is defined next, in accordance with the definition given in [14,15] for rooted binary triplets. Let a, b, c be three leaf labels in U.

- The rooted binary triplet $ab|c$ is *consistent with* U if and only if U contains a junction common ancestor w of a and b as well as a junction common ancestor z of c and w such that there are four directed paths of non-zero length from w to a, from w to b, from z to w, and from z to c that are vertex-disjoint except for in the vertices w and z.
- The rooted fan triplet $a|b|c$ is *consistent with* U if and only if U contains a vertex w and three directed paths from w to a, from w to b, and from w to c that are vertex-disjoint except for in the common start vertex w.

To illustrate, $ab|c$ and $bc|a$ are consistent with the network in Fig. 2. In Fig. 3, $ab|c$ and $a|b|c$ are consistent with the network in (A), $bc|a$ and $a|b|c$ are consistent with the network in (B), and $a|b|c$ is consistent with the network in (C).

Observe that whenever U is a tree, the concepts of a lowest common ancestor and a junction common ancestor between two leaves coincide, and the definitions of rooted triplet consistency in a tree reduce to the definitions given in Section 1.

3.2. The rooted triplet distance

In this article, the rooted triplet distance between two phylogenetic networks is defined as:

Definition 1. Let U_1, U_2 be two phylogenetic networks on the same leaf label set L. The *rooted triplet distance between U_1 and U_2*, denoted by $d_{rt}(U_1, U_2)$, is the number of rooted fan triplets and rooted binary triplets with leaf labels from L that are consistent with exactly one of U_1 and U_2.

This is the canonical extension of d_{rt} suggested by Gambette and Huber [11] from the phylogenetic tree model to the phylogenetic network model. We remark that this definition differs slightly from the one restricted to trees in [2,3,8,9], which counts the number of "bad" cardinality-3 subsets L' of L for which the rooted triplet with leaf set L' consistent with U_1 differs from the rooted triplet with leaf set L' consistent with U_2. When dealing with phylogenetic networks, Definition 1 may be more suitable than simply counting the number of bad cardinality-3 subsets of L because it distinguishes between cases such as: (i) $a|b|c$ and $bc|a$ are consistent with U_1 whereas only $bc|a$ is consistent with U_2; and (ii) $a|b|c$ is consistent with U_1 and $bc|a$ is consistent with U_2. (Only $a|b|c$ will contribute to $d_{rt}(U_1, U_2)$ in case (i), but both $a|b|c$ and $bc|a$ will contribute in case (ii).) One important feature of phylogenetic networks is their ability to induce more than just one rooted triplet for any three given leaf labels, thereby making it possible for two networks to *partially* disagree on three leaf labels, and Definition 1 takes this into account.

Also note that when restricted to trees, the value of d_{rt} in Definition 1 is exactly two times the value of d_{rt} from [2,3,8,9] because every bad cardinality-3 subset will contribute twice to our d_{rt} (one time for the rooted triplet in U_1 and one time for the rooted triplet in U_2) rather than once. Obviously, Definition 1 can be normalized by dividing it by two but then d_{rt} will no longer always be an integer in the non-tree case.

3.3. Galled trees

Here, we recall the definition of the class of phylogenetic networks called the *galled tree* [12,14], and explore some of its structural properties. This kind of phylogenetic network was first considered by Wang et al. [23] and later by Gusfield et al. [12] and others (see, e.g., [14]).

A *reticulation vertex* in a phylogenetic network is any vertex of indegree greater than 1. For any phylogenetic network U, define *its underlying undirected graph* as the undirected graph obtained by replacing every directed edge in U by an undirected edge. A *cycle* C in a phylogenetic network is any subgraph with at least three edges whose corresponding subgraph in the underlying undirected graph is isomorphic to a cycle, and the vertex of C that is an ancestor of all vertices on C is called the *root* of C. A phylogenetic network is called a *galled tree* if all of its cycles are vertex-disjoint [12,14]. Clearly, every reticulation vertex in a galled tree must have indegree 2. Every cycle C in a galled tree has one root and one reticulation vertex, and C consists of two directed, internally disjoint paths from its root to its reticulation vertex. Also, any directed path from the root of the galled tree to a vertex on such a cycle passes through the root of the cycle.

Let U be a galled tree. From U, we can construct two trees $U^{\searrow}$ and $U^{\swarrow}$ as follows. For each cycle C in U, arbitrarily term one of the two edges on C incident to the reticulation vertex as *a left reticulation edge* and the other one as *a right reticulation edge*. Let $U^{\searrow}$ be the tree obtained from U by taking a copy of U and removing all right reticulation edges. Define $U^{\swarrow}$ in the same way, but removing all left reticulation edges instead. Although $U^{\searrow}$ and $U^{\swarrow}$ are not uniquely defined by U, for any such pair of resulting trees it holds that every reticulation edge of U occurs in exactly one $U^{\searrow}$ and $U^{\swarrow}$.

The next lemma summarizes some useful properties of galled trees:

Lemma 2. *Let U be a galled tree with n leaves and let u and v be any two vertices in U. Then*:

1. *There is exactly one lowest common ancestor of u and v in U.*
2. *There are at most two junction common ancestors of u and v in U.*
3. *If there are two junction common ancestors of u and v in U then both of them lie on the same cycle C in U. Furthermore, one of them is the root of C and the other one is the lowest common ancestor of u and v in U.*
4. *The number of vertices in U as well as the number of edges in U is $O(n)$.*
5. *All junction common ancestors of pairs of vertices in U can be listed in $O(n^2)$ time.*

Proof.

1. Suppose, for the purpose of obtaining a contradiction, that there were two different lca's w_1 and w_2 of u and v in U. Consider any path from w_1 to u in U and any path from w_2 to u in U. Since both paths lead to u, they must meet at some ancestor u' of u which then has indegree larger than 1, where u' is a proper descendant of w_1 and also a proper descendant of w_2. Symmetrically, there exists an ancestor v' of v with indegree larger than 1 which is a proper descendant of both w_1 and w_2, with $u' \neq v'$. Now let x be an lca of w_1 and w_2 in U. In the underlying undirected graph of U, there is a cycle containing x and u' and another cycle containing x and v', i.e., two non-vertex-disjoint cycles, contradicting the definition of a galled tree. Thus, property 1 holds.
2. Consider the two trees $U^{\searrow}$ and $U^{\swarrow}$ defined above. First observe that every jca of u and v in U is an lca of u and v in at least one of $U^{\searrow}$ and $U^{\swarrow}$. (To see this, let w be a jca of u and v in U. If w belongs to a cycle C then at most one of the two disjoint paths from w to u and v can pass through the reticulation vertex on C; therefore, w will still be a jca, and hence an lca, of u and v in at least one of $U^{\searrow}$ and $U^{\swarrow}$. Otherwise, w does not belong to a cycle and then w is an lca of u and v in both $U^{\searrow}$ and $U^{\swarrow}$.) Since u and v have exactly one lca in $U^{\searrow}$ and exactly one lca in $U^{\swarrow}$ (possibly the same vertex in U), property 2 follows.
3. Suppose there are two different jca's w_1 and w_2 of u and v in U. By the observation in the previous paragraph, w_1 and w_2 are the lca's of u and v in the two trees $U^{\searrow}$ and $U^{\swarrow}$. If w_1 and w_2 do not belong to the same cycle in U then the lca of u and v in $U^{\searrow}$ is the same vertex as the lca of u and v in $U^{\swarrow}$, so w_1 cannot be different from w_2, which is a contradiction. Hence, w_1 and w_2 must belong to the same cycle in U. Denote this cycle by C. If neither w_1 nor w_2 is the root of C then either w_1 and w_2 lie on the same path from the root of C to the reticulation vertex of C, or on different paths. The former case is impossible because it would imply that the two paths to u and v from the uppermost jca (either w_1 or w_2) overlap, and the latter case is impossible because then w_1 and w_2 could not both be jca's. Thus, either w_1 or w_2 is the root of C. Next, according to the definitions, if w is an lca of u and v in U then w is also a jca of u and v in U. There are at most two jca's of u and v in U by property 2, so any such w must be equal to either w_1 or w_2, which yields property 3.
4. To upper-bound the number of vertices in U, construct a binary galled tree U' (where every vertex has outdegree at most 2) by repeatedly selecting any vertex w with outdegree larger than 2 and replacing any two of its outgoing edges (w, c_1) and (w, c_2) by a single edge (w, x) and two edges (x, c_1) and (x, c_2) where x is a newly created vertex, until no vertex with outdegree larger than 2 remains. This will not introduce any vertices having both indegree 1 and outdegree 1, and U' is still a galled tree with n leaves, but U' contains at least as many vertices as U. According to Lemma 3 in [7], the total number of vertices in any binary galled tree U' with n leaves is $O(n)$, so this also gives an upper bound for U. Furthermore, any vertex in a galled tree can have indegree at most 2 (otherwise, there would exist two non-vertex-disjoint cycles in the underlying undirected graph), so the total number of edges in U is $O(n)$.
5. Since the trees $U^{\searrow}$ and $U^{\swarrow}$ can be preprocessed in linear time to answer ancestor or descendant queries as well as *lca* queries in constant time [13], and lca's in a tree are unique, property 5 follows. □

3.4. Characterizations of rooted triplet consistency in a galled tree

When restricted to galled trees, the definitions of consistency of a rooted binary triplet $ab|c$ or a rooted fan triplet $a|b|c$ with a phylogenetic network can be expressed as in Lemmas 3 and 5 below. See Figs. 2 and 3 for some examples. These two lemmas enable our main algorithm in Section 4 to count the number of rooted triplets shared by two galled trees.

We first characterize rooted binary triplet consistency.

For any galled tree U and two vertices u, v in U, a junction common ancestor z of u and v in U is said to *rely on* another vertex w if, after the removal of w from U, the vertex z is no longer a junction common ancestor of u and v.

Lemma 3. *Let U be a galled tree. For any three leaves a, b, c in the leaf label set of U, the rooted binary triplet $ab|c$ is consistent with U if and only if U contains a junction common ancestor w of a and b as well as a different junction common ancestor z of c and w such that if both w and z belong to the same cycle C of U then at least one of them does not rely on the reticulation vertex of C.*

Proof. The necessity of the condition stated in the lemma follows directly from the definition of consistency of $ab|c$ with U. It remains to show the sufficiency.

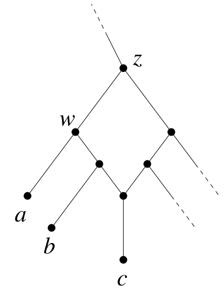

Fig. 2. Illustrating Lemma 3. w is a jca of a and b that does not rely on the reticulation vertex, and z is a jca of c and w, so Lemma 3 gives us the rooted binary triplet $ab|c$. Note that Lemma 3 also correctly identifies $bc|a$.

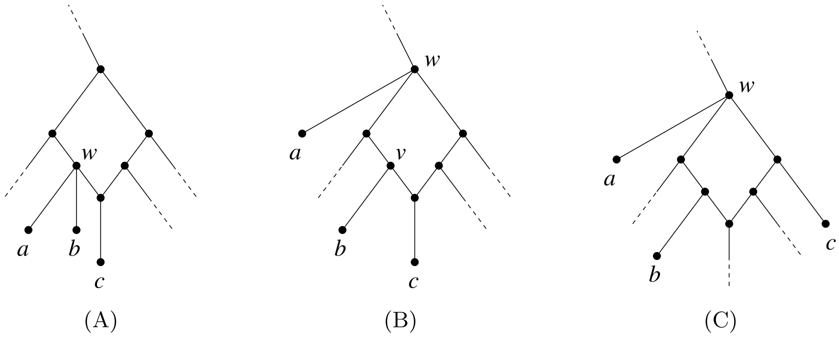

Fig. 3. Illustrating Lemma 5. In each of the diagrams, U is a (partially depicted) galled tree. In (A), $a|b|c$ is of type 1 and the conditions in Lemma 5-1 (i) hold with $w \neq \mathit{lca}^{U^{\swarrow}}(a,c) = \mathit{lca}^{U^{\swarrow}}(b,c)$. In (B), $a|b|c$ is of type 1 and Lemma 5-1 (ii) holds with $w \neq \mathit{lca}^{U}(b,c) = \mathit{lca}^{U^{\searrow}}(b,c)$. In (C), $a|b|c$ is of type 2 and Lemma 5-2 holds. In addition to the above, Lemma 3 also identifies $ab|c$ in (A) and $bc|a$ in (B).

The proof is by contradiction. First of all, the path from z to w crosses neither the path from w to a nor the path from w to b since U is an acyclic directed graph. Next, if the path from z to c would cross the one from w to a (or the one from w to b) in an inner vertex x then z and w would lie on a common cycle whose reticulation vertex is x, and both would rely on x. We obtain a contradiction. □

The rest of this subsection deals with rooted fan triplet consistency.

Suppose that $a|b|c$ is a rooted fan triplet that is consistent with a galled tree U. By definition, U contains a vertex w and three directed, internally vertex-disjoint paths from w to each of the leaves a, b, and c. There can only be one such vertex w in U (otherwise, there would be two non-vertex-disjoint cycles in U's underlying undirected graph, contradicting the definition of a galled tree), and we call w *the root of* $a|b|c$ *in* U.

Lemma 4. *Suppose that* $a|b|c$ *is a rooted fan triplet that is consistent with a galled tree* U. *Then* $a|b|c$ *is consistent with at least one of* $U^{\searrow}$ *and* $U^{\swarrow}$.

Proof. Let w be the root of $a|b|c$ in U and let P_a, P_b, and P_c be any three directed, internally vertex-disjoint paths in U from w to a, from w to b, and from w to c, respectively.

For any cycle C in U, the reticulation vertex of C is reachable from its root in both $U^{\searrow}$ and $U^{\swarrow}$. Also, any subpath in U starting at a vertex on C that is not the root of C and ending at the reticulation vertex of C is present in exactly one of $U^{\searrow}$ and $U^{\swarrow}$. Therefore, if it holds for each cycle in U that at most one of P_a, P_b, and P_c contains edges from that cycle, then a, b, and c will still be reachable by internally vertex-disjoint paths from w in at least one of the two trees. Otherwise, some cycle C in U overlaps with two of P_a, P_b, and P_c, and then w must be the root of C; noting that at most one of the left and the right reticulation edges of C can be contained in P_a, P_b, and P_c (otherwise two paths would overlap in C's reticulation vertex, which contradicts them being internally vertex-disjoint), we deduce that a reticulation edge of C not contained in any of P_a, P_b, and P_c can be removed without affecting the reachability of C's reticulation vertex from w, i.e., a, b, and c are reachable by internally vertex-disjoint paths from w in at least one of $U^{\searrow}$ and $U^{\swarrow}$. This shows that $a|b|c$ is consistent with at least one of the two trees. □

It will be convenient to partition the rooted fan triplets in any galled tree U into two types based on Lemma 4. Let $a|b|c$ be a rooted fan triplet consistent with U. If $a|b|c$ is consistent with exactly one of the two trees $U^{\searrow}$ and $U^{\swarrow}$ then $a|b|c$ is said to be of *type* 1 *in* U; otherwise, by Lemma 4, $a|b|c$ must be consistent with both of $U^{\searrow}$ and $U^{\swarrow}$ and we say that $a|b|c$

is of *type* 2 *in* U. For example, $a|b|c$ in Fig. 3 (A) is of type 1 and $a|b|c$ in Fig. 3 (B) is also of type 1. On the other hand, $a|b|c$ in Fig. 3 (C) is of type 2.

The next (somewhat technical) lemma characterizes the occurrences of rooted fan triplets of type 1 and type 2 in U in terms of relations between lowest common ancestors in U, $U^{\searrow}$, and $U^{\swarrow}$. By Lemma 2, any pair of vertices u, v in a galled tree U has a unique lowest common ancestor, which will be denoted by $lca^U(u, v)$. Similarly, for any tree T and vertices u, v in T, $lca^T(u, v)$ is the unique lowest common ancestor of u and v in T.

Lemma 5. *Let a, b, c be three leaf labels in a galled tree U. It holds that*:

1. *The rooted fan triplet $a|b|c$ is consistent with U and $a|b|c$ is of type* 1 *in U if and only if either*:
   (i) *$lca^U(a, b) = lca^U(a, c) = lca^U(b, c) = w$ for some vertex w, w is equal to all of $lca(a, b)$, $lca(a, c)$, and $lca(b, c)$ in one of $U^{\searrow}$ and $U^{\swarrow}$, and w is equal to exactly one of $lca(a, b)$, $lca(a, c)$, and $lca(b, c)$ in the other*; *or*
   (ii) *For two (x, y) among $\{(a, b), (a, c), (b, c)\}$, it holds that $lca^U(x, y) = lca^{U^{\searrow}}(x, y) = lca^{U^{\swarrow}}(x, y) = w$ for some vertex w, and for the third pair (x, y), there exists a proper descendant v of w in U such that $v = lca^U(x, y)$ and $v = lca(x, y)$ in exactly one of $U^{\searrow}$ and $U^{\swarrow}$ while $w = lca(x, y)$ in the other.*
2. *The rooted fan triplet $a|b|c$ is consistent with U and $a|b|c$ is of type* 2 *in U if and only if $lca^{U^{\searrow}}(a, b) = lca^{U^{\swarrow}}(a, b) = lca^{U^{\searrow}}(a, c) = lca^{U^{\swarrow}}(a, c) = lca^{U^{\searrow}}(b, c) = lca^{U^{\swarrow}}(b, c)$.*

Proof. $\Rightarrow$) Suppose that $a|b|c$ is consistent with U and let w be the root of $a|b|c$ in U. Let P_a, P_b, and P_c be any three directed, internally vertex-disjoint paths in U from w to a, from w to b, and from w to c. A case analysis reveals that the following distinct cases are possible:

1. w does not lie on a cycle: Then the three paths in $U^{\searrow}$ from w to a, b, and c (not necessarily the same as P_a, P_b, and P_c) are internally vertex-disjoint, i.e., w is still a jca of all pairs of leaves in $\{a, b, c\}$ in $U^{\searrow}$, and $a|b|c$ is consistent with $U^{\searrow}$. The argument can be repeated for $U^{\swarrow}$ so $a|b|c$ is also consistent with $U^{\swarrow}$. Furthermore, we have $w = lca^U(a, b) = lca^{U^{\searrow}}(a, b) = lca^{U^{\swarrow}}(a, b) = lca^U(a, c) = lca^{U^{\searrow}}(a, c) = lca^{U^{\swarrow}}(a, c) = lca^U(b, c) = lca^{U^{\searrow}}(b, c) = lca^{U^{\swarrow}}(b, c)$.
2. w lies on a cycle C but is not the root of C: Then at most one P_a, P_b, and P_c (say P_c) contains edges from C. Thus, $w = lca^U(a, b) = lca^{U^{\searrow}}(a, b) = lca^{U^{\swarrow}}(a, b) = lca^U(a, c) = lca^U(b, c)$. There are two subcases:
   (a) If P_c contains no reticulation edges of C then $a|b|c$ is consistent with both $U^{\searrow}$ and $U^{\swarrow}$ as in case 1, and $w = lca^{U^{\searrow}}(a, c) = lca^{U^{\swarrow}}(a, c) = lca^{U^{\searrow}}(b, c) = lca^{U^{\swarrow}}(b, c)$.
   (b) Otherwise, P_c contains exactly one reticulation edge of C (see Fig. 3 (A) for an example where this occurs). If it is the left one then $a|b|c$ is consistent with $U^{\searrow}$ but not $U^{\swarrow}$, and $lca^{U^{\searrow}}(a, c) = lca^{U^{\searrow}}(b, c) = w$ and $lca^{U^{\swarrow}}(a, c) = lca^{U^{\swarrow}}(b, c) = r$, where r is the root of C. If it is the right one then $a|b|c$ is consistent with $U^{\swarrow}$ but not $U^{\searrow}$, and $lca^{U^{\searrow}}(a, c) = lca^{U^{\searrow}}(b, c) = r$ and $lca^{U^{\swarrow}}(a, c) = lca^{U^{\swarrow}}(b, c) = w$.
3. w is the root of a cycle C: At most two of the three paths P_a, P_b, and P_c can contain edges from C because U is a galled tree. Assume without loss of generality that P_a does not contain any edges from C. Then no path from w to a can intersect P_b or P_c except for in the starting vertex w, so $lca^U(a, b) = lca^U(a, c) = w$ must hold and also $lca^{U^{\searrow}}(a, b) = lca^{U^{\swarrow}}(a, b) = lca^{U^{\searrow}}(a, c) = lca^{U^{\swarrow}}(a, c) = w$. There are two subcases:
   (a) If $lca^U(b, c) = w$ (see Fig. 3 (C) for an example where this occurs) then since w is the root of C, $w = lca^{U^{\searrow}}(b, c) = lca^{U^{\swarrow}}(b, c)$ and $a|b|c$ is consistent with both of $U^{\searrow}$ and $U^{\swarrow}$.
   (b) Otherwise, $lca^U(b, c) = v$ for some vertex v on C with $v \neq w$ (see Fig. 3 (B) for an example where this occurs). Exactly one of b and c (say c) is a descendant of C's reticulation vertex, so the subpath of P_c starting at w and ending at C's reticulation vertex exists in exactly one of $U^{\searrow}$ and $U^{\swarrow}$. It follows that $a|b|c$ is consistent with one of $U^{\searrow}$ and $U^{\swarrow}$, and one of $lca^{U^{\searrow}}(b, c)$ and $lca^{U^{\swarrow}}(b, c)$ is equal to v while the other one is equal to w.

Now, we see that $a|b|c$ is of type 1 in U if and only if case 2 (b) or case 3 (b) occurs, and $a|b|c$ is of type 2 in U if and only if case 1, case 2 (a), or case 3 (a) occurs. Furthermore, condition 1 (i) in the lemma statement is equivalent to case 2 (b), and condition 1 (ii) is equivalent to case 3 (b).

$\Leftarrow$) The rooted fan triplet $a|b|c$ is consistent with $U^{\searrow}$ if and only if $lca^{U^{\searrow}}(a, b) = lca^{U^{\searrow}}(a, c) = lca^{U^{\searrow}}(b, c)$, and analogously for $U^{\swarrow}$. Therefore, if either condition 1 (i) or condition 1 (ii) in the lemma statement is true then $a|b|c$ is consistent with exactly one of $U^{\searrow}$ and $U^{\swarrow}$, i.e., $a|b|c$ is of type 1 in U, and if $lca^{U^{\searrow}}(a, b) = lca^{U^{\swarrow}}(a, b) = lca^{U^{\searrow}}(a, c) = lca^{U^{\swarrow}}(a, c) = lca^{U^{\searrow}}(b, c) = lca^{U^{\swarrow}}(b, c)$ then $a|b|c$ is of type 2 in U. □

4. Computing the rooted triplet distance between galled trees

In this section, we apply the triangle counting techniques from Section 2 to obtain a subcubic-time algorithm for computing the rooted triplet distance between two galled trees U_1 and U_2 with the same set L of leaf labels. We first explain how to compute the number of rooted fan triplets consistent with both networks in Section 4.1 and then the number of rooted binary triplets consistent with both networks in Section 4.2. Combining these two results gives us our main result (Theorem 4) in Section 4.3.

4.1. Counting the number of shared rooted fan triplets

To count the number of rooted fan triplets consistent with both of the two galled trees U_1 and U_2, we apply Theorems 2 and 3 from Section 2 as detailed below. As a warm-up, we first present a simple reduction from the problem of counting rooted fan triplets shared by two *trees* to the problem of counting monochromatic triangles in a graph.

Lemma 6. *Let T_1 and T_2 be two phylogenetic trees with the same set L of n leaf labels. The number of rooted fan triplets consistent with both T_1 and T_2 can be computed in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time.*

Proof. Create an auxiliary undirected, edge-colored, complete graph $\mathcal{G} = (L, E)$ in which every edge is assigned a color of the form (v_1, v_2), where v_1 is a vertex of T_1 and v_2 is a vertex of T_2, as follows.

- For each edge $\{u, v\} \in E$: Let x_1 be the lca of u and v in T_1, let x_2 be the lca of u and v in T_2, and color the edge $\{u, v\}$ in $\mathcal{G}$ with the color (x_1, x_2).

The key observation is that for any three leaf labels a, b, c in L, the rooted fan triplet $a|b|c$ is consistent with T_1 if and only if the lca's in T_1 of a and b, of a and c, and of b and c are identical. The same holds for T_2. Therefore, $a|b|c$ is consistent with both T_1 and T_2 if and only if all three edges $\{a, b\}$, $\{a, c\}$, $\{b, c\}$ have the same color in $\mathcal{G}$. It follows that the number of rooted fan triplets common to both trees equals the number of monochromatic triangles in $\mathcal{G}$.

By Lemma 2, $\mathcal{G}$ can be constructed in $O(n^2)$ time. By Theorem 2, we can compute the number of rooted fan triplets that are consistent with both T_1 and T_2 in $O(n^{(3+\omega)/2})$ time. □

Next, we address the more complicated *galled tree* case.

Recall from Sections 3.3 and 3.4 that $U^{\searrow}$ and $U^{\swarrow}$ are two trees obtained from a galled tree U by removing edges incident to the reticulation vertices, and that every rooted fan triplet in U is partitioned into one of two types depending on if it is consistent with one or two of $U^{\searrow}$ and $U^{\swarrow}$. For $p, q \in \{1, 2\}$, let $T_{p,q}$ denote the number of rooted fan triplets consistent with both U_1 and U_2 that are of type p in U_1 and of type q in U_2. Our next goal is to compute the sum $T_{1,1} + T_{1,2} + T_{2,1} + T_{2,2}$, which equals the number of rooted fan triplets consistent with both U_1 and U_2.

First, construct the four trees $U_1^{\searrow}$, $U_1^{\swarrow}$, $U_2^{\searrow}$, and $U_2^{\swarrow}$ defined in Section 3.3 in $O(n)$ time. Then, Lemma 7 below accomplishes our goal by computing $T_{1,1} + 2T_{1,2} + 2T_{2,1} + 4T_{2,2}$ and subtracting multiples of $T_{1,2}$, $T_{2,1}$, and $T_{2,2}$.

Proposition 1. *The sum $T_{1,1} + 2T_{1,2} + 2T_{2,1} + 4T_{2,2}$ can be computed in $O(n^{(3+\omega)/2})$ time.*

Proof. For $i \in \{1, 2\}$, each rooted fan triplet of type 1 in U_i is consistent with one of $U_i^{\searrow}$ and $U_i^{\swarrow}$, while each rooted fan triplet of type 2 in U_i is consistent with both of them. If we sum the number of rooted fan triplets shared between $U_1^{\searrow}$ and $U_2^{\searrow}$, between $U_1^{\searrow}$ and $U_2^{\swarrow}$, between $U_1^{\swarrow}$ and $U_2^{\searrow}$, and between $U_1^{\swarrow}$ and $U_2^{\swarrow}$ then every rooted fan triplet consistent with U_1 and U_2 that is of type 1 in both U_1 and U_2 is counted once, while any shared rooted fan triplet that is of different types in U_1 and U_2 is counted twice, and any shared rooted fan triplet of type 2 in both U_1 and U_2 is counted four times. Hence, the computed sum equals $T_{1,1} + 2T_{1,2} + 2T_{2,1} + 4T_{2,2}$.

Using Lemma 6, we can find the number of shared rooted fan triplets between $U_1^{\searrow}$ and $U_2^{\searrow}$, between $U_1^{\searrow}$ and $U_2^{\swarrow}$, between $U_1^{\swarrow}$ and $U_2^{\searrow}$, and between $U_1^{\swarrow}$ and $U_2^{\swarrow}$ in $O(n^{(3+\omega)/2})$ time. □

Proposition 2. *The value of $T_{2,2}$ can be computed in $O(n^{(3+\omega)/2})$ time.*

Proof. We adapt the idea used in the proof of Lemma 6. As before, create an auxiliary complete graph $\mathcal{G} = (L, E)$, but now assign the edge colors in $\mathcal{G}$ as follows.

- For each edge $\{u, v\} \in E$: For $i \in \{1, 2\}$, let y_i be the lca of u and v in $U_i^{\searrow}$ and z_i the lca of u and v in $U_i^{\swarrow}$. If $y_1 = z_1$ and $y_2 = z_2$ then color $\{u, v\}$ with the color (y_1, y_2); otherwise, color $\{u, v\}$ with a null color that never occurs again in $\mathcal{G}$.

According to Lemma 5-2, any three leaf labels a, b, c in a galled tree U form a rooted fan triplet of type 2 in U if and only if $lca^{U^{\searrow}}(a, b) = lca^{U^{\swarrow}}(a, b) = lca^{U^{\searrow}}(a, c) = lca^{U^{\swarrow}}(a, c) = lca^{U^{\searrow}}(b, c) = lca^{U^{\swarrow}}(b, c)$. Hence, by applying Theorem 2 to count the number of monochromatic triangles in $\mathcal{G}$, we get the number of shared rooted fan triplets that are of type 2 in both U_1 and U_2 in $O(n^{(3+\omega)/2})$ time. □

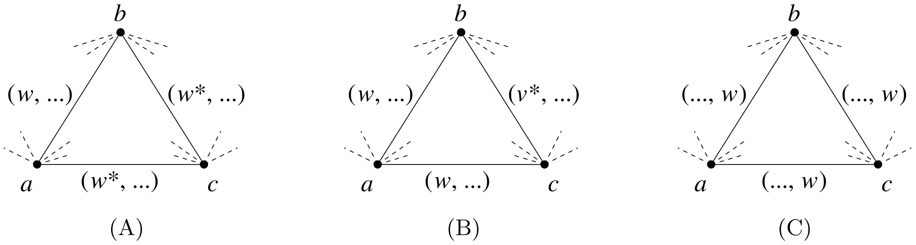

Fig. 4. In the proof of Proposition 3, the edges in the auxiliary graph $\mathcal{G}$ are colored so that any three vertices a, b, c form an R-chromatic triangle if and only if $a|b|c$ is consistent with both U_1 and U_2 and is of type 1 in U_1 and of type 2 in U_2. (A) occurs when U_1 satisfies the conditions in Lemma 5-1 (i), (B) occurs when U_1 satisfies the conditions in Lemma 5-1 (ii), and (C) occurs when U_2 satisfies the conditions in Lemma 5-2.

Proposition 3. *The values of $T_{1,2}$ and $T_{2,1}$ can be computed in $O(n^{(3+\omega)/2})$ time.*

Proof. To compute $T_{1,2}$, we again use the idea from the proof of Lemma 6. Create an auxiliary undirected, complete graph $\mathcal{G} = (L, E)$ and color each edge $\{u, v\} \in E$ as described next. Recall from Lemma 2 that lca's in a galled tree are unique.

- Let x_1 be the lca of u and v in U_1, let y_2 be the lca of u and v in $U_2^{\searrow}$, and let z_2 be the lca of u and v in $U_2^{\swarrow}$. If x_1 is the lca of u and v in both of $U_1^{\searrow}$ and $U_1^{\swarrow}$ while $y_2 = z_2$ then $\{u, v\}$ is colored with (x_1, y_2). On the other hand, if x_1 is the lca of u and v in exactly one of $U_1^{\searrow}$ and $U_1^{\swarrow}$ while $y_2 = z_2$ then $\{u, v\}$ is colored with $(x_1{}^*, y_2)$. Otherwise, $\{u, v\}$ is colored with a null color that never occurs again in $\mathcal{G}$.

Here, colors containing an asterisk symbol are used to indicate that the lca in U_1 of two leaves in L is also the lca in one of, but not both of, $U_1^{\searrow}$ and $U_1^{\swarrow}$. Lemma 5-1 implies that $a|b|c$ is a rooted fan triplet of type 1 in U_1 if and only if either:

(i) two of the three edges $\{a, b\}$, $\{a, c\}$, and $\{b, c\}$ in $\mathcal{G}$ are assigned a color of the form $(w^*, \ldots)$ while the third one is assigned a color of the form $(w, \ldots)$, as in Fig. 4 (A); or
(ii) two of them are assigned a color of the form $(w, \ldots)$ and the third one a color of the form $(v^*, \ldots)$, where v is a proper descendant of w, as in Fig. 4 (B).

Moreover, by Lemma 5-2, $a|b|c$ is a rooted fan triplet of type 2 in U_2 if and only if $lca^{U_2^{\searrow}}(a, b) = lca^{U_2^{\swarrow}}(a, b) = lca^{U_2^{\searrow}}(a, c) = lca^{U_2^{\swarrow}}(a, c) = lca^{U_2^{\searrow}}(b, c) = lca^{U_2^{\swarrow}}(b, c)$, i.e., if all of the three edges $\{a, b\}$, $\{a, c\}$, and $\{b, c\}$ have a color of the form $(\ldots, w)$, as in Fig. 4 (C).

Thus, if we define a binary relation R on the edge colors of $\mathcal{G}$ by:

- $(i_1, i_2) R (k_1, k_2)$ holds if and only if: (i) either $i_1 = k_1{}^*$ or $k_1 = j^*$, where j is a proper descendant of i_1; and (ii) $i_2 = k_2$.

Then the number of R-chromatic triangles in $\mathcal{G}$ equals $T_{1,2}$.

The four trees $U_1^{\searrow}$, $U_1^{\swarrow}$, $U_2^{\searrow}$, and $U_2^{\swarrow}$ can be preprocessed in $O(n)$ time to support $O(1)$-time *lca* queries [13], and we can spend $O(n^2)$ time to build a data structure supporting $O(1)$-time proper descendant queries for U_1. After that, we can apply Theorem 3 to $\mathcal{G}$ to obtain $T_{1,2}$ in $O(n^{(3+\omega)/2})$ time.

The value of $T_{2,1}$ can be computed in $O(n^{(3+\omega)/2})$ time in the same way. □

By combining Propositions 1, 2, and 3, the sum $T_{1,1} + T_{1,2} + T_{2,1} + T_{2,2}$ can be computed in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time, which yields the next lemma.

Lemma 7. *Let U_1 and U_2 be two galled trees with the same set of n leaf labels. The number of rooted fan triplets consistent with both U_1 and U_2 can be computed in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time.*

4.2. Counting the number of shared rooted binary triplets

Let U_1 and U_2 be two galled trees with the same set L of leaf labels. This subsection describes how to compute the number of rooted binary triplets consistent with both U_1 and U_2.

First, for every pair of leaf labels in $L \times L$, compute their junction common ancestors in U_1 as well as in U_2. For each such jca, store information about if it is located on a cycle and if so, whether it is the root of the cycle and whether it relies on the reticulation vertex of the cycle. By applying Lemma 2, this can be done in $O(n^2)$ time.

Next, partition the set of pairs of distinct leaf labels from L into *classes* according to their jca's in U_1 and U_2 as follows. Define the class $C_{v_1^{f_1}, v_2^{f_2}}$, where v_i is a vertex in U_i for $i \in \{1,2\}$ and $f_1, f_2 \in \{0,1\}$, so that any pair $(a,b) \in L \times L$ with $a \neq b$ belongs to $C_{v_1^{f_1}, v_2^{f_2}}$ if and only if the following three conditions hold for both $i \in \{1,2\}$:

1. v_i is a jca of a and b in U_i;
2. $f_i = 1$ if and only if v_i is located on a cycle of U_i and v_i relies on the reticulation vertex of the cycle; and
3. if v_i is the root of a cycle in U_i then there is no other jca of a and b in U_i.

For example, if U_1 is the galled tree in Fig. 3 (B) and w_1 denotes the vertex labeled w in this figure, and U_2 is the galled tree in Fig. 3 (C) and w_2 denotes its vertex labeled w, then we have $(a,b) \in C_{w_1^0, w_2^0}$, $(a,c) \in C_{w_1^1, w_2^0}$, and $(b,c) \in C_{v^1, w_2^0}$.

The third condition in the definition is needed to ensure that the resulting classes are disjoint. According to Lemma 2, any pair of leaves a and b in a galled tree has at most two jca's. Moreover, if there are two such jca's v and w then they are located on the same cycle and one of them (say w) will be the root of the cycle. Since any path from an ancestor of w ending at w can be extended to reach v, from the point of view of a rooted binary triplet $ab|c$, it is sufficient to consider the jca v which is not the root of the cycle. By obeying condition 3, (a,b) will only be placed in some class whose index involves v and not in any class whose index involves w. We obtain the following proposition.

Proposition 4. *The classes $C_{v_1^{f_1}, v_2^{f_2}}$ are disjoint. They can be constructed in $O(n^2)$ time by integer sorting.*

Next, construct two binary matrices M_1 and M_2 from U_1 and U_2, respectively. For $i \in \{1,2\}$, M_i contains two rows for each vertex v in U_i (indexed by v^0 and v^1, and ordered so that row v^0 precedes row v^1) and n columns corresponding to the leaf labels in L. Define the matrix entries according to:

- $M_i[v^0, c] = 1$ if and only if there exists a jca w in U_i of v and the leaf c such that $v \neq w$.
- $M_i[v^1, c] = 1$ if and only if v belongs to a cycle in U_i and there exists a jca w in U_i of v and the leaf c such that $v \neq w$ and w does not rely on the reticulation vertex of the cycle that v belongs to.

Observe that $M_i[v^1, c] = 1$ implies $M_i[v^0, c] = 1$. Also observe that:

Proposition 5. *For every $c \in L$, if some matrix entry $M_i[v_i^{f_i}, c] = 1$ then c cannot occur in any leaf label pair belonging to a class of the form $C_{v_1^{f_1}, v_2^{f_2}}$.*

Proof. Let a be any leaf label in L with $a \neq c$ and let $C_{v_1^{f_1}, v_2^{f_2}}$ be the class that contains the pair (a,c). By definition, there are two paths in U_1 from v_1 to a and from v_1 to c that are vertex-disjoint except for in the common start vertex v_1. There are two cases:

1. $f_1 = 0$: Then either (i) v_1 does not lie on a cycle, or (ii) v_1 lies on a cycle and does not rely on the cycle's reticulation vertex. In both (i) and (ii), there is no jca w in U_1 of v_1 and c with $w \neq v_1$, so $M_1[v_1^0, c] = 0$.
2. $f_1 = 1$: Then v_1 lies on a cycle and relies on the reticulation vertex of the cycle, which means that exactly one of a and c is a descendant of the reticulation vertex. If it is a then there exists no jca w in U_1 of v_1 and c with $w \neq v_1$; if it is c then the root of the cycle is a jca w of v_1 and c with $w \neq v_1$, but both v_1 and w rely on the cycle's reticulation vertex. Thus, $M_1[v_1^1, c] = 0$.

Analogous arguments hold for U_2. Therefore, $M_i[v_i^{f_i}, c] = 0$ for $i \in \{1,2\}$. □

In the next step, compute the matrix product $Q = M_1 \times M_2^t$ in $O(n^{\omega})$ time. From the definitions of M_1 and M_2, we have:

Proposition 6. *Each entry $Q[v_1^{f_1}, v_2^{f_2}]$ equals the number of leaf labels in L having a junction common ancestor with v_1 in U_1 not equal to v_1 as well as a junction common ancestor with v_2 in U_2 not equal to v_2 that, furthermore, do not rely on the reticulation vertex of the cycle which v_i lies on if $f_i = 1$ for $i \in \{1,2\}$.*

Now, consider any specified vertex v_1 in U_1, vertex v_2 in U_2, and $f_1, f_2 \in \{0,1\}$. Suppose that $c \in L$ is a leaf label that contributes to the value of $Q[v_1^{f_1}, v_2^{f_2}]$ in Proposition 6, i.e., $M_1[v_1^{f_1}, c] = M_2[v_2^{f_2}, c] = 1$, and that (a,b) is a pair in $C_{v_1^{f_1}, v_2^{f_2}}$. (According to Proposition 5, c does not occur in any pair belonging to $C_{v_1^{f_1}, v_2^{f_2}}$.) For each $i \in \{1,2\}$, if v_i lies on a cycle in U_i and $f_i = 0$ then v_i does not rely on the reticulation vertex of this cycle by the definition of $C_{v_1^{f_1}, v_2^{f_2}}$; if v_i lies on a cycle in U_i and $f_i = 1$ then the jca of v_i and c does not rely on the reticulation vertex by the definition of $M_i[v_i^1, c]$. It follows

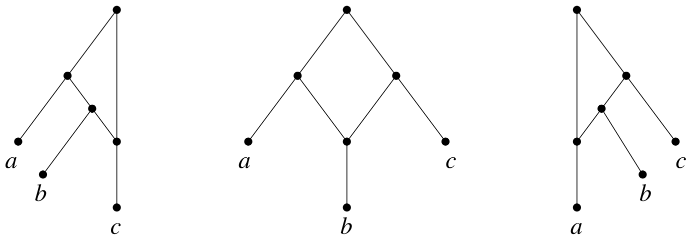

Fig. 5. An example given in Fig. 5 in [11]. The two rooted triplets $ab|c$ and $bc|a$ are consistent with each of the three non-isomorphic galled trees shown above. Note that there is one cycle of length 4 in each network.

from Lemma 3 that the rooted binary triplet $ab|c$ is consistent with both U_1 and U_2. Also, for $\{f_1, f_2\} \neq \{f'_1, f'_2\}$, it holds by Proposition 4 that $C_{v_1^{f_1}, v_2^{f_2}} \cap C_{v_1^{f'_1}, v_2^{f'_2}} = \emptyset$. Thus, the sum

$$\sum_{f_1, f_2 \in \{0,1\}} |C_{v_1^{f_1}, v_2^{f_2}}| \cdot Q\left[v_1^{f_1}, v_2^{f_2}\right]$$

equals the number of rooted binary triplets of the form $ab|c$ consistent with both U_1 and U_2 that use v_i as a jca of a and b in U_i for $i \in \{1, 2\}$, with the exception of the case where v_1 or v_2 is the root of a cycle and there is another jca of a and b that is a descendant of this vertex. Given the $C_{v_1^{f_1}, v_2^{f_2}}$-classes and Q, it suffices to compute the sum

$$\sum_{v_1 \in U_1} \sum_{v_2 \in U_2} \sum_{f_1, f_2 \in \{0,1\}} |C_{v_1^{f_1}, v_2^{f_2}}| \cdot Q\left[v_1^{f_1}, v_2^{f_2}\right]$$

to obtain the total number of rooted binary triplets consistent with both U_1 and U_2, which takes $O(n^2)$ time.

In summary, we have shown the following lemma:

Lemma 8. *Let U_1 and U_2 be two galled trees with the same set of n leaf labels. The number of rooted binary triplets consistent with both U_1 and U_2 can be computed in $O(n^{\omega}) \leqslant o(n^{2.373})$ time.*

4.3. Computing the rooted triplet distance

By combining the results in the previous two subsections, we obtain:

Theorem 4. *Let U_1 and U_2 be two galled trees with the same set of n leaf labels. The rooted triplet distance $d_{rt}(U_1, U_2)$ can be computed in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time.*

Proof. For $i \in \{1, 2\}$, let F_i be the set of rooted fan triplets consistent with U_i, and let B_i be the set of rooted binary triplets consistent with U_i. We have $d_{rt}(U_1, U_2) = \sum_{i=1}^{2}(|F_i| + |B_i|) - 2|F_1 \cap F_2| - 2|B_1 \cap B_2|$. Compute $|F_i \cap F_i| = |F_i|$ and $|F_1 \cap F_2|$ in $O(n^{(3+\omega)/2}) \leqslant o(n^{2.687})$ time using Lemma 7, and compute $|B_i \cap B_i| = |B_i|$ and $|B_1 \cap B_2|$ in $O(n^{\omega}) \leqslant o(n^{2.373})$ time using Lemma 8. □

5. Concluding remarks

We have demonstrated that the rooted triplet distance can be computed in subcubic time for a well-known class of phylogenetic networks called galled trees [12,14]. More precisely, we have presented a new $o(n^{2.687})$-time algorithm for computing the rooted triplet distance between two input galled trees with n leaves each and identical leaf label sets (Theorem 4). We have also derived three results on counting triangles in a graph (Theorems 1–3) that may have other applications. The first two triangle counting results are generalizations of their known (weaker) detection counterparts from [1] and [24], respectively.

Some criteria for when a phylogenetic network is uniquely defined by a set of rooted triplets were established by Gambette and Huber in [11]. Significantly, Corollary 1 in [11] states that if U is a binary galled tree (i.e., a galled tree whose vertices have outdegree at most 2 and whose reticulation vertices have indegree 2 and outdegree 1) containing b cycles of length 4, then there are 3^b non-isomorphic binary galled trees that are consistent with the same set of rooted triplets as U. Fig. 5 shows an example from [11]. Since two non-isomorphic galled trees U_1, U_2 may satisfy $d_{rt}(U_1, U_2) = 0$, it immediately follows that the rooted triplet distance is not a metric for the class of galled trees. However, Corollary 2 in [11] shows that d_{rt} is a metric for the subclass of binary galled trees that do not have any cycles of length 4.

An alternative extension of the rooted triplet distance from phylogenetic trees to phylogenetic networks was proposed by Cardona et al. in [4]. It works for the class of *tree-child time consistent phylogenetic networks* [4]. For each cardinality-3 subset L' of the leaf label set, instead of looking at the rooted binary triplets or rooted fan triplets over L' in the two networks, it considers the subnetworks induced by the so-called least common semistrict ancestors of the leaves in L'; consequently, "rooted triplets" of different types than rooted binary triplets and rooted fan triplets are possible. This leads to an $O(n^5)$-time, breadth-first-search-based algorithm [4]. Although Cardona et al.'s [4] extension of the rooted triplet distance is more involved than Gambette and Huber's [11] canonical extension studied in this paper (Definition 1) and is still not a metric for the class of galled trees (see Fig. 19 in [4]), it has some other nice mathematical properties. It might be beneficial to examine the relationship between the two alternatives and try to speed up the $O(n^5)$-time algorithm of [4] by using fast triangle counting or matrix multiplication techniques.

Nielsen et al. [20] showed how to compute the *unrooted quartet distance* between two *unrooted* phylogenetic trees with n leaves in $o(n^{2.687})$ time. Interestingly, they also rely on matrix multiplication. Their method does not count triangles in an auxiliary graph as we have done here, but uses matrix multiplication to count so-called *shared* and *different butterflies* between the two input trees directly. In some sense, their problem may be inherently "easier" than ours as it does not involve cycles. Indeed, much of the conceptual complexity in Section 4 above stems from the non-uniqueness of junction common ancestors in galled trees; for example, it is more complicated to prove Lemma 7 than Lemma 6 because of this issue.

Finally, we list some open problems.

1. Does the problem of computing the rooted triplet distance $d_{rt}(U_1, U_2)$ between two galled trees U_1, U_2 admit a quadratic-time algorithm or not?
2. Can our method be extended to even larger classes of phylogenetic networks than the galled trees? For example, can it be extended to *level-k phylogenetic networks* [7,14] for any positive integer k? As shown in [21], the class of galled trees forms a subset of the class of level-1 phylogenetic networks, with equality occurring when restricted to networks whose vertices have outdegree at most 2 and whose reticulation vertices have indegree 2 and outdegree 1.
3. From a practical point of view, an experimental analysis based on real data or simulations would be useful to validate the rooted triplet distance as a means to compare phylogenetic networks. In particular, how well does the rooted triplet distance capture the notion of dissimilarity between galled trees in practice?

References

[1] N. Alon, R. Yuster, U. Zwick, Finding and counting given length cycles, Algorithmica 17 (3) (1997) 209–223.

[2] M.S. Bansal, J. Dong, D. Fernández-Baca, Comparing and aggregating partially resolved trees, Theor. Comput. Sci. 412 (48) (2011) 6634–6652.

[3] G.S. Brodal, R. Fagerberg, T. Mailund, C.N.S. Pedersen, A. Sand, Efficient algorithms for computing the triplet and quartet distance between trees of arbitrary degree, in: Proceedings of the 24th Annual ACM–SIAM Symposium on Discrete Algorithms, SODA 2013, 2013, pp. 1814–1832.

[4] G. Cardona, M. Llabrés, R. Rosselló, G. Valiente, Metrics for phylogenetic networks II: Nodal and triplets metrics, IEEE/ACM Trans. Comput. Biol. Bioinform. 6 (3) (2009) 454–469.

[5] G. Cardona, M. Llabrés, R. Rosselló, G. Valiente, Comparison of galled trees, IEEE/ACM Trans. Comput. Biol. Bioinform. 8 (2) (2011) 410–427.

[6] H.-L. Chan, J. Jansson, T.-W. Lam, S.-M. Yiu, Reconstructing an ultrametric galled phylogenetic network from a distance matrix, J. Bioinform. Comput. Biol. 4 (4) (2006) 807–832.

[7] C. Choy, J. Jansson, K. Sadakane, W.-K. Sung, Computing the maximum agreement of phylogenetic networks, Theor. Comput. Sci. 335 (1) (2005) 93–107.

[8] D.E. Critchlow, D.K. Pearl, C. Qian, The triples distance for rooted bifurcating phylogenetic trees, Syst. Biol. 45 (3) (1996) 323–334.

[9] A.J. Dobson, Comparing the shapes of trees, in: Proceedings of the Third Australian Conference on Combinatorial Mathematics (Combinatorial Mathematics III), in: Lecture Notes in Mathematics, vol. 452, Springer, 1975, pp. 95–100.

[10] J. Felsenstein, Inferring Phylogenies, Sinauer Associates, Inc., Sunderland, MA, 2004.

[11] P. Gambette, K.T. Huber, On encodings of phylogenetic networks of bounded level, J. Math. Biol. 65 (1) (2012) 157–180.

[12] D. Gusfield, S. Eddhu, C. Langley, Optimal, efficient reconstruction of phylogenetic networks with constrained recombination, J. Bioinform. Comput. Biol. 2 (1) (2004) 173–213.

[13] D. Harel, R.E. Tarjan, Fast algorithms for finding nearest common ancestors, SIAM J. Comput. 13 (2) (1984) 338–355.

[14] D.H. Huson, R. Rupp, C. Scornavacca, Phylogenetic Networks: Concepts, Algorithms and Applications, Cambridge University Press, 2010.

[15] L. van Iersel, S. Kelk, Constructing the simplest possible phylogenetic network from triplets, Algorithmica 60 (2) (2011) 207–235.

[16] J. Jansson, N.B. Nguyen, W.-K. Sung, Algorithms for combining rooted triplets into a galled phylogenetic network, SIAM J. Comput. 35 (5) (2006) 1098–1121.

[17] M.K. Kuhner, J. Felsenstein, A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates, Mol. Biol. Evol. 11 (3) (1994) 459–468.

[18] D. Morrison, Introduction to Phylogenetic Networks, RJR Productions, 2011.

[19] L. Nakhleh, T. Warnow, D. Ringe, S.N. Evans, A comparison of phylogenetic reconstruction methods on an Indo-European dataset, Trans. Philol. Soc. 103 (2) (2005) 171–192.

[20] J. Nielsen, A.K. Kristensen, T. Mailund, C.N.S. Pedersen, A sub-cubic time algorithm for computing the quartet distance between two general trees, Algorithms Mol. Biol. 6 (2011) (Article 15).

[21] F. Rosselló, G. Valiente, All that glisters is not galled, Math. Biosci. 221 (1) (2009) 54–59.

[22] C. Semple, M. Steel, Phylogenetics, Oxford Lecture Series in Mathematics and its Applications, vol. 24, Oxford University Press, 2003.

[23] L. Wang, B. Ma, M. Li, Fixed topology alignment with recombination, Discrete Appl. Math. 104 (1–3) (2000) 281–300.

[24] V. Vassilevska, R. Williams, R. Yuster, Finding heaviest H-subgraphs in real weighted graphs, with applications, ACM Trans. Algorithms 6 (3) (2010) (Article 44).

[25] V. Vassilevska Williams, Multiplying matrices faster than Coppersmith–Winograd, in: Proceedings of the 44th ACM Symposium on Theory of Computing, STOC 2012, 2012, pp. 887–898.