# Computing the Rooted Triplet Distance between Galled Trees by Counting Triangles

Jesper Jansson[1,⋆] and Andrzej Lingas[2,⋆⋆]

[1] Laboratory of Mathematical Bioinformatics, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611-0011, Japan
jj@kuicr.kyoto-u.ac.jp
[2] Department of Computer Science, Lund University, 22100 Lund, Sweden
Andrzej.Lingas@cs.lth.se

**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 $o(n^{2.688})$ time. Our upper bound is obtained by reducing the problem of computing the rooted triplet distance 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 related results that may be of independent interest.

## 1 Introduction

Phylogenetic trees and their generalization to non-treelike structures, *phylogenetic networks*, are commonly used by scientists to describe evolutionary relationships among a set of objects such as biological species or natural languages [2, 3, 6–8, 10–14]. Various metrics for measuring the (dis-)similarity of two given phylogenetic trees have been proposed and analyzed in the literature; see, e.g., [2] and the references therein. In this paper, we consider an extension of one particular, well-known method called the *rooted triplet distance* [2, 6] to the phylogenetic network model and describe how to compute it efficiently.

The rooted triplet distance between two phylogenetic trees provides an intuitive measure of their dissimilarity and exhibits many attractive mathematical properties [2, 6]. It counts the number of substructures (more precisely, subtrees induced by three leaves) that differ between the two trees. More formally, it 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 *non-binary* rooted triplet leaf-labeled by $\{a, b, c\}$ is called a *rooted fan triplet* and is denoted by $a|b|c$ (see the leftmost tree in Fig. 1), and a *binary* rooted triplet

⋆ Funded by The Hakubi Project and KAKENHI grant number 23700011.
⋆⋆ Research supported in part by VR grant 621-2008-4649.

J. Kärkkäinen and J. Stoye (Eds.): CPM 2012, LNCS 7354, pp. 385–398, 2012.

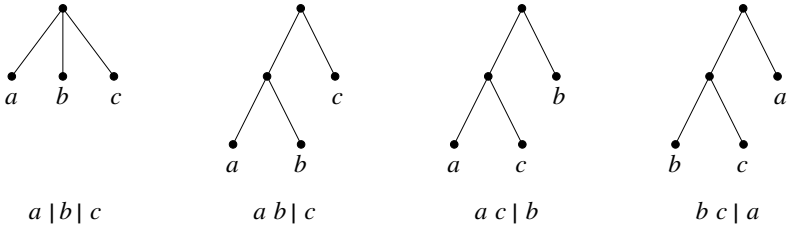


**Fig. 1.** The rooted fan triplet $a|b|c$ and the rooted proper triplets $ab|c$, $ac|b$, and $bc|a$

is called a *rooted proper 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 also Fig. 1). A rooted triplet $t$ is said to be *consistent with* a rooted phylogenetic tree $T$ if $t$ is an embedded subtree of $T$. [1] 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 naive algorithm for computing $d_{rt}(T_1, T_2)$ between two 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 [9, 17], 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.* [6] 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.

Due to the recently increasing focus on phylogenetic networks (see, e.g., the two new textbooks [10, 13]), it is compelling to consider generalizations of the rooted triplet distance to the network case. For this case, 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* [8, 10] admit fast algorithms for the rooted triplet distance. Galled trees are structurally restricted phylogenetic networks in which all underlying cycles are vertex-disjoint (for a detailed definition, refer to Section 2.2 below). This kind of phylogenetic network was first considered by Wang *et al.* [18] and later by Gusfield *et al.* [8], and is also known in the literature as a *level*-1 *phylogenetic network* [4, 10]. Galled trees have turned out to be useful in certain settings where reticulation events do occur but are known to be rare. [2] As a consequence, a number of algorithms for building galled trees from different kinds of data have been published [3, 8, 10–12].

[1] There are several equivalent ways to define this. For example, for any two leaf labels $x, y$, let $lca^T(x, y)$ denote the lowest common ancestor in $T$ of the leaves labeled by $x$ and $y$. Then $a|b|c$ 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 $lca^T(a, c) = lca^T(b, c)$. See also Section 2.1 below.

[2] See [8] for a discussion about the biological relevance of galled trees.

### 1.1 New Results

The main contribution of our paper is an $o(n^{2.688})$-time algorithm for computing the rooted triplet distance between two galled trees with $n$ leaves each [Theorem 4]. From a computational complexity point of view, this is significant because it breaks the natural $O(n^3)$-time barrier for any kind of non-tree phylogenetic networks for the first time. The precise running time is $O(n^{(3+\omega)/2})$, where $\omega$ denotes the exponent in the running time of the fastest existing method for matrix multiplication. It is well known that $\omega < 2.376$ [5], and recent developments [16, 20] suggest slightly tighter bounds on $\omega$.

Our main result 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 efficiently, we strengthen a technique based on matrix multiplication used in [1] and [19] 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 graph with $m$ edges can be computed in $O(m^{\frac{2\omega}{\omega+1}}) \leq o(m^{1.408})$ 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 $O(n^{(3+\omega)/2}) \leq o(n^{2.688})$ time [Theorem 2].

We also need to relax the concept of a monochromatic triangle to what we call an $R$-chromatic triangle (see Section 3 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 $O(n^{(3+\omega)/2}) \leq o(n^{2.688})$ time [Theorem 3].

Our 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.

## 2 Preliminaries

### 2.1 Basic Definitions

A (rooted) *phylogenetic network* $U$ is a directed acyclic graph with a single root vertex and a set $L$ of distinctly labeled leaves, and no vertices having both indegree 1 and outdegree 1. 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 non-zero length paths from $w$ to $u$ and $v$, 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 $v$. As an example, in Fig. 2 (i), 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 $c$.

We now define rooted triplet consistency for a phylogenetic network $U$. Following [10, 11], for any three leaf labels $a, b, c$, say that the rooted proper 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 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 $z$. Secondly, say that the rooted fan triplet $a|b|c$ is *consistent with* $U$ if and only if $U$ contains a vertex $w$ such that there are 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 $w$. Observe that in the special case where $U$ 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 thus reduce to the definitions in footnote 1.

Next, we define the rooted triplet distance between phylogenetic networks 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 proper triplets with leaf labels from $L$ that are consistent with exactly one of $U_1$ and $U_2$.*

This definition of $d_{rt}$ differs slightly from the one restricted to trees in [2, 6]. The definition in [2, 6] 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 $U_2$. Therefore, when restricted to trees, our definition of $d_{rt}$ is exactly two times $d_{rt}$ from [2, 6] because each "bad" subset will contribute twice to our $d_{rt}$ (once for the rooted triplet in $U_1$ and once for the rooted triplet in $U_2$); obviously, our definition of $d_{rt}$ could be normalized by dividing by two but then $d_{rt}$ would no longer always be an integer in the non-tree case. We believe that our definition is more suitable in the context of phylogenetic networks because it allows us to distinguish between cases such as: (i) $ab|c$ and $bc|a$ are consistent with $U_1$ whereas only $bc|a$ is consistent with $U_2$; and (ii) $ab|c$ is consistent with $U_1$ and $bc|a$ is consistent with $U_2$.

### 2.2 Galled Trees

Here, we recall the definition of the class of phylogenetic networks called the *galled tree* [8, 10], and investigate some of its properties.

A *reticulation vertex* of 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 phylogenetic network $U$ is called a *galled tree* if all cycles

in its underlying undirected graph are vertex-disjoint [8, 10]. A *cycle* $C$ in a galled tree is any set of vertices that induce a cycle in the underlying undirected graph, and the vertex of $C$ in $U$ that is an ancestor of all vertices on $C$ is called the *root* of $C$. Thus, every cycle $C$ in a galled tree has exactly one root and one reticulation vertex, and $C$ consists of two directed 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 must pass through the root of the cycle. The next lemma summarizes some useful properties of galled trees:

**Lemma 1.** *Let $U$ be a galled tree with $n$ leaves and let $u, v$ be any two vertices in $U$. Then:*

1. *The lowest common ancestor in $U$ of $u$ and $v$ is unique.*
2. *There are at most two different junction common ancestors of $u$ and $v$.*
3. *If there are two junction common ancestors of $u$ and $v$ 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.* To prove property 1, suppose there were two different lowest common ancestors $w_1$ 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 $w_2$. In the same way, 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 a lowest common ancestor 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.

Next, we prove properties 2 and 3. For each cycle in $U$, arbitrarily term one of the two edges on $C$ incident to the reticulation vertex as *the left reticulation edge* and the other one as *the right reticulation edge*. Let $U_L$ be the tree obtained from $U$ by removing all right reticulation edges in $U$ and define $U_R$ symmetrically. Then, every junction common ancestor of $u$ and $v$ in $U$ is a lowest common ancestor of $u$ and $v$ in at least one of $U_L$ and $U_R$. Property 2 follows. According to the definitions, if $w$ is a lowest common ancestor of $u$ and $v$ in $U$ then $w$ is also a junction common ancestor of $u$ and $v$ in $U$, which yields property 3.

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 [4], the number of vertices in any binary

galled tree $U'$ with $n$ leaves is $O(n)$, so this gives an upper bound for $U$ as well. 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 $O(n)$. Thus, property 4 holds.

Finally, since the trees $U_L$, $U_R$ can be preprocessed in linear time to answer ancestor or descendant queries as well as *lca* queries in constant time [9, 17], and lca's in a tree are unique, property 5 follows. □

When the phylogenetic network $U$ is a galled tree, the definitions of consistency of a rooted proper triplet $ab|c$ or a rooted fan triplet $a|b|c$ with $U$ can be expressed as in Lemma 2 and Lemma 3 below. (These two key lemmas are used by our main algorithm in Section 4 to efficiently count the number of shared rooted triplets in two galled trees.) See Fig. 2 for some examples.

A junction common ancestor $z$ of two vertices $u, v$ in $U$ is said to *use* 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 2.** *Let $U$ be a galled tree. For any three leaves $a, b, c$ in the leaf label set of $U$, the rooted proper 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 use 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 of this condition for galled trees.

The proof is by contradiction. First of all, the path from $z$ to $w$ crosses neither that from $w$ to $a$ or that from $w$ to $b$ since $U$ is an acyclic directed graph. Next, if the path from $z$ to $c$ had to cross that from $w$ to $a$ (or $b$, respectively) in an inner vertex $x$ then $z$ and $w$ would lie on a common cycle whose reticulation vertex is exactly $x$, and both would use $x$. We obtain a contradiction. □

**Lemma 3.** *Let $U$ be a galled tree, and let $a, b, c$ be three leaf labels in $U$. The rooted fan triplet $a|b|c$ is consistent with $U$ if and only if there exists a vertex $w$ in $U$ such that: (1) $w$ is a junction common ancestor of all three pairs of leaves $\{a, b\}, \{a, c\}, \{b, c\}$; and (2) $w$ is the lowest common ancestor of at least two pairs of leaves among $\{a, b\}, \{a, c\}, \{b, c\}$.*

*Proof.* $\Rightarrow$) Suppose $a|b|c$ is consistent with $U$. Then $U$ contains a vertex $w$ such that there are three directed paths $P_a$, $P_b$, and $P_c$ from $w$ to $a$, $b$, and $c$, respectively, that are vertex-disjoint except for in the common start vertex $w$. Thus, property (1) always holds. Next, since $U$ is a galled tree, at most two of the three paths $P_a$, $P_b$, and $P_c$ overlap with edges from the same cycle in $U$. Clearly, if none of them overlap with the same cycle then $lca(a, b) = lca(a, c) = lca(b, c) = w$; on the other hand, if w.l.o.g. $P_a$ and $P_b$ overlap with the same cycle then no path from $w$ to $c$ can intersect $P_a$ or $P_b$ except for in the starting vertex $w$, so $lca(a, c) = lca(b, c) = w$.

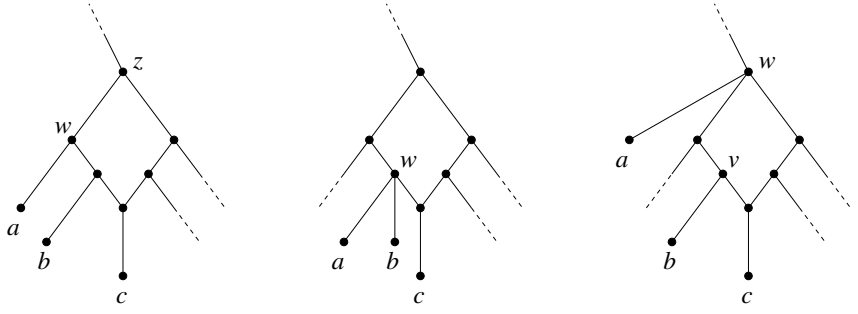


**Fig. 2.** Illustrating Lemmas 2 and 3. In (i), $w$ is a jca of $a$ and $b$ that does not use the reticulation vertex, and $z$ is a jca of $c$ and $w$, so Lemma 2 gives us the rooted proper triplet $ab|c$. In (ii), $w$ is a jca and also the lca for all three pairs $\{a, b\}, \{a, c\}, \{b, c\}$, so the network is consistent with $a|b|c$ according to Lemma 3. Similarly, in (iii), $w$ is a jca for all three pairs $\{a, b\}, \{a, c\}, \{b, c\}$ and the lca for exactly two pairs $\{a, b\}, \{a, c\}$, so the network is consistent with $a|b|c$ according to Lemma 3. Note that in addition to the above, Lemma 2 also correctly identifies $bc|a$ in (i), $ab|c$ in (ii), and $bc|a$ in (iii).

$\Leftarrow$) Suppose there exists a vertex $w$ that satisfies properties (1) and (2). There are two cases.

- First case: $w$ is the lca of all three pairs of leaves in $\{a, b, c\}$, i.e., $w = lca(a, b) = lca(a, c) = lca(b, c)$. By the definition of a lowest common ancestor, no proper descendant of $w$ can be an ancestor of any two of the three leaves $\{a, b, c\}$. Hence, there are three internally vertex-disjoint paths $w \rightsquigarrow a$, $w \rightsquigarrow b$, and $w \rightsquigarrow c$, i.e., $a|b|c$ is consistent with $U$. See also Fig. 2 (ii).
- Second case: $w$ is the lca of exactly two pairs of leaves in $\{a, b, c\}$, say $w = lca(a, b) = lca(a, c)$ but $w \neq lca(b, c) = v$. Then there are two junction common ancestors of $b$ and $c$, namely $w$ and $v$, so by Lemma 1, both $v$ and $w$ lie on the same cycle $C$ in $U$. Note that exactly one of the leaves $b$ and $c$ is a descendant of the reticulation vertex of $C$. Let $P_b$ and $P_c$ be two internally disjoint paths from $w$ to $b$ and $c$, respectively, where one of $P_b$ and $P_c$ passes through the reticulation vertex of $C$ and the other one passes through $v$. Since $w = lca(a, b)$, there is no path from $w$ to $a$ that intersects $P_b$. In the same way, since $w = lca(a, c)$, there is no path from $w$ to $a$ that intersects $P_c$. Thus, there are three internally vertex-disjoint paths $w \rightsquigarrow a$, $w \rightsquigarrow b$, and $w \rightsquigarrow c$, so $a|b|c$ is consistent with $U$. See also Fig. 2 (iii). □

## 3 Counting Monochromatic and Almost-Monochromatic Triangles

A *triangle* in an undirected graph is a cycle of length 3. In [1], Alon *et al.* showed how to determine if a connected, undirected graph with $m$ edges contains a

triangle, and if so, how to find a triangle in $O(m^{\frac{2\omega}{\omega+1}}) \leq o(m^{1.408})$ time. In the same paper, they also showed how to *count* the number of triangles in an undirected graph with $n$ vertices in $O(n^\omega) \leq o(n^{2.376})$ time. We first improve their technique to count the number of triangles more efficiently in case $m \ll n^2$:

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

*Proof.* First, we count the number of triangles in $G$ whose three vertices all have degree at least $t$ in $G$, where $t$ is a threshold parameter that will be set later. To do this, we take the subgraph of $G$ induced by all vertices of degree $\geq t$, and apply the triangle counting method from [1] which runs in $O(|V|^\omega)$ time for any graph with $|V|$ vertices. Since the number of vertices with degree $\geq t$ is $O(\frac{m}{t})$, the aforementioned method takes $O(\frac{m^\omega}{t^\omega})$ time. Let $N_\Delta$ be the computed number of triangles in the subgraph.

Secondly, we count the number of triangles with at least one vertex of degree strictly less than $t$. For this purpose, we enumerate the set $E_t$ of edges in $G$ with at least one endpoint of degree $< t$, and for $i = 1, \ldots, |E_t|$, iterate the following:

- Pick an endpoint $v$ of the $i$-th edge $e_i$ in $E_t$ of degree less than $t$; for each edge $e$ incident to $e_i$ at $v$, check if $e_i$ and $e$ induce a triangle in $G$ which does not include any edge $e_j \in E_t$ where $j < i$; if yes then increase $N_\Delta$ by one.

The above steps can be implemented in $O(t)$ time, so counting the remaining triangles takes $O(mt)$ time. By solving the equation $\frac{m^\omega}{t^\omega} = mt$, we obtain and set $t = m^{\frac{\omega-1}{\omega+1}}$. □

Next, we similarly refine the part of Theorem 1.8 in [19] which states that a monochromatic triangle in a connected, undirected, edge-colored graph with $n$ vertices can be found (if one exists) in $O(n^{(3+\omega)/2}) \leq o(n^{2.688})$ 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 $O(n^{(3+\omega)/2}) \leq o(n^{2.688})$ time.*

*Proof.* For each color $i \in C$, let $E_i$ be the set of edges in $G$ colored by $i$. Following [19], we say that $i$ is *heavily used* if $|E_i| \geq n^{(\omega+1)/2}$. For each heavily used color, we count the number of monochromatic triangles by directly applying the triangle counting method from [1] to the subgraph induced by edges colored with $i$ in $O(n^\omega)$ time. This takes $O(n^2/n^{(\omega+1)/2}) \cdot O(n^\omega) = O(n^{2-(\omega+1)/2+\omega}) = O(n^{(3+\omega)/2})$ 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 [19], we observe that the total time taken by the non-heavily used colors $i \in C$ is maximized if $|E_i| = \Theta(n^{(\omega+1)/2})$ holds for each of them, and thus there are $\Theta(n^{2-(\omega+1)/2})$ of them. Since $O(n^{2-(\omega+1)/2}) \cdot O((n^{(\omega+1)/2})^{\frac{2\omega}{\omega+1}}) = O(n^{(3-\omega)/2}) \cdot O(n^\omega) = O(n^{(3+\omega)/2})$, this shows that the total time taken by counting the remaining monochromatic triangles is $O(n^{(3+\omega)/2})$, too. □

Finally, 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 the following technical generalization of Theorem 1:

**Lemma 4.** *Suppose that an undirected graph $G$ with colored edges is preprocessed so that for any color edge $i$, the subgraph induced by the edges of color $i$ can be extracted in $O(m_i)$ time, where $m_i$ is the number of edges with color $i$ in $G$. Let $R$ be a binary relation on the colors of $G$ computable in constant time. The number of $R$-chromatic triangles with at least two edges of color $i$ in $G$ can be computed in $O(m_i^{\frac{2\omega}{\omega+1}})$ time.*

*Proof.* First, extract the subgraph $G_i$ induced by the edges of $G$ with color $i$ in $O(m_i)$ time. Then, run the method of Theorem 1 on $G_i$ with the following modifications which do not affect the asymptotic time complexity:

1. Once the square $C_i$ of the adjacency matrix of the subgraph of $G_i$ consisting of all vertices of degree at least $t$ is computed then for each entry $C_i[k,l]$ we check if $(k,l)$ is an edge of $G$ whose color $j$ is in the relation $R$ with the color $i$, i.e., $R(i,j)$ holds. Only in this case we increase the count of triangles by the arithmetic value of $C[k,l]$ (in case $(k,l)$ is an edge whose color is also $i$ and $R(i,i)$ holds, we increase the count of triangles by $C[k,l]/3$ only).
2. When we scan the edges $e$ of $G_i$ with at least one vertex $v$ of degree smaller than $t$, then for each edge $e'$ of $G$ incident to $e$ at $v$, we check if these two edges induce an $R$-chromatic triangle that was not counted before. If so, we increase the count by one. □

We now generalize Theorem 2 to $R$-chromatic triangles by applying Lemma 4:

**Theorem 3.** *Let $G$ be a connected, undirected graph with $n$ vertices and colored edges, and let $R$ be a binary relation on the colors of $G$ computable in constant time. The number of $R$-chromatic triangles in $G$ can be computed in $O(n^{(3+\omega)/2}) \leq o(n^{2.688})$ time.*

*Proof.* First construct the graphs $G_i$ induced by the sets $E_i$ of edges with color $i$. This takes $O(n^2)$ time in total. Next, proceed as in the proof of Theorem 2. For each heavily used color $i$, i.e., satisfying $|E_i| \geq n^{(\omega+1)/2}$, count the number of $R$-chromatic triangles with at least two edges with color $i$ by squaring the adjacency matrix of $G_i$ and testing, for each entry $C_i[k,l]$ of the resulting matrix, if $(k,l)$ is an edge whose color is in the relation $R$ with $i$ (analogously as in (1) in the proof of Lemma 4). This takes $O(n^{2-(\omega+1)/2+\omega}) = O(n^{(3+\omega)/2})$ time in total. To count the remaining $R$-chromatic triangles, each with at least two edges colored with a non-heavily used color $i$, use Lemma 4, which takes time $O(|E_i|^{2\omega/(\omega+1)})$ for any given color $i$. By an argument analogous to one in the proof of Theorem 2, the total time to count the remaining monochromatic triangles is $O(n^{(3+\omega)/2})$. □

# 4 Computing the Rooted Triplet Distance between Galled Trees

In this section, we apply the triangle counting techniques from Section 3 to obtain a subcubic-time algorithm for computing the rooted triplet distance between two galled trees. 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 proper 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 two given galled trees, we use Theorems 2 and 3 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 5.** *Let $U_1$, $U_2$ be two trees on the same set $L$ of $n$ leaves. The number of rooted fan triplets consistent with both $U_1$ and $U_2$ can be computed in $O(n^{(3+\omega)/2}) \leq o(n^{2.688})$ time.*

*Proof.* Form an auxiliary undirected complete graph $G = (L, E)$ in which every edge is assigned a color of the form $(v_1, v_2)$, where $v_1$ is a vertex of $U_1$ and $v_2$ is a vertex of $U_2$, as follows: For each edge $\{u, v\} \in E$, let $j_i$ for $i = 1, 2$ be the unique junction common ancestor of $u$ and $v$ in $U_i$, and color the edge $\{u, v\}$ in $G$ with the color $(j_1, j_2)$. By Lemma 1, $G$ can be constructed in $O(n^2)$ time.

For any $\{a, b, c\} \subseteq L$, the rooted fan triplet $a|b|c$ is consistent with $U_1$ if and only if the junction common ancestors in $U_1$ of $a$ and $b$, of $a$ and $c$, and of $b$ and $c$ are identical. The same holds for $U_2$. Therefore, $a|b|c$ is consistent with both $U_1$ and $U_2$ if and only if all three edges $\{a, b\}$, $\{a, c\}$, $\{b, c\}$ have the same color in $G$. It follows that the number of rooted fan triplets which are common to both trees equals the number of monochromatic triangles in $G$. By Theorem 2, we can compute the number of rooted fan triplets that are consistent with both $U_1$ and $U_2$ in $O(n^{(3+\omega)/2})$ time. □

Next, we adapt the reduction in the proof of Lemma 5 to the more complicated *galled tree* case:

**Lemma 6.** *Let $U_1$, $U_2$ be two galled trees on the same set $L$ of $n$ leaves. The number of rooted fan triplets consistent with both $U_1$ and $U_2$ can be computed in $O(n^{(3+\omega)/2}) \leq o(n^{2.688})$ time.*

*Proof.* By Lemma 3, we can distinguish two classes of rooted fan triplets in a galled tree $U$: those where for each of its three pairs of leaves, the lca is equal to the shared junction common ancestor as in the example in Fig. 2 (ii) (henceforth referred to as "class 1"), and those where the equality holds for two pairs only, as in Fig. 2 (iii) (henceforth referred to as "class 2"). For the sake of the proof,

we need to consider a slightly different two-partition of rooted fan triplets in $U$. We shall say that a rooted fan triplet in $U$ is of *type 1* iff it belongs to the class 1 and the unique lca of each pair of leaves in the triplet is also their lca in each of the trees $U_L$, $U_R$. All remaining rooted fan triplets in $U$ are said to be of *type 2*.

For $i = 1, 2$, consider the trees $(U_i)_L$, $(U_i)_R$ defined as in the proof of Lemma 1. By Lemma 5, we can compute the number of shared rooted fan triplets between $(U_1)_A$ and $(U_2)_B$ for any $A, B \in \{L, R\}$ in $O(n^{(3+\omega)/2})$ time. Note that each rooted fan triplet of type 1 in $U_1$ occurs in both $(U_1)_L$ and $(U_1)_R$, while each rooted fan triplet of type 2 in $U_1$ occurs in only one of the trees. The reason for the distinction is that a rooted fan triplet of type 2 contains exactly one pair of leaves whose lca in $U_1$ relies on one of the two edges directed to a reticulation vertex of a cycle. Hence, the lca of the pair occurs in exactly one of the trees $(U_1)_L$ and $(U_1)_R$, and consequently the rooted fan triplet also occurs in exactly one of (not necessarily the same as above) $(U_1)_L$ and $(U_1)_R$. Analogous observations hold for $U_2$. Hence, if we sum the number of shared rooted fan triplets between $(U_1)_A$ and $(U_2)_B$ over all $A, B \in \{L, R\}$, then each rooted fan triplet that is of type 1 both in $U_1$ and $U_2$ is counted four times, while those that are of different types in $U_1$ and $U_2$ are counted twice, and finally those of type 2 both in $U_1$ and $U_2$ are counted only once. Hence, if for $p,\ q \in \{1, 2\}$, $T_{p,q}$ denotes the number of rooted shared fan triplets that are of type $p$ in $U_1$ and of type $q$ in $U_2$ then the computed sum equals $4T_{1,1} + 2T_{1,2} + 2T_{2,1} + T_{2,2}$.

In fact, we can also determine $T_{1,1}$ in $O(n^{(3+\omega)/2})$ time in the same way as we have done for a pair of trees in Lemma 5. While constructing the auxiliary complete graph, we require $j_i$ to be both the lca of $u$ and $v$ in $(U_i)_L$ and $(U_i)_R$), as well as a junction common ancestor of $u$ and $v$ in $U$. Then, we use Theorem 2 to determine the number of monochromatic triangles analogously.

It remains to determine the number of shared rooted fan triplets for $U_1$ and $U_2$ that are of different types in $U_1$ and $U_2$ in order to cancel repetitions in the aforementioned sum, i.e., $T_{1,2} + T_{2,1}$. To compute, say $T_{2,1}$, we again form the auxiliary complete graph on the set $L$ of leaves and color each edge $\{u, v\}$ as described next. Recall that lca's are unique in a galled tree. For $i = 1, 2$, let $j_i$ be the unique lca of of $u$ and $v$ in $U_i$. If, for $i = 1, 2$, $j_i$ is also a junction common ancestor of $u$ and $v$ in $U_i$ and it is the lca of $u$ and $v$ in both trees $(U_i)_L$ and $(U_i)_R$, then $\{u, v\}$ is colored with $(j_1, j_2)$ as before. Next, if for $i = 1, 2$, $j_i$ is also a junction common ancestor of $u$ and $v$ in $U_i$, and $j_1$ is the lca of $u$ and $v$ in exactly one of the trees $(U_1)_L$ and $(U_1)_R$ while $j_2$ is the lca of $u$ and $v$ in both trees $(U_2)_L$ and $(U_2)_R$ then $\{u, v\}$ is colored with $((j_1)^*, j_2)$. Otherwise, $\{u, v\}$ is colored with the null color. To use Theorem 3, we define the relation $R$ by:

- $(j_1, j_2)R(l_1, l_2)$ holds iff $l_1 = (k)^*$, where $k$ is a proper descendant of $j_1$ or $j_1 = k$, and $j_2 = l_2$.

The trees $(U_i)_A$ for $i = 1, 2$, $A \in \{L, R\}$, can be preprocessed to support $O(1)$-time lca queries in $O(n)$ time [9, 17]. By using $(U_1)_L, (U_1)_R$, we can spend $O(n^2)$ time to build a data structure supporting $O(1)$-time proper descendant queries.

Now, we apply Theorem 3 to the auxiliary graph to obtain the number $T_{2,1}$ of rooted shared triplets of type 2 in $U_1$ and type 1 in $U_2$ in $O(n^{(3+\omega)/2})$ time.

The number $T_{1,2}$ of rooted shared triplets of type 1 in $U_1$ and type 2 in $U_2$ is obtained in $O(n^{(3+\omega)/2})$ time in the same way. □

### 4.2 Counting the Number of Shared Rooted Proper Triplets

**Lemma 7.** *Let $U_1$, $U_2$ be two galled trees on the same set $L$ of $n$ leaves. The number of rooted proper triplets consistent with both $U_1$ and $U_2$ can be computed in $O(n^\omega) \leq o(n^{2.376})$ time.*

*Proof.* First, for $i = 1, 2$, for each pair of leaves in $U_i$, compute their junction common ancestors (if they exist) along with the information if the respective junction common ancestor is located on a cycle of $U_i$, if it is the root of the cycle, and if it uses the reticulation vertex of the cycle. By a straightforward modification of the proof of Lemma 1, this takes $O(n^2)$ time.

For each vertex $v_i$ of $U_i$ form two copies $v_i^0$, $v_i^1$. Next, for the set of pairs of distinct leaves in $L$, form the classes $C_{v_1^{b_1}, v_2^{b_2}}$, where $\{b_1, b_2\} \subset \{0, 1\}$ and $v_i$ is a vertex of $U_i$ for $i = 1, 2$, such that $(a, b) \in C_{v_1^{b_1}, v_2^{b_2}}$ if and only if the following three conditions hold for $i = 1, 2$: (1) $v_i$ is a junction common ancestor of $a$ and $b$ in $U_i$; (2) $v_i$ is located on a cycle of $U_i$ and uses the reticulation vertex of the cycle iff $b_i = 1$; and (3) if $v_i$ is the root of a cycle in $U_i$ then there is no other junction common ancestor of $a$ and $b$. By Lemma 1, any pair of leaves $a, b$ in a galled tree can have at most two junction common ancestors. Moreover, if there are two then they are located on the same cycle and one of them will be the root of the cycle. Hence, from the point of view of a rooted triplet $ab|c$, it is sufficient to consider the junction common ancestor of $a, b$ that is a descendant of the root vertex of the cycle in this case, since any path from an ancestor of the root of the cycle can be extended to reach the descendant junction common ancestor. This explains the third condition, which implies that the classes $C_{v_1^{b_1}, v_2^{b_2}}$ are pairwise disjoint. These classes can be formed in $O(n^2)$ time by integer sorting.

Furthermore, for $i = 1, 2$, form matrices $M_i$ such that the rows of $M_i$ correspond to the copies of vertices in $U_i$, the columns of $M_i$ correspond to the leaves in $L$, and $M_i[v_i^{b_i}, c] = 1$ if and only if there is a junction common ancestor of $v_i$ and $c$ in $U_i$ which in case $b_i = 1$ does not use the reticulation vertex of the cycle on which $v_i$ lies in $U_i$. Importantly, if $M_i[v_i^{b_i}, c] = 1$ then $c$ cannot occur in any pair in a class of the form $C_{v_1^{b_1}, v_2^{b_2}}$. Simply, in this case, an ancestor of $c$ or $c$ itself would be a reticulation vertex used by both $v_i$ and any junction common ancestor of $v_i$ and $c$. This in particular would imply $b_i = 1$. Hence, $M_i[v_i^{b_i}, c]$ would be set to 0, and we obtain a contradiction.

Next, compute $Q = M_1 \times M_2^t$ in $O(n^\omega)$ time. By the definitions of $M_1$ and $M_2$, the value of $Q[v_1^{b_1}, v_2^{b_2}]$ is exactly the number of leaves $c$ in $L$ that have a junction common ancestor with $v_i$ in $U_i$ not using the reticulation vertex of the cycle on which $v_i$ lies if $b_i = 1$, for $i = 1, 2$. Note that for $\{b_1, b_2\} \neq \{b_1', b_2'\}$, $C_{v_1^{b_1}, v_2^{b_2}} \cap C_{v_1^{b_1'}, v_2^{b_2'}} = \emptyset$ and the aforementioned leaves $c$ cannot occur in any pair in $C_{v_1^{b_1}, v_2^{b_2}}$. By Lemma 2, the sum $\sum_{\{b_1, b_2\} \subset \{0,1\}} |C_{v_1^{b_1}, v_2^{b_2}}| Q[v_1^{b_1}, v_2^{b_2}]$

equals the number of proper rooted triplets $ab|c$ consistent with both $U_1$ and $U_2$ that use $v_i$ as a junction common ancestor of $a$ and $b$ in $U_i$, for $i = 1, 2$, with the exception of the case when $v_1$ or $v_2$ is the root vertex of a cycle in its galled tree and there is another junction common ancestor of $a$ and $b$ which is a descendant of the root vertex in the galled tree. Due to the latter, for different pairs of $v_1, v_2$, the sum counts different sets of the proper rooted triplets $ab|c$ consistent with both $U_1$ and $U_2$. Thus, it is sufficient to compute the sum $\sum_{v_1 \in U_1} \sum_{v_2 \in U_2} \sum_{\{b_1,b_2\} \subset \{0,1\}} |C_{v_1^{b_1}, v_2^{b_2}}| Q[v_1^{b_1}, v_2^{b_2}]$ to obtain the total number of rooted triplets consistent with both $U_1$ and $U_2$. This takes $O(n^2)$ time. □

### 4.3 Computing the Rooted Triplet Distance

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

**Theorem 4.** *Let $U_1$, $U_2$ be two galled trees on the same set $L$ of $n$ leaves. The rooted triplet distance $d_{rt}(U_1, U_2)$ can be computed in $O(n^{(3+\omega)/2}) \leq o(n^{2.688})$ time.*

*Proof.* For $i = 1, 2$, let $F_i$ denote the set of rooted fan triplets consistent with $U_i$, and let $P_i$ denote the set of rooted proper triplets consistent with $U_i$. We have $d_{rt}(U_1, U_2) = \sum_{i=1}^{2}(|F_i| + |P_i|) - 2|F_1 \cap F_2| - 2|P_1 \cap P_2|$. Compute $|F_i \cap F_i| = |F_i|$ and $|F_1 \cap F_2|$ in $O(n^{(3+\omega)/2}) \leq o(n^{2.688})$ time by Lemma 6, and compute $|P_i \cap P_i| = |P_i|$ and $|P_1 \cap P_2|$ in $O(n^{\omega}) \leq o(n^{2.376})$ time by Lemma 7. □

## 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 [8, 10]. More precisely, we have presented a new $o(n^{2.688})$-time algorithm for computing the rooted triplet distance between two input galled trees with $n$ leaves each [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 [19], respectively.

Recently, Nielsen *et al.* [15] showed how to compute the *unrooted quartet distance* between two *unrooted* phylogenetic trees with $n$ leaves in $o(n^{2.688})$ 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 seems inherently "easier" than ours as it does not involve cycles. A lot of the conceptual complexity in our paper stems from the non-uniqueness of junction common ancestors in galled trees; compare the proofs of Lemmas 5 and 6, for example.

It is an open question whether the problem of computing the rooted triplet distance $d_{rt}(U_1, U_2)$ between two galled trees $U_1, U_2$ admits a quadratic-time algorithm or not. Another important question is if our method can be enhanced to include even larger classes of phylogenetic networks than galled trees.

## References


1. Alon, N., Yuster, R., Zwick, U.: Finding and counting given length cycles. Algorithmica 17(3), 209–223 (1997)
2. Bansal, M.S., Dong, J., Fernández-Baca, D.: Comparing and aggregating partially resolved trees. Theoretical Computer Science 412(48), 6634–6652 (2011)
3. Chan, H.-L., Jansson, J., Lam, T.-W., Yiu, S.-M.: Reconstructing an ultrametric galled phylogenetic network from a distance matrix. Journal of Bioinformatics and Computational Biology 4(4), 807–832 (2006)
4. Choy, C., Jansson, J., Sadakane, K., Sung, W.-K.: Computing the maximum agreement of phylogenetic networks. Theoretical Computer Science 335(1), 93–107 (2005)
5. Coppersmith, D., Winograd, S.: Matrix Multiplication via Arithmetic Progressions. Journal of Symbolic Computation 9, 251–280 (1990)
6. Critchlow, D.E., Pearl, D.K., Qian, C.: The triples distance for rooted bifurcating phylogenetic trees. Systematic Biology 45(3), 323–334 (1996)
7. Felsenstein, J.: Inferring Phylogenies. Sinauer Associates, Inc., Sunderland (2004)
8. Gusfield, D., Eddhu, S., Langley, C.: Optimal, efficient reconstruction of phylogenetic networks with constrained recombination. Journal of Bioinformatics and Computational Biology 2(1), 173–213 (2004)
9. Harel, D., Tarjan, R.E.: Fast algorithms for finding nearest common ancestors. SIAM Journal on Computing 13(2), 338–355 (1984)
10. Huson, D.H., Rupp, R., Scornavacca, C.: Phylogenetic Networks: Concepts, Algorithms and Applications. Cambridge University Press (2010)
11. van Iersel, L., Kelk, S.: Constructing the Simplest Possible Phylogenetic Network from Triplets. Algorithmica 60(2), 207–235 (2011)
12. Jansson, J., Nguyen, N.B., Sung, W.-K.: Algorithms for Combining Rooted Triplets into a Galled Phylogenetic Network. SIAM Journal on Computing 35(5), 1098–1121 (2006)
13. Morrison, D.: Introduction to Phylogenetic Networks. RJR Productions (2011)
14. Nakhleh, L., Warnow, T., Ringe, D., Evans, S.N.: A comparison of phylogenetic reconstruction methods on an Indo-European dataset. Transactions of the Philological Society 103(2), 171–192 (2005)
15. Nielsen, J., Kristensen, A.K., Mailund, T., Pedersen, C.N.S.: A sub-cubic time algorithm for computing the quartet distance between two general trees. Algorithms for Molecular Biology 6, Article 15 (2011)
16. Stothers, A.J.: On the Complexity of Matrix Multiplication. PhD thesis, University of Edinburgh (2010)
17. Tarjan, R.E.: Applications of path compression on balanced trees. Journal of the ACM 26(4), 690–715 (1979)
18. Wang, L., Ma, B., Li, M.: Fixed topology alignment with recombination. Discrete Applied Mathematics 104(1-3), 281–300 (2000)
19. Vassilevska, V., Williams, R., Yuster, R.: Finding Heaviest $H$-Subgraphs in Real Weighted Graphs, with Applications. ACM Transactions on Algorithms 6(3), Article 44 (2010)
20. Vassilevska Williams, V.: Breaking the Coppersmith-Winograd barrier. UC Berkely and Stanford University (2011) (manuscript)