Optimal Completion and Comparison of Incomplete Phylogenetic Trees Under Robinson-Foulds Distance

Keegan Yao
Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA

Mukul S. Bansal
Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA

Abstract

The comparison of phylogenetic trees is a fundamental task in phylogenetics and evolutionary biology. In many cases, these comparisons involve trees inferred on the same set of leaves, and many distance measures exist to facilitate such comparisons. However, several applications in phylogenetics require the comparison of trees that have non-identical leaf sets. The traditional approach for handling such comparisons is to first restrict the two trees being compared to just their common leaf set. An alternative, conceptually superior approach that has shown promise is to first complete the trees by adding missing leaves so that the completed trees have identical leaf sets. This alternative approach requires the computation of optimal completions of the two trees that minimize the distance between them. However, no polynomial-time algorithms currently exist for this optimal completion problem under any standard phylogenetic distance measure.

In this work, we provide the first polynomial-time algorithms for the above problem under the widely used Robinson-Foulds (RF) distance measure. This hitherto unsolved problem is referred to as the RF(+) problem. We (i) show that a recently proposed linear-time algorithm for a restricted version of the RF(+) problem is a 2-approximation for the RF(+) problem, and (ii) provide an exact \(O(nk^2)\)-time algorithm for the RF(+) problem, where \(n\) is the total number of distinct leaf labels in the two trees being compared and \(k\), bounded above by \(n\), depends on the topologies and leaf set overlap of the two trees. Our results hold for both rooted and unrooted binary trees.

We implemented our exact algorithm and applied it to several biological datasets. Our results show that completion-based RF distance can lead to very different inferences regarding phylogenetic similarity compared to traditional RF distance. An open-source implementation of our algorithms is freely available from https://compbio.engr.uconn.edu/software/RF_plus.

1 Introduction

Phylogenetic trees, or simply phylogenies, are leaf-labeled trees that depict the evolutionary relationships between different species, genes, or other biological entities such as cells in an organism or individuals from a population. In phylogenetic trees, leaf nodes represent extant entities while internal nodes represent hypothetical ancestors. Many different methodologies,
Optimal Phylogenetic Tree Completion

algorithms, and data types exist for estimating phylogenies, and there is often considerable uncertainty and error in their inference, with different methods or data types suggesting different evolutionary relationships between the same extant entities. Many distance (or similarity) measures have therefore been developed for systematically comparing different phylogenetic trees, including the widely used Robinson-Foulds distance [29], triplet and quartet distances [14, 17], nearest neighbor interchange (NNI) and subtree prune and regraft (SPR) distances [31, 18, 34], maximum agreement subtrees [19, 2, 15], nodal distance [9], geodesic distance [23] and others. However, these distance measures implicitly assume that the two trees being compared have identical leaf sets, an assumption that is often violated in practice. Indeed, several applications, such as supertree construction [24, 6, 10, 32, 1], phylogenetic database search [28, 30, 11, 25], and clustering of phylogenies [20, 35], require the computation of distances between trees with partially overlapping leaf sets.

The traditional approach to comparing two trees with only partially overlapping leaf sets is to first restrict (i.e., prune down) both trees to their shared leaf set. This restriction based approach, though simple to conceptualize and compute, can result in the loss of valuable topological information through scrapping of leaves that are not common to both trees. An alternative approach to comparing trees with non-identical leaf sets is to complete or fill in each of the input trees to the union of their leaf sets in a way which minimizes the distance between them, and then compute their distance. This approach, though conceptually more complex, successfully incorporates all topological information in both the trees being compared. In addition to its more complete use of topological information, the completion based approach also has the benefit of a larger range of attainable values due to comparisons over larger extended trees rather than smaller induced trees. Despite these advantages, no polynomial-time algorithms currently exist for completion based comparison under any standard phylogenetic distance measure. In this work, we provide the first polynomial-time algorithms for optimal completion and comparison of incomplete phylogenetic trees under the widely used Robinson-Foulds (RF) distance measure. Following existing literature [4], we refer to completion based RF distance as RF(+), the traditional restriction based RF distance as RF(-), and the problem of computing the RF(+) distance between two trees as the RF(+) problem. Figure 1 illustrates the difference between RF(-) and RF(+) distances.

Previous work. The idea of completion based Robinson-Foulds distance arose at least a decade ago when Cotton and Wilkinson introduced majority-rule supertrees [13] and defined two variants, majority-rule(-) and majority-rule(+) supertrees, based on RF(-) and RF(+), respectively. Completion based majority-rule(+) supertrees and some variants were subsequently shown to have many desirable properties [16]. Later, Kupczok [22] characterized the RF(+) distance for the restricted special case where the leaf set of one tree is a subset of the leaf set of the other in terms of incompatible splits between the two trees. For this restricted special case, referred to as the One Tree RF(+) (OT-RF(+)) problem [4], an $O(n^2)$-time algorithm was proposed by Christensen et. al. in 2017 [12], where $n$ is the total number of distinct leaf labels in the two trees being compared. More recently, Bansal proposed an optimal $O(n)$-time algorithm for this OT-RF(+) problem [3, 4]. Bansal also proposed a restricted formulation of the RF(+) problem, called the Extraneous-Clade-Free RF(+) (EF-RF(+)) problem, which is based on computing optimal completions that avoid the creation of any subtrees formed by joining together two subtrees unique to each one of the two input trees. Essentially, the EF-RF(+) problem disallows certain types of completions; specifically, it ignores how subtrees exclusive to one input tree impact the overall optimal position where subtrees from the other input tree should be added. Bansal showed that the EF-RF(+) problem can be solved in $O(n)$ time [4]. These linear-time algorithms for the OT-RF(+) and EF-RF(+) problems are applicable to both rooted and unrooted trees.
Figure 1 RF(-) and RF(+) distances. The figure shows a "base" tree $S$ and two other trees $U$ and $V$, with $Le(U) = Le(V)$, being compared to $S$. $S_*$, $U_*$ and $V_*$ represent the trees $S$, $U$ and $V$, respectively, when restricted to the common leaf set. $U^*$ and $V^*$ are the optimal RF(+) completions of $U$ and $V$ with respect to $S$. $S_U^*$ and $S_V^*$ are the optimal RF(+) completions of $S$ with respect to $U$ and $V$, respectively. Filled in nodes represent matched nodes (Definition 2.2). Here, $RF(S_*, U^*) = 2$ and $RF(S_*, V^*) = 4$ while $RF(S_U^*, U^*) = 8$ and $RF(S_V^*, V^*) = 4$. Thus, in this example, $U$ is closer to $S$ than $V$ under RF(-) but $V$ is closer to $S$ than $U$ under RF(+).

Our Contributions. In this work, we provide the first polynomial-time algorithms for the RF(+) problem for both rooted and unrooted trees. Specifically, we make the following contributions: First, we show that the EF-RF(+) distance between two trees is a 2-approximation for the RF(+) distance between those trees. Since the EF-RF(+) problem can be solved in $O(n)$ time, this yields a linear time 2-approximation algorithm for the RF(+) problem. Second, we provide an $O(nk^2)$-time exact algorithm for the RF(+) problem, where $k$, bounded above by $n$, is the number of maximal subtrees exclusive to one input tree. And third, we perform an extensive experimental study which demonstrates that the use of RF(+) distance can lead to very different inferences regarding phylogenetic similarity compared to RF(-) distance. We also find that, in practice, EF-RF(+) distances are often very close to RF(+) distances, suggesting that the linear-time algorithm for computing EF-RF(+) distances could be an excellent heuristic for estimating RF(+) distances between large trees.

The rest of this manuscript is organized as follows: Preliminaries and problem definitions appear in the next section. We describe the linear time 2-approximation algorithm in Section 3, and the exact algorithm in Section 4. Section 5 shows how our algorithms can be
Optimal Phylogenetic Tree Completion

extended to unrooted trees, and Section 6 describes the results of our experimental study. Concluding remarks appear in Section 7. Proofs of all lemmas and theorems from Sections 3 and 4 appear in the Appendix.

2 Definitions and Preliminaries

We follow basic definitions and problem formulations from [4]. All trees will be unordered. Given a tree $T$, we denote its node set, edge set, and leaf set by $V(T)$, $E(T)$, and $Le(T)$, respectively. The set of all non-leaf (i.e., internal) nodes of $T$ is denoted by $I(T)$. If $T$ is rooted, the root node of $T$ is denoted by $rt(T)$, the parent of a node $v \in V(T)$ by $pa_T(v)$, its set of children by $Ch_T(v)$, and the (maximal) subtree of $T$ rooted at $v$ by $T(v)$. If two nodes in $T$ have the same parent, they are called siblings of each other. If $pa_T(v)$ has exactly two children, then we will denote the sibling of $v$ as $sib_T(v)$. The least common ancestor, denoted $lca_T(L)$, of a set $L \subseteq Le(T)$ in $T$ is defined to be the node $v \in V(T)$ such that $L \subseteq Le(T(v))$ and $L \not\subseteq Le(T(u))$ for any child $u$ of $v$. For convenience, given a collection of vertices $a_1, \ldots, a_m$ in $T$, we will define $lca_T(a_1, \ldots, a_m) = lca_T(Le(T(a_1)) \cup \cdots \cup Le(T(a_m)))$. Given a rooted tree $T$ and $a, b \in V(T)$, we say that $a \preceq b$ if $a \in V(T(b))$, and $a \prec b$ if $a \in V(T(b))$ and $a \neq b$. A rooted tree is binary if all of its internal nodes have exactly two children, while an unrooted tree is binary if all its nodes have degree either 1 or 3. Throughout this work, the term tree refers to binary trees with uniquely labeled leaves.

Let $T$ be a rooted or unrooted tree. Given a set $L \subseteq Le(T)$, let $T_L$ be the minimal subtree of $T$ with leaf set $L$. We define the leaf induced subtree $T[L]$ of $T$ on leaf set $L$ to be the tree obtained from $T_L$ by successively removing each non-root node of degree two and adjoining its two neighbors.

▶ Definition 2.1 (Completion of a tree). Given a tree $T$ and a set $L'$ such that $Le(T) \subseteq L'$, a completion of $T$ on $L'$ is a tree $T'$ such that $Le(T') = L'$ and $T'[Le(T)] = T$.

If $T$ is a rooted tree, for each node $v \in V(T)$, the clade $C_T(v)$ is defined to be the set of all leaf nodes in $T(v)$; i.e. $C_T(v) = Le(T(v))$. We denote the set of all clades of a rooted tree $T$ by $Clade(T)$. This concept can be extended to unrooted trees as follows. If $T$ is an unrooted tree, each edge $(u, v) \in E(T)$ defines a partition of the leaf set of $T$ into two disjoint subsets $Le(T_u)$ and $Le(T_v)$, where $T_u$ is the subtree containing node $u$ and $T_v$ is the subtree containing node $v$, obtained when edge $(u, v)$ is removed from $T$. The partition induced by any edge $(u, v) \in E(T)$ is called a split and is represented by the set $\{Le(T_u), Le(T_v)\}$. The set of all splits in an unrooted tree $T$ is denoted by $Split(T)$.

▶ Definition 2.2 (Matched and mismatched nodes). Given rooted trees $S$ and $T$, and a node $v \in V(S)$, we call $v$ a matched node with respect to $T$ if $C_S(v) \in Clade(T)$, and a mismatched node otherwise. Analogously, $C_S(v)$ is called a matched clade if $C_S(v) \in Clade(T)$, and a mismatched clade otherwise.

The symmetric difference of two sets $A$ and $B$, denoted by $A \Delta B$, is the set $(A \setminus B) \cup (B \setminus A)$. We now define the Robinson-Foulds distance and the two problems that we solve in this paper.

▶ Definition 2.3 (Robinson-Foulds distance). The Robinson-Foulds (RF) distance, $RF(S, T)$, between two trees $S$ and $T$ is defined to be $|Clade(S) \Delta Clade(T)|$ if $S$ and $T$ are rooted trees, and $|Split(S) \Delta Split(T)|$ if $S$ and $T$ are unrooted trees.
Problem 1 (Rooted RF(+) (R-RF(+))). Given two rooted binary trees $S$ and $T$, compute a binary completion $S^*$ of $S$ on $\text{Le}(S) \cup \text{Le}(T)$ and a binary completion $T^*$ of $T$ on $\text{Le}(S) \cup \text{Le}(T)$ such that $\text{RF}(S^*, T^*)$ is minimized.

Problem 2 (Unrooted RF(+) (U-RF(+))). Given two unrooted binary trees $S$ and $T$, compute a binary completion $S^*$ of $S$ on $\text{Le}(S) \cup \text{Le}(T)$ and a binary completion $T^*$ of $T$ on $\text{Le}(S) \cup \text{Le}(T)$ such that $\text{RF}(S^*, T^*)$ is minimized.

These problems can equivalently be viewed as maximizing the number of matched clades or minimizing the number of mismatched clades between completions of the input trees. Our algorithms for the problems above rely on first computing exact solutions for restricted variants of those problems. These restricted variants of R-RF(+) and U-RF(+) were first proposed and defined in [4] and are referred to as the Extraneous-Clade-Free R-RF(+) (EF-R-RF(+) and Extraneous-Split-Free U-RF(+) (EF-U-RF(+)) problems. These restricted variants are based on computing optimal completions that do not contain any subtrees formed by joining together two subtrees unique to each one of the two input trees. Next, we first define extraneous clades and extraneous splits, and then state the EF-R-RF(+) and EF-U-RF(+) problems.

Definition 2.4 (Extraneous clade [4]). Suppose $S$ and $T$ are rooted trees. Given completions $S'$ and $T'$ of $S$ and $T$, respectively, on $\text{Le}(S) \cup \text{Le}(T)$, we define a clade of $S'$ or $T'$ to be an extraneous clade if it contains leaves from both $S$ and $T$ but no leaves that are common to $S$ and $T$.

An extraneous split is simply the analogous notion for unrooted trees and we refer the reader to [4] for a formal definition. The corresponding problem variants can now be defined as follows:

Problem 3 (Extraneous-Clade-Free R-RF(+) (EF-R-RF(+)) [4]). Given two rooted trees $S$ and $T$, compute a completion $S'$ of $S$ on $\text{Le}(S) \cup \text{Le}(T)$ and a completion $T'$ of $T$ on $\text{Le}(S) \cup \text{Le}(T)$ such that $S'$ and $T'$ do not contain any extraneous clades and $\text{RF}(S', T')$ is minimized.

Problem 4 (Extraneous-Split-Free U-RF(+)) (EF-U-RF(+)) [4]. Given two unrooted trees $S$ and $T$ such that $|\text{Le}(S) \cap \text{Le}(T)| \geq 2$, compute a completion $S'$ of $S$ on $\text{Le}(S) \cup \text{Le}(T)$ and a completion $T'$ of $T$ on $\text{Le}(S) \cup \text{Le}(T)$ such that $S'$ and $T'$ do not contain any extraneous splits and $\text{RF}(S', T')$ is minimized.

Figure 2 provides examples of completions with and without extraneous clades. Both the EF-R-RF(+) and EF-U-RF(+) problems can be solved optimally in linear time [4].

Note. In the remainder of this section, as well as in Sections 3 and 4 we focus on only the rooted version of RF(+), i.e., on the R-RF(+) problem, and implicitly assume that the two trees being compared, $S$ and $T$, are rooted.

Node coloring scheme for rooted trees. For ease of presentation, we assign a color to some of the nodes of the two rooted input trees as follows. These node colorings can also be used to define red and green subtrees.

Definition 2.5 (Red and Green Nodes). Let $S$ and $T$ be two arbitrary rooted trees. A node $v \in V(S)$ is called a red node (with respect to $T$) if $\text{Le}(S(v)) \subseteq \text{Le}(S) \setminus \text{Le}(T)$. Analogously, a node $v \in V(T)$ is called a green node (with respect to $S$) if $\text{Le}(T(v)) \subseteq \text{Le}(T) \setminus \text{Le}(S)$.
EF-RF(+) and RF(+) completions. \( S', T' \) are optimal EF-R-RF(+) completions (without extraneous clades) of \( S \) and \( T \), respectively, and completions \( S^*, T^* \) are optimal RF(+) completions. Nodes labeled with downward and upward pointing triangles are red and green nodes, respectively, as defined in Definition 2.5. Filled in nodes correspond to matched clades.

**Definition 2.6 (Red and Green Subtrees).** A subtree \( S(u) \), where \( u \in V(S) \), is called a red subtree of \( S \) if \( u \) is a red node. A subtree \( T(u) \), where \( u \in V(T) \), is called a green subtree of \( T \) if \( u \) is a green node. A subtree \( S(u) \), where \( u \in V(S) \), is called a maximal red subtree of \( S \) if \( S(u) \) is a red subtree and either \( u = rt(S) \) or \( pa_S(u) \) is not red. A subtree \( T(u) \), where \( u \in V(T) \), is called a maximal green subtree of \( T \) if \( T(u) \) is a green subtree and either \( u = rt(T) \) or \( pa_T(u) \) is not green. Note that all nodes in a red (green) subtree must be red (green).

Under this node coloring, completing a tree \( S \) with respect to tree \( T \) entails adding all the green leaves of \( T \) into \( S \) and completing a tree \( T \) with respect to tree \( S \) entails adding, or grafting, all the red leaves of \( S \) into \( T \). Importantly, as we show later in Theorem 3.1, under R-RF(+) problem, there exist optimal completions of \( S \) and \( T \) in which all grafted subtrees are maximal red or green subtrees. In other words, to optimally complete \( S \) we must only add the maximal green subtrees of \( T \) to \( S \), and vice versa.

Notational conventions. \( S \) and \( T \) will denote the two given (input) trees to be completed/compared. Going forward, we will generally use \( S' \) and \( T' \) to represent completions (optimal or non-optimal) with no extraneous clades, and \( S^* \) and \( T^* \) to represent completions that may include extraneous clades.

### 3 EF-R-RF(+) is a 2-Approximation for R-RF(+)

Observe that any optimal pair of R-RF(+) completions can be modified into a pair of (not necessarily optimal) EF-R-RF(+) completions by breaking apart any existing extraneous clades and reinserting the red/green leaves in a manner that avoids forming extraneous clades. In this section, we will show how to perform such a modification of optimal R-RF(+) completions so that the resulting increase in RF distance is appropriately bounded. This will establish that EF-R-RF(+) distance is a 2-approximation for R-RF(+) distance and will yield a linear-time 2-approximation algorithm for the R-RF(+) problem. We will first establish the presence of canonical optimal R-RF(+) completions that satisfy some desirable structural properties.
**Notation and terminology.** Given completions $S^*$ and $T^*$ of $S$ and $T$, if there exists an extraneous clade $C_{T^*}(v)$ for some vertex $v \in T^*$, then we will call the subtree $T^*(v)$ an **extraneous subtree**. If the children $s$ and $t$ of $v$ satisfy $C_{T^*}(s) \in \text{Clade}(S)$ and $C_{T^*}(t) \in \text{Clade}(T)$, then we will denote the extraneous subtree by $\{s, t\}$. To simplify notation, we will write $pa_{T^*}\{s, t\}$ to express the parent $pa_{T^*}(lca_{T^*}(s, t))$ of the root node of the extraneous subtree $\{s, t\}$ in completion $T^*$. Likewise, we will write $sib_{T^*}\{s, t\}$ to express $sib_{T^*}(lca_{T^*}(s, t))$, i.e., the sibling of the root node of extraneous subtree $\{s, t\}$ in $T^*$.

Next, we show that there always exists an optimal pair of R-RF(+) completions in which all extraneous clades are of the form $\{s, t\}$, and any such extraneous clade appears in both completions. We refer to such optimal R-RF completions $S^*$ and $T^*$ of $S$ and $T$ as **canonical optimal R-RF(+) completions**.

**Theorem 3.1.** Let $S$ and $T$ be rooted binary trees. Then, there exist optimal completions $S^*$ and $T^*$ under the R-RF(+) problem with the following properties:

1. Every subtree inserted into $S^*$ is a maximal green subtree of $T$, and every subtree inserted into $T^*$ is a maximal red subtree of $S$.
2. Every extraneous subtree in $S^*$ and $T^*$ is of the form $\{s, t\}$, where $s$ is the root of a maximal red subtree in $S$ and $t$ is the root of a maximal green subtree in $T$.
3. Every extraneous subtree $\{s, t\}$ which is a subtree of $S^*$ is also a subtree of $T^*$ and vice versa.

**Decomposition of canonical optimal R-RF(+) completions.** Given an extraneous subtree $\{s, t\}$ in canonical optimal R-RF(+) completions $S^*, T^*$ of $S$ and $T$, where $s \in V(S)$ and $t \in V(T)$, we define a **decomposition** of the extraneous subtree $\{s, t\}$ as a modification of the completions $S^*$ and $T^*$, yielding new completions $S'$ and $T'$ with strictly fewer extraneous subtrees, as follows:

1. If either none or both of the nodes $pa_S\{s, t\}$ and $pa_T\{s, t\}$ are matches (in $S^*$ and $T^*$), then the decomposition occurs as described below.
   - In tree $T^*$, prune out the grafted subtree $S(s)$ and regraft it at the parent edge of node $sib_{T^*}\{s, t\}$.
   - In tree $S^*$, prune out the grafted subtree $T(t)$ and regraft it at the parent edge of node $pa_S\{s, t\}$. If $pa_S\{s, t\} = rt(S^*)$, then create a new root node with children $t$ and $pa_S\{s, t\}$.

2. Otherwise, if exactly one of the nodes $pa_S\{s, t\}$ and $pa_T\{s, t\}$ is a matched node (in $S^*$ and $T^*$), then the decomposition occurs as described below. Without loss of generality, assume that $pa_S\{s, t\}$ is a match and $pa_T\{s, t\}$ a mismatch.
   - In tree $S^*$, prune out the grafted subtree $T(t)$ and regraft it at the parent edge of node $sib_S\{s, t\}$.
   - In tree $T^*$, prune out the grafted subtree $S(s)$ and regraft it at the parent edge of that unique node $u \in V(T^*)$ for which $C_{T^*}(u) = C_{S^*}(pa_S\{s, t\})$. If $u = rt(S^*)$, then create a new root node with children $s$ and $pa_S\{s, t\}$. Note that $u$ must exist since $pa_S\{s, t\}$ is a matched node.

This decomposition is illustrated in Figure 3. The following lemma characterises how the RF distance between $S^*$ and $T^*$ is impacted as their extraneous subtrees are decomposed.
TwoTreeCompletion

the R-RF(+) problem. Let under the EF-R-RF(+) problem. Then, We will refer to the placement of a maximal colored subtree under the Algorithm as a 

As shown above, optimal EF-R-RF(+) completions 2-approximate RF(+) distance. We now

Lemma 3.2. Let \( S' \) and \( T' \) denote the trees obtained by decomposing extraneous subtree \( \{s, t\} \) in completions \( S^* \) and \( T^* \), respectively.

1. If \( pa_S, \{s, t\} \) and \( pa_T, \{s, t\} \) are both matched nodes then \( RF(S', T') = RF(S^*, T^*) \).
2. If exactly one of \( pa_S, \{s, t\} \) and \( pa_T, \{s, t\} \) is a matched node then \( RF(S', T') = RF(S^*, T^*) \).
3. If neither \( pa_S, \{s, t\} \) nor \( pa_T, \{s, t\} \) is a matched node then \( RF(S', T') = RF(S^*, T^*) + 2 \).

The 2-approximation now follows by appropriately bounding the number of extraneous subtrees \( \{s, t\} \) that fall in category 3 of the above lemma.

Theorem 3.3. Let \( S^* \) and \( T^* \) represent optimal completions of \( S \) and \( T \), respectively, under the R-RF(+) problem. Let \( S' \) and \( T' \) represent optimal completions of \( S \) and \( T \) respectively under the EF-R-RF(+) problem. Then, \( RF(S', T') \leq 2 \cdot RF(S^*, T^*) \).

4 An Efficient Exact Algorithm for R-RF(+) Distance

As shown above, optimal EF-R-RF(+) completions 2-approximate RF(+) distance. We now show how to construct optimal R-RF(+) completions by modifying optimal EF-R-RF(+) completions.

Notation and terminology. We refer to EF-R-RF(+) completions resulting from the TwoTreeCompletion Algorithm of [4] as canonical EF-R-RF(+) completions. This is due to the way that maximal red and green subtrees are topologically well placed in such completions. We will refer to the placement of a maximal colored subtree under the TwoTreeCompletion Algorithm as a canonical EF-R-RF(+) position. The placement of each maximal red subtree \( R \) of \( S \), rooted at \( r \), in canonical EF-R-RF(+) completion \( T' \) of \( T \) has the useful property that all leaves \( a \in Le(S) \cap Le(T) \) where \( lca_S(a, r) = pa_S(r) \) also satisfy \( lca_T(a, r) = pa_T(r) \), and all leaves \( b \in Le(S) \cap Le(T) \) where \( lca_T(b, r) > pa_T(r) \) also satisfy \( lca_S(b, r) > pa_S(r) \).
By Theorem 3.1, we know that there exists an optimal pair of R-RF(+) completions where the only extraneous subtrees are of the form \{s, t\}. We will first show that a canonical pair of R-RF(+) completions can be constructed by taking a canonical pair of EF-R-RF(+) completions and pairing up extraneous subtrees of the form \{s, t\} in an optimal manner. We will then design a recurrence relation which computes the best possible change to the RF distance caused by pairwise extraneous subtrees of the form \{s, t\}, and show that this change to the RF distance can be computed in near linear time depending on the leaf-set overlap between the input trees.

**Lemma 4.1.** There exist canonical R-RF(+) completions \(S^*\) and \(T^*\) of rooted binary trees \(S\) and \(T\) such that every subtree grafted into \(S^*\) and \(T^*\) is either in an extraneous subtree or in its canonical EF-R-RF(+) position.

In the remainder of this section, let \(S', T'\) and \(S^*, T^*\) represent canonical EF-R-RF(+) and R-RF(+) completions of \(S\) and \(T\), respectively. We will soon define the subproblems that are the basis of our dynamic programming algorithm. Before doing so, we motivate the dynamic programming recurrence relation with the following lemma, which describes a new useful tree \(T''\) that is easier to construct from \(T'\) and preserves the important topological structure of \(T^*\). Our dynamic programming algorithm actually constructs \(T''\), and we can then easily use \(T''\) to generate \(S^*\) and \(T^*\).

**Lemma 4.2.** Let \(T''\) be the tree obtained by taking \(T^*\) and regrafting every extraneous subtree \(\{s, t\}\) along the parent edge of \(lca_T(\{lca_T(\text{Le}(sib_S(s)), t)\})\). Then \(RF(S', T'') = RF(S^*, T^*) + 2m\), where \(m\) is the number of extraneous subtrees \(\{s, t\}\) contained in \(T^*\).

Note that \(T''\) itself may not be a completion of \(T\). In particular, in the construction of \(T''\), pruning and regrafting the maximal green subtree \(T(t)\) is necessary if the extraneous subtree \(\{s, t\}\) is formed and \(lca_T(\{s, t\}) \neq p\sigma_T(\{s, t\})\). Moving any subtree of \(T\) in \(T'\) changes \(T'\) to no longer be a completion of \(T\). Figure 4 shows a concrete example.

**Definition 4.3.** Let the colors red and green be associated with the binary values 0 and 1, respectively. For \(v \in V(T')\) and \(c \in \{0, 1\}\), let \(c\text{Max}(c, v)\) be the total number of maximal subtrees of color \(c\) in \(T'(v)\). Moreover, let \(m\) be an integer such that \(0 \leq m \leq c\text{Max}(c, v)\). We define \(Cost(v, m, c)\) to be \(\min_{c'}(RF(S', T') - 2p - RF(S^*, T^*))\), where \(T'\) is obtained from \(T''\) by regrafting maximal red and green subtrees in \(T'(v)\) under the constraint that each extraneous subtree \(\{s, t\}\) is grafted along the parent edge of \(lca_T(v)(s, t)\) and exactly \(m\) maximal \(c\)-colored subtrees in \(T'(v)\) have been regrafted along the parent edge of \(v\), excluding extraneous subtrees (see Figure 5 for an example), and \(p\) denotes the number of extraneous subtrees of the form \(\{s, t\}\) in \(\hat{T}\).

In the trivial case when \(v\) is the root of a maximal \(c\)-colored subtree, we will say that it is possible to push one red subtree up to the parent edge of \(v\) or down from the parent edge of \(v\).

Note that the \(Cost()\) subproblem builds the optimal RF(+) distance. However, the cost is defined based on Lemma 4.2 by constructing \(T''\) and subtracting out the extraneous subtrees as they are produced. Moreover, we subtract the constant term \(RF(S', T')\) to express the cost as the change in RF distance.

We point out that the choice of \(\hat{T}\) implying \(Cost(\sigma_T(T'), 0, 0)\) is exactly \(T''\) by Lemmas 4.1 and 4.2. Furthermore, for any internal node \(v\) in \(T'\), and for the choice of \(m, c\) which imply the optimal cost value of \(Cost(\sigma_T(T'), 0, 0)\) via the upcoming recurrence relation, the tree \(\hat{T}(v)\) which admits \(Cost(v, m, c)\) is exactly equal to \(T''(v)\). In this sense, each \(\hat{T}\) captures an entire subtree of \(T''\). Note that on a local scale, in any specific \(\hat{T}\) there may be a red or green
The tree $T''$. This figure shows the relationship between $T'$, $T''$, and $T^*$. In this example, observe that there is exactly one extraneous subtree $\{s,t\}$ in the optimal completions $S^*$ and $T^*$, and that $RF(S', T'') = RF(S^*, T^*) + 2$. Moreover, $T''$ in this example cannot be a completion of $T$ since the green leaf $i$ has been regrafted. But constructing $T''$ is simply an intermediary step for constructing completions $S^*$ and $T^*$. Matches are denoted by filled in nodes.

subtree regrafted outside of an extraneous subtree and outside of its canonical EF-R-RF(+) position. However, it can be concluded that either eventually these red and green subtrees will be paired in extraneous subtrees for some later $\hat{T}$, or the particular cost value does not imply the optimal $Cost(rt(T''), 0, 0)$.

The next lemma provides a recurrence relation that can compute each $Cost(v, m, c)$ efficiently. In this recurrence relation, a subscript of $L$ or $R$ denotes the left or right child, respectively. For example, if a vertex $v$ is an internal node in $T$ then $v_L$ is the left child of $v$, and if $c$ is a color associated with vertex $v$ then $c_L$ is a color associated with vertex $v_L$. Note that the trees are unordered, so we use “left” and “right” here only to distinguish between the two children of an internal node.

Lemma 4.4. Let $f(m_i, v_i, c_i)$ equal 2 when $m_i > 0$ and $v_i$ is a match with color other than $c_i$, and 0 otherwise. Let $g_c(m_L, m_R, c_L, c_R)$ equal $2 \cdot \min \{m_L, m_R\}$ when $c_L \neq c_R$, and 0 when $c_L = c_R = c$. Then,

$$Cost(v, m, c) = \min_{m_L, m_R, c_L, c_R} \left\{ Cost(v_L, m_L, c_L) + Cost(v_R, m_R, c_R) + f(m_L, v_L, c_L) + f(m_R, v_R, c_R) - g_c(m_L, m_R, c_L, c_R) \right\}$$

if $v$ is an internal node of $T'$, and $Cost(v, m, c) = 0$ if $v$ is a leaf of $T'$, where:

(a) $c, c_L, c_R \in \{0, 1\}$, and either $c_L \neq c_R$ or $c_L = c_R = c$,

(b) $0 \leq m \leq c\text{Max}(c, v)$,

(c) If $c_L \neq c_R$, then $m_i - m_j = m$ for $i, j \in \{L, R\}$, $i \neq j$ satisfying $c_i = c$,

(d) If $c_L = c_R = c$, then $m_L + m_R = m$

The functions $f$ and $g_c$ from Lemma 4.4 both track local changes in matched and mismatched nodes. In particular, $f$ tracks a local change between $RF(S', T')$ and $RF(S', T'')$ while $g_c$ tracks a local change between $RF(S', T'')$ and $RF(S^*, T^*)$. We now provide our dynamic programming algorithm for computing the R-RF(+) distance between $S$ and $T$. 

![Figure 4](image-url)
Figure 5 Illustration of tree $\hat{T}$. The figure shows an example of what the tree $\hat{T}$ might look like after computing $\text{Cost}(u, 2, 1)$, where $c$ and $d$ have both been regrafted iteratively along the parent edge of $u$ and not regrafted into an extraneous subtree. Note that the extraneous subtree $\{e, f\}$ has also been regrafted along the parent edge of $u$, though it does not contribute to the value of $m = 2$. In particular, $u = \text{lca}_{T'}(e, f)$, so the extraneous subtree $\{e, f\}$ will appear at the same position in $\hat{T}$ and $T''$. Moreover, $f$ is not included as one of the two maximal green subtrees grafted onto the parent edge of $u$ since it is a part of an extraneous subtree. For each choice of vertex $v$, integer $m$ and color $c$ implying to the minimum $\text{Cost}(\text{rt}(T'), 0, 0)$ value, the corresponding optimal $\hat{T}$ provides the topological structure of $T''$ when restricted to the subtree rooted at $v$.

Algorithm 1 Compute-R-RF+(S,T).

1: Compute the EF-R-RF(+) completions $S'$ and $T'$ of $S$ and $T$.
2: for $v$ in $T'$ in postorder do
3: if $v$ is a leaf then
4: Set $\text{Cost}(v, 0, 0) = \text{Cost}(v, 0, 1) = 0$.
5: if $v$ is the root of a maximal red (0) or green (1) subtree then
6: Set $\text{Cost}(v, 1, c_v) = 0$, where $c_v$ is the color of $v$.
7: else
8: for each color $c$ and value $0 \leq m \leq c_{\text{Max}}(c, v)$ do
9: Compute $\text{Cost}(v, m, c)$ using the recurrence relation from Lemma 4.4
10: return $\text{RF}(S', T') + \text{Cost}(\text{rt}(T'), 0, 0)$

The algorithm above can be easily augmented to compute optimal completions by backtracking and determining the optimal values of $m$ and $c$ at each vertex of $T'$ implying $\text{Cost}(\text{rt}(T'), 0, 0)$. Using these optimal $m$ and $c$ values, we can determine when opposite colored subtrees converge and construct $T''$. From $T''$, we simply move each extraneous subtree $\{s, t\}$ into the canonical EF-R-RF(+) position for $T(t)$ to build $T^*$ and form the same extraneous subtrees in $S'$ to build $S^*$.

Theorem 4.5. The RF(+) distance between two rooted binary trees $S$ and $T$ can be computed in $O(nk^2)$ time, where $n = |\text{Le}(S) \cup \text{Le}(T)|$ and $k$ is the number of maximal red and green subtrees in $S$ and $T$.

5 Extension to Unrooted Trees

Our algorithm for the R-RF(+) problem can be easily adapted for the U-RF(+) problem. Specifically, the following algorithm computes the unrooted RF(+) distance between two unrooted input trees $S$ and $T$ with at least one leaf in common.
Optimal Phylogenetic Tree Completion

Algorithm 2 Compute-U-RF+(S, T).
1: Let l be any leaf from Le(S) ∩ Le(T). Produce two rooted trees ̂S and ̂T by rooting S and T, respectively, on the edge which connects l to the rest of each tree.
2: Compute the RF(+) distance d between ̂S and ̂T using Algorithm Compute-R-RF+(S, T).
3: Return d

The correctness of this algorithm is easy to establish based on the well-understood association between rooted and unrooted RF distances [10, 4], and further technical details and proofs are therefore omitted. This yields the following two theorems.

Theorem 5.1. The U-RF(+) problem can be solved in O(nk²) time, where n = |Le(S) ∪ Le(T)| and k is the number of maximal red and green subtrees in the corresponding EF-U-RF(+) completion of S or T.

Theorem 5.2. Let S* and T* represent optimal completions of unrooted trees S and T, respectively, under the U-RF(+) problem. Let S' and T' represent optimal completions of S and T, respectively, under the EF-U-RF(+) problem. Then, RF(S*, T*) ≤ 2 · RF(S', T').

6 Experimental Evaluation

We implemented our exact algorithm and performed experiments to assess the impact of using RF(+) distance instead of RF(-) distance on inferences related to tree similarity. We also conducted experiments to assess how well the linear-time algorithm for computing EF-RF(+) distances approximates RF(+) distances in practice. All our experiments were performed using real biological phylogenetic tree datasets on marsupials [8] (158 trees), legumes [33] (22 trees), and placental mammals [7] (726 trees).

Experiment 1: Conflicts between RF(+) and RF(-). Given two trees S and T, let RF+(S, T) and RF−(S, T), respectively, denote the RF(+) and RF(-) distances between them. We used the above datasets to measure the number of times that for any “base” tree S, there is a tree T1 which is closer to S than T2 under one of RF(+) or RF(-) but not closer under the other distance measure. This motivates the following definitions to describe each possible case of a change in order.

Type-1 Triples: Triple (S, T1, T2) is Type-1 if RF−(S, T1) < RF−(S, T2) but RF+(S, T1) > RF+(S, T2), or RF−(S, T2) < RF−(S, T1) but RF+(S, T2) > RF+(S, T1). A Type-1 triple indicates when the ordering of T1 and T2 by distance from S strictly changes as the distance function changes between RF(-) and RF(+).

Type-2 Triples: Triple (S, T1, T2) is Type-2 if RF−(S, T1) = RF−(S, T2) but RF+(S, T1) ≠ RF+(S, T2). A Type-2 triple indicates when T1 and T2 have equal distance to S under RF(-) but not under RF(+).

Type-3 Triples: Triple (S, T1, T2) is Type-3 if RF−(S, T1) ≠ RF−(S, T2) but RF+(S, T1) = RF+(S, T2). A Type-3 triple indicates when T1 and T2 have equal distance to S under RF(+) but not under RF(-).

Observe that the magnitude of difference between RF(+) and RF(-) distances depends on the level of overlap between the trees being compared. To account for this effect, we define the leaf-overlap ratio of a pair of trees (S, T) to be the following ratio: |Le(S) ∩ Le(T)| divided by min{|Le(S)|, |Le(T)|}, and the leaf-overlap ratio of a triple of trees S, T1, and T2 to be the minimum pairwise leaf-overlap ratio between (S, T1) and (S, T2).
Figure 6  Fraction of conflicting triples for different leaf-overlap ratios. The figure contains three plots, one for each dataset, which each show the fraction of triples of type-1, type-2, and type-3 for different ranges of leaf-overlap ratio, among all triples of trees within the same leaf-overlap ratio range in that dataset. The dotted line represents the total number of conflicting triples (i.e., all triples of type 1, 2, or 3). x-axis labels denote the center of each interval of size 0.1. Each leaf-overlap ratio range is a closed interval and includes the boundary, e.g., x-axis label 0.15 represents the range \([0.1 - 0.2]\).

We performed this experiment for each subset of three trees from each dataset, and Figure 6 shows its results. As the figure shows, the proportion of conflicting triples (type-1, 2, or 3) tends to increase as the triple leaf-overlap ratio increases. In particular, at least 10% of all triples show a conflict (either of type-1, 2, or 3) when the leaf-overlap ratio is 0.7 or greater. Even for leaf-overlap ratio as small as 0.4, we find that 5% of all triples show a conflict. This demonstrates that RF(+) and RF(-) frequently differ starkly in their assessments of relative similarities between trees. Observe that the results on the Legumes dataset are vastly different from the results on the other two datasets. This is mainly because the Legumes dataset consists of only 22 trees, which is significantly smaller than the 158 tree and 726 tree datasets. For instance, the number of triples within each leaf overlap ratio range (interval size 0.1) is between 8,214,518 and 50,815,687 for the placental mammals dataset, between 3,287 and 1,652,701 for the Marsupials dataset, but only 6, 16, 5, and 0, respectively, for the Legumes dataset for leaf overlap ratio ranges \([0.5 - 0.6]\), \([0.6 - 0.7]\), \([0.7 - 0.8]\), and \([0.8 - 0.9]\).

Experiment 2: Impact on phylogenetic database search and clustering. Next, we assessed the potential impact of using RF(+) distance on applications related to phylogenetic database search and clustering. Specifically, we assessed how, for each “query” tree in each dataset, the sets of the “closest” trees to it differed under RF(+) and RF(-). Specifically, we measured how the sets of (i) the most similar trees and (ii) the most similar 10% of trees (i.e., top 10% closest matches) differed when using RF(+) and RF(-) distances. To avoid any ambiguity in defining these sets, we include all trees with equal distance, even if that results in sets of different sizes under RF(+) and RF(-).

For our comparison of the most similar trees, we found that the sets of closest trees under RF(+) and RF(-) all had a distance of 0 to the query tree and were identical, for all choices of the query tree in all datasets. To perform a more meaningful comparison, we therefore required a minimum leaf-overlap ratio of 0.7, i.e., only those trees with a minimum leaf-overlap ratio of 0.7 with the query tree could be compared with the query tree. Likewise,
Figure 7: Difference between sets of closest trees under $RF(+)$ and $RF(-)$. Plots in the left column show the number of query trees where the set of closest trees with a minimum leaf-overlap ratio of 0.7 differ under $RF(+)$ and $RF(-)$ distances for each of the three biological data sets. Plots in the right column show the number of query trees where the set of closest 10% of trees with a minimum leaf-overlap ratio of 0.5 differ under $RF(+)$ and $RF(-)$ distances. Results are presented for varying levels of difference between the sets (labels on the $x$-axes). The sizes of the datasets, in order from top to bottom, are 158 trees, 22 trees and 726 trees. Each tree in each of these datasets was used as a query tree for this analysis.

for our comparison of the most similar 10% of trees, we found that the sets of closest 10% of trees were generally identical under $RF(+)$ and $RF(-)$ if no minimum leaf-overlap ratio was imposed. We therefore imposed a minimum leaf-overlap ratio of 0.5 for the analysis, which was the smallest ratio for which a non-negligible fraction of query trees returned differing sets under $RF(+)$ and $RF(-)$. Figure 7 shows the results of both these analyses. We find that there are several query trees in each dataset for which there is a large difference (normalised symmetric difference greater than, say, 0.4) between their sets of closest trees under $RF(+)$ and $RF(-)$. For the sets of closest 10% of trees, we find that over 25% of trees in the marsupials dataset, 18% of trees in the legumes dataset, and almost 15% of trees in the placental mammals dataset return different sets of closest 10% of trees under $RF(+)$ and $RF(-)$ distances. These results demonstrate how using $RF(+)\text{distance}$ can substantially impact phylogenetic database search and phylogenetic tree clustering, especially when the trees under consideration have a sufficiently large overlap in their leaf sets.

Experiment 3: Comparison of $EF-RF(\text{+})$ and $RF(\text{+})$. Finally, we used simulated and real datasets to compare the distances inferred under $EF-RF(\text{+})$ and $RF(\text{+})$, and to study the runtime and scalability of our implementation. For our analysis with simulated data, we generated two datasets of random trees using the birth-death model implemented in SaGePhy [21] (specific parameter values: height of tree = 1.0, birth rate = 5.0 and death
rate = 0.05). The first simulated dataset consisted of 100 randomly generated trees, each with between 200 and 300 leaves. The second simulated dataset also consisted of 100 randomly generated trees, but each with between 900 and 1000 leaves. The average leaf-set sizes for these two datasets were 244.95 and 941.14, respectively, and the average pairwise leaf-overlap ratio for both datasets was approximately 0.5. For each pair of trees in each dataset, we measured how close the EF-RF(+) distance is to the RF(+) distance for that pair. Figure 8 plots the distribution of the ratio of RF(+) distance to EF-RF(+) distance for the two datasets. As that figure shows, the ratio of RF(+) distance to EF-RF(+) distance is approximately 0.92, on average, and roughly follows a Gaussian distribution.

For the three biological datasets, we found that the ratio of RF(+) distance to EF-RF(+) distance was equal to one for an overwhelmingly large proportion of pairs of trees within all three datasets. Specifically, for the marsupials, legumes, and placental mammals datasets, the average ratios of RF(+) distance to EF-RF(+) distance were 0.998, 0.993, and 0.995, respectively. In fact, 99.07%, 93.81%, and 96.82% of the pairs in these datasets, respectively, had identical EF-RF(+) and RF(+) distances. Even when the trees being compared were restricted to have at least 0.4 leaf-overlap ratio, 95.97%, 78.79%, and 95.59% of the pairs in marsupials, legumes, and placental mammals datasets, respectively, had identical EF-RF(+) and RF(+) distances. This discrepancy between results for simulated data and real data is not surprising since we expect any pair of randomly generated trees to have smaller maximal red and green subtrees and greater RF(-) distance, presenting more opportunities to improve the distance by creating extraneous clades. Together, these results on simulated and real datasets show that EF-RF(+) distance, which is linear-time computable, is generally very close to RF(+) distance in practice.

Runtime comparison. We also measured the runtimes of the two algorithms and found that, on average, computing EF-RF(+) distances took 0.06 seconds for the first simulated dataset and 0.25 seconds for the second simulated dataset. Corresponding average runtimes for computing RF(+) distances were 0.17 seconds and 1.04 seconds, respectively. All timed experiments were run on a single core of a 2.1 GHz Intel Xeon processor.

7 Conclusion

Completion based comparison of incomplete phylogenetic trees is an emerging, promising approach for tree comparison. In this work, we developed the first polynomial-time exact algorithm for the RF(+) problem. We also established a linear-time 2-approximation algorithm for the problem. These algorithms allow for more principled comparison of
incomplete phylogenetic trees than was hitherto possible, and our experimental analysis shows that RF(+) distance can lead to very different inferences regarding phylogenetic similarity compared to traditional RF distance. Moreover, our results suggest that the linear-time 2-approximation algorithm for the RF(+) problem almost always computes optimal or near-optimal RF(+) distances in practice.

In addition to their utility for improved tree comparison and clustering, our solutions for the RF(+) problem also have implications for phylogenomics. Many modern phylogenomics methods for reconstructing evolutionary histories and understanding genome-scale patterns of evolution are designed to work with complete phylogenies from genomic loci across the genomes of the considered species [5, 26, 27, 20, 12], and loci that yield incomplete phylogenies are often discarded, resulting in only a fraction of the available genomic sequence information being used for the phylogenomic analysis. Thus, problems related to optimal completion of incomplete phylogenies (i.e., imputation of complete phylogenies) arise naturally in phylogenomics. Our algorithms for the RF(+) problem may provide a principled way to impute such complete phylogenies.

The current work is restricted to comparison of binary trees under the Robinson-Foulds metric, and it can be extended in many useful ways. A possible next step could include consideration of non-binary trees in computing distances between incomplete trees. Future work could also entail development of similar completion based methods under other distance/similarity measures such as triplet/quartet distances [14, 17], nearest neighbor interchange (NNI) and subtree prune and regraft (SPR) distances [31, 18, 34], and nodal distance [9]. Furthermore, the idea of computing optimal completions could be extended to multi-labeled trees, which arise frequently in practice due to evolutionary events such as gene duplication.

**References**


Suppose, without loss of generality, that \( r_2 \) is the only subtree removed (matches except for one representing the smallest match containing \( R \)). Furthermore, the matched clade \( Le(R_2) \) is formed in both \( S^* \) and \( T^* \), which counteracts this lost match.

### Appendix

**Proof of Theorem 3.1.** Let \( S^* \) and \( T^* \) be arbitrarily chosen optimal completions of \( S \) and \( T \) under R-RF(+). We will modify \( S^* \) and \( T^* \) to be of the desired form. To do so, we first show that any maximal red subtree in \( S \) and any maximal green subtree of \( T \) can be made subtrees of \( S^* \) and \( T^* \) without increasing the RF distance between them (condition 1). Suppose there exist two maximal matched red subtrees \( R_1 \) and \( R_2 \) of \( S^* \) and \( T^* \) which neighbor each other in the original tree \( S \). Let \( r_1 \) and \( r_2 \) be the roots of \( R_1 \) and \( R_2 \).

1. Suppose both \( C_{T^*}(pa_{T^*}(r_1)) \setminus C_{T^*}(r_1) \) and \( C_{T^*}(pa_{T^*}(r_2)) \setminus C_{T^*}(r_2) \) contain some non-green leaves. Observe that every matched clade in \( T^* \) containing \( C_{T^*}(r_1) \cup C_{T^*}(r_2) \) must also contain \( C_{T^*}(lca_{T^*}(r_1, r_2)) \) because \( R_1 \) and \( R_2 \) neighbor each other in \( S \) by assumption. Therefore, we can regraft \( R_2 \) to neighbor \( R_1 \) in \( T^* \) without increasing the RF distance between \( S^* \) and \( T^* \). Moreover, if there are any green subtrees inserted along the path from \( R_1 \) to \( R_2 \) in \( S^* \), then they can be regrafted along the parent edge of \( lca_{T^*}(r_1, r_2) \) without increasing the Robinson-Foulds distance.

2. Suppose, without loss of generality, that \( C_{T^*}(pa_{T^*}(r_2)) \setminus C_{T^*}(r_2) \) contains only green leaves. That is, suppose \( R_2 \) is contained in an extraneous subtree, whose root could be a match without anestorcing \( R_1 \). First, regraft \( R_2 \) in \( T^* \) to neighbor \( R_1 \). Then, regraft all green subtrees from the path in \( S^* \) connecting \( R_2 \) and \( R_1 \) to the parent edge of \( lca_{T^*}(r_1, r_2) \), preserving the topological structure of the green leaves. This does not increase the RF distance between \( S^* \) and \( T^* \). Notice that any originally matched clades containing \( Le(R_2) \) are mismatched. However, preserving the topological structure of the green leaves from any matched clades containing \( Le(R_2) \) also retains the same number of matches except for one representing the smallest match containing \( R_2 \). This is because the only subtree removed (in both \( S^* \) and \( T^* \)) from these matched extraneous subtrees is \( R_2 \).
If this is done iteratively for all such $R_1$ and $R_2$, we then conclude that there exist optimal completions $S^*$ and $T^*$ where every maximal red subtree in $S$ is also a subtree of $S^*$ and $T^*$. The same argument applies for maximal green subtrees.

Now we will show that $S^*$ and $T^*$ can be modified to only contain extraneous subtrees of the form $\{s, t\}$ without increasing the RF distance (condition 2). We will simultaneously show that an extraneous subtree $\{s, t\}$ is a subtree of $S^*$ if and only if it is a subtree of $T^*$ by construction (condition 3). Observe that if $Le(U) \cap Le(V) \cap Le(S) \neq \emptyset$ for two maximal extraneous subtrees $U$ and $V$ of $S^*$ and $T^*$ respectively, then $Le(U) \cap Le(V) \cap Le(S) \subseteq Le(R)$ for a single maximal red subtree $R$ of $S$. Likewise if $Le(U) \cap Le(V) \cap Le(T) \neq \emptyset$, then $Le(U) \cap Le(V) \cap Le(T) \subseteq Le(Y)$ for a single maximal green subtree $Y$ of $T$. Therefore, every maximal extraneous subtree in $S^*$ or $T^*$ satisfies one of the following two cases.

1. Without loss of generality, let $U$ be a maximal extraneous subtree of $S^*$ rooted at $u$ such that for every maximal extraneous subtree $V$ of $T^*$, $Le(U) \cap Le(V) \cap Le(S) \neq \emptyset$ or $Le(U) \cap Le(V) \cap Le(T) \neq \emptyset$. Then, every extraneous clade contained in $Le(U)$ must be a mismatch. Hence, every maximal green subtree of $U$ can be regrafted along the parent edge of $pa_{S^*}(u)$ without increasing the Robinson-Foulds distance from $T^*$. This results in destroying all extraneous subtrees contained in $U$ because $pa_{S^*}(u)$ is an ancestor of a maximal extraneous subtree and therefore possesses uncolored descendants.

2. Let $U$ and $V$ be maximal extraneous subtree of $S^*$ and $T^*$, rooted at $u$ and $v$ respectively, satisfying $Le(U) \cap Le(V) \cap Le(S) \neq \emptyset$ and $Le(U) \cap Le(V) \cap Le(T) \neq \emptyset$. Then, every matched extraneous clade contained in $Le(U)$ and $Le(V)$ must contain elements of $Le(U) \cap Le(V) \cap Le(S)$ and $Le(U) \cap Le(V) \cap Le(T)$. Every maximal green subtree of $U$ with no leaves in $Le(U) \cap Le(V) \cap Le(T)$ can be regrafted along the parent edge of $u$ without increasing the RF distance. Likewise, every maximal red subtree of $V$ with no leaves in $Le(U) \cap Le(V) \cap Le(S)$ can be regrafted along the parent edge of $v$ without increasing the RF distance. Moreover, as described before, $Le(U) \cap Le(V) \cap Le(S) \subseteq Le(R)$ and $Le(U) \cap Le(V) \cap Le(T) \subseteq Le(Y)$ for a single maximal red subtree $R$ of $S$ and a single maximal green subtree $Y$ of $T$. Hence, we are only left with the extraneous subtree $\{rt_{S^*}(R), rt_{T^*}(Y)\}$ in $S^*$ and $\{rt_{T^*}(R), rt_{T^*}(Y)\}$ in $T^*$.

Once every maximal extraneous subtree in $S^*$ and $T^*$ is handled according to the appropriate case above, we are left with two optimal completions $S^*$ and $T^*$ of the desired form. ▶

**Proof of Lemma 3.2**. Case 1: In this case, both $pa_{S^*}\{s, t\}$ and $pa_{T^*}\{s, t\}$ are matched nodes. Here, we must have $Le(S^*(pa_{S^*}\{s, t\})) = Le(T^*(pa_{T^*}\{s, t\}))$. This holds because $C_{S^*}(pa_{S^*}\{s, t\})$ and $C_{T^*}(pa_{T^*}\{s, t\})$ are both matches, and the smallest proper super-subs of $C_{T^*}(s) \cup C_{T^*}(t)$ in $S^*$ and $T^*$ respectively. By definition, the decomposition replaces the matched clades $C_{S^*}(s) \cup C_{S^*}(t)$ and $C_{T^*}(s) \cup C_{T^*}(t)$ with $C_{S^*}(pa_{S^*}\{s, t\}) \setminus C_{S^*}(t)$ and $C_{T^*}(pa_{T^*}\{s, t\}) \setminus C_{T^*}(t)$ in $S^*$ and $T^*$ respectively. Since $Le(S^*(pa_{S^*}\{s, t\})) = Le(T^*(pa_{T^*}\{s, t\}))$, we conclude that $C_{S^*}(pa_{S^*}\{s, t\}) \setminus C_{S^*}(t)$ and $C_{T^*}(pa_{T^*}\{s, t\}) \setminus C_{T^*}(t)$ are then matches in the resulting trees $S'$ and $T'$.

Case 2: We now consider the case when exactly one of the nodes $pa_{S^*}\{s, t\}$ and $pa_{T^*}\{s, t\}$ is a matched node. Without loss of generality, suppose $pa_{S^*}\{s, t\}$ is a match and $pa_{T^*}\{s, t\}$ is not a match. For convenience, let $x$ denote $pa_{S^*}\{s, t\}$, $y$ denote $pa_{T^*}\{s, t\}$, and let $u$ be the element of $V(T^*)$ such that $C_{S^*}(x) = C_{T^*}(u)$. Then, observe that $C_{S^*}(x) \supset C_{T^*}(y)$, i.e., $y \prec u$ in $T^*$. Moreover, every node $v$ along the path from $y$ to $u$ in $T^*$ must be a mismatch since $C_{T^*}(t) \subset C_{T^*}(v)$ and $C_{S^*}(t) \cap C_{S^*}(sib_{S^*}\{s, t\}) = \emptyset$ but $C_{T^*}(v) \cap C_{S^*}(sib_{S^*}\{s, t\}) \neq \emptyset$ for arbitrary choice of $v$. Now, applying the decomposition of extraneous subtree $\{s, t\}$ to $S^*$ and $T^*$ yields the modified trees $S'$ and $T'$. Observe that this modification changes exactly the $\{s, t\}$ clade, and all clades along the path from $y$ to $u$ in $T^*$. In $S'$, the new clade
formed at the subtree rooted at \(pa_s'(t)\) must be a matched node since \(C_{S'}(pa_s'(t)) = C_T(u)\). Moreover, in \(T'\), all clades \(C_T(v)\) along the path from \(y\) to \(u\) remain mismatches except for \(C_T(u)\) because it still holds that \(C_T(t) \subseteq C_T(v)\) and \(C_{S'}(t) \cap C_{S'}(sib_{S'}(s, t)) = \emptyset\) but \(C_T(v) \cap C_s'(sib_{S'}(s, t)) \neq \emptyset\) for arbitrary choice of \(v\) along the path. Thus, after the decomposition, the number of matched clades in \(S'\) (w.r.t. \(T'\)) remains the same as the number of matched clades in \(S^*\) (w.r.t. \(T^*\)).

**Case 3:** If neither \(pa_{S'}(s, t)\) nor \(pa_{T'}(s, t)\) is a matched node, then, following the same argument as in Case 1, \(S'\) will have one less matched node (w.r.t. \(T'\)) than \(S^*\) (w.r.t. \(T^*\)). Namely, the clades \(C_{S'}(pa_{S'}(s, t)) \setminus C_{S'}(t)\) and \(C_{T'}(pa_{T'}(s, t)) \setminus C_T(t)\) are mismatched clades in \(S'\) and \(T'\) respectively. Consequently, \(T'\) will have one less matched node as well. Thus, \(RF(S', T') = RF(S^*, T^*) + 2\).

**Proof of Theorem 3.3.** Let \(d = \frac{1}{2}RF(S^*, T^*)\) and let \(e\) be the number of extraneous clades in \(S^*\). Then, we have that \(d\) is also the number of mismatches in \(S^*\), or equivalently in \(T^*\). Observe that at most \(d\) of the \(e\) extraneous clades have mismatched parent nodes in both trees. Thus, by Lemma 3.2, decomposing all \(e\) extraneous clades will increase the RF distance by at most \(2d = RF(S^*, T^*)\). Therefore, the decomposed extraneous clade free completion will have an RF distance of at most \(2 \cdot RF(S^*, T^*)\).

**Proof of Lemma 4.1.** Consider arbitrary canonical \(R-RF(+)\) completions \(S^*\) and \(T^*\). We will show that any grafted subtree in \(S^*\) and \(T^*\) that is not in its canonical \(EF-R(+)\) position or in an extraneous subtree can be regrafted into its canonical \(EF-R(+)\) position without increasing the RF distance. Without loss of generality, suppose there exists a maximal red subtree \(R\), with \(r\) denoting \(rt(R)\), in \(T^*\) such that \(R\) is neither in its canonical \(EF-R(+)\) position nor in an extraneous subtree. Let \(u\) represent the canonical \(EF-R(+)\) position of subtree \(R\) in completion \(T^*\). Thus, \(u \neq pa_T(r)\). Then, we have two possible cases: either \(pa_T(r)\) is an ancestor of \(u\) or not \((pa_T(r) > u\) or \(pa_T(r) \neq u\).

1. Suppose \(pa_T(r) > u\). We will prove that \(pa_T(r)\) can be regrafted in position \(u\) without increasing the RF distance. Since \(pa_T(r) > u\), for any arbitrary node \(c\) on the path from \(pa_T(r)\) to \(u\), there exists a subtree \(C\) of \(T^*(c)\) rooted at one of the children of \(c\) (the subtree not containing \(u\)) satisfying \(pa_T(r) > c = lca_T(u, Le(C)) > u\) and \(pa_{S'}(r) < lca_{S'}(r, Le(C))\). Since \(pa_T(r) > lca_T(u, Le(C)) > u\), we have that \(pa_T(r) > lca_T(Le(C), a) > a\) for all leaves \(a \in Le(S) \cap Le(T)\) such that \(a < pa_{S'}(r)\). Since for each such \(a\), we have that \(a < pa_{S'}(r) < lca_{S'}(a, Le(C))\) and \(a < lca_T(a, Le(C)) = c < pa_T(r)\), every match containing \(Le(C)\) must also contain \(Le(R)\). In particular, \(c\) is not a match. This is true for every node \(c\) along the path from \(pa_T(r)\) to \(u\). We can therefore regraft \(R\) at position \(u\) without increasing the RF distance because every node along the path from \(pa_T(r)\) to \(u\) is already mismatched.

2. Now suppose \(pa_T(r) \neq u\). We will prove that \(R\) can be regrafted along the parent edge of \(lca_T(pa_T(r), u)\) (equivalent position to \(u\) if \(u\) is an ancestor of \(pa_T(r)\)) without increasing the RF distance. This will then reduce the case where \(pa_T(r)\) is not an ancestor of \(u\) to the previous case where \(pa_T(r)\) is an ancestor of \(u\). If \(pa_T(r)\) is not an ancestor of \(u\), then there exist some \(a_1, \ldots, a_k \in Le(S) \cap Le(T)\) such that \(pa_T(r) > a_i\) and \(lca_T(pa_T(r), a_i) > pa_T(r)\) for all values of \(i\). Therefore, \(pa_T(r)\) is not a match, as well as every node on the same path up to the node \(lca_T(pa_T(r), a_1, \ldots, a_k)\) which contains every \(a_i\) in its clade \(C_T(lca_T(pa_T(r), a_1, \ldots, a_k))\). Then, regrafting \(R\) at the parent edge of \(lca_T(a_1, \ldots, a_k, pa_T(r)) = lca_T(pa_T(r), u)\) will not increase the RF distance since there are no matches to become mismatched.
Proof of Lemma 4.2. For binary trees $U$ and $V$, let $\mathcal{M}_U^V$ denote the LCA map from $U$ to $V$. That is, on input $u \in V(U)$, $\mathcal{M}_U^V(u)$ returns $\text{lea}_V(C_U(u))$. We will show that $RF(S', T'') - RF(S', T') = RF(S'', T'') - RF(S', T') + 2m$. Observe that the only changes from $S'$ and $T'$ to $S''$, $T''$ and $T''$ are the formations of the extraneous subtrees $\{s, t\}$. Then, it suffices to confirm that for every extraneous subtree $\{s, t\}$, the number of mismatched clades in $T''(\text{pa}_{T''}(s, t))$ equals the number of mismatched clades in $T''(\mathcal{M}_{S''}^T(\text{pa}_{S''}(s, t)))$ plus the number of extraneous subtrees. For an arbitrary extraneous subtree $\{s, t\}$ in $T''$, we first count the mismatched clades in $T''(\text{pa}_{T''}(s, t))$. Then, we count the mismatched clades in $T''(\mathcal{M}_{S''}^T(\text{pa}_{S''}(s, t)))$ and compare.

1. Suppose $v$ lies along the path from $\text{pa}_{T''}(s, t)$ to the parent of the canonical EF-R-RF(+) position for $T(t)$ in $T''$. Moreover, suppose $u$ lies along the path from $\text{pa}_{T''}(s, t)$ to the parent of the canonical EF-R-RF(+) position for $S(s)$ in $T''$. Then $C_{S''}(\mathcal{M}_{T''}^{S''}(v)) \supseteq C_{T''}(v) \cup C_{S''}(t)$ since $v$ is an ancestor of the canonical EF-R-RF(+) position of $T(t)$ in $T''$ and hence $\mathcal{M}_{T''}^{S''}(v)$ is an ancestor of the canonical EF-R-RF(+) position of $T(t)$ in $S''$. Moreover, $C_{T''}(v) \cap C_{S''}(t) = \emptyset$ if $v \neq \text{pa}_{T''}(s, t)$ by construction of $T''$. Hence if $v \neq \text{pa}_{T''}(s, t)$, then $v$ is mismatched with respect to $S'$. Likewise, $C_{S''}(\mathcal{M}_{T''}^{S''}(u)) \supseteq C_{T''}(u) \cup C_{S''}(s)$ and $C_{T''}(u) \cap C_{S''}(s) = \emptyset$ if $u \neq \text{pa}_{T''}(s, t)$. Hence if $u \neq \text{pa}_{T''}(s, t)$, then $u$ is mismatched with respect to $S'$. Note that by construction, $C_{T''}(\text{pa}_{T''}(s, t)) = C_T(\text{lea}_T(s, t))$. Hence $\text{pa}_{T''}(s, t)$ has a match with respect to $S'$ if and only if $\text{lea}_T(s, t)$ is, and every other node along either path is mismatched. Note that the only remaining node impacted in the formation of $\{s, t\}$ is the root of the extraneous subtree in $T''$. This node must be matched with respect to $S'$ since $S'$ is an extraneous free completion.

2. Now suppose $v$ lies along the path from $\text{pa}_{T''}(s, t)$ (the canonical EF-R-RF(+) position for $T(t)$ in $T''$) to $\mathcal{M}_{S''}^T(\text{pa}_{S''}(s, t))$ (the least common ancestor of the EF-R-RF(+) positions in $T''$). Moreover, suppose $u$ lies along the path from $\mathcal{M}_{S''}^T(\text{pa}_{S''}(s, t))$ to the parent of the canonical EF-R-RF(+) position for $S(s)$ in $T''$. Observe that $\mathcal{M}_{T''}^{S''}(v)$ is an ancestor of the extraneous subtree $\{s, t\}$ in $S''$, and therefore $\mathcal{M}_{S''}^T(u)$ is an ancestor of the canonical EF-R-RF(+) position for $S(s)$ in $S''$. Then $C_{S''}(\mathcal{M}_{T''}^{S''}(v)) \supseteq C_{T''}(v) \cup C_{S''}(\text{sib}_{S''}(s, t))$, where $C_{T''}(v) \cap C_{S''}(\text{sib}_{S''}(s, t)) = \emptyset$ if $v \neq \mathcal{M}_{S''}^T(\text{pa}_{S''}(s, t))$. Additionally, notice that $\mathcal{M}_{S''}^T(u)$ is an ancestor of the canonical EF-R-RF(+) position for $S(s)$ in $S''$, and therefore $\mathcal{M}_{S''}^T(u)$ is an ancestor of the extraneous subtree $\{s, t\}$. Then $C_{S''}(\mathcal{M}_{T''}^{S''}(u)) \supseteq C_{T''}(u) \cup C_{S''}(s)$, where $C_{T''}(u) \cap C_{S''}(s) = \emptyset$ if $u \neq \mathcal{M}_{S''}^T(\text{pa}_{S''}(s, t))$. It follows that if $u \neq \mathcal{M}_{S''}^T(\text{pa}_{S''}(s, t))$, then $u$ is a mismatched node. Likewise, if $v \neq \mathcal{M}_{S''}^T(\text{pa}_{S''}(s, t))$, then $v$ is a mismatched node. Furthermore, $C_{T''}(\mathcal{M}_{S''}^T(\text{pa}_{S''}(s, t)))$ is a matched clade with respect to $S'$ if and only if $C_{T''}(\text{lea}_T(s, t))$ is a matched clade with respect to $S'$. Note, again, that the only remaining node impacted in the formation of $\{s, t\}$ is the root of the extraneous subtree $\{s, t\}$. Since $S'$ and $T''$ are canonical R-RF(+) completions, we know that this node must be matched in $S''$ and $T''$. Now, observe that the union of paths connecting the canonical EF-R-RF(+) positions for $S(s)$ and $T(t)$ to $\text{pa}_{T''}(s, t)$ in $T''$ is the same size as the union of paths connecting the canonical EF-R-RF(+) positions for $S(s)$ and $T(t)$ to $\text{pa}_{T''}(s, t)$ in $T''$. Moreover, every node in each union of paths (except the common ancestor) is mismatched. Finally, the root of $\{s, t\}$ is mismatched in $T''$ but matched in $T''$. Since the choice of $\{s, t\}$ was arbitrary, we conclude with $RF(S', T'') - RF(S', T') = RF(S', T'') - RF(S', T') + 2m$, where $m$ is the number of extraneous subtrees in $T''$. Equivalently, $RF(S', T'') = RF(S', T') + 2m$. ▶
Proof of Lemma 4.4. Let $S, T$ be two input binary rooted trees, and let $S', T'$ be their canonical EF-R-RF(+) completions. By the proof of Lemma 4.1, we observe two important points: First, it can only be beneficial to move a maximal red or green subtree if the maximal subtree is eventually paired in an extraneous subtree. And second, a maximal red or green subtree will increase the RF distance by a lower amount if it is paired in an extraneous subtree closer to the canonical EF-R-RF(+) position. The recurrence relation follows by induction.

Base Case: No extraneous clades can be formed at a leaf node and there are no matches to become mismatched. Hence, the cost at each leaf is indeed zero.

Inductive Step: Assume we have computed all $\text{Cost}(x, \cdot, \cdot)$ for all descendants $x$ of an internal node $v$. Let $c \in \{0, 1\}$ and $0 \leq m \leq c \text{Max}(c, v)$ be arbitrarily given. We first show that twice the maximal number of new extraneous subtrees $\{s, t\}$ that can be formed at $v$ given $c_L, c_R, m_L$ and $m_R$ is equal to $g_c(m_L, m_R, c_L, c_R)$. There are two cases to consider: 1. $c_L = c_R = c$ and 2. $c_L \neq c_R$ (at least one of $c_L$ and $c_R$ must equal $c$).

1. Suppose $c_L = c_R = c$ and let $m_L, m_R$ be arbitrary nonnegative values such that $m_L + m_R = m$. Then by the first observation above, the condition $m_L + m_R = m$ is optimal to regraft $m$ subtrees of color $c_L = c_R = c$ along the parent edge of $v$. By the second observation above, if there are any extraneous subtrees that can be paired at $v$ then it is optimal to do so at $v$. We cannot pair any maximal red and green subtrees at $v$ because $c_L = c_R = c$, which means that all $m$ subtrees regrafted along the parent edge of $v$ are the same color. Hence, twice the number of new extraneous subtrees that can be formed at $v$ is equal to $g_c(m_L, m_R, c_L, c_R) = 0$ when $c_L = c_R = c$.

2. Now suppose without loss of generality that $c_L \neq c_R$ and let $m_L, m_R$ be arbitrary nonnegative values such that $|m_L - m_R| = m$. Then by the second observations above, the condition $|m_L - m_R| = m$ is optimal to regraft the $m_L + m_R$ subtrees on the parent edge of $v$. By the second observation above, if there are any extraneous subtrees that can be paired at $v$ then it is optimal to do so at $v$. Note that since $c_L \neq c_R$, we can pair exactly $\min\{m_L, m_R\}$ extraneous subtrees at $v$. Hence, twice the number of new extraneous subtrees that can be formed at $v$ is equal to $g_c(m_L, m_R, c_L, c_R) = 2 \min\{m_L, m_R\}$.

We now show that, regardless of the choice of colors $c_L$ and $c_R$, the new increase in RF distance between $S'$ and $T'$ only by regrafting $m_L$ and $m_R$ subtrees from $T'(v_L)$ and $T'(v_R)$ at the parent edge of $v$, respectively, is equal to $f(m_L, v_L, c_L) + f(m_R, v_R, c_R)$. Once a subtree is regrafted at the parent edge of $v_L$, the only clade that can become mismatched by regrafting the subtree on the parent edge of $v$ is $C_T(v_L)$. This clade only becomes mismatched if it is a matched clade and it is not contained in a maximal $c_L$-colored subtree. Once the clade is mismatched, regrafting all remaining $m_L$ maximal subtrees on the parent edge of $v$ cannot make $v$ mismatched again. Therefore, the act of pruning and regrafting $m_L$ maximal $c_L$-colored subtrees from the parent edge of $v_L$ to the parent edge of $v$ increases the RF distance between $S'$ and $T'$ by $f(m_L, v_L, c_L)$, one for each of $S'$ and $T'$ if a match becomes mismatched. By symmetry, the new increase in RF distance between $S'$ and $T'$ from pruning and regrafting $m_R$ maximal $c_R$-colored subtrees from $v_R$ to $v$ is equal to $f(m_R, v_R, c_R)$.

We have determined that the maximal number of new extraneous subtrees which can be formed is equal to $g_c(m_L, m_R, c_L, c_R)$, and the new increase in RF distance is $f(m_L, v_L, c_L) + f(m_R, v_R, c_R)$. Then the change in cost from $v_L$ and $v_R$ to $v$ is equal to $f(m_L, v_L, c_L) + f(m_R, v_R, c_R) - g_c(m_L, m_R, c_L, c_R)$. Note if a maximal $c_L$-colored subtree of $T'(v_L)$ is regrafted along the parent edge of $v$, it must first already be regrafted along parent edge...
of $v_L$ by construction. Then, the cost of regrafting $m_L$ subtrees at the parent edge of $v_L$ must be $\text{Cost}(v_L, m_L, c_L)$. By symmetry, the right subtree adds a cost of $\text{Cost}(v_R, m_R, c_R)$. Moreover, the cost values also subtract the number of extraneous subtrees formed in $T'(v_L)$ and $T'(v_R)$.

Hence, the value of $\text{RF}(S', \hat{T}) - 2p - \text{RF}(S', T')$ given fixed $c_L, c_R, m_L$ and $m_R$ is $\text{Cost}(v_L, m_L, c_L) + \text{Cost}(v_R, m_R, c_R) + f(m_L, v_L, c_L) + f(m_R, v_R, c_R) - g_c(m_L, m_R, c_L, c_R)$. By definition, the cost $\text{Cost}(v, m, c)$ is equal to the minimum over all methods of moving maximal colored subtrees in $T'(v)$ while leaving $m$ maximal colored subtrees regrafted along the parent edge of $v$ and unpaired in an extraneous subtree. Then, taking the minimum over all possible $c_L, c_R, m_L$ and $m_R$ values provides the optimal cost value.

Proof of Theorem 4.5. We note that a pair of canonical extraneous free completions can be computed in $O(n)$ time. To compute the optimal cost values at each vertex of an EF-R-RF(+) completion, Algorithm $\text{Compute-R-RF+}(S, T)$ has a total of three nested loops, over (1) the postorder traversal, (2) the values of $c$ and $m$, and (3) the values of $c_L, c_R, m_L$ and $m_R$ when the recurrence relation is invoked. The total time complexity is then the product of the sizes of each nested loop. Note there are a constant number of colors.

1. The postorder traversal has $O(n)$ nodes to parse.
2. Notice $m$ must be bounded above by $\max\{c\text{Max}(0, v), c\text{Max}(1, v)\} \leq c\text{Max}(0, \text{rt}(T')) + c\text{Max}(1, \text{rt}(T')) = k$ for any vertex $v$. Hence, we have another multiplicative $O(k)$ factor.
3. For each $\text{Cost}(v, m, c)$ value, we observe that the number of possible values of $m_L$ and $m_R$ considered is again bounded above by $k$, adding another multiplicative $O(k)$ factor.

Thus, the total runtime to compute all cost values is $O(nk^2)$. Once all cost values are computed, the RF(+) distance can be computed in $O(1)$ time.