

## RESEARCH

# A Generalized Robinson-Foulds Distance for Labeled Trees

Samuel Briand<sup>1</sup>, Christophe Dessimoz<sup>2,4,5,6,7</sup>, Nadia El-Mabrouk<sup>1\*</sup>, Manuel Lafond<sup>3</sup> and Gabriela Lobinska<sup>4</sup>

\*Correspondence:

[mabrouk@iro.umontreal.ca](mailto:mabrouk@iro.umontreal.ca)

<sup>1</sup>Computer Science Department,  
Université de Montréal, Canada  
Full list of author information is  
available at the end of the article

## Abstract

The Robinson-Foulds (RF) distance is a well-established measure between phylogenetic trees. Despite a lack of biological justification, it has the advantages of being a proper metric and being computable in linear time. For phylogenetic applications involving genes, however, a crucial aspect of the trees ignored by the RF metric is the type of the branching event (e.g. speciation, duplication, transfer, etc). Here, we extend RF to trees with labeled internal nodes by including a node *flip* operation, alongside edge contractions and extensions. We explore properties of this extended RF distance in the case of a binary labeling. In particular, we show that contrary to the unlabeled case, an optimal edit path may require contracting “good” edges, i.e. edges shared between the two trees. We provide a 2-approximation algorithm which is shown to perform well empirically. Looking ahead, computing distances between labeled trees opens up a variety of new algorithmic directions.

Implementation and simulations available at

<https://github.com/DessimozLab/pylabeledrf>.

**Keywords:** edit distance; labeled trees; Robinson-Foulds; tree metric

## 1 Introduction

Phylogenetic trees represent the evolutionary relationship between sets of genetic elements or taxa, where the elements of a set are in one-to-one relationship with the leaves of the corresponding tree [1]. Different phylogenetic inference methods may lead to different trees, and each method, typically exploring a large space of trees, can also result in multiple equally likely solutions for the same dataset. It follows that comparing trees is an essential task for finding out how inferred trees are far from one another, or how an inferred tree is far from a simulated tree or from a gold standard tree for the same datasets.

Designing appropriate tree metrics is a widely explored branch of research. A variety of measures have been designed for different types of trees, rooted or unrooted, some restricted to comparing tree shapes [2], others considering multilabeled trees, i.e. trees with repeated leaf labels [3] and yet others considering information on edge length [4]. In particular, a large number of pairwise measures of similarity or dissimilarity have been developed for comparing two topologies on the same leafset. Among them are the methods based on counting the structural differences between the two trees in terms of path length, bipartitions or quartets for unrooted trees, clades or triplets for rooted trees [5–7], or those based on minimizing a number of rearrangements that disconnect and reconnect subpieces of a tree, such as nearest

neighbour interchange (NNI), subtree-pruning-regrafting (SPR) or Tree-Bisection-Reconnection (TBR) moves [8–10]. While the latter methods are NP-hard [11], the former are typically computable in polynomial time. In particular, the Robinson-Foulds (RF) distance, defined in terms of bipartition dissimilarity for unrooted trees, and clade dissimilarity for rooted trees [12], can be computed in linear [13], and even sublinear time [14].

Despite several drawbacks such as lack of robustness (a small change in a tree may cause a disproportional change in the distance), skewed distribution [15–17], and a lack of biological rationale, RF remains the most widely used measure, not only in phylogenetics, but also in other fields such as in linguistics. To increase robustness, improved versions of the RF distance have also been developed [11, 18].

In addition of being efficiently computable, RF has the merit of being a true metric. It was originally defined on unrooted trees, in terms of edit operations on the tree edges: the minimum number of edge contraction and extension needed to transform one tree into the other [19]. Interestingly, the same metric, expressed in terms of node deletion and insertion, has been widely used in the context of data featuring hierarchical dependencies, modeled as trees with labeled nodes. In this case, the standard Tree Edit Distance (TED) is defined in terms of a minimum cost path of node deletion, node insertion and node relabeling (label substitution) transforming one tree to the other, for two trees sharing the same set of node labels (i.e. each label is present exactly once in each tree). While the less constrained version of the problem on unordered labeled trees is NP-complete [20], most variants are solvable in polynomial time [21–23].

Even though this kind of hierarchical node labeling has limited applicability for phylogenetic trees, other types of labeling can be used in the context of genetic data comparison. In the case of gene trees, it is important to identify the evolutionary event (duplication, speciation, transfer, etc) that has led to a given bifurcation. For example, information on duplication and speciation node labeling is provided for the trees of the Ensembl Compara database [24] (reconciled with *TreeBest* [25]). Therefore, being able to compare labeled phylogenies is important in the context of gene tree reconstruction and analysis.

This paper is the first effort towards extending the RF distance to labeled trees involving, in addition to edge contraction and extension (operations that can alternatively be defined as node insertion and deletion), a node substitution or “relabeling” operation. Importantly, our extended RF remains a metric in the mathematical sense.

While the formulation of the RF distance in terms of edit operations is known, the bipartition and clade formulations are often those that are used in the literature. Though similar, the three formulations present some differences depending on whether the trees are rooted or unrooted. We begin by making these differences explicit. We then explore, in Section 3, some properties of the extended RF distance in the case of two labels (e.g. speciation and duplication). In particular, we show that, in contrast to the RF distance for unlabeled trees, an optimal edit path for labeled trees may involve contracting good edges, i.e. edges representing common bipartitions of the two compared trees, which makes the extended RF much harder to compute than the basic RF. In Section 4, we explore various avenues for computing the extended RF. We give an exact algorithm for contracting “mixed subtrees”,

i.e. subtrees with alternating labels, and a bounded heuristic for general trees that achieves a factor 2 approximation. In Section 5, the heuristic is shown, on simulated datasets, to be efficient, by plotting the number of tree edits against the computed RF distance. Finally, in Section 6, we explore some avenues for improvement. All proofs are given in the Appendix.

## 2 Notations and concepts

Let  $T$  be a tree with a node set  $V(T)$  and an edge set  $E(T)$ . Given a node  $x$  of  $T$ , the *degree of  $x$*  is the number of edges incident to  $x$ . We denote by  $L(T) \subseteq V(T)$  the set of *leaves of  $T$* , i.e. the set of nodes of  $T$  of degree one. A node of  $V(T) \setminus L(T)$  is called an *internal node*. A tree with a single internal node is called a *star tree*. An edge connecting two internal nodes is called an *internal edge*; otherwise, it is a *terminal edge*. Moreover, a *rooted tree* admits a single internal node  $r(T)$  considered as the root.

Let  $x$  and  $y$  be two nodes of a rooted tree  $T$ ;  $y$  is an *ancestor* of  $x$  if  $y$  is on the path from  $x$  to the root (possibly  $y$  itself);  $y$  is a *descendant* of  $x$  if  $y$  is on the path from  $x$  to a leaf (possibly  $y$  itself) of  $T$ . For a rooted tree, we may write  $(x, y)$  for an edge between  $x$  and  $y$  where  $x$  is closer to the root. We say that  $y$  is a *child* of  $x$ . If  $T$  is unrooted, we call the set  $\{y : \{x, y\} \in E(T)\}$  the set of children of  $x$  (this is an unusual definition, but defining a notion of children for both rooted and unrooted trees will be useful later). For a rooted or an unrooted tree  $T$ , we denote by  $Ch(x)$  the set of children of an internal node  $x$  of  $T$ .

A tree  $T$  representing the evolution of a set  $\mathcal{L}$  of entities (usually taxa or genes) is a tree with a one-to-one mapping between  $L(T)$  and  $\mathcal{L}$ . We simply write  $\mathcal{L} = L(T)$  and say that  $T$  is a *tree for  $\mathcal{L}$* . An internal node represents an ancestral event (classically a speciation or a duplication) leading from one to many different entities. Moreover rooting a tree amounts to determining the common ancestor of all entities, i.e. determining the direction of evolution. Accordingly, internal nodes of an evolutionary tree (which are the trees considered in this paper) should be of degree at least 3, except the root which is of degree at least 2. An internal node  $x \neq r(T)$  of a tree  $T$  is *binary* if and only if  $x$  is of degree 3 and  $r(T)$  is *binary* if and only if  $r(T)$  is of degree 2. A tree  $T$  is said *binary* if and only if all its internal nodes are binary.

A *subtree  $S$*  of  $T$  is a tree such that  $V(S) \subseteq V(T)$ ,  $E(S) \subseteq E(T)$  and any edge of  $E(S)$  connects two nodes of  $V(S)$ . A *chain* of  $T$  is a subtree  $C$  with a node set  $V(C) = \{x_1, \dots, x_k\}$  and an edge set  $E(C) = \{e_1, \dots, e_{k-1}\}$  such that for each  $1 \leq i \leq k$ ,  $e_i$  is incident to  $x_i$  and  $x_{i+1}$ .

If  $T$  is an unrooted tree, *rooting  $T$*  requires choosing an internal node as the root, or creating a new node  $r(T)$  on an edge  $e = \{x, y\}$  of  $T$ , namely removing  $e$  and adding two edges  $\{r(T), x\}$  and  $\{r(T), y\}$ . If  $T$  is a rooted tree then the *unrooted version* of  $T$  is simply  $T$  (ignoring the description of  $r(T)$  as the root) if  $r(T)$  is non-binary; otherwise it is the tree obtained from  $T$  by removing  $r(T)$  and its two incident edges going to its neighbors  $u$  and  $v$ , and adding an edge between  $u$  and  $v$ .

For a rooted tree  $T$ , we denote by  $T_x$  the subtree of  $T$  rooted at  $x \in V(T)$ , i.e. the subtree of  $T$  containing all the descendants of  $x$ . We call  $L(T_x)$  the *clade of  $x$* . A clade is *non-trivial* if it corresponds to an internal node of  $T$ . We denote by  $\mathcal{C}(T)$  the set of non-trivial clades of  $T$ . It can be seen as a subset of the power set of  $\mathcal{L}$ .

The *bipartition* of an unrooted tree  $T$  corresponding to an internal edge  $e = \{x, y\}$  is the unordered pair of clades  $L(T_x)$  and  $L(T_y)$  where  $T_x$  and  $T_y$  are the two subtrees rooted respectively at  $x$  and  $y$  obtained by removing  $e$  from  $T$ . A bipartition is *non-trivial* if it corresponds to an internal edge of  $T$ , and trivial otherwise. We denote by  $\mathcal{B}(T)$  the set of non-trivial bipartitions of  $T$ . Note that bipartitions are sometimes called *splits* in the literature.

### 2.1 The Robinson-Foulds distance

**Definition 1** (edit operations) *Two edit operations on the edges of a tree  $T$  (rooted or unrooted) are defined as follows:*

- *Let  $e = \{x, y\}$  be an internal edge of  $E(T)$ . An edge contraction  $Cont(T, e)$  is an operation transforming the tree  $T$  into the tree  $T'$  obtained from  $T$  by removing the edge  $e$  of  $T$  and identifying  $x$  and  $y$ ; in other words,  $T'$  is obtained by adding the edge  $\{x, z\}$  for each  $z \in Ch(y) \setminus \{x\}$ , and then removing  $y$  and its incident edges (including  $\{x, y\}$ ).*
- *Let  $x$  be a non-binary internal node of  $V(T)$  and  $X = \{y_1, \dots, y_t\} \subsetneq Ch(x)$  be a subset of  $Ch(x)$  such that  $|X| \geq 2$ . A node extension  $Ext(T, x, X)$  is an operation transforming the tree  $T$  into the tree  $T'$  obtained from  $T$  by removing the edges  $\{x, y_i\}$ , for  $1 \leq i \leq t$ , creating a node  $y$  and a new edge  $e = \{x, y\}$  adjacent to  $x$ , and creating new edges  $\{y, y_i\}$ , for  $1 \leq i \leq t$ .*

The function  $\delta(T_1, T_2)$  assigning to each pair of rooted or each pair of unrooted trees the length of a minimum sequence of edit operations transforming  $T_1$  into  $T_2$  has been shown to be a metric, called the *Edit distance* or *Robinson-Foulds distance* between  $T_1$  and  $T_2$  [19].

For unrooted trees  $T_1$  and  $T_2$ , this distance corresponds to the symmetric difference between the bipartitions of the two trees. More precisely,  $\delta(T_1, T_2) = |\mathcal{B}(T_1) \setminus \mathcal{B}(T_2)| + |\mathcal{B}(T_2) \setminus \mathcal{B}(T_1)|$ . In fact, to transform  $T_1$  into  $T_2$ , edit operations are needed on *bad edges* representing bipartitions which are not shared by the two trees, i.e. edges of  $T_1$  (respec.  $T_2$ ) defining bipartitions in  $T_1$  (respec.  $T_2$ ) which are not in  $\mathcal{B}(T_2)$  (respec. in  $\mathcal{B}(T_1)$ ). An edge which is not bad is said to be *good* (see Figure 2). Terminal edges are always good.

In the case of rooted trees  $T_1$  and  $T_2$ , the Robinson-Foulds distance, that we denote in this case  $\delta_R(T_1, T_2)$ , is usually defined in the literature as the symmetric difference between the clades of the two trees. More precisely, for two rooted trees  $T_1$  and  $T_2$ ,  $\delta_R(T_1, T_2) = |\mathcal{C}(T_1) \setminus \mathcal{C}(T_2)| + |\mathcal{C}(T_2) \setminus \mathcal{C}(T_1)|$ .

The only thing that can make bipartitions and clades differ in number is rooting into a bad edge. In this case, the same bipartition, corresponding to the two edges adjacent to the root, would be counted twice. The link between this distance, defined in terms of clades (that we write  $\delta_R$ ) and the edit distance (that we write  $\delta$ ), has been established through the defined relation between the bipartition system (or split system) and the clade system (or cluster system) [26].

Although our extended distance is more likely useful for rooted trees, algorithmic analyses are simpler for unrooted trees, as in this case all internal nodes can be treated in the same way. Here, we make the link between the rooted and unrooted case, and then focus, for the rest of the paper, on unrooted trees.

Let  $T^r$  be a rooted version of an unrooted tree  $T$ , with a binary root. Denote by  $e_1, e_2$  the two edges adjacent to  $r(T^r)$ . As  $e_1$  and  $e_2$  define the same bipartition of  $\mathcal{B}(T)$ , these edges are either both good or both bad. These notations are used in the following lemma.

**Lemma 1** (Link between rooted and unrooted trees) *Let  $T_1$  and  $T_2$  be two unrooted trees, and  $T'_1$ , respectively  $T'_2$ , be a rooting of  $T_1$ , respectively  $T_2$ .*

- *If  $T'_1$  and  $T'_2$  are both rooted into existing nodes of  $T_1$  and  $T_2$  or both rooted into good edges of  $T_1$  and  $T_2$ , then  $\delta_R(T'_1, T'_2) = \delta(T_1, T_2)$ ;*
- *If  $T'_1$  and  $T'_2$  are both rooted into bad edges of  $T_1$  and  $T_2$ , then  $\delta_R(T'_1, T'_2) = \delta(T_1, T_2) + 2$ ;*
- *If exactly one among  $T'_1$  and  $T'_2$  is rooted into a bad edge of  $T_1$  or  $T_2$ , then  $\delta_R(T'_1, T'_2) = \delta(T_1, T_2) + 1$ .*

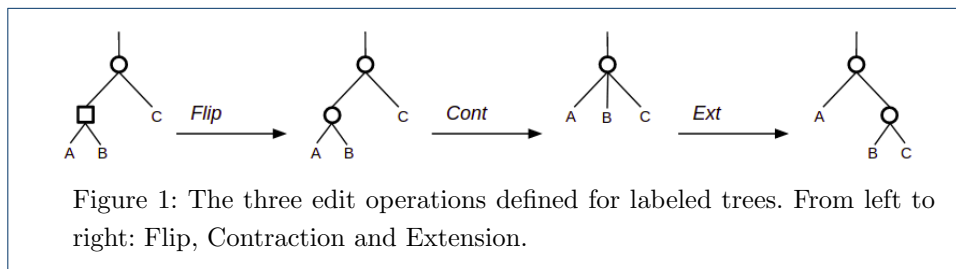
The edit distance between two trees (rooted or unrooted) can be computed in linear time with the algorithm proposed by Day [13] in 1984. Our goal is to extend this distance to labeled trees.

### 2.2 Labeled trees

Given a finite set of labels  $\Lambda$ ,  $T$  is labeled if and only if each internal node  $x$  of  $T$  has a unique label  $\lambda(x) \in \Lambda$ .

Contraction and extension operations are generalized to labeled trees as follows: The node  $y$  created from an edge extension  $Ext(T, x, X)$  is such that  $\lambda(y) = \lambda(x)$ ; an edge contraction is only defined on edges  $\{x, y\}$  for which  $\lambda(x) = \lambda(y)$ . It follows that a third edit operation should be introduced for labeled trees. Let  $x$  be a node of a labeled tree  $T$  with label  $\lambda = \lambda(x)$ . A *node flip*  $Flip(x, \lambda')$  is an operation assigning a new label  $\lambda'$  to  $x$ , i.e. a label  $\lambda' \in \Lambda$  such that  $\lambda' \neq \lambda$ . Those operations are depicted in Figure 1.

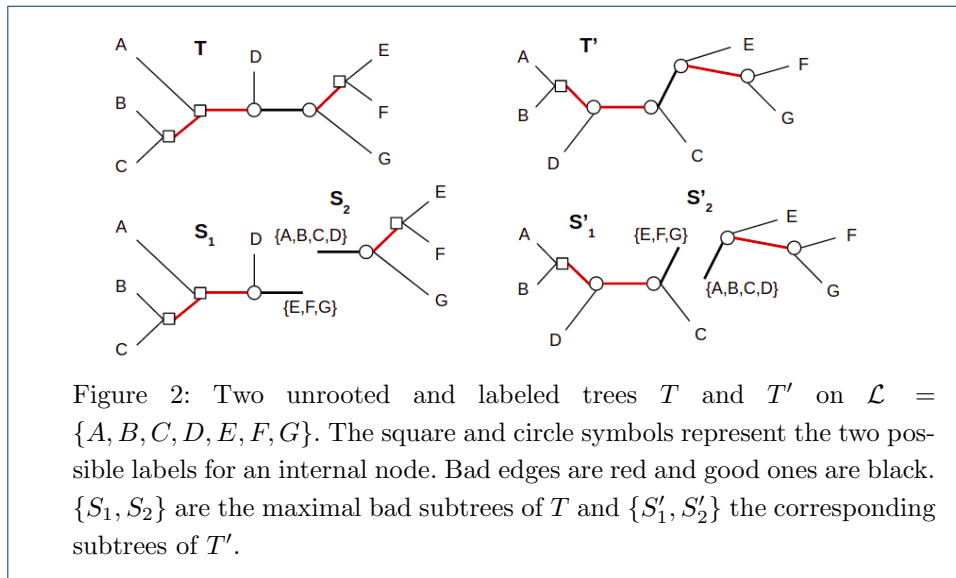
A node flip is required before contracting a *mixed edge*, i.e. an edge with its two extremities being differently labeled. A tree is said to be a *mixed tree* if all its edges are mixed edges.



Let  $\mathcal{T}$  be the set of trees on  $\mathcal{L}$ , all trees being of the same type, i.e. all rooted or unrooted, all labeled or unlabeled. The following lemma (holding for all these cases) shows that introducing the flip operation does not prevent  $\delta$  from being a distance.

**Lemma 2** (Edit distance) *The function  $\delta(T_1, T_2)$  assigning to each pair  $(T_1, T_2) \in \mathcal{T}^2$  the minimum length of a sequence of edit operations transforming  $T_1$  into  $T_2$  defines a distance on  $\mathcal{T}$ .*

In this paper,  $\Lambda$  is restricted to two labels. They are illustrated by a circle and a square in Figure 2. The two labels can, for example, represent speciation and duplication events. Notice however that labeling is not constrained to be consistent with a species tree [27, 28]. In other words, the intermediate trees in an optimal path transforming a tree to another are not required to be feasible according to the speciation/duplication labeling. Algorithmic analyses are made independently of the nature of the two node labels. However, for notation purpose, we write  $\Lambda = \{Spe, Dup\}$ .



### 3 Results on labeled trees

We focus now on unrooted trees. Using Lemma 1, our results can then be easily extrapolated to rooted trees. Consider  $\mathcal{T}$  as the set of unrooted and labeled trees on  $\mathcal{L}$ . The goal is to compute the edit distance  $\delta(T, T')$  for any pair  $T, T'$  of trees of  $\mathcal{T}$ , that is the number of operations in an *optimal sequence*, i.e a sequence of edit operations of minimum length transforming  $T$  into  $T'$ .

#### 3.1 Reduction to maximal bad subtrees

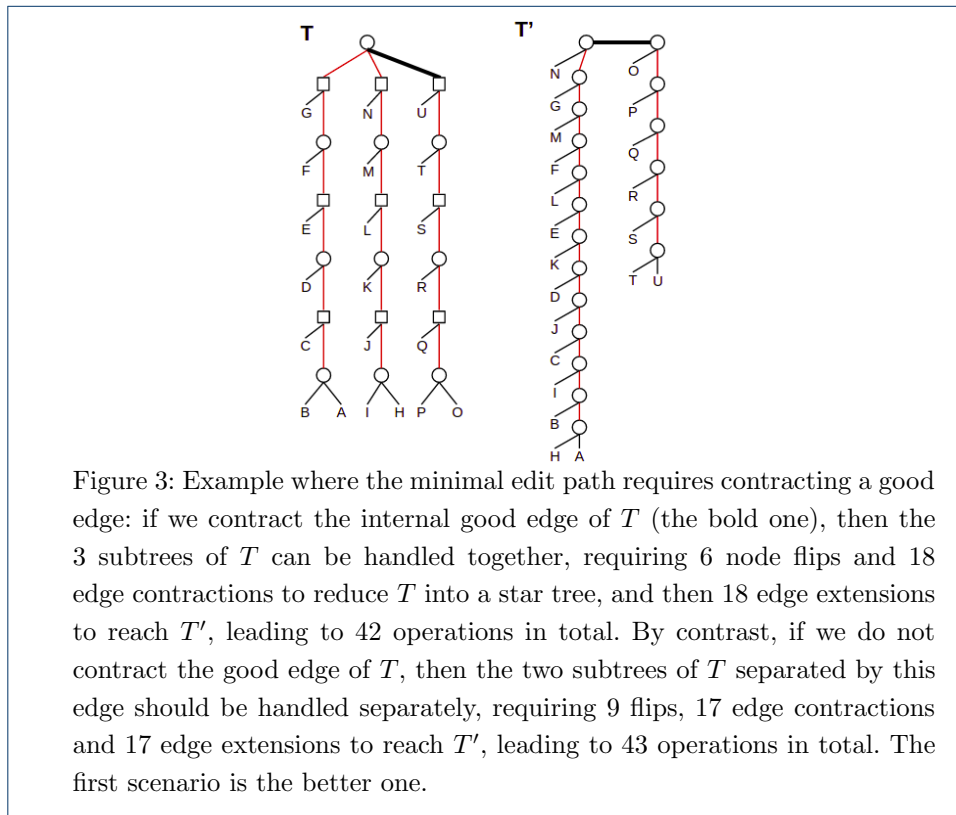
Let  $S$  be a subtree of  $T$ . Let  $\{e_i = \{x_i, y_i\}, \text{ for } 1 \leq i \leq k\}$  be the set of terminal edges of  $S$ , with each  $y_i$  being a leaf of  $S$ , and  $\{X_i, Y_i\}$  being the bipartition corresponding to  $e_i$ . Each leaf  $y_i$  of  $S$  is said to be *mapped* to  $Y_i$ . Notice that  $\cup_{1 \leq i \leq k} Y_i = \mathcal{L}$ .

We say that  $S$  is a *bad subtree of  $T$*  if and only if  $S$  contains only bad edges, except the terminal edges of  $S$  which are all good edges of  $T$ . In other words,  $S$  is maximal in the sense that no more bad internal edges can be added into it. Intuitively,  $S$  can be obtained by taking a subtree with only bad edges, and adding edges adjacent to bad edges of  $S$  iteratively until the process stops. As a result, every terminal edge  $e_i$  of  $S$  will be good, i.e. there is an edge  $e'_i = \{x'_i, y'_i\}$  in  $T'$  corresponding to  $e_i = \{x_i, y_i\}$ , that determine the same bipartition  $\{X_i, Y_i\}$ . Note that a maximal bad subtree may contain no bad edge at all (i.e. it is a star tree centered on good edges).

**Lemma 3** (Pairs of maximal bad subtrees) *Let  $S$  be a maximal bad subtree of  $T$  with the set  $\{e_i\}_{1 \leq i \leq k}$  of terminal edges, and let  $\{e'_i\}_{1 \leq i \leq k}$  be the corresponding set of edges in  $T'$ . Then the subtree  $S'$  of  $T'$ , containing all  $e'_i$  edges as terminal edges, is unique. Moreover, it is a maximal bad subtree of  $T'$ .*

Let  $\{S_1, S_2, \dots, S_k\}$  be the set of maximal bad subtrees of  $T$  and  $\{S'_1, S'_2, \dots, S'_k\}$  be the corresponding subtrees of  $T'$  (see Figure 2 for an example). For  $1 \leq i \leq m$ , let  $\mathcal{P}_i$  be an optimal sequence transforming  $S_i$  into  $S'_i$ . Then the sequence  $\mathcal{P}$  obtained by performing consecutively  $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_m$  transforms  $T$  into  $T'$ .

Although the traditional RF distance can be deduced from the above observation, in our case such a sequence is not necessarily optimal. In fact, in contrast with unlabeled trees, optimal sequences for labeled trees may involve contracting good edges, as illustrated in Figure 3.



### 3.2 Reduction to mixed bad subtrees

In the next section, we will describe an exact algorithm for optimally contracting a mixed tree. Before reaching this step, the question is how to obtain such a tree. The next lemma shows that non-mixed bad edges can be contracted first. The idea of the proof is that any optimal solution must eventually contract a non-mixed bad edge  $\{x, y\}$ . We can thus contract  $\{x, y\}$  first into a single node  $z$ , and “reproduce” all the events of the optimal solution by treating  $z$  as either  $x$  or  $y$ .

**Lemma 4** (Contract non-mixed bad edges) *Let  $e$  be any non-mixed bad edge of  $T$ , and let  $T_c$  be the tree obtained from  $T$  by contracting  $e$ . Then  $\delta(T_c, T') = \delta(T, T') - 1$ .*

According to this lemma, we can safely start by contracting all non-mixed bad edges of  $T$  and  $T'$  first, since there is always an optimal sequence of edit operations that also does this. The resulting trees  $T_c$  and  $T'_c$  can then be subdivided into pairs of maximal bad subtrees, all such bad subtrees being mixed subtrees.

### 4 Algorithms

We first consider a general framework which entails performing all required edge contractions first, and then all node extensions.

---

**Methodology 1** ( $T, T'$ )

---

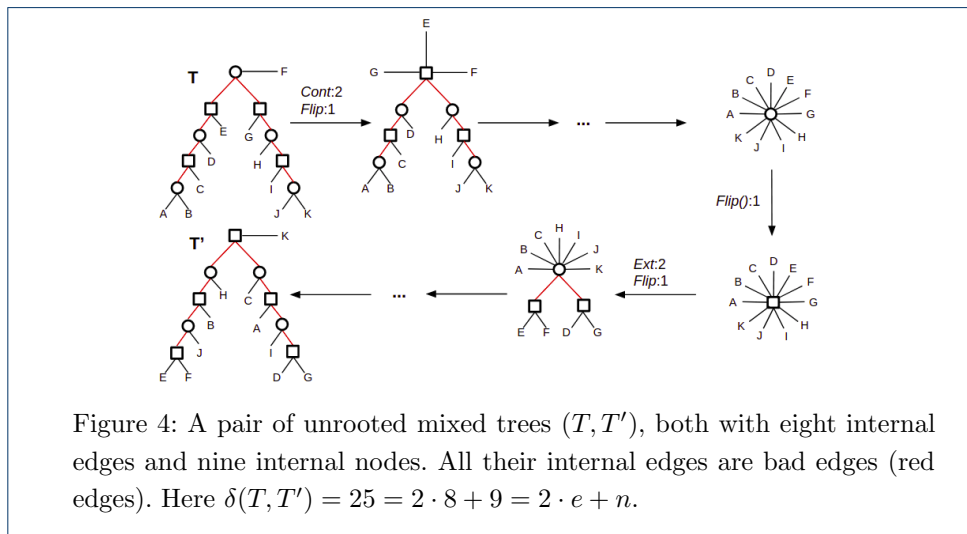
Contract non-mixed bad edges of  $T$  and  $T'$ , leading to  $T_c$  and  $T'_c$ ;  
**for** each pair  $S, S'$  of maximal bad subtrees of  $T_c, T'_c$  **do**  
    Perform a sequence of flip and contraction operations leading from  $S$  to a star tree  $S_*$ ;  
    Perform a sequence of flip and extension operations leading from  $S_*$  to  $S'$ ;  
**end for**

---

This general framework leads to the following upper bound for  $\delta(T, T')$ .

**Lemma 5** (Upper bound  $\delta$ ) *Let  $T$  and  $T'$  be two unrooted and labeled trees with  $n$  internal nodes each and let  $e$  (resp.  $e'$ ) be the number of internal bad edges of  $T$  (resp.  $T'$ ). Then  $\delta(T, T') \leq e + e' + n$ .*

Notice that if both  $T$  and  $T'$  are binary, then  $e = e'$ . Moreover, in this this case  $2e + n$  is a tight bound as it can be reached in some cases (see an example in Figure 4).



The first step of Methodology 1 leads to a star tree  $T_*$ . Instead of then extending nodes to reach  $T'$ , a symmetric way would be to transform  $T'$  into a star tree  $T'_*$ . The difference between  $T_*$  and  $T'_*$  may be in the label of the single node of each of these trees, which would then need an additional flip operation to reconstruct a corresponding path from  $T$  to  $T'$ . This second methodology is given below, where *Contract-Tree*( $T, T_*$ ) takes as input a tree  $T$  and returns a sequence of operations *contracting a tree*  $T$ , i.e. transforming  $T$  into a star tree, and the star tree  $T_*$  resulting from this optimal contraction.



Methodology 2 is clearly simpler to handle and will be explored in the next section. The next lemma shows that it may overestimate an optimal sequence returned by Methodology 1 by at most one operation for each pair of maximal bad subtrees.

---

**Methodology 2** ( $T, T'$ )

---

Contract non-mixed bad edges of  $T$  and  $T'$ , leading to  $T_c$  and  $T'_c$ ;  
**for** each pair  $S, S'$  of maximal bad subtrees of  $T_c, T'_c$  **do**  
     $Contract-Tree(S, S_*)$ ;  
     $Contract-Tree(S', S'_*)$ ;  
    Perform a final flip if required;  
**end for**

---

**Lemma 6** (Compare Meth.1 and Meth.2) *Let  $S$  and  $S'$  be a pair of maximal bad subtrees of  $T_c$  and  $T'_c$ , obtained similarly by Methodology 1 and Methodology 2. Let  $M_1(S, S')$  (respec.  $M_2(S, S')$ ) be the number of operations performed by the **for** loop of Methodology 1 (respec. Methodology 2). Moreover, let  $S_*$  (respec.  $S'_*$ ) be the star tree returned by  $Contract-Tree$  on  $S$  (respec. on  $S'$ ).*

- 1 If  $S_* = S'_*$  (same node label), then  $M_2(S, S') = M_1(S, S')$ ;
- 2 Otherwise,  $M_1(S, S') \leq M_2(S, S') \leq M_1(S, S') + 1$

#### 4.1 An optimal algorithm for contracting a tree

The remaining problem is the one of finding an optimal sequence of contraction and flip operations contracting a mixed tree  $T$ . For any such sequence, the number of contraction operations is just the number of internal edges of  $T$ . Therefore, the problem reduces to finding the minimum number of flip operations  $\phi(T)$  in such an optimal sequence. Notice that the problem does not reduce to performing the minimum number of flips leading to the same label for all nodes, which would just be  $\min\{nb_{spe}, nb_{dup}\}$  with  $nb_{spe}$  (respec.  $nb_{dup}$ ) being the number of *Spe* (respec. *Dup*) nodes of  $T$ . For example, for the tree  $T$  of Figure 3,  $\min\{nb_{spe}, nb_{dup}\} = 9$ . However, proceeding by an alternating sequence of flip and contraction operations (the top node flipped to *Dup*, then the three top edges contracted, then the next top node flipped to a *Spe* node, then the three top edges contracted, etc.) leads to a total of 6 flips rather than 9.

We will proceed iteratively by starting a sequence of contraction operations from the center of a tree  $T$ , i.e. the midpoint of the longest mixed chain of  $T$ . The *diameter*, denoted  $diam(T)$ , of a tree  $T$  is the length of its longest chain (determined in terms of the number of edges). Note that any longest chain in a tree has two leaves at its extremities, as otherwise we could extend the chain. Assume that  $T$  has at least two terminal edges, so that  $diam(T) \geq 2$ . We show that  $\phi(T)$  is equal to  $\lceil diam(T)/2 \rceil - 1$ . For a node  $v$ , let  $ecc_T(v)$  denote the maximum distance from  $v$  to a leaf of  $T$  (this is known as the *eccentricity* of  $v$ ) [1].

**Lemma 7** (Optimal path contracting a mixed tree) *The minimum number of flips in an optimal sequence of operations transforming a mixed tree  $T$  into a star tree is  $\lceil diam(T)/2 \rceil - 1$ .*

---

*Algorithm Contract-Tree*( $T$ ) (where  $T$  is a mixed tree)

---

Let  $P = (w_1, w_2, \dots, w_k)$  be a longest chain of  $T$ ;  
 Let  $w = w_{\lceil k/2 \rceil}$  be a midpoint of  $P$ ; ( $w$  has minimum eccentricity)  
**while**  $w$  has a non-leaf neighbor **do**  
     Flip  $w$ ;  
     Contract the internal edges incident to  $w$ ;  
**end while**

---

Lemma 7 immediately lead to Algorithm *Contract-Tree*. The fact that the algorithm contracts  $T$  into a star tree using  $\phi(T)$  flips follows from the proof of Lemma 7.

**Theorem 1** For  $T$  being a mixed tree, Algorithm *Contract-Tree* returns the length of an optimal sequence of operations contracting  $T$ .

One should note that if  $T$  has even diameter, then there are two possible midpoints, i.e. two nodes with minimum eccentricity. This means that it is possible to choose the label of the internal node of the resulting star tree. This guarantees that when contracting a pair of bad subtrees  $T$  and  $T'$ , we can always avoid a final flip by choosing the appropriate final label if either  $T$  or  $T'$  has even diameter. We cannot guarantee that this final flip is avoidable if both subtrees have odd diameter.

We now show that Methodology 2 has a guaranteed approximation ratio of 2 when using Algorithm *Contract-Tree* as a subroutine. The idea behind the approximation is to show that any optimal solution must contract all the bad edges and perform at least one flip or good edge contraction per bad subtree. Our algorithm only contracts bad edges, and we can show that the number of flips performed is at most the number of bad edges plus twice the number of bad subtrees.

**Theorem 2** (Upper bound Meth.2) Let  $d$  be the number of operations performed by Methodology 2 when tree contractions are done by Algorithm *Contract-Tree*. Then  $d \leq 2\delta(T, T')$ .

## 5 Experimental results

We implemented a heuristic following Methodology 2, using the *Contract-Tree* algorithm. To test it on simulated data, we retrieved the TP53 gene family from Ensembl release 96 (542 genes), including the speciation and duplication labels, and introduced an increasing number of random edit operations, on 30 replicates. A random edit was introduced as follows: with probability 0.3, the label of one random internal node was flipped; the rest of the probability mass function was evenly distributed among all potential internal edges (which could be potentially contracted) and all nodes of degree  $> 3$  (in which a new edge could be expanded).

After each edit, we computed the classical RF distance and its extension to labeled trees using our heuristic (Fig. 5). Because it accounts for labels, the latter tracked more closely the true number of edits. At the same time, the estimated distances were never higher than the actual number of edits, which suggests that the heuristic

---

<sup>[1]</sup>The radius of  $T$  is a well-known graph parameter and is defined as the minimum eccentricity of a node of  $T$ . In a tree, the radius turns out to be  $\lceil \text{diam}(T)/2 \rceil$ .

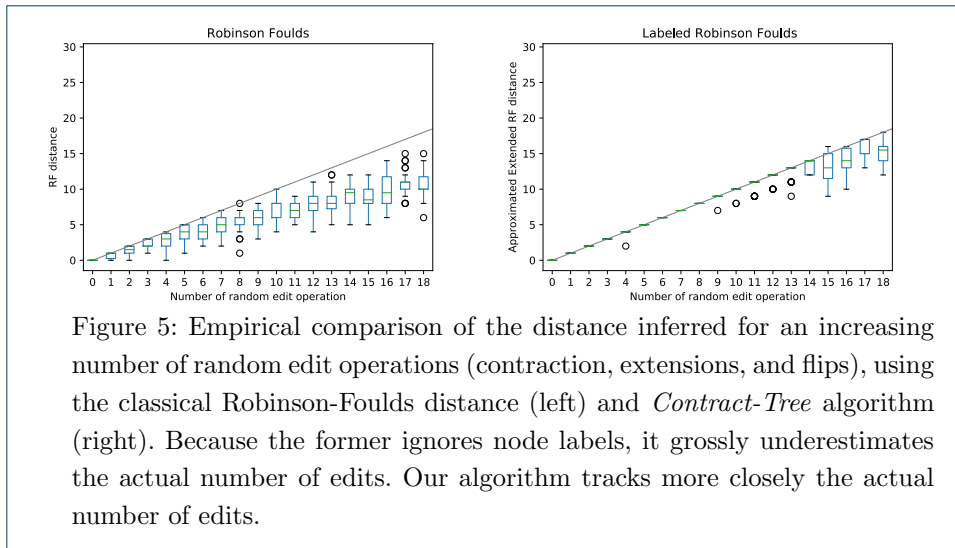


Figure 5: Empirical comparison of the distance inferred for an increasing number of random edit operations (contraction, extensions, and flips), using the classical Robinson-Foulds distance (left) and *Contract-Tree* algorithm (right). Because the former ignores node labels, it grossly underestimates the actual number of edits. Our algorithm tracks more closely the actual number of edits.

can identify a minimum edit path when the total number of edit operations is relatively low. The implementation, including the function to mutate labeled trees, is available as an open source Python library (PyPI package `pylabeledrf`, also available at <https://github.com/DessimozLab/pylabeledrf>).

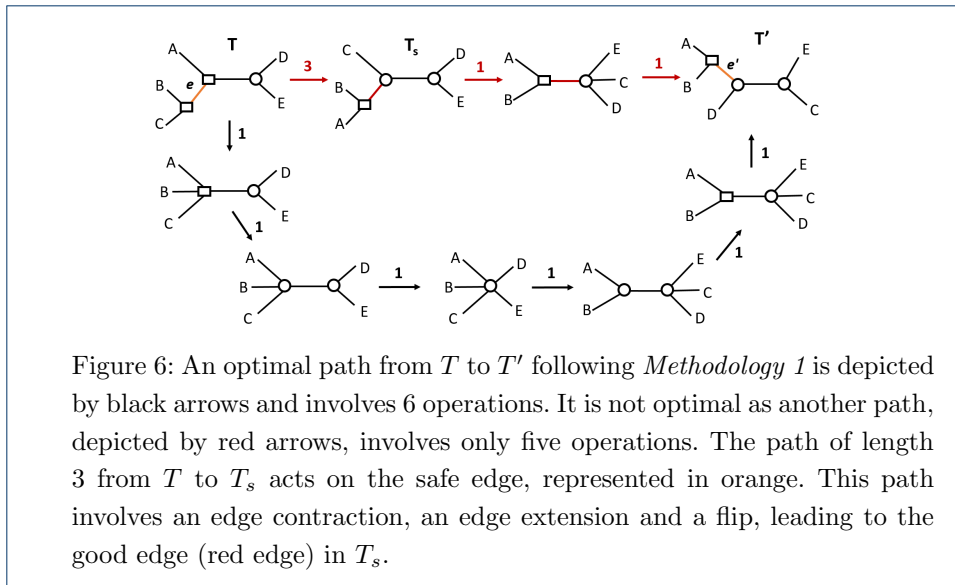
## 6 Discussion

In this paper, we have considered what we thought was the simplest and most natural extension of the Robinson-Foulds distance to labeled trees. Although its theoretical complexity is unknown and remains an open problem, this extension appears to be much harder to compute than the classical RF distance for unlabeled trees.

Despite the optimality of Algorithm *Contract-Tree* for contracting a mixed tree, neither *Methodology 1*, nor *Methodology 2* are guaranteed to lead to an optimal solution. This is due to two main reasons. The first one is that, as shown in Figure 3, an optimal path contracting a tree  $T$  may require contracting good edges, i.e. edges common to both trees, which is not the case for unlabeled trees. The second reason is that an optimal path from a tree  $T$  to a tree  $T'$  may not be one with all edge contraction events preceding all edge extension. An example, given in Figure 6, shows that it may be better to convert a given bad edge into a good edge rather than contracting all bad edges. It can be observed from this example that going from  $T$  to  $T'$  following the red path entails performing a nearest-neighbour interchange (NNI) operation on the edge  $e$  of  $T$ . A future direction for improving the algorithm will be to consider such “safe” edges, i.e. edges admitting an NNI leading to a bipartition of the target tree.

Still, we have implemented a heuristic which constitutes a better baseline solution to quantifying differences between labeled tree topologies than the conventional RF measure, which is blind to labels. For instance, this implementation could be useful in the context of orthology benchmarking, to compare inferred labeled trees with reference curated ones [29].

Looking ahead, we envision several potential future directions. We see potential in identifying the good edges that should be contracted and characterizing classes



of trees that may be resolved optimally. In particular, it would be interesting to restrict the study to the class of labeled trees consistent with a species tree (which is not the case of the trees of Figure 3).

Another direction would be to consider an alternative extension of the RF distance. In this paper, edge contraction and edge extension, the two edit operations defining the classical RF, were re-defined in the context of labeled nodes, by constraining them to occur on edges with the same labels on their extremities. Another direction would be to consider edit operations on nodes, as for the Tree Edit Distance (TED) for hierarchical trees, i.e. node deletion, insertion and relabeling. In addition to the theoretical complexity and computational efficiency, it would be important to evaluate the robustness of these two RF extensions with respect to small changes in the topology or tree labeling. Although we do not expect robustness to be much better than the classical RF, knowing which extension is better can orient the study towards future improvements. Finally another direction would be to extend the study to an arbitrary set of possible labels.

More generally, we think that computing the distance between labeled trees conceals many new problems and opens a variety of new algorithmic directions.

## 7 Declarations

Abbreviations

RF: Robinson-Foulds

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

### Availability of data and materials

The software implemented during the current study is available in the pylabeledrf repository, <https://github.com/DessimozLab/pylabeledrf>.

### Competing interests

The authors declare that they have no competing interests.

### Funding

Publication of this work is funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), Fonds de Recherche Nature et Technologies of Quebec (FRQNT), and Swiss National Science Foundation (SNSF).

### Authors' contributions

SB, CD, NEM, and ML devised the proofs and algorithms and wrote the paper. CD and GL implemented the distance software introduced in the manuscript. All authors read and approved the final manuscript.

### Acknowledgements

Not applicable.

### Author details

<sup>1</sup>Computer Science Department, Université de Montréal, Canada. <sup>2</sup>Department of Computational Biology, University of Lausanne, Switzerland. <sup>3</sup>Computer Science Department, Université de Sherbrooke, Canada. <sup>4</sup>Department of Genetics Evolution and Environment, University College London, United Kingdom. <sup>5</sup>Center for Integrative Genomics, University of Lausanne, Switzerland. <sup>6</sup>Swiss Institute of Bioinformatics, Switzerland. <sup>7</sup>Department of Computer Science, University College London, United Kingdom.

### References

1. Semple, C., Steel, M., *et al.*: Phylogenetics vol. 24. Oxford University Press on Demand, ??? (2003)
2. Colijn, C., Plazzotta, G.: A metric on phylogenetic tree shapes. *Syst. Biol.* **67**(1), 113–126 (2018)
3. Lafond, M., El-Mabrouk, N., Huber, K.T., Moulton, V.: The complexity of comparing multiply-labelled trees by extending phylogenetic-tree metric. *Theoretical Computer Science* **760**, 15–34 (2019)
4. Bryant, D., Scornavacca, C.: An  $O(n \log N)$  Time Algorithm for Computing the Path-length Distance Between Trees
5. Cardona, G., Lladrés, M., Rosselló, F., Valiente, G.: Nodal distances for rooted phylogenetic trees. *Journal of mathematical biology* **61**(2), 253–276 (2010)
6. Estabrook, G.F., McMorris, F., Meacham, C.A.: Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. *Systematic Zoology* **34**(2), 193–200 (1985)
7. Critchlow, D.E., Pearl, D.K., Qian, C.: The triples distance for rooted bifurcating phylogenetic trees. *Systematic Biology* **45**(3), 323–334 (1996)
8. Jiang, B.D.X.H.T., Li, M., Tromp, J., Zhang, L.: On computing the nearest neighbor interchange distance. In: *Discrete Mathematical Problems with Medical Applications: DIMACS Workshop Discrete Mathematical Problems with Medical Applications*, December 8–10, 1999, DIMACS Center, vol. 55, p. 125 (2000). American Mathematical Soc.
9. Hickey, G., Dehne, F., Rau-Chaplin, A., Blouin, C.: Spr distance computation for unrooted trees. *Evolutionary Bioinformatics* **4**, 419 (2008)
10. Allen, B.L., Steel, M.: Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of combinatorics* **5**(1), 1–15 (2001)
11. Lin, Y., Rajan, V., Moret, B.M.: A metric for phylogenetic trees based on matching. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* **9**(4), 1014–1022 (2012)
12. Mittal, S., Munjal, G.: Tree mining and tree validation metrics: A review. *IOSR: Journal of Computer Engineering*, 31–36 (2015)
13. Day, W.H.: Optimal algorithms for comparing trees with labeled leaves. *Journal of classification* **2**(1), 7–28 (1985)
14. Pattengale, N.D., Gottlieb, E.J., Moret, B.M.: Efficiently computing the robinson-foulds metric. *Journal of Computational Biology* **14**(6), 724–735 (2007)
15. Steel, M.A., Penny, D.: Distributions of tree comparison metrics—some new results. *Systematic biology* **42**(2), 126–141 (1993)
16. Bryant, D., Steel, M.: Computing the distribution of a tree metric. *IEEE/ACM transactions on computational biology and bioinformatics* **6**(3), 420–426 (2009)
17. Chaudhary, R., Burleigh, J.G., Fernandez-Baca, D.: Fast local search for unrooted robinson-foulds supertrees. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* **9**(4), 1004–1013 (2012)

18. Moon, J., Eulenstein, O.: Cluster matching distance for rooted phylogenetic trees. In: International Symposium on Bioinformatics Research and Applications, pp. 321–332 (2018). Springer
19. Robinson, D.F., Foulds, L.R.: Comparison of phylogenetic trees. *Mathematical biosciences* **53**(1-2), 131–147 (1981)
20. Zhang, K., Statman, R., Shasha, D.: On the editing distance between unordered labeled trees. *Information processing letters* **42**(3), 133–139 (1992)
21. Zhang, K.: A new editing based distance between unordered labeled trees. In: Annual Symposium on Combinatorial Pattern Matching, pp. 254–265 (1993). Springer
22. Zhang, K.: A constrained edit distance between unordered labeled trees. *Algorithmica* **15**(3), 205–222 (1996)
23. Schwarz, S., Pawlik, M., Augsten, N.: A new perspective on the tree edit distance. In: International Conference on Similarity Search and Applications, pp. 156–170 (2017). Springer
24. Vilella, A.J., Severin, J., Ureta-Vidal, A., Heng, L., Durbin, R., Birney, E.: EnsemblCompara gene trees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome Research* **19**, 327–335 (2009)
25. Schreiber, F., Patricio, M., Muffato, M., Pignatelli, M., Bateman, A.: Treefam v9: a new website, more species and orthology-on-the-fly. *Nucleic Acids Research* (2013). doi: 10.1093/nar/gkt1055
26. Dress, A.: Towards a theory of holistic clustering. *DIMACS Ser. Discrete Math. Theoret. Comput. Sci* **37**, 271–289 (1997)
27. Hernandez-Rosales, M., Hellmuth, M., Wieseke, N., Huber, K.T., Moulton, V., Stadler, P.F.: From event-labeled gene trees to species trees. In: *BMC Bioinformatics*, vol. 13, p. 6 (2012). BioMed Central
28. Lafond, M., El-Mabrouk, N.: Orthology and paralogy constraints: satisfiability and consistency. *BMC genomics* **15**(6), 12 (2014)
29. Altenhoff, A.M., Boeckmann, B., Capella-Gutierrez, S., Dalquen, D.A., DeLuca, T., Forslund, K., Huerta-Cepas, J., Linard, B., Pereira, C., Prysycz, L.P., Schreiber, F., da Silva, A.S., Szklarczyk, D., Train, C.-M., Bork, P., Lecompte, O., von Mering, C., Xenarios, I., Sjölander, K., Jensen, L.J., Martin, M.J., Muffato, M., Quest for Orthologs consortium, Gabaldón, T., Lewis, S.E., Thomas, P.D., Sonnhammer, E., Dessimoz, C.: Standardized benchmarking in the quest for orthologs. *Nature methods* **13**(5), 425–430 (2016). doi:10.1038/nmeth.3830

## 8 Appendix

### 8.1 Proof of Lemma 1 (Link between rooted and unrooted trees):

There is a one-to-one relationship between the set of non-trivial bipartitions and the set of internal edges of an unrooted tree  $T$ . Similarly, for a rooted tree  $T'$ , there is a one-to-one relationship between the set of internal edges of  $T'$  and its set of non-trivial clades, excluding the clade  $L(T')$ . However, the number of edges may differ between a tree  $T$  and a rooting  $T'$  of  $T$ .

- If  $T_1$  and  $T_2$  are both rooted into existing nodes, then  $T_1$  and  $T'_1$  (respec.  $T_2$  and  $T'_2$ ) have exactly the same edge sets, and we conclude from what precedes that there is a one-to-one relationship between the set of non-trivial bipartitions of  $T_1$  (respec.  $T_2$ ) and the set of non-trivial clades excluding  $L(T_1)$  (respec.  $L(T_2)$ ) of  $T'_1$  (respec.  $T'_2$ ). As  $L(T_1) = L(T_2)$ , this clade does not contribute to the symmetric difference computation of  $\delta_R(T'_1, T'_2)$ , and thus  $\delta_R(T'_1, T'_2) = \delta(T_1, T_2)$ .
- If both  $T_1$  and  $T_2$  are rooted into good edges, then  $T'_1$  (respec.  $T'_2$ ) has one edge more than  $T_1$  (respec.  $T_2$ ). But these new edges are good edges and therefore do not contribute to the symmetric difference computation of the  $\delta_R$  distance, and thus  $\delta_R(T'_1, T'_2) = \delta(T_1, T_2)$ .
- If both  $T_1$  and  $T_2$  are rooted into bad edges, then  $T'_1$  (respec.  $T'_2$ ) has one edge more than  $T_1$  (respec.  $T_2$ ). These two new edges are bad edges, and thus contribute to the symmetric difference computation of the  $\delta_R$  distance by adding two clades, and thus  $\delta_R(T'_1, T'_2) = \delta(T_1, T_2) + 2$ .
- If exactly one among  $T_1$  and  $T_2$  is rooted into a bad edge, than only one new edge contributes to the symmetric difference computation of the  $\delta_R$  distance by adding one clade, thus  $\delta_R(T'_1, T'_2) = \delta(T_1, T_2) + 1$ .

### 8.2 Proof of Lemma 2 (Edit distance):

The non-negative and identity conditions are obvious. For the symmetric condition, notice that we can reverse every edit operation in an optimal sequence from  $T_1$  to  $T_2$  to obtain a sequence from  $T_2$  to  $T_1$  with the same number of events, and vice-versa (extensions and contractions are inverses of each other, and any flip can be reversed by a flip). We thus have  $\delta(T_2, T_1) \leq \delta(T_1, T_2)$  and  $\delta(T_1, T_2) \leq \delta(T_2, T_1)$ , and equality follows.

Finally, we prove the triangular inequality condition: for 3 trees  $T_1$ ,  $T_2$  and  $T_3$ , to transform  $T_1$  into  $T_2$ , we may take any edit sequence from  $T_1$  to  $T_3$ , followed by any edit sequence from  $T_3$  to  $T_2$ . It follows that  $\delta(T_1, T_2) \leq \delta(T_1, T_3) + \delta(T_3, T_2)$ .

### 8.3 Proof of Lemma 3 (Pairs of maximal bad subtrees):

As  $\cup_i Y_i = \mathcal{L}$ ,  $\{e'_i\}_{1 \leq i \leq k}$  are the only terminal edges of any subtree  $S'$  of  $T'$  containing the set  $\{e'_i\}_{1 \leq i \leq k}$  as terminal edges. As  $T'$  is a tree, for any  $1 \leq i \neq j \leq k$ , there is only one possible path from  $x'_i$  to  $x'_j$ . Uniqueness follows.

Suppose that such a subtree  $S'$  is not a bad subtree. Then it contains an internal good edge  $e' = (x', y')$ . In other words, there is a non-trivial bipartition of  $\{Y_i\}_{1 \leq i \leq k}$  which is also a bipartition in  $S$ . This contradicts the fact that  $S$  is a bad subtree of

$T$ . Finally, as all terminal edges of  $S'$  are good edges of  $T'$ , it follows that  $S'$  is a maximal bad subtree of  $T'$ .

#### 8.4 Proof of Lemma 4 (Contract non-mixed bad edges):

We first introduce a definition that will be of use later in the proof. For two rooted trees  $S_1$  and  $S_2$ , define the *union* of  $S_1$  and  $S_2$  as the tree obtained by identifying their roots, i.e. by removing the root of  $S_2$  and making all its children now children of the root of  $S_1$ .

Let  $e = \{u, v\}$  be a non-mixed bad edge and assume, without loss of generality, that both  $u$  and  $v$  have the label *Spe* (recall that  $\Lambda = \{\textit{Spe}, \textit{Dup}\}$ ). Notice that any sequence of operations turning  $T$  into  $T'$ , at some point, must contract the  $\{u, v\}$  edge, as otherwise, the (bad) bipartition corresponding to  $\{u, v\}$  would remain in the transformed tree and we would not obtain  $T'$  (noting that extensions cannot remove bipartitions). We now prove the Lemma by induction over  $\delta(T, T')$ . As a base case, suppose that  $\delta(T, T') = 1$ . Then  $\{u, v\}$  must be the only bad edge of  $T$  and the single operation is to contract it, proving the base case.

Now assume that for any tree  $\tilde{T}$  satisfying  $\delta(\tilde{T}, T') < \delta(T, T')$ , contracting any non-mixed bad edge of  $\tilde{T}$  reduces its distance to  $T'$  by 1. Let  $Q = (q_1, \dots, q_l)$  be an optimal sequence of operations transforming  $T$  into  $T'$  (here each  $q_i$  denotes either a contraction, extension or flip). Let  $q_j$  be the event that contracts  $\{u, v\}$ . If  $q_1 = q_j$ , then we are done, so assume otherwise. We make the assumption that whenever there is a contraction involving  $u$  prior to  $q_j$ , the contracted node is still called  $u$ . Furthermore, we assume that if an extension prior to  $q_j$  splits the neighbors of  $u$ , the node  $v$  is still a neighbor of  $u$  after the operation. All the same assumptions hold for  $v$ . This just changes the names we give to nodes and does not alter the scenario, but observe that this means that  $\{u, v\}$  is in every tree obtained before the first  $j$  operations.

For each  $i \in \{1, \dots, l\}$ , let  $T_i$  be the tree obtained after applying  $q_1, \dots, q_i$  on  $T$ , and define  $T_0 = T$ . Furthermore, for  $i \in \{0, 1, \dots, j-1\}$ , denote by  $T_i^u$  and  $T_i^v$  the two trees obtained from  $T_i$  by removing the edge  $\{u, v\}$ , where  $u$  is in  $T_i^u$  and  $v$  is in  $T_i^v$ . Define  $T^u = T_0^u$  and  $T^v = T_0^v$ . We will assign  $u$  and  $v$  as the respective roots of each  $T_i^u$  and  $T_i^v$ . Notice that for each  $i \in \{1, \dots, j-1\}$ ,  $q_i$  only modifies either the subtree  $T_{i-1}^u$  or  $T_{i-1}^v$ . Therefore, if events  $q_i$  and  $q_{i+1}$  modify  $T_{i-1}^u$  and  $T_i^v$ , respectively, we could apply  $q_{i+1}$  before  $q_i$  and  $T_{i+1}$  would still be the same tree. This lets us assume that we may reorder events such that all events affecting  $T^u$  (prior to  $q_j$ ) occur before those affecting  $T^v$ . That is, there is some  $h$  such that  $q_1, \dots, q_h$  only affects the  $T^u$  subtree,  $q_{h+1}, \dots, q_{j-1}$  only affects the  $T^v$  subtree, so that  $T_h^u = T_{h+1}^u = \dots = T_{j-1}^u$  and  $T^v = T_1^v = \dots = T_h^v$ .

Suppose first that  $u$  is labeled *Spe* in  $T_h$ , and thus also in  $T_{j-1}$ . Then  $v$  is also labeled *Spe* in  $T_{j-1}$  (and also in  $T_h$  since  $v$  was untouched until  $q_{h+1}$ ). Let  $\hat{T}$  be the tree obtained after contracting  $\{u, v\}$  in  $T$ , and let  $z$  be the resulting node. Observe that if we interpret  $z$  as  $u$ , then we may apply the events  $q_1, \dots, q_h$  on  $\hat{T}$ , since these events only affected the  $T^u$  subtrees. To be formal, we “reproduce”  $q_1$  through  $q_h$  on  $\hat{T}$  by applying the events  $Q' = (q'_1, \dots, q'_h)$  on  $\hat{T}$ , defining  $\hat{T}_i$  as the tree obtained after the  $i$ -th event of  $Q'$ , where each  $q'_i$  in  $Q'$  is defined as follows:



- if  $q_i$  contracts  $\{x, y\}$  in  $T_{i-1}$ , then  $q'_i$  contracts  $\{x, y\}$  in  $\hat{T}_{i-1}$  if  $x, y \neq u$ , otherwise if, say,  $x = u$ , then  $q'_i$  contracts  $\{z, y\}$  (and calls the resulting node  $z$ );
- if  $q_i$  flips  $x$  in  $T_{i-1}$ , then  $q'_i$  flips  $x$  in  $\hat{T}_{i-1}$  if  $x \neq u$ , or flips  $z$  otherwise;
- if  $q_i$  is an extension and splits the neighborhood of  $x$ , then  $q'_i$  does the same if  $x \neq u$  (replacing  $u$  by  $z$  if needed). If  $x = u$ , then let  $X$  be the set of neighbors of  $v$  in  $T_{i-1}$ , excluding  $u$ . If  $Ch(u)$  is split into  $A$  and  $B$  by  $q_i$ , where  $v \in B$ , then  $q'_i$  splits the neighbors  $A \cup (B \setminus \{v\}) \cup X$  of  $z$  into  $A$  and  $(B \setminus \{v\}) \cup X$  (and  $z$  is the neighbor of  $(B \setminus \{v\}) \cup X$  and the newly created node).

One can verify the following that the following invariant holds on each  $\hat{T}_i$ ,  $i \in \{1, \dots, h\}$ : if we take  $T_i$  and contract the edge  $\{u, v\}$ , ignoring the labels and keeping the label of  $u$ , then we obtain  $\hat{T}_i$  (the invariant is also true for  $T$  and  $\hat{T}$ ).

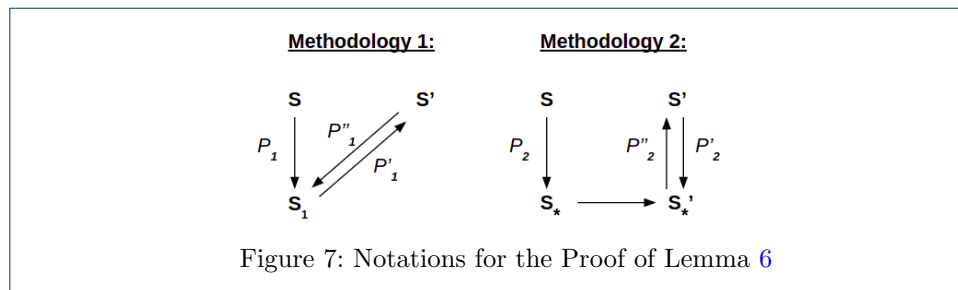
The resulting tree  $\hat{T}_h$  obtained from applying  $q'_1, \dots, q'_h$  on  $\hat{T}$  will therefore contain  $z$  as a *Spe* node, and will be the union of  $T_h^u$  and  $T_0^v$ . From this point, in a similar fashion, we may interpret  $z$  as  $v$  and apply  $q_{h+1}, \dots, q_{j-1}$  on  $\hat{T}_h$ , resulting a tree that is the union of  $T_h^u = T_{j-1}^u$  and  $T_{j-1}^v$ . The corresponding events are the same as above, we omit the formal details. Since  $T_j$  is obtained from  $T_{j-1}$  by contracting  $\{u, v\}$ , this means that  $\hat{T}_{j-1} = T_j$ , which we have attained with  $j$  events but contracting  $\{u, v\}$  first, which proves this case.

Suppose instead that  $u$  is labeled *Dup* in  $T_h$ . Then  $v$  is a *Dup* node in  $T_{j-1}$ . We may further assume that  $v$  is a *Spe* node in  $T_{h+1}, \dots, T_{j-2}$ , since whenever we flip  $v$  into a *Dup*, we may assume by induction that  $\{u, v\}$  gets contracted. Therefore,  $q_{j-1}$  flips  $v$  from *Spe* to *Dup*, and for the first time. We may then do the following: first apply the events  $q_{h+1}, \dots, q_{j-2}$  on  $\hat{T}$ , interpreting  $z$  as  $v$ . The resulting tree  $\hat{T}'$  contains  $z$  as a *Spe* node, and is the union of  $T_{j-2}^v$  and  $T_0^u$ . We may now apply  $q_1, \dots, q_h$  on  $\hat{T}'$  by interpreting  $u$  as  $z$ , resulting in a tree  $\hat{T}''$  that contains  $z$  as a *Dup* node and is the union of  $T_h^u = T_{j-1}^u$  and  $T_{j-1}^v$ . We have thus attained  $T_j$ , but this time without the  $q_{j-1}$  flip on  $v$ , contradicting the optimality of  $Q$ . This concludes the proof.

### 8.5 Proof of Lemma 5 (Upper bound $\delta$ ):

Methodology 1 performs  $e$  contractions and  $e'$  extensions. As for the number of flips, we have to flip at most all the nodes belonging to the smallest label group, which means at most half the nodes in each tree, and thus at most  $n$  flips in total.

### 8.6 Proof of Lemma 6 (Compare Meth.1 and Meth.2):



We denote by  $Cont(T)$  the minimum length of a sequence of operations contracting  $T$ , and by  $l(\mathcal{P})$  the length of a sequence  $\mathcal{P}$  of edit operations.

Let  $\mathcal{P}_2$  be an optimal sequence contracting  $S$  to  $S_*$  and  $\mathcal{P}'_2$  be an optimal sequence contracting  $S'$  to  $S'_*$ . As each operation is reversible,  $\mathcal{P}'_2$  leads to a corresponding sequence  $\mathcal{P}''_2$  of the same length between  $S'_*$  and  $S'$ . Thus,  $\mathcal{P}_2$ , concatenated with a possible flip operation transforming  $S_*$  to  $S'_*$ , concatenated with  $\mathcal{P}''_2$  is a sequence from  $S$  to  $S'$  following Methodology 1, and thus  $M_1(S, S') \leq M_2(S, S')$  (R1).

Conversely, let  $\mathcal{P}$  be an optimal sequence following Methodology 1. Then this sequence can be subdivided into a sequence  $\mathcal{P}_1$  from  $S$  to a star tree  $S_1$ , and  $\mathcal{P}'_1$  from  $S_1$  to  $S'$ . As each operation is reversible,  $\mathcal{P}'_1$  leads to a corresponding sequence  $\mathcal{P}''_1$  of the same length between  $S'$  and  $S_1$ . In other words,  $M_1(S, S') = l(\mathcal{P}_1) + l(\mathcal{P}'_1) = l(\mathcal{P}_1) + l(\mathcal{P}''_1) \geq Cont(S) + Cont(S')$ .

- 1 If  $S_* = S'_*$ , then  $M_2(S, S') = Cont(S) + Cont(S')$  and thus  $M_1(S, S') \geq M_2(S, S')$ , and the result follows from (R1).
- 2 Otherwise,  $S_*$  and  $S'_*$  are different and  $M_2(S, S') = Cont(S) + Cont(S') + 1$ . Thus  $M_1(S, S') \geq Cont(S) + Cont(S') = M_2(S, S') - 1$ , and thus  $M_2(S, S') \leq M_1(S, S') + 1$ .

### 8.7 Proof of Lemma 7 (Optimal path contracting a mixed tree):

We first show that at least  $\lceil diam(T)/2 \rceil - 1$  flips are needed, by induction over the diameter of  $T$ . When  $diam(T) = 2$ ,  $T$  is a star tree and  $0 = diam(T)/2 - 1$  flips are needed. For the induction step, we assume that any tree  $T'$  with  $diam(T') < diam(T)$  requires at least  $\lceil diam(T')/2 \rceil - 1$  flips. Take any optimal sequence of events  $S$ , and observe that in  $S$ , when we flip a node  $v$  of  $T$ , by Lemma 4 we may assume that  $S$  contracts all the incident edges to  $v$  until we obtain another mixed tree. Let  $T_1, T_2, \dots, T_k$  be the sequence of mixed trees encountered when applying  $S$ , i.e. each  $T_i$  is obtained after flipping a node and contracting its incident edges. Define  $T_0 = T$ . Let  $i$  be the smallest index such that  $diam(T_i) < diam(T)$ . Then in  $T_{i-1}$ , there was a longest chain  $P = (u_1, \dots, u_l)$  of length  $diam(T)$ . The flip-and-contract operations from  $T_{i-1}$  to  $T_i$  can reduce the length of  $P$  by at most 2 since we flip one node and only its incident edges, of which there are at most two on  $P$ . Hence  $diam(T_i) \geq diam(T) - 2$ . We deduce by induction that the number of required flips is at least  $1 + \lceil (diam(T) - 2)/2 \rceil - 1 = \lceil diam(T)/2 \rceil - 1$ .

We now turn to the converse bound  $\phi(T) \leq \lceil diam(T)/2 \rceil - 1$ . Fix any node  $v$  of  $T$ , and suppose that we run the following procedure: as long as  $T$  is not a star tree, flip  $v$  and contract its incident internal edges. Since each flip-and-contraction iteration reduces the length from  $v$  to any leaf by 1 (except its neighbors),  $ecc_T(v)$  is reduced by 1 each round. We stop when  $ecc_T(v) = 1$ , in which case only terminal edges remain, and in the end, this means that  $ecc_T(v) - 1$  flips are needed.

To see why this proves our bound, we show that there always exists a node with eccentricity  $\lceil diam(T)/2 \rceil$ . Consider a longest chain  $P$  of  $T$  with nodes  $w_1, \dots, w_k$ . Observe that  $diam(T) = k - 1$  (recall that distances are counted in terms of edges). Consider a midpoint node  $w := w_{\lceil k/2 \rceil}$  on  $P$ . We claim that  $ecc_T(w) = \lceil diam(T)/2 \rceil$ . It is easy to check that  $w$  has distance at most  $\lceil diam(T)/2 \rceil$  and at least  $\lfloor diam(T)/2 \rfloor$  to the leaves  $w_1$  and  $w_k$  on  $P$ . Assume for contradiction that  $w$  is at distance at least  $\lceil diam(T)/2 \rceil + 1$  from some leaf  $l$  of  $T$  not in  $P$ . Then either we can form a chain from  $w_1$  to  $w$  and then to  $l$ , or a chain from  $w_k$  to  $w$  and then to  $l$ . This chain has length at least  $\lfloor diam(T)/2 \rfloor + \lceil diam(T)/2 \rceil + 1 > diam(T)$ , a contradiction. This shows that  $ecc_T(w) = \lceil diam(T)/2 \rceil$  and concludes the proof.

8.8 Proof of Theorem 2 (Upper bound Meth.2):

Consider a given instance  $(T, T')$ . Take any leaf of  $T$  and assign it as the root, and do the same for  $T'$ . Although we have assumed roots of degree at least two so far, we use this rooting only for our analysis in order fix a parent-child relationship between nodes. Let  $Q$  be an optimal sequence of operations turning  $T$  into  $T'$ . We may assume that  $Q$  first contracts every non-mixed edge, and our algorithm does the same. Therefore, we suppose that  $T$  and  $T'$  contain no non-mixed edges. Assume for our purposes that whenever a contraction takes place in  $Q$  between a node  $u$  and a child  $v$ , the  $u$  node stays in the tree and  $v$  gets removed (here the notion of a child is in the rooted sense with respect to our rooting above). Also assume that when there is an extension splitting a node  $u$ , then the newly created node becomes a child of  $u$  and  $u$  retains the same parent. It is easily checked that this only alters the name of nodes and not the sequence itself.

Call an internal node  $v$  of  $T$  a *good child* if the edge between  $v$  and its parent is good. Note that  $v$  has a unique corresponding node in  $T'$  which we denote  $v'$  (i.e.  $v'$  is the root of the same clade as the subtree rooted at  $v$ ). Further, call  $v$  a *bad-good* child if  $v$  is a good child, but either the label of  $v$  differs from that of  $v'$ , or  $v$  is incident to at least one bad edge. Note that every maximal bad subtree of  $T$  has a (good) terminal edge with one endpoint being a bad-good child. Also note that a bad-good child  $v$  that is incident to only good edges is a particular case of a maximal bad subtree (i.e.  $v$  just has the wrong label).

We already know that  $\delta(T, T')$  is at least the number of bad edges in  $T$  and  $T'$ . Let  $Q'$  be the set of operations of  $Q$  that are either flips, or contraction of good edges. We argue that  $|Q'|$  is at least the number of bad-good children in  $T$ . To see this, let  $v$  be a bad-good child. Assume first that  $v$  is not incident to any bad edge. If we never flip  $v$  nor remove it by contracting its parent edge, then  $Q$  cannot transform  $T$  into  $T'$ , as  $v$  and its underlying clade remain present in every tree from  $T$  to  $T'$ , but with the wrong label (because a contraction not removing  $v$  cannot remove the  $v$  clade, and extensions can create clades but not remove them). So we may assume that  $v$  gets flipped or that its parent edge gets contracted. A flip must be in  $Q'$  and, observing that at any point the parent edge of  $v$  must be good, a contraction removing  $v$  must also be in  $Q'$ . Assume instead that  $v$  is incident to at least one bad edge  $\{v, w\}$ , with  $w$  a child of  $v$ . If  $v$  is never flipped nor removed owing to a contraction of its parent edge, then at some point  $w$  must be flipped so that the  $\{v, w\}$  edge gets contracted. Otherwise, if  $v$  gets removed, then its parent edge was contracted, again implying the contraction of a good edge. Either case implies an operation in  $Q'$ . Importantly, observe that the operations in  $Q'$  identified above are all distinct, since each one implies a flip or a the removal of a node in a different bad subtree of  $T$ .

Now, let  $T_1, \dots, T_k$  be the maximal bad subtrees of  $T$  and  $T'$ , and for each  $i \in \{1, \dots, k\}$ , let  $t_i$  be the number of bad edges in  $T_i$ . Further denote  $b = \sum_{i=1}^k t_i$ . Since bad subtrees form pairs, our arguments above imply that  $Q'$  has at least  $k/2$  operations (because  $|Q'|$  is at least the number of maximal bad trees in  $T$ , which is half the number of bad subtrees). The contraction of bad edges plus the operations of  $Q'$  show that  $Q$  has at least  $\sum_{i=1}^k t_i + k/2 = b + k/2$  operations. Our algorithm contracts  $b$  edges in total. To count the number of flips, take any

bad subtree  $T_i$ . Then  $t_i \geq \text{diam}(T_i) - 2$  and the number of flips we perform is at most  $\lceil \text{diam}(T_i)/2 \rceil - 1 = \lceil (\text{diam}(T_i) - 2)/2 \rceil \leq t_i/2 + 1$ . Note that this also holds when  $T_i$  contains no bad edge. Therefore, the number of operations that we perform is at most  $b + \sum_{i=1}^k (t_i/2 + 1) = 3b/2 + k$ . Our approximation ratio is therefore  $\frac{3b/2+k}{b+k/2} \leq \frac{2b+k}{b+k/2} = 2$ .