Briand *et al. BMC Genomics* 2020, **21**(Suppl 10):779
https://doi.org/10.1186/s12864-020-07011-0

BMC Genomics

RESEARCH Open Access

A generalized Robinson-Foulds distance for labeled trees

Samuel Briand[1], Christophe Dessimoz[2,3,4,5,6*], Nadia El-Mabrouk[1*], Manuel Lafond[7] and Gabriela Lobinska[3]

From The 18th Asia Pacific Bioinformatics Conference
Seoul, Korea. 18-20 August 2020

Abstract

Background: 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).

Results: 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.

Conclusions: 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

Background

Phylogenic 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

*Correspondence: c.dessimoz@ucl.ac.uk; mabrouk@iro.umontreal.ca
[2]Department of Computational Biology, University of Lausanne, Lausanne, Switzerland
[1]Computer Science Department, Université de Montréal, Montreal, Canada
Full list of author information is available at the end of the article

[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 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. We then 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 the following section, the heuristic is shown, on simulated datasets, to be efficient, by plotting the number of tree edits against the computed RF distance. Finally, we explore some avenues for improvement. All proofs are given in the Appendix.

Methods

We first start with notations and concepts, and then describe the Robinson Foulds distance and the extension to labeled trees.

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 $V(C) = \{x_1, \cdots, x_k\}$ and an edge set $E(C) = \{e_1, \cdots, e_{k-1}\}$ 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, an unrooted version of T can just be T ignoring the root status of $r(T)$. To avoid having nodes of degree two, we rather define the *unrooted tree T' corresponding to T* as the unrooted tree obtained from T by adding a dummy leaf R and an edge $e = (r(T), R)$.

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.

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 $X = \{y_1, \cdots, y_t\} \subsetneq Ch(x)$ 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*. 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 link between the distance defined in terms of clades (that we write δ_R) and the edit distance (that we write δ) 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, T_2 be two rooted trees, and T'_1, T'_2 be the corresponding unrooted trees. Then*

$$\delta_R(T_1, T_2) = \delta(T'_1, T'_2)$$

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.

Labeled trees

Given a finite set of labels Λ, *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 λ' to x, i.e. a label $\lambda' \in \Lambda$ such that $\lambda' \neq \lambda$. Those operations are depicted in Fig. 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 δ 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, Λ is restricted to two labels. They are illustrated by a circle and a square in Fig. 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 the speciation/duplication labeling. Algorithmic analyses are made independently of the nature of the two node labels. However, for notation purpose, we write $\Lambda = \{Spe, Dup\}$.

Results

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'.

Note that although we focus on unrooted trees, our results can then be easily extrapolated to rooted trees using Lemma 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 $\{S_1, S_2, \cdots, S_k\}$ be the set of maximal bad subtrees of T and $\{S'_1, S'_2, \cdots, S'_k\}$ be the corresponding subtrees of T' (see Fig. 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 $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_m$ 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 Fig. 3.

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 $\delta(T_c, T') = \delta(T, T') - 1$.*

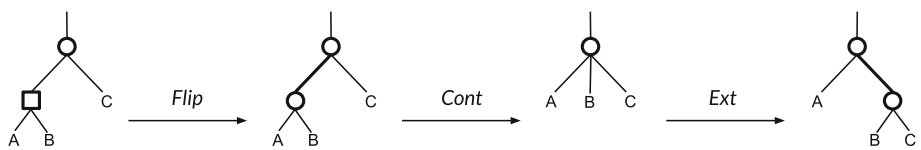

Fig. 1 The three edit operations defined for labeled trees. From left to right: Flip, Contraction and Extension

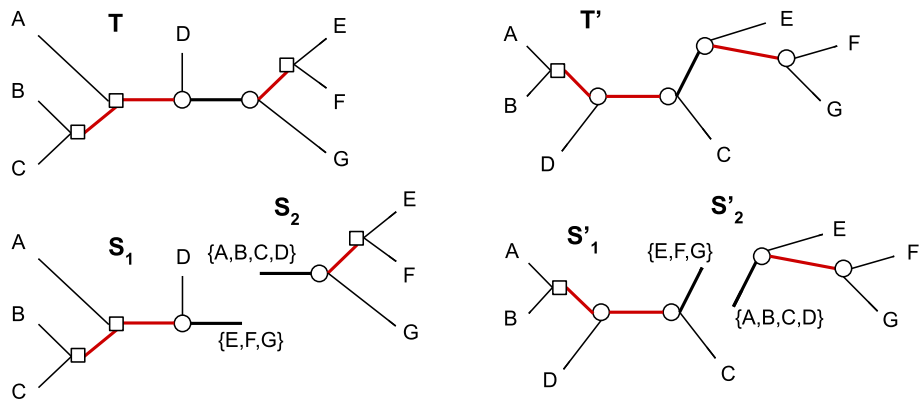

Fig. 2 Two unrooted and labeled trees T and T' on $\mathcal{L} = \{A, B, C, D, E, F, G\}$. The square and circle symbols represent the two possible labels for an internal node. Bad edges are red and good ones are black. $\{S_1, S_2\}$ are the maximal bad subtrees of T and $\{S'_1, S'_2\}$ the corresponding subtrees of T'

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.

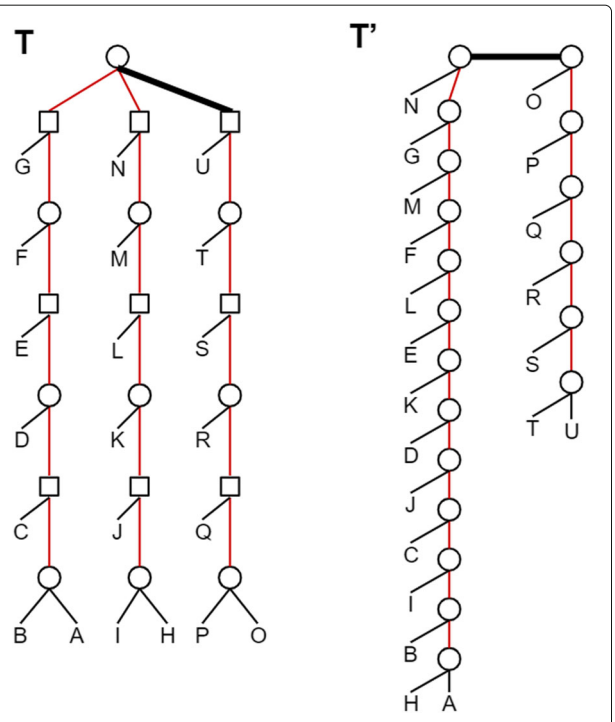

Fig. 3 Example where the minimal edit path requires contracting a good edge: if we contract the internal good edge of T (the bold one), then the 3 subtrees of T can be handled together, requiring 6 node flips and 18 edge contractions to reduce T into a star tree, and then 18 edge extensions to reach T', leading to 42 operations in total. By contrast, if we do not contract the good edge of T, then the two subtrees of T separated by this edge should be handled separately, requiring 9 flips, 17 edge contractions and 17 edge extensions to reach T', leading to 43 operations in total. The first scenario is the better one

Algorithms

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

Algorithm 1 ***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 δ) *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 case $2e + n$ is a tight bound as it can be reached in some cases (see an example in Fig. 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

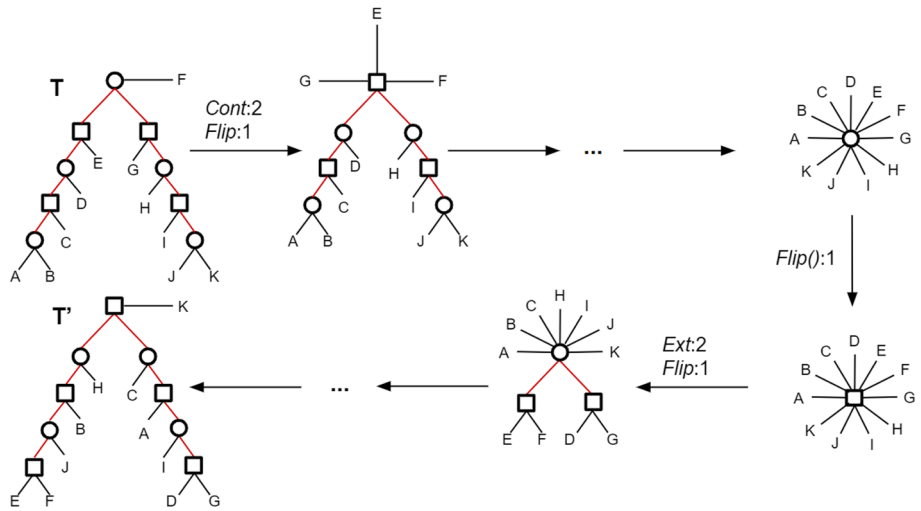

Fig. 4 A pair of unrooted mixed trees (T, T'), both with eight internal edges and nine internal nodes. All their internal edges are bad edges (red edges). Here $\delta(T, T') = 25 = 2 \cdot 8 + 9 = 2 \cdot e + n$

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.

Algorithm 2 ***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, $M_1(S, S') \leq M_2(S, S') \leq M_1(S, S') + 1$*

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 Fig. 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]

[1] 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 $\lceil diam(T)/2 \rceil$.

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

Let $P = (w_1, w_2, \ldots, w_k)$ 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* $d \leq 2\delta\left(T, T'\right)$.

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 internal edges connecting nodes of the same type (which could be potentially contracted) and all nodes of degree > 3 (in which a new edge could potentially 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 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).

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 Fig. 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 Fig. 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

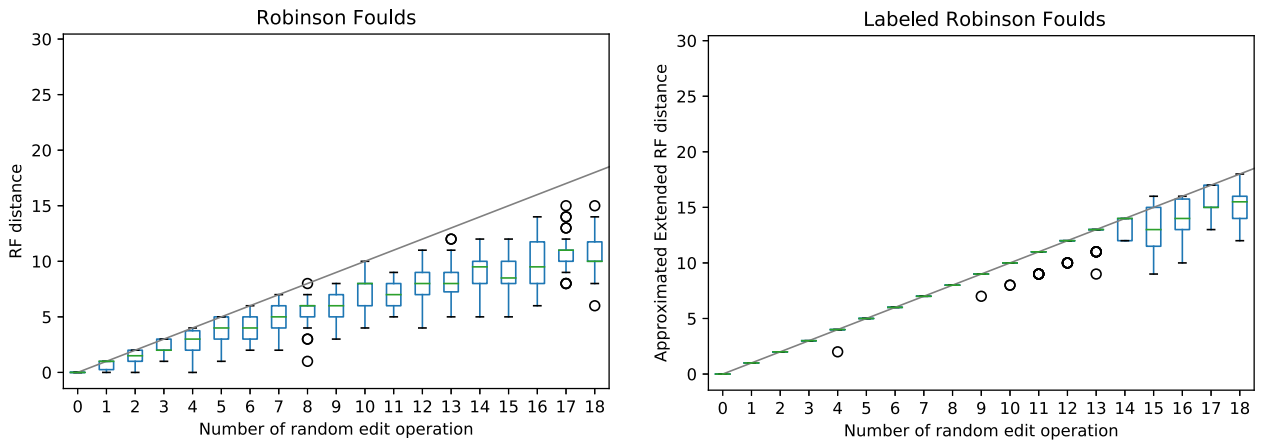

Fig. 5 Empirical comparison of the distance inferred for an increasing number of random edit operations (contraction, extensions, and flips), using the classical Robinson-Foulds distance (left) and *Contract-Tree* algorithm (right). Because the former ignores node labels, it grossly underestimates the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

orthology benchmarking, to compare inferred labeled trees with reference curated ones [29].

Conclusion

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 Fig. 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.

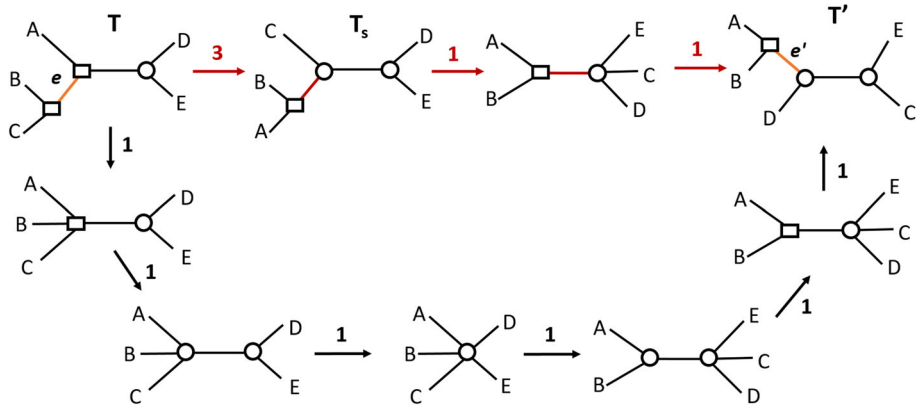

Fig. 6 An optimal path from T to T' following *Methodology 1* is depicted by black arrows and involves 6 operations. It is not optimal as another path, depicted by red arrows, involves only five operations. The path of length 3 from T to T_s acts on the safe edge, represented in orange. This path involves an edge contraction, an edge extension and a flip, leading to the good edge (red edge) in T_s

Appendix

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

Let T_1 and T_2 be two rooted trees and T_1' and T_2' be the corresponding unrooted trees, i.e. $V(T_1') = V(T_1) \cup \{R\}$, $V(T_2') = V(T_2) \cup \{R\}$, $E(T_1') = E(T) \cup \{(r(T_1), R)\}$ and $E(T_2') = E(T) \cup \{(r(T_2), R)\}$.

We first show that any bad bipartition of T_1', i.e. any bad edge of T_1', corresponds to a bad clade of T_1 (a clade which is not present in T_2). Let e_1' be a bad edge of T_1'. Then e_1' should be a non-terminal edge of T_1', thus different from $(r(T_1), R)$), and therefore it has a corresponding edge $e_1 = (x_1, y_1)$ in T_1. Then, for one of the two nodes adjacent to e_1' that we denote y_1', we have $L(T'_{1y_1'}) = L(T_{1y_1}) = C$. If e_1' is a bad edge of T_1', then C should be a bad clade of T_1 not present in T_2. This is because otherwise C would be a non-trivial clade of T_2 rooted at an internal node y_2 adjacent to an edge $e_2 = (x_2, y_2)$ and thus also equal to $L(T'_{2y_2'})$ for a given edge $e_2' = (x_2', y_2')$. This contradicts the fact that e_1' is a bad edge. Therefore, each bad bipartition of T_1' corresponds to a bad clade of T_1. Moreover, two disjoint bad bipartitions of T_1' correspond to two different bad edges of T_1', with the corresponding edges of T_1 associated to two disjoint clades. Thus we have $|\mathcal{B}(T_1')| \leq |\mathcal{C}(T_1)|$.

Conversely, a bad clade C of T_1 corresponds to an internal node y_1 of T_1. Let $e_1 = (x_1, y_1)$ in T_1, where x_1 is the parent of y_1. Then the corresponding edge e_1' in T_1' is a bad edge. Moreover, two disjoint clades of T_1 correspond to two disjoint edges of T_1'. It follows that $|\mathcal{C}(T_1)| \leq |\mathcal{B}(T_1')|$. Combining this result with the result above, we deduce that $|\mathcal{C}(T_1)| = |\mathcal{B}(T_1')|$. As T_2 and T_2' can be considered similarly, the result follows.

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 $\delta(T_1, T_2) \leq \delta(T_1, T_3) + \delta(T_3, T_2)$.

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

As $\cup_i Y_i = Ł$, $\{e_i'\}_{1 \leq i \leq k}$ 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'.

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 $\Lambda = \{Spe, Dup\}$). 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 $Q = (q_1, \ldots, q_l)$ 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 $i \in \{1, \ldots, l\}$, let T_i be the tree obtained after applying $q_1, \ldots, q_i$ on T, and define $T_0 = T$. Furthermore, for $i \in \{0, 1, \ldots, j-1\}$, 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 $i \in \{1, \ldots, j-1\}$, 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 $q_1, \ldots, q_h$ only affects the

T^u subtree, $q_{h+1}, \ldots, q_{j-1}$ only affects the T^v subtree, so that $T^u_h = T^u_{h+1} = \ldots = T^u_{j-1}$ and $T^v = T^v_1 = \ldots = T^v_h$.

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 $q_1, \ldots, q_h$ 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 $Q' = (q'_1, \ldots, q'_h)$ 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 $A \cup B \cup X$ of z into A and $B \cup X$ (and z is the neighbor of $B \cup X$ and the newly created node).

One can verify the following that the following invariant holds on each $\hat{T}_i$, $i \in \{1, \ldots, h\}$: 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 $q'_1, \ldots, q'_h$ on $\hat{T}$ will therefore contain z as a *Spe* node, and will be the union of T^u_h and T^v_0. From this point, in a similar fashion, we may interpret z as v and apply $q_{h+1}, \ldots, q_{j-1}$ on $\hat{T}_h$, resulting a tree that is the union of $T^u_h = T^u_{j-1}$ and T^v_{j-1}. 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 $T_{h+1}, \ldots, T_{j-2}$, 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 $q_{h+1}, \ldots, q_{j-2}$ on $\hat{T}$, interpreting z as v. The resulting tree $\hat{T}'$ contains z as a *Spe* node, and is the union of T^v_{j-2} and T^u_0. We may now apply $q_1, \ldots, q_h$ 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 $T^u_h = T^u_{j-1}$ and T^v_{j-1}. 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.

Proof of Lemma 5 (Upper bound δ):
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.

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 $l(¶)$ the length of a sequence ¶ of edit operations (Fig. 7).

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

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

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 $M_2(S, S') \leq M_1(S, S') + 1$.

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 $T_1, T_2, \ldots, T_k$ 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 $P = (u_1, \ldots, u_l)$ 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

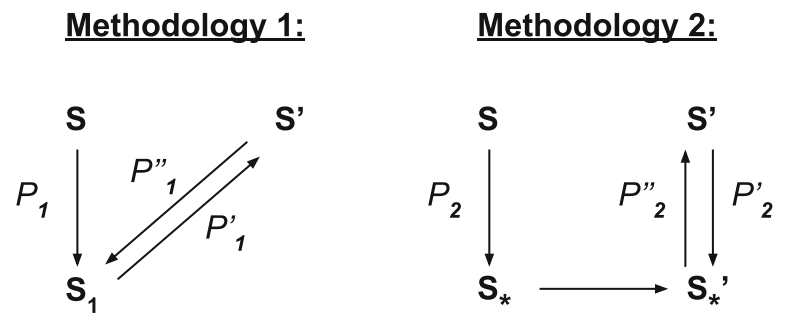

Fig. 7 Notations for the Proof of Lemma 6

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 $w_1, \ldots, w_k$. 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.

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 (yes, children are capable of being both bad and good). Note that every bad subtree of T is rooted at a bad-good child, and observe that here we say that a bad-good child v that is incident to only good edges is a particular case of a 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 cases imply an operation in Q'. Importantly, observe that the operations in Q' identified above are all distinct, since each one implies a flip or a node removal of a node in a different bad subtree of T.

Now, let $T_1, \ldots, T_k$ be the bad subtrees of T and T', and for each $i \in \{1, \ldots, k\}$, 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 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 $t_i \geq diam(T_i) - 2$ and the number of flips we perform is at most $\lceil diam(T_i)/2 \rceil - 1 = \lceil (diam(T_i) - 2)/2 \rceil \leq t_i/2 + 1$. 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 $\frac{3b/2+k}{b+k/2} \leq \frac{2b+k}{b+k/2} = 2$.

Abbreviations
RF: Robinson-Foulds

Acknowledgements
The authors thank the Program Committee of APBC2020 for the three constructive peer-reviews.

About this supplement
This article has been published as part of *BMC Genomics Volume 21 Supplement 10, 2020: Selected articles from the 18th Asia Pacific Bioinformatics Conference (APBC 2020): genomics*. The full contents of the supplement are available online at https://bmcgenomics.biomedcentral.com/articles/supplements/volume-21-supplement-10.

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 author(s) read and approved the final manuscript.

Funding
Christophe Dessimoz is supported by the Swiss National Science Foundation (SNSF) Professorship grant 183723. Nadia El-Mabrouk is supported by Natural Sciences and Engineering Research Council of Canada (NSERC), and Fonds de Recherche Nature et Technologies of Quebec (FRQNT). Manuel Lafond is supported by Natural Sciences and Engineering Research Council of Canada (NSERC). Publication costs were funded by the Swiss National Science Foundation. The funding sources had no role in the design and conduct of the study, nor in the writing, review or approval of the manuscript.

Availability of data and materials
The software implemented during the current study is available in the pylabeledrf repository, https://github.com/DessimozLab/pylabeledrf.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1]Computer Science Department, Université de Montréal, Montreal, Canada. [2]Department of Computational Biology, University of Lausanne, Lausanne, Switzerland. [3]Department of Genetics Evolution and Environment, University College London, London, UK. [4]Center for Integrative Genomics, University of Lausanne, Lausanne, Switzerland. [5]Swiss Institute of Bioinformatics, Lausanne, Switzerland. [6]Department of Computer Science, University College London, London, UK. [7]Computer Science Department, Université de Sherbrooke, Sherbrooke, Canada.

Published: 18 November 2020

References
1. Semple C, Steel M, et al. Phylogenetics vol. 24. Oxford: Oxford University Press on Demand; 2003.
2. Colijn C, Plazzotta G. A metric on phylogenetic tree shapes. Syst Biol. 2018;67(1):113–26.
3. Lafond M, El-Mabrouk N, Huber KT, Moulton V. The complexity of comparing multiply-labelled trees by extending phylogenetic-tree metric. Theor Comput Sci. 2019;760:15–34.
4. Bryant D, Scornavacca C. An $O(n \log N)$ time algorithm for computing the path-length distance between trees. Algorithmica. 2019;81(9):3692–706.
5. Cardona G, Llabrés M, Rosselló F, Valiente G. Nodal distances for rooted phylogenetic trees. J Math Biol. 2010;61(2):253–76.
6. Estabrook GF, McMorris F, Meacham CA. Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. Syst Zool. 1985;34(2):193–200.
7. Critchlow DE, Pearl DK, Qian C. The triples distance for rooted bifurcating phylogenetic trees. Syst Zool. 1996;45(3):323–34.
8. Jiang BDXHT, 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. Providence: American Mathematical Soc.; 2000. p. 125.
9. Hickey G, Dehne F, Rau-Chaplin A, Blouin C. Spr distance computation for unrooted trees. Evol Bioinforma. 2008;4:419.
10. Allen BL, Steel M. Subtree transfer operations and their induced metrics on evolutionary trees. Ann Comb. 2001;5(1):1–15.
11. Lin Y, Rajan V, Moret BM. A metric for phylogenetic trees based on matching. IEEE/ACM Trans Comput Biol Bioinforma (TCBB). 2012;9(4):1014–22.
12. Mittal S, Munjal G. Tree mining and tree validation metrics: A review. IOSR: J Comput Eng. 2015;2:31–36.
13. Day WH. Optimal algorithms for comparing trees with labeled leaves. J Classif. 1985;2(1):7–28.
14. Pattengale ND, Gottlieb EJ, Moret BM. Efficiently computing the robinson-foulds metric. J Comput Biol. 2007;14(6):724–35.
15. Steel MA, Penny D. Distributions of tree comparison metric–some new results. Syst Biol. 1993;42(2):126–41.
16. Bryant D, Steel M. Computing the distribution of a tree metric. IEEE/ACM Trans Comput Biol Bioinforma. 2009;6(3):420–6.
17. Chaudhary R, Burleigh JG, Fernandez-Baca D. Fast local search for unrooted robinson-foulds supertrees. IEEE/ACM Trans Comput Biol Bioinforma (TCBB). 2012;9(4):1004–13.
18. Moon J, Eulenstein O. Cluster matching distance for rooted phylogenetic trees. In: International Symposium on Bioinformatics Research and Applications. Springer; 2018. p. 321–32. https://doi.org/10.1007/978-3-319-94968-0_31.
19. Robinson DF, Foulds LR. Comparison of phylogenetic trees. Math Biosci. 1981;53(1-2):131–47.
20. Zhang K, Statman R, Shasha D. On the editing distance between unordered labeled trees. Inf Process Lett. 1992;42(3):133–9.
21. Zhang K. A new editing based distance between unordered labeled trees. In: Annual Symposium on Combinatorial Pattern Matching. Berlin: Springer; 1993. p. 254–65.
22. Zhang K. A constrained edit distance between unordered labeled trees. Algorithmica. 1996;15(3):205–22.
23. Schwarz S, Pawlik M, Augsten N. A new perspective on the tree edit distance. In: International Conference on Similarity Search and Applications. Springer; 2017. p. 156–70. https://doi.org/10.1007/978-3-319-68474-1_11.

24. Vilella AJ, Severin J, Ureta-Vidal A, Heng L, Durbin R, Birney E. EnsemblCompara gene trees: Complete, duplication-aware phylogenetic trees in vertebrates. Genome Res. 2009;19:327–35.
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 Res. 2013;42(D1):D922–D925. https://doi.org/10.1093/nar/gkt1055.
26. Dress A. Towards a theory of holistic clustering. DIMACS Ser Discrete Math Theoret Comput Sci. 1997;37:271–89.
27. Hernandez-Rosales M, Hellmuth M, Wieseke N, Huber KT, Moulton V, Stadler PF. From event-labeled gene trees to species trees. BMC Bioinformatics. 2012;13:6. BioMed Central.
28. Lafond M, El-Mabrouk N. Orthology and paralogy constraints: satisfiability and consistency. BMC Genomics. 2014;15(6):12.
29. Altenhoff AM, Boeckmann B, Capella-Gutierrez S, Dalquen DA, DeLuca T, Forslund K, Huerta-Cepas J, Linard B, Pereira C, Pryszcz LP, Schreiber F, da Silva AS, Szklarczyk D, Train C-M, Bork P, Lecompte O, von Mering C, Xenarios I, Sjölander K, Jensen LJ, Martin MJ, Muffato M, Quest for Orthologs consortium, Gabaldón T, Lewis SE, Thomas PD, Sonnhammer E, Dessimoz C. Standardized benchmarking in the quest for orthologs. Nature methods. 2016;13(5):425–30. https://doi.org/10.1038/nmeth.3830.

Publisher's Note