A Linear Time Solution to the Labeled Robinson-Foulds Distance Problem

Samuel Briand[1], Christophe Dessimoz[2,3,4,5,6], Nadia El-Mabrouk[1], and Yannis Nevers[2,3,6]

1. Département d'informatique et de recherche opérationnelle (DIRO), Université de Montréal
2. Department of Computational Biology, University of Lausanne
3. Center for Integrative Genomics, University of Lausanne
4. Centre for Life's Origins and Evolution, Genetics Evolution and Environment, University College London
5. Department of Computer Science, University College London
6. SIB Swiss Institute of Bioinformatics

Abstract

Motivation: Comparing trees is a basic task for many purposes, and especially in phylogeny where different tree reconstruction tools may lead to different trees, likely representing contradictory evolutionary information. While a large variety of pairwise measures of similarity or dissimilarity have been developed for comparing trees with no information on internal nodes, very few address the case of inner node-labeled trees. Yet such trees are common; for instance reconciled gene trees have inner nodes labeled with the type of event giving rise to them, typically speciation or duplication. Recently, we proposed a formulation of the Labeled Robinson Foulds edit distance with edge extensions, edge contractions between identically labeled nodes, and node label flips. However, this distance proved difficult to compute, in particular because shortest edit paths can require contracting "good" edges, i.e. edges present in the two trees.

Results: Here, we report on a different formulation of the Labeled Robinson Foulds edit distance — based on node insertion, deletion and label substitution — which we show can be computed in linear time. The new formulation also maintains other desirable properties: being a metric, reducing to Robinson Foulds for unlabeled trees and maintaining an intuitive interpretation.

The new distance is computable for an arbitrary number of label types, thus making it useful for applications involving not only speciations and duplications, but also horizontal gene transfers and further events associated with the internal nodes of the tree. To illustrate the utility of the new distance, we use it to study the impact of taxon sampling on labeled gene tree inference, and conclude that denser taxon sampling yields better trees.

Availibility and implementation: The software written in Python is available in the pylabeledrf repository at `https://github.com/DessimozLab/pylabeledrf`.

1 Introduction

Gene trees are extensively used, not only for inferring phylogenetic relationships between corresponding taxa, but also for inferring the most plausible scenario of evolutionary events leading to the observed gene family from a single ancestral gene copy. This has important implications towards elucidating the functional relationship between gene copies. For this purpose, reconciliation methods [reviewed in 4] embed a given gene tree into a known species tree. This process results in the labeling of the internal nodes of the gene tree with the type of events which gave rise to them, typically speciations and duplications, but also horizontal gene transfers or possibly other events (whole genome duplication, gene convergence, etc). For example, information on duplication and speciation node labeling is provided for the trees of the Ensembl Compara database [25].

The existence of a variety of different phylogenetic inference methods leading to different, potentially inconsistent, trees for the same dataset, brings forward the need for appropriate tools for comparing them. Although comparing labeled gene trees remains a largely unexplored field, a large variety of pairwise measures of similarity or dissimilarity have been developed for comparing unlabeled evolutionary trees. Among them are the methods based on counting the structural differences between the two trees in terms of path size, bipartitions or quartets for unrooted trees, clades or triplets for rooted trees [6, 10, 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 [2, 11, 13]. While the latter methods are NP-hard [15], 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 [16], can be computed in linear [9], and even sublinear time [20].

On the other hand, metrics have also been developed for node-labeled trees (rooted, and sometimes with an order on nodes) arising from many different applications in various fields (parsing, RNA structure comparison, computer vision, genealogical studies, etc), where node labels in a given tree are pairwise different (i.e. no repeated labels). For such trees, the standard Tree Edit Distance (TED) [29], defined in terms of a minimum cost path of node deletion, node insertion and node change (label substitution) transforming one tree to another, has been widely used. While the general version of the problem on unordered labeled trees with a non-constant cost function on edit operations is NP-complete [30], most variants are solvable in polynomial time [27, 28, 23].

The metric we developed in [5], referred to as $ELRF$, is the first effort towards comparing labeled gene trees, expressed in terms of trees with a binary node labeling (typically speciation and duplication). $ELRF$ is an extension of the RF distance, one of the most widely used tree distance, not only in phylogenetics, but also in other fields such as in linguistics, for its computational efficiency, intuitive interpretation and the fact that it is a true metric. Improved versions of the RF distance have also been developed [15, 17] to address the distance's drawbacks, which are lack of robustness (a small change in a tree may cause a disproportional change in the distance) and skewed distribution. Classically defined in terms of bipartition or clade dissimilarity, the RF distance can similarly be defined in terms of edit operations on tree edges: the minimum number of edge contraction and extension needed to transform one tree into the other [22]. In [5], this definition of the RF distance was extended to trees with binary node labeling by including a node *flip* operation, alongside edge contractions and extensions. While remaining a metric, $ELRF$ turned out to be much more challenging to compute. As a result, only a heuristic could be proposed to compute it.

In this paper, we explore a different extension of RF to node-labeled trees with labels belonging to a set of label types, directly derived from TED [29], which is a reformulation of the RF distance in terms of edit operations on tree nodes rather than on tree edges. We show that this new distance is computable in linear time for an arbitrary number of label types, thus making it useful for applications involving not only speciations and duplications, but also horizontal gene transfers and further events associated with the internal nodes of the tree. We show that the new distance compares favourably to RF and $ELRF$ by performing simulations on labeled gene trees of 182 leaves. Finally, we use our new distance in the purpose of measuring the impact of taxon sampling on labeled gene tree inference, and conclude that denser taxon sampling yields better predictions.

2 Notation and Concepts

The Robinson-Foulds (RF) distance is defined in the literature for rooted and unrooted trees. Moreover, as mentioned in [5], the problem of computing the RF distance for two rooted trees can be reduced to computing the RF distance for the two corresponding unrooted trees obtained by grafting an edge linking the root to a dummy leaf. Therefore, in this paper we restrict ourselves to unrooted trees. We begin by introducing few required notations.

Let T be a tree with node set $V(T)$ and edge set $E(T)$. Given a node x of T, the *degree of* x is the number of edges incident to x. In this paper, the considered trees are unrooted with all internal nodes being of degree at least 3. An internal node of degree 3 is said to be *binary*.

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. In particular, given a set $\mathcal{L}$ (let us say taxa or genetic elements), a tree T on $\mathcal{L}$ is a tree with leafset $L(T) = \mathcal{L}$.

A node of $V(T) \setminus L(T)$ is called an *internal node*. A tree with a single internal node x is called a *star tree*, and x is called a *star node*. 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.

We call $N(x) = \{y : \{x, y\} \in E(T)\}$ the set of neighbours of an internal node x of T.

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 $V(S)$.

The *bipartition* of an unrooted tree T corresponding to an 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. We denote by $\mathcal{B}(T)$ the set of non-trivial bipartitions of T, i.e. those corresponding to internal edges of T.

2.1 The Robinson-Foulds distance

Given two unrooted trees T and T' on the leafset $\mathcal{L}$, the Robinson-Foulds (RF) distance between T and T' is the size of the symmetric difference between the bipartitions of the two trees. More precisely,

$$RF(T, T') = |\mathcal{B}(T) \setminus \mathcal{B}(T')| + |\mathcal{B}(T') \setminus \mathcal{B}(T)|$$

The RF distance is equivalently defined in terms of an edit distance on edges. However, as for node-labeled trees an additional substitution operation on node labels will be required, for the sake of standardization, we reformulate the edit operations to operate on nodes rather than on edges.

Definition 1 (node edit operations)**.** *Two edit operations on the nodes of a tree T are defined as follows:*

- ***Node deletion:*** *Let x be an internal node of T which is not a star node and y be a neighbour of x which is not a leaf. Deleting x with respect to y means making the neighbours of x become the neighbours of y. More precisely, $Del(T, x, y)$ is an operation transforming the tree T into the tree T' obtained from T by removing the edge $\{x, z\}$ for each $z \in N(x)$, creating the edge $\{y, z\}$ for each $z \in N(x) \setminus \{y\}$, and then removing node x.*

- ***Node insertion:*** *Let y be an internal node of $V(T)$ of degree at least 3. Inserting x as a neighbour of y entails making x the neighbour of a subset $Z \subsetneq N(y)$ such that $|Z| \geq 2$. More precisely, $Ins(T, x, y, Z)$ is an operation transforming the tree T into the tree T' obtained from T by removing the edges $\{y, z_i\}$, for all $z_i \in Z$, creating a node x and a new edge $e = \{x, y\}$, and creating new edges $\{x, z_i\}$, for all $z_i \in Z$.*

Notice the one-to-one correspondence between operations on nodes and operations on edges. In fact, deleting a node x by an operation $Del(T, x, y)$ results in removing the edge $\{x, y\}$, while inserting a node x by an operation $Ins(T, x, y, Z)$ results in inserting the edge $\{x, y\}$. Here, we define the *RF* distance in terms of edit operations on nodes. Formally, let T and T' be two trees on the same leafset $\mathcal{L}$. The *Robinson-Foulds* or *Edit distance* [22] $RF(T, T')$ between T and T' is the size of a shortest path of edge edit operations (i.e. *edge extensions* and *edge contractions*) transforming T into T'. This distance measure, equivalently defined as the size of the symmetric difference between the non-trivial bipartitions of the two trees, has been shown to be a metric.

Call a *bad edge* of T with respect to T' (or similarly of T' with respect to T; if there is no ambiguity, we will omit the "with respect to" precision) an edge representing bipartitions which are not shared by the two trees, i.e. an edge of T (respec. T') defining a bipartition of $\mathcal{B}(T)$ (respec. $\mathcal{B}(T')$) which is not in $\mathcal{B}(T')$ (respec. in $\mathcal{B}(T)$). An edge which is not bad is said to be *good*. Terminal edges are always good.

3 Generalizing the Robinson-Foulds distance to Labeled Trees

A tree T is *labeled* if and only if each internal node x of T has a label $\lambda(x) \in \Lambda$, Λ being a finite set of labels. For gene trees, labels usually represent the type of event leading to the bifurcation, typically duplications and speciations, although other events, such as horizontal gene transfers, may be considered. The metric defined in this paper holds for an arbitrary number of labels. We generalize the RF distance to labeled trees by generalizing the edit operations defined above. This is simply done by introducing a third operation for node labels editing.

Definition 2 (Labeled node edit operations)**.** *Three edit operations on internal nodes of a labeled tree T are defined as follows:*

- ***Node deletion:*** $Del(T, x, y)$ *is an operation deleting an internal node x of T with respect to a neighbour y of x which is not a leaf, defined as in Definition 1.*
- ***Node insertion:*** $Ins(T, x, y, Z, \lambda)$ *is an operation inserting an internal node x as a new neighbour of a non-binary node y, and moving $Z \subsetneq N(y)$ such that $|Z| \geq 2$, to be the neighbours of x, as defined in Definition 1. In addition, the inserted node x receives a label $\lambda \in \Lambda$.*
- ***Node label substitution:*** $Sub(T, x, \lambda)$ *is an operation substituting the label of the internal node x of T with $\lambda \in \Lambda$.*

These operations are illustrated in Figure 1.

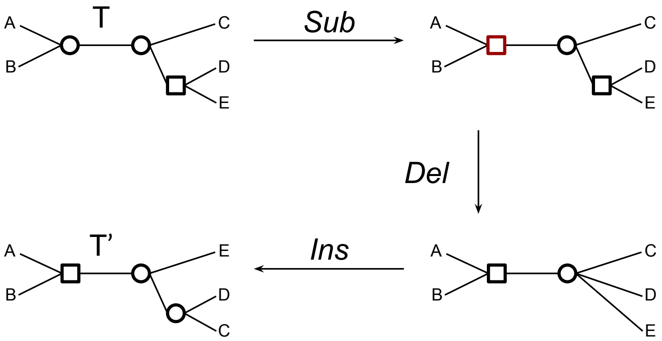

Figure 1: The transformation of a tree T into a tree T' depicting the three edit operations on nodes. From top to bottom: node label substitution (leading to the red label), node deletion (the parent of D and E) and node insertion (the parent of D and C).

Let $\mathcal{T}_\mathcal{L}$ be the set of unrooted and labeled trees on the leafset $\mathcal{L}$. For two trees T, T' of $\mathcal{T}_\mathcal{L}$, we call the *Labeled Robinson Foulds* distance between T and T' and denote by $LRF(T, T')$ the size of a shortest path of labeled node edit operations transforming T into T' (or vice versa). The two following lemmas state that, similarly to RF, LRF is a true metric. Moreover, LRF is exactly RF for unlabeled trees (or similarly labeled with a single label).

In the following, the *unlabeled version of* a tree $T \in \mathcal{T}_\mathcal{L}$ is simply T ignoring its node labels.

Lemma 1. *The function $LRF(T,T')$ assigning to each pair $(T,T') \in \mathcal{T}_{\mathcal{L}}^2$ the size of a shortest path of node edit operations transforming T into T' defines a distance on $\mathcal{T}_{\mathcal{L}}$.*

Proof. The non-negative, identity and triangular inequality conditions are obvious. For the symmetric condition, notice that we can reverse every edit operation in a path from T to T' to obtain a path from T' to T with the same number of events, and vice versa (insertions and deletions are symmetrical operations, and any substitution can be reversed by a substitution). We thus have $LRF(T',T) \leq LRF(T,T')$ and $LRF(T,T') \leq LRF(T',T)$, and equality follows. □

The next lemma directly follows from the fact that node substitutions are never applied in case of a label set restricted to a single label.

Lemma 2. *If Λ is restricted to a single label, then for each pair $(T,T') \in \mathcal{T}_{\mathcal{L}}^2$, $LRF(T,T') = RF(T,T')$.*

A previous extension of RF to labeled trees, based on edit operations on edges rather than on nodes, was introduced in [5]. This distance, which we call $ELRF$, was defined on three operations:

- Edge extension $Ext(T,x,X)$ creating an edge $\{x,y\}$ and defined as a node insertion $Ins(T,y,x,X,\lambda(x))$ inserting a node y as a neighbour of x and assigning to y the label of x;
- Edge contraction $Cont(T,\{x,y\})$ is equal to a node deletion $Del(T,y,x)$ deleting y, but contrary to LRF, requires that $\lambda(x) = \lambda(y)$;
- Node flip $Flip(T,x,\lambda)$ assigning the label λ to x.

Given two labeled trees T and T' of $\mathcal{T}_{\mathcal{L}}$, $ELRF(T,T')$ is the size of the shortest path of edge extension, edge contraction and label flip required to transform T to T'.

The following lemma makes the link between LRF and $ELRF$.

Lemma 3. *For any pair $(T,T') \in \mathcal{T}_{\mathcal{L}}^2$,*

$$LRF(T,T') \leq ELRF(T,T')$$

Proof. Let $\mathcal{P}$ be a path of edge edit operations and label flip transforming T into T' such that $|\mathcal{P}| = ELRF(T,T')$. Then the sequence $\mathcal{P}'$ obtained from $\mathcal{P}$ by replacing each edge extension by the corresponding node insertion, each edge contraction by the corresponding node deletion and each node flip by the corresponding node substitution is clearly a path of node edit operations of size $|\mathcal{P}'| = |\mathcal{P}| = ELRF(T, T')$ transforming T into T'. And thus $LRF(T, T') \leq ELRF(T, T')$. Figure 2 depicts an example were the inequality is strict. □

The rest of this paper is dedicated to computing the edit distance $LRF(T, T')$ for any pair (T, T') of trees of $\mathcal{T}_{\mathcal{L}}$.

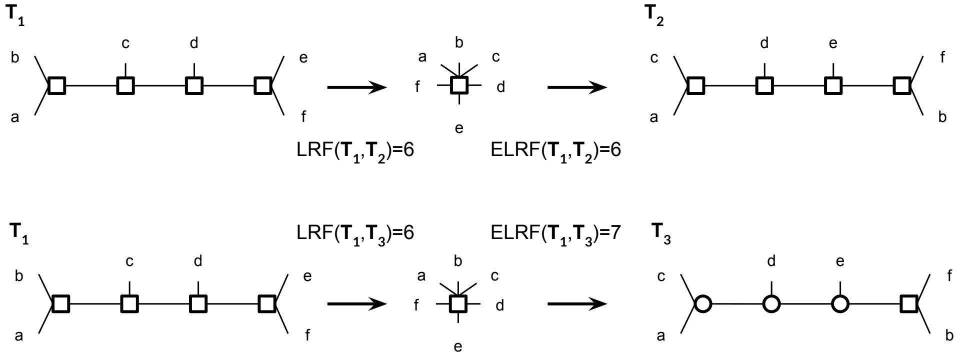

Figure 2: The transformation of a tree T_1 into a tree T_2 (respect. T_3) depicting where the equality (respect. the inequality) is strict between $LRF(T_1, T_2)$ and $ELRF(T_1, T_2)$ (respect. $LRF(T_1, T_3)$ and $ELRF(T_1, T_3)$).

3.1 Reduction to Islands

In this section, we define a subdivision of the two trees into pairs of maximum subtrees that can be treated separately.

While a good edge e in T has a corresponding good edge e' in T' (the one defining the same bipartition), a bad edge in T has no corresponding edge in T'. However, these bad edges may be grouped into pairs of corresponding *islands* (called maximum bad subtrees in [5]), as defined bellow.

Definition 3 (Islands)**.** *An* island *of T is a maximum subtree I (i.e. a subtree with a maximum number of edges) such that no internal edge of I is a good edge of T, and all terminal edges of I are good edges of T. The* size *of I, denoted $\epsilon(I)$, is its number of internal edges.*

In other words, an island of T is a maximum subtree with all internal edges (if any) being bad edges of T, and all terminal edges being good edges of T. Notice that an island I of T may have no internal edge at all, i.e. it may be restricted to a star tree (if $\epsilon(I) = 0$). Notice also that each bad edge of T belongs to a single island, while each good edge belongs to exactly two islands of T if it is an internal edge of T, or to a single island if it is a terminal edge of T.

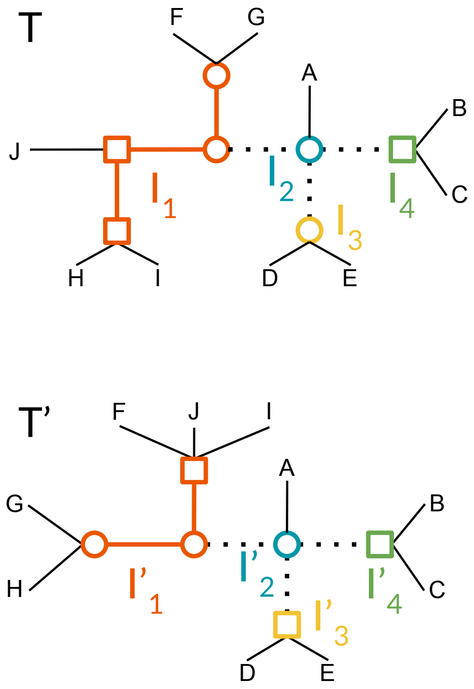

Figure 3: Two trees T and T' on $\mathcal{T}_{\mathcal{L}}$ for $\mathcal{L} = \{A, B, C, D, E, F, I, J\}$, with a binary labeling of internal nodes (squares and circles). Dotted lines represent good internal edges, solid lines represent bad edges and thin lines represent terminal edges (which are good edges). This representation highlights the subdivision of the two trees into the island pairs $\mathcal{I}_{(T,T')} = \{(I_1, I'_1), (I_2, I'_2), (I_3, I'_3), (I_4, I'_4)\}$.Notice that each dotted line is a terminal edge of its two adjacent islands

Finally, the following lemma (lemma 3 from [5]) shows that there is a one-to-one correspondence between the islands of T and those of T'.

Lemma 4. *Let I be an island 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 I' of T', containing all e'_i edges as terminal edges, is unique. Moreover, it is an island of T'.*

Proof. As $\cup_i Y_i = \mathcal{L}$, $\{e'_i\}_{1\leq i\leq k}$ are the only terminal edges of any subtree I' 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 I' is not an island. 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 I. This contradicts the fact that I is an island of T. Finally, as all terminal edges of I' are good edges of T', it follows that I' is an island of T'. □

For any island I of T, let I' be the corresponding island of T'. We call (I, I') an *island pair* of (T, T'). See Figure 3 for an example.

Now, let $\mathcal{I}_{(T,T')} = \{(I_1, I'_1), (I_2, I'_2), \cdots, (I_n, I'_n)\}$ be the set of island pairs of (T, T'). For $1 \leq i \leq n$, let $\mathcal{P}_i$ be a shortest path of labeled node edit operations transforming I_i into I'_i. Then the path $\mathcal{P}$ obtained by performing consecutively $\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n$ (that we represent later as $\mathcal{P}_1.\mathcal{P}_2.\cdots.\mathcal{P}_n$) clearly transforms T into T'. Therefore we have

$$LRF(T, T') \leq \sum_{i=1}^{n} LRF(I_i, I'_i)$$

As described in [5], one major issue with ELRF is that good edge contractions may not be avoided in a shortest path of edit operations transforming T into T', resulting in island merging. In other words, treating island pairs separately may not result in an optimal scenario of edit operations under $ELRF$, preventing the above inequality from being an equality. Interestingly, the equality holds for the LRF distance, as we show in the next section.

3.2 Computing the LRF distance on islands

We require an additional definition. Two trees I and I' of an island pair are said to *share a common label* $l \in \Lambda$ if there exist $x \in V(I)$ and $x' \in V(I')$ such that $\lambda(x) = \lambda(x') = l$. If I and I' do not share any common label, then (I, I') is called a *label-disjoint* island pair. For example, the pair (I_3, I'_3) in Figure 3 or the pair (I, I') in Figure 4 are label-disjoint.

Now let (I, I') be an island pair. Transforming I into I' can be done by reducing I into a star tree by performing a sequence of node deletions (if any, i.e. if I is not already a star tree), and then raising the star tree by inserting the required nodes to reach I'. Only the unique node not deleted during the first step might require a label substitution; for all inserted nodes, the label can be chosen to match that of I'. However, if I and I' share a common label l among their internal nodes, then the deletions can be done in a way such that the surviving node x of I is one with label $\lambda(x) = l$, thus avoiding the need for any substitution. The number of required operations is thus $\epsilon(I)$ deletions, followed by zero or one substitution, followed by $\epsilon(I')$ insertions. Alternatively, the problem can be seen as one of reducing the two trees

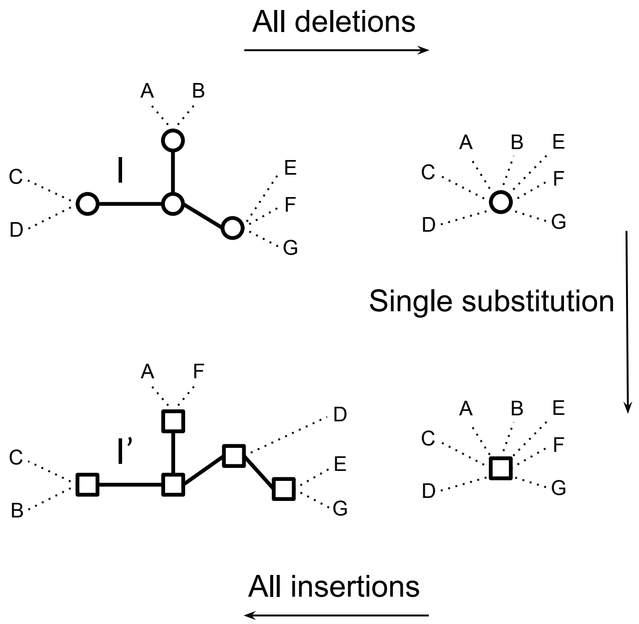

Figure 4: An optimal sequence of edit operations for the island pair (I, I').

into star trees by performing $\epsilon(I) + \epsilon(I')$ deletions, in a way reducing the two islands into two star trees sharing the same label, if possible. Figure 4 depicts an example of such tree editing for a label-disjoint island pair.

The following lemmas show that the sequential way of doing described above is optimal.

Lemma 5. *Let (I, I') be an element of $\mathcal{I}_{(T,T')}$. Then:*

- *If I and I' share a common label, then $LRF(I, I') = \epsilon(I) + \epsilon(I')$.*
- *Otherwise $LRF(I, I') = \epsilon(I) + \epsilon(I') + 1$.*

Proof. The scenario depicted above for transforming I into I' clearly requires $\epsilon(I) + \epsilon(I')$ node insertions and deletions, and an additional node label substitution in case I ans I' are label-disjoint. We can conclude that $LRF(I, I') \leq \epsilon(I) + \epsilon(I')$ if I and I' share a common label and $LRF(I, I') \leq \epsilon(I) + \epsilon(I') + 1$, if I and I' are label-disjoint.

On the other hand, as all the edges of I are bad edges, they should be all removed, before reinserting those of I'. Now, since an edit operation can remove or insert at most one edge, and the only operations removing an edge are node removal or node insertion, we clearly require at least $\epsilon(I) + \epsilon(I')$ node removals and insertions to transform the unlabeled form of the tree I into the unlabeled form of I'. Furthermore, as deletions do not affect star nodes, at least one node in I should survive (i.e. not be affected by a node deletion). Thus, if the two trees are label-disjoint, then at least one node label substitution is required. We can then conclude that $LRF(I, I') \geq \epsilon(I) + \epsilon(I')$ if I and I' share a common label and $LRF(I, I') \geq \epsilon(I) + \epsilon(I') + 1$, if I and I' are label-disjoint, which concludes the proof. □

We have obviously $LRF(T, T') \leq \sum_{(I,I') \in \mathcal{I}_{(T,T')}} LRF(I, I')$. It remains to show that the symmetrical inequality also holds, i.e. we cannot do better by merging islands, and thus pairs of islands can be considered separately. The following lemma states that we can always find a sequence of operations, at each step maintaining or increasing the number of islands, i.e. never merging islands.

For a path $\mathcal{P} = (o_1, o_2, \cdots o_p)$ transforming a tree T into a tree T' and $1 \leq k \leq p$, denote by T_k the tree obtained from T after performing the sub-sequence of operations $\mathcal{P}_k = (o_1, \cdots o_k)$.

Lemma 6. *Let T and T' be two trees of $\mathcal{T}_{\mathcal{L}}$. There is a shortest path $\mathcal{P} = (o_1, o_2, \cdots o_p)$ of edit operations transforming T into T' such that for each k, $2 \leq k \leq p$, $|\mathcal{I}(T_{k-1}, T')| \leq |\mathcal{I}(T_k, T')|$.*

Proof. Let $\mathcal{P} = (o_1, o_2, \cdots o_p)$ be a shortest path transforming T into T', Denote $\epsilon(T_k, T') = \sum_{(I,I') \in \mathcal{I}(T_k,T')} \epsilon(I) + \epsilon(I')$, and $\xi(T_k, T')$ the number of label-disjoints pairs of $\mathcal{I}(T_k, T')$.

Assume $\mathcal{P}$ contains an operation reducing the number of islands of T_{i-1}, and let o_i be the last operation of that form, i.e. $|\mathcal{I}(T_{i-1}, T')| > |\mathcal{I}(T_i, T')|$. Such an operation can only be a deletion $Del(T_{i-1}, x, y)$ where $e = \{x, y\}$ is a good edge, thus merging the two islands I_x, I_y containing this good edge.

As, by assumption, o_i is the last operation merging two islands, at that point each pair of islands is treated separately, and we deduce from the fact that $\mathcal{P}$ is a shortest path that $LRF(T_i, T') = \sum_{(I,I') \in \mathcal{I}(T_i,T')} LRF(I, I')$, and thus $LRF(T_{i-1}, T') = 1 + \sum_{(I,I') \in \mathcal{I}(T_i,T')} LRF(I, I')$. Then, from Lemma 5, $LRF(T_{i-1}, T') = 1 + \epsilon(T_i, T') + \xi(T_i, T')$.

On the other hand, there is a path from T_{i-1} to T' of size $c(T_{i-1}, T') = \epsilon(T_{i-1}, T') + \xi(T_{i-1}, T')$.

As o_i is a deletion of a good edge $e = \{x, y\}$, it destroys the bipartition defined by this edge in T_{i-1}, consequently the corresponding edge in T' becomes a bad edge. Therefore $\epsilon(T_{i-1}, T') = \epsilon(T_i, T') - 1$.

On the other hand, let $\delta = \xi(T_i, T') - \xi(T_{i-1}, T')$ be the difference between the number of label-disjoint pairs of islands after performing the operation o_i merging two pairs of islands (I_1, I'_1) and (I_2, I'_2).

- If both pairs (I_1, I'_1) and (I_2, I'_2) share a common label, then the merged pair also shares a common label, and thus $\delta = 0$;

- If both pairs (I_1, I'_1) and (I_2, I'_2) are label-disjoint, then after the merging the resulting pair of islands may or may not share a common label and thus $-2 \leq \delta \leq -1$;

- If only one of the two pairs (I_1, I'_1) and (I_2, I'_2) share a common label, then after the merging, the resulting pair of islands may or may not share a common label and thus $-1 \leq \delta \leq 0$;

Therefore, in all cases we have $\xi(T_i, T') \leq \xi(T_{i-1}, T') \leq \xi(T_i, T') + 2$.

Recall $c(T_{i-1}, T') = \epsilon(T_{i-1}, T') + \xi(T_{i-1}, T') = \epsilon(T_i, T') - 1 + \xi(T_{i-1}, T')$. Thus $c(T_{i-1}, T') \leq \epsilon(T_i, T') - 1 + \xi(T_i, T') + 2 = \epsilon(T_i, T') + \xi(T_i, T') + 1$ $= LRF(T_{i-1}, T')$. As $LRF(T_{i-1}, T')$ is the size of the shortest path from T_{i-1} to T', we should have $c(T_{i-1}, T') = LRF(T_{i-1}, T')$.

Therefore, replacing the sequence of operations $(o_i, \cdots o_p)$ on T_{i-1} by a sequence of operations solving each pair if islands separately leads to the same number of operations.

□

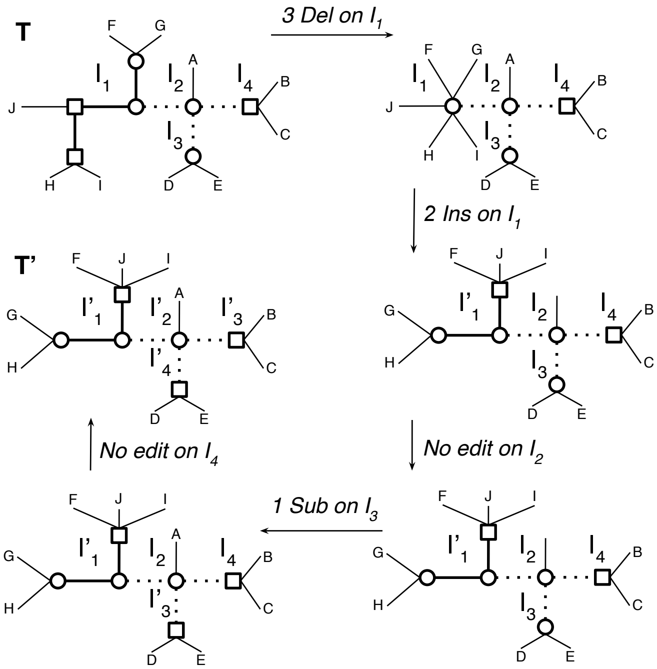

Figure 5: A path $\mathcal{P}$ transforming T into T' of the form $\mathcal{P}_1.\mathcal{P}_2.\mathcal{P}_3.\mathcal{P}_4$, each $\mathcal{P}_i$ being a shortest path for the island pair (I_i, I'_i). Here $|\mathcal{P}_1| = 6$, $|\mathcal{P}_2| = 0$, $|\mathcal{P}_3| = 1$, and $|\mathcal{P}_4| = 0$.

We are now ready to prove the equality leading to the efficient computation of the LRF distance of two trees (see Figure 5 for an example).

Theorem 1. *Let $\mathcal{I}_{(T,T')} = \{(I_1, I'_1), (I_2, I'_2), \cdots, (I_n, I'_n)\}$ be the island pairs of T and T'. Then*

$$LRF(T,T') = \sum_{i=1}^{n} LRF(I_i, I'_i)$$

Proof. Let $\mathcal{P}$ be a shortest path transforming T into T' verifying the condition of Lemma 6, i.e. not involving any removal of good edges. As islands can only share good edges, and good edges are never removed by any operation of $\mathcal{P}$, islands are never merged during the process of transforming T into T', and thus can be treated separately. Let $\mathcal{P}_i$, $1 \leq i \leq n$, be the subpath of edit operations transforming I_i into I'_i. Each $\mathcal{P}_i$ should be a shortest path from I_i to I'_i as otherwise it can be replaced by a shortest path, contradicting the fact that $\mathcal{P}$ is a shortest path.

□

The next result directly follows from Lemma 5 and Theorem 1.

Corollary 1. *Let $\mathcal{I}_{(T,T')} = \{(I_1, I'_1), (I_2, I'_2), \cdots, (I_n, I'_n)\}$ be the island pairs of T and T' and δ be the number of label-disjoint pairs. Then*

$$LRF(T,T') = \sum_{i=1}^{n} (\epsilon(I_i) + \epsilon(I'_i)) \; + \; \delta$$

4 Algorithm

We present our algorithm for computing the LRF distance (Algorithm 1). The input is a pair of trees T_1, T_2 of $\mathcal{T}_{\mathcal{L}}$. We show that $LRF(T_1, T_2)$ can be computed in time $\mathcal{O}(n)$, where $n = |\mathcal{L}|$.

4.1 The $LRF()$ function

We start with the identification of good edges. Lines 1 and 2 of Algorithm 1 retrieve the non-trivial bipartitions for each input tree and Line 3 intersects the obtained bipartitions of T_1 and T_2 to generate the set of good edges shared by the two input trees. This is the same procedure as to compute the conventional (unlabeled) Robinson-Foulds distance, so we do not detail it here. It is sufficient to know that it can be done in time $\mathcal{O}(n)$ [9].

Next the algorithm identifies and characterises the islands of T_1 and T_2 (lines 4 and 5). This is performed using an auxiliary function $getIslands()$ (Algorithm 2), which we describe in detail below. As we shall see, it runs in $\mathcal{O}(n)$.

Next, we process the matching islands of T_1 and T_2 by iterating over the good edges (of which there are $\mathcal{O}(n)$). We retrieve for a good edge the two islands to which it belongs in T_1 (line 8) and in T_2 (line 9). This is achieved in constant time using bipartition-to-island-pair mappings obtained during the tree traversal of $getIslands()$ below.

For each of the matching island of T_1 and T_2 (line 10), the algorithm checks whether the pair has already been visited in a previous iteration of the loop (the same island pair can be visited from multiple good edges). If not, the current distance is updated by adding the number of bad edges in each island. Since these sizes are also pre-computed by $getIslands()$, this operation is in constant time as well.

The iteration over all good edges ends with lines 13-14, which account for a potentially required single substitution between corresponding islands, in case they have no label in common (i.e. they form a label-disjoint island pair). These operations can also be performed in constant time, giving an overall $\mathcal{O}(n)$ runtime for the for-loop.

Finally, lines 16-19 are needed to handle the special case where there is no good edge between T_1 and T_2. In such a case, there is only one island per tree, which is matching.

4.2 The $getIsland()$ function

We now detail the auxiliary function $getIslands()$ (Algorithm 2). Recall that its goal is to identify the islands of an input tree, given a list of good edges. This is achieved through a single traversal of the tree in pre-order (we assume that the tree is arbitrarily rooted, and that the dummy root node has no label). In doing so, we identify the islands, which are separated by good edges, and keep track of (i) the set of labels found in each island (array $islLabels$); (ii) the number of bad edge in each island (array $islSizes$;, (iii) the pair of islands associated with each bipartition ($bipart2isl$). These three data structures are initialised in lines 1-3. Note that the initial island is initialised to -1 because it will be incremented to 0 at the first step of the traversal.

Lines 4-20 define the recursive function used to traverse the tree. Because each good edge belongs to exactly two islands (Sect. 3), good edges can be used to identify the transition between two islands. By contrast, adjacent bad edges are by definition part of the same island. In our traversal, we thus check whether a particular node is attached to the previous island by a good edge (lines 7-15) or a bad edge (lines 16-20). If it is the former, we have just transitioned to a new island, and thus append new elements to the $islLabels$ (line 8) and $islSizes$ arrays (line 9).

Furthermore, we update the bipartition-to-island table with references to the two islands which are deliminated by the good edge. Since by definition a good edge induces the same bipartition in T_1 and T_2, the bipartition bitmask (a binary vector of the length n with 1 for all leaves present in the clade attached to the good edge) should either be the same for the good edge in T_1 and T_2, or bit-wise complementary if the rooting between T_1 and T_2 is on different sides of the good edge (as the tree data structure is arbitrarily rooted). We store the associated islands using both bitmasks, ensuring that the island which on same side as the root is listed first (lines 12-13).

If we encounter a node which is attached to the previous island by a bad edge, then it is still part of it, so we just update the set of leaf of the previous island (line 17), and increment its size counter by one (line 18).

All operations performed at each internal node are constant time, and the number of internal nodes is $\mathcal{O}(n)$, so the time complexity of the tree traversal is done in time $\mathcal{O}(n)$.

Algorithm 1 LRF(T_1, T_2)

```
1:  $bipartitions1 = getBiparitions(T_1)$
2:  $bipartitions2 = getBiparitions(T_2)$
3:  $goodEdges = bipartitions_1 \cap bipartitions_2$
4:  $islLabels1, islSizes1, bipart2isl1 = getIslands(T_1, goodEdges)$
5:  $islLabels2, islSizes2, bipart2isl2 = getIslands(T_2, goodEdges)$
6:  $distance = 0$
7:  for $i \in goodEdges$:
8:      $i11, i12 = bipart2isl1[i.bitmask]$
9:      $i21, i22 = bipart2isl2[i.bitmask]$
10:     for $(j1, j2) \in [(i11, i21), (i12, i22)]$:
11:         if $j1.visited == False$:
12:             $distance{+}{=}islSizes1[j1] + islSizes2[j2]$
13:             if $islLabels1[j1] \cap islLabels2[j2] == \emptyset$:
14:                 $distance{+}{=}1$
15:             $j1.visited = True$
16: if $goodEdges == \emptyset$ :
17:     $distance{+}{=}\ islSizes1[1] + islSizes2[1]$
18:     if $islLabels1[1] \cap islLabels2[1] == \emptyset$:
19:         $distance\ {+}{=}\ 1$
20: return $distance$
```

We provide an open source implementation of *LRF* in Python as part of the pyLabeledRF package (https://github.com/DessimozLab/pylabeledrf).

Algorithm 2 getIslands($T, goodEdges$)

```
$islLabels = [\{\}]$
$islSizes = [-1]$
$bipart2isl = NewHashTable()$
function traverseT($t, oldIsland$):
    if $t$ is a leaf:
        return
    else if $t.rootedge.bipartition \in goodEdges$:
        $islLabels.append(\{t.label\})$
        $islSizes.append(1)$
        $newIsland = islSizes.length$
        $mask = t.rootedge.bipartition.bitmask$
        $bipart2isl[mask] = (oldIsland, newIsland)$
        $bipart2isl[BitComplement(mask)] = (newIsland, oldIsland)$
        for $c \in t.children$:
            return $traverseT(c, newIsland)$
    else:
        $islLabels[oldIsland].add(t.label)$
        $islSize[oldIsland]{+}{=}1$
        for $c \in t.children$:
            return $traverseT(c, oldIsland)$
$traverseT(T, 1)$
return $islLabels, islSizes, bipart2isl$
```

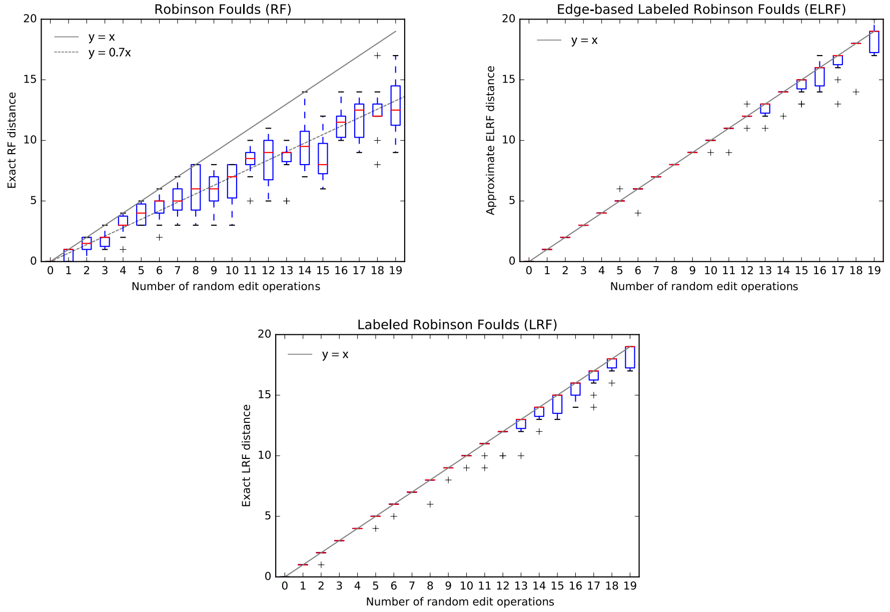

Figure 6: Empirical comparisons of the distance inferred for an increasing number of random edit operations (node insertion, deletion, substitution) on the NOX4 gene tree (182 leaves), using the classical RF distance (top), the ELRF approximation ([5]; middle), and the LRF exact distance (bottom).

5 Experimental results

To illustrate the usefulness of LRF, we performed two experiments. First, we compared LRF with RF and $ELRF$ on a labeled gene tree with random edits. Second, we used LRF to tackle an open question in orthology inference: does labeled gene tree inference benefit from denser taxon sampling?

5.1 Empirical comparison of LRF with RF and $ELRF$

To get a first sense of LRF's ability to measure the actual number of edits between two trees, we performed a simulation study alongside RF and $ELRF$. We retrieved the labeled tree associated with human gene NOX4 from Ensembl release 99 [26], containing 182 genes, including speciation and duplication nodes. Next, we introduced a varying number of random edits, with 10 replicates, as follows: with probability 0.3, the label of one random internal node was substituted (from a speciation label into a duplication one or vice versa); the rest of the probability mass function was evenly distributed among all internal edges (each implying a potential node deletion) and all nodes of degree > 3 (each providing the opportunity of a potential node insertion). For $ELRF$, consistent with its underlying model, we added the requirement that edge removal only affect edges with adjacent nodes with the same label.

For each of RF, LRF and $ELRF$, we provide the distance as a function of the number of random edits (Fig. 6). As expected, the conventional RF distance returns the smallest values because it ignores labels; it however tracks quite well the expected number of node insertion and/or removal (dashed line). The two labeled RF alternatives performed similarly, but the heuristic for $ELRF$ occasionally exceeded the true number of edit operations — a shortcoming that we do not have with LRF, as we have an exact algorithm for this distance. Both labeled RF variants tracked better the actual number of changes, until around 13 edits for LRF or $ELRF$, after which the minimum edit path starts to be often shorter than the actual sequence of random edits.

5.2 The effect of denser taxon sampling on labeled gene tree inference

We used LRF to assess the effect of species sampling for the purpose of labeled gene tree reconstruction. Consider the problem of reconstructing a labeled tree corresponding to homologous genes from 10 species. Our question is: is it better to infer and label the tree using these 10 species alone, or is it better to use more species to infer and label the tree, and then prune the resulting tree to only contain the leaves corresponding to the original 10 species? While denser taxon sampling is known to improve unlabeled phylogenetic inference [19], we are not aware of any previous study on labeled gene tree inference.

First, using ALF [8], we simulated the evolution of the genomes of 100 extant species from a common ancestor genome containing 100 genes (*Parameters*: root genome with 100 genes of 432 nucleic acids each; species tree sampled from a birth-death model with default parameters; sequences evolved using the WAG model, with Zipfian gap distribution; duplication and loss events rate of 0.001). In the simulation, genes can mutate, be duplicated or lost. All the genes in the extant species can thus be traced back to one of these 100 ancestral genes and be assigned to the corresponding gene family. The 100 true gene trees, including speciation and duplication labels, are known from the simulation. However, in our run, one tree ended up containing only two genes (due to losses on early branches) and was thus excluded from the rest of the analysis.

To evaluate the inference process, among the 100 species, we randomly selected nested groups of 10, 20, 30, 40, 50, 60, 70, 80 and 90 species. We considered the 10 species in the first group as the species of interest. All other species were used to potentially improve the reconstruction of the gene trees for the first 10 genomes. Then, for each group, we aligned protein

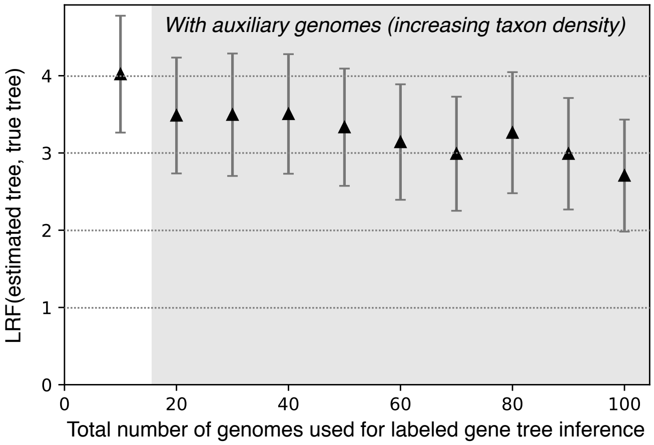

Figure 7: Denser taxon sampling decreases labeled tree estimation error: labeled gene trees reconstructed with an increasing number of auxiliary genomes (i.e. obtained by including the additional genomes during tree inference and labeling, followed by pruning) have a smaller LRF distance to the true trees. Error bars depict 95% confidence intervals around the mean.

sequences translated from homologous genes using MAFFT L-INS-i [14], inferred phylogenetic trees from the alignments using FastTree [21], and annotated their nodes using the species overlap algorithm [24] as implemented in the ETE3 python library [12]. Finally, we pruned both the inferred gene trees and the true trees to include only proteins corresponding to the 10 species of interest.

We used LRF to assess the distance between the estimated and true labeled trees, for the various number of auxiliary genomes considered. For each scenario, we computed the mean LRF distance over all gene trees (Fig. 7). The mean error (expressed in LRF distance) decreases as the number of auxiliary species increases. This simple simulation study suggests that denser species sampling improves labeled gene tree inference.

6 Discussion and Conclusion

The LRF distance introduced here overcomes the major drawback of $ELRF$, namely the lack of an exact polynomial algorithm for the latter. Indeed, with $ELRF$, minimal edit paths can require contracting "good" edges, i.e., edges present in the two trees [5]. By contrast, with LRF, we demonstrated that there is always a minimal path which does not contract good edges. Better yet, we proved that LRF can be computed exactly in linear time. The new formulation also maintains other desirable properties: being a metric, even for an arbitrary number of label types, and reducing to the conventional

Robinson Foulds distance in the presence of trees with only one type of label.

Our experimental results provide a relationship between the number of random edits and the computed edit distances. At first sight, it may seem surprising that in a tree of 182 leaves, the minimum edit path under *LRF* or *ELRF* already starts underestimating the actual number of random edit operations after around 13 operations. However, this can be explained by the "birthday paradox" [1]: to be able to reconstruct the actual edit path, no two random edits should affect the same node. Yet the odds of having, among 13 random edits, at least two edits affecting the same internal node (among 179) is in fact substantial — approximately 36% in our case — just like the odds of having two people with the same birthday in a given group is higher than what most people intuit.

It has to be noted that *LRF* has the same limitations as *RF* regarding lack of robustness and skewed distribution. Moreover, like *RF* and *ELRF*, the main limitation of *LRF* is the lack of biological realism. For one thing, there is no justification to assign equal weight to the three kinds of edits in all circumstances. For instance, it is typically highly implausible to introduce a speciation node at the root of a subtree containing multiple copies of a gene in the same species. However, *LRF* complement analyses performed using more realistic models are either unavailable or too onerous to compute. In particular, the ability of *LRF* to support an arbitrary number of labels makes it applicable to gene trees containing more than just speciations and duplications, such as horizontal gene transfers or gene conversion events.

Finally, *LRF* constitutes a clear improvement over *RF* in the context of gene tree benchmarking, where trees inferred by various reconciliation models are compared using a distance measure [3, 18]. Such an application was illustrated in the simulation study of the previous section, in which we observed that denser taxon sampling improved labeled tree inference computed using the widely used species overlap method. More work will be needed to assess the generality of this result.

Acknowledgements

We warmly thank Céline Scornavacca, Barbara Holland, Gabriel Cardona, Jean-Baka Domelevo Entfellner, and an anonymous reviewer for their detailed and constructive peer-reviews. The work was supported by Swiss National Science Foundation (SNSF) Professorship grant 183723 (to CD), by the Natural Sciences and Engineering Research Council of Canada (NSERC, to NEM), and Fonds de Recherche Nature et Technologies of Quebec (FRQNT, to NEM).

References

[1] Morton Abramson and W O J Moser. More Birthday Surprises. *The American mathematical monthly: the official journal of the Mathematical Association of America*, 77(8):856–858, 1970.

[2] Benjamin L Allen and Mike Steel. Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of combinatorics*, 5(1):1–15, 2001.

[3] Adrian M Altenhoff, Brigitte Boeckmann, Salvador Capella-Gutierrez, Daniel A Dalquen, Todd DeLuca, Kristoffer Forslund, Jaime Huerta-Cepas, Benjamin Linard, Cécile Pereira, Leszek P Pryszcz, et al. Standardized benchmarking in the quest for orthologs. *Nature methods*, 13(5):425–430, 2016.

[4] Bastien Boussau and Celine Scornavacca. Reconciling gene trees with species trees. *Phylogenetics in the Genomic Era*, p. 3.2:1–3.2:23, 2020.

[5] Samuel Briand, Christophe Dessimoz, Nadia El-Mabrouk, Manuel Lafond, and Gabriala Lobinska. A generalized Robinson-Foulds distance for labeled trees. *BMC Genomics*, 2020.

[6] Gabriel Cardona, Mercè Llabrés, Francesc Rosselló, and Gabriel Valiente. Nodal distances for rooted phylogenetic trees. *Journal of mathematical biology*, 61(2):253–276, 2010.

[7] Douglas E Critchlow, Dennis K Pearl, and Chunlin Qian. The triples distance for rooted bifurcating phylogenetic trees. *Systematic Biology*, 45(3):323–334, 1996.

[8] Daniel A Dalquen, Maria Anisimova, Gaston H Gonnet, and Christophe Dessimoz. ALF–a simulation framework for genome evolution. *Molecular biology and evolution*, 29(4):1115–1123, April 2012.

[9] William HE Day. Optimal algorithms for comparing trees with labeled leaves. *Journal of classification*, 2(1):7–28, 1985.

[10] George F Estabrook, FR McMorris, and Christopher A Meacham. Comparison of undirected phylogenetic trees based on subtrees of four evolutionary units. *Systematic Zoology*, 34(2):193–200, 1985.

[11] Glenn Hickey, Frank Dehne, Andrew Rau-Chaplin, and Christian Blouin. Spr distance computation for unrooted trees. *Evolutionary Bioinformatics*, 4:EBO–S419, 2008.

[12] Jaime Huerta-Cepas, François Serra, and Peer Bork. ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. *Molecular biology and evolution*, 33(6):1635–1638, June 2016.

[13] Bhaskar DasGupta Xin He Tao Jiang, Ming Li, John Tromp, and Louxin Zhang. 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*, volume 55, p. 125. American Mathematical Soc., 2000.

[14] Kazutaka Katoh and Daron M Standley. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4):772–780, April 2013.

[15] Yu Lin, Vaibhav Rajan, and Bernard ME Moret. A metric for phylogenetic trees based on matching. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(4):1014–1022, 2012.

[16] S. Mittal and G. Munjal. Tree mining and tree validation metrics: A review. *IOSR: Journal of Computer Engineering*, pp. 31-36, 2015.

[17] Jucheol Moon and Oliver Eulenstein. Cluster matching distance for rooted phylogenetic trees. In *International Symposium on Bioinformatics Research and Applications*, pp. 321–332. Springer, 2018.

[18] Benoit Morel, Alexey M. Kozlov, Alexandros Stamatakis, and Gergely J. Szöllősi. Generax: A tool for species tree-aware maximum likelihood based gene tree inference under gene duplication, transfer, and loss. *bioRxiv*, 2019.

[19] Ahmed Ragab Nabhan and Indra Neil Sarkar. The impact of taxon sampling on phylogenetic inference: a review of two decades of controversy. *Briefings in bioinformatics*, March 2011.

[20] Nicholas D Pattengale, Eric J Gottlieb, and Bernard ME Moret. Efficiently computing the robinson-foulds metric. *Journal of Computational Biology*, 14(6):724–735, 2007.

[21] Morgan N Price, Paramvir S Dehal, and Adam P Arkin. FastTree 2–approximately maximum-likelihood trees for large alignments. *PloS one*, 5(3):e9490, 2010.

[22] David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. *Mathematical biosciences*, 53(1-2):131–147, 1981.

[23] Stefan Schwarz, Mateusz Pawlik, and Nikolaus Augsten. A new perspective on the tree edit distance. In *International Conference on Similarity Search and Applications*, pp. 156–170. Springer, 2017.

[24] René T J M van der Heijden, Berend Snel, Vera van Noort, and Martijn a Huynen. Orthology prediction at scalable resolution by phylogenetic tree analysis. *BMC bioinformatics*, 8:83, January 2007.

[25] A.J. Vilella, J. Severin, A. Ureta-Vidal, L. Heng, R. Durbin, and E. Birney. EnsemblCompara gene trees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome Research*, 19:327-335, 2009.

[26] Andrew D Yates, Premanand Achuthan, Wasiu Akanni, James Allen, Jamie Allen, Jorge Alvarez-Jarreta, M Ridwan Amode, Irina M Armean, Andrey G Azov, Ruth Bennett, et al. Ensembl 2020. *Nucleic acids research*, 48(D1):D682–D688, 2020.

[27] Kaizhong Zhang. A new editing based distance between unordered labeled trees. In *Annual Symposium on Combinatorial Pattern Matching*, pp. 254–265. Springer, 1993.

[28] Kaizhong Zhang. A constrained edit distance between unordered labeled trees. *Algorithmica*, 15(3):205–222, 1996.

[29] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262, 1989.

[30] Kaizhong Zhang, Rick Statman, and Dennis Shasha. On the editing distance between unordered labeled trees. *Information processing letters*, 42(3):133–139, 1992.