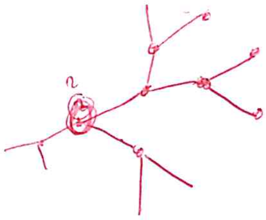


The existence of a variety of different phylogenetic inference methods, leading to different, potentially inconsistent trees for the same dataset, brings forward the need for appropriate tools for comparing them. Although comparing labeled gene trees remains a largely unexplored field, a large variety 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 length, bipartitions or quartets for unrooted trees, clades or triplets for rooted trees (Cardona *et al.*, 2010; Estabrook *et al.*, 1985; Critchlow *et al.*, 1996), 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 (Jiang *et al.*, 2000; Hickey *et al.*, 2008; Allen and Steel, 2001). While the latter methods are NP-hard (Lin *et al.*, 2012), 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 (Mittal and Munjal, 2015), can be computed in linear (Day, 1985), and even sublinear time (Pattengale *et al.*, 2007).

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. For such trees, the standard Tree Edit Distance (TED) (Zhang and Shasha, 1989), 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 less constrained version of the problem on unordered labeled trees is NP-complete (Zhang *et al.*, 1992), most variants are solvable in polynomial time (Zhang, 1993, 1996; Schwarz *et al.*, 2017).

The metric we developed in Briand *et al.* (2020), 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 (Lin *et al.*, 2012; Moon and Eulenstein, 2018) to address the distance 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 (Robinson and Foulds, 1981). In Briand *et al.* (2020), this definition of the RF distance was extended to node labeled trees 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, even for binary node labels. As a result, only a heuristic could be proposed to compute it.

(approx.)

In this paper, we explore a different extension of RF to node labeled trees, directly derived from TED (Zhang and Shasha, 1989), 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

← nowhere here is said that we don't accept internal nodes other than the root with degree 2

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. 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~~. Now an internal node x is *binary* if x is of degree ~~3 and $r(T)$~~ is *binary* if $r(T)$ is of degree 2.

rephrase.

rephrase. Any node is binary if it has degree 3, except the root.

Let x and y be two nodes of a rooted tree T; y is a *descendant* of x if y is on the path from x to a leaf (possibly y itself) of T. If T is rooted, we say that y is a *child* of x if $e = \{x, y\}$ is an ~~edge~~ element of $E(T)$ with y being a descendant of x. If T is unrooted, we call the set $\{y : \{x, y\} \in E(T)\}$ the set of ~~children~~ neighbours of x. For a rooted or an unrooted tree T, we denote by $Ch(x)$ the set of children of an internal node x of T.

rephrase.

r(T) not on the path

allows not connecting

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 $V(S)$. 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 x and all its descendants ~~all the descendants of x~~. We call $L(T_x)$ the *clade of* x.

The *bipartition* of a 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 + symmetric difference between the bipartitions of the two trees. More precisely,

+size of the

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

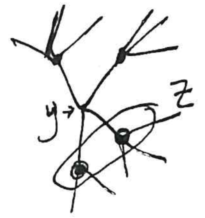
As recalled in Briand *et al.* (2020), the RF distance is equivalently defined in terms of an edit distance on edges. However, as for labeled trees an additional substitution operation on node labels will be required, for the sake of standardization, we reformulate the edit operations to operate on nodes rather than on edges.

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

- ***Node deletion:*** *Let x be an internal node of T which is neither the root nor a star node, and let y be the parent of x if T is rooted, or y be a given child of x which is not a leaf if T is unrooted (such a~~n~~ y exists from the fact that x is not a star node). Deleting x means making the children of x become the children 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 $z \in Ch(x)$, creating the edge $\{y, z\}$ for each $z \in Ch(x) \setminus \{y\}$, and then removing node x.*
- ***Node insertion:*** *Let y be a non-binary internal node of $V(T)$. Inserting x as a child of y entails making x the parent of a subset $Z \subsetneq Ch(y)$ 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 $z_i \in Z$.*

→ cf upper wrt. internal nodes of deg. 2

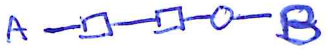
Notice the one-to-one correspondence between operations on nodes and operations on edges. In fact, deleting a node x by an operation



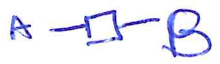
n taxa

binary unrooted : $\begin{cases} 2n-2 \text{ nodes} \\ 2n-3 \text{ edges} \end{cases}$

⚠ can go from



to



but not the other way !

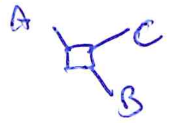
Ⓑ ajout de noeud "au milieu d'une branche" :
+ 1 noeud,
+ 1 branche

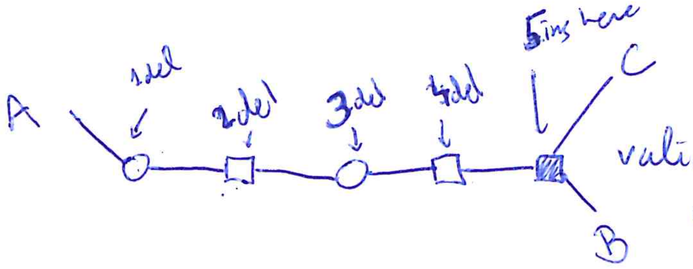
Ⓐ ajout de feuille direct sur un noeud
↳ +1 noeud,
+1 edge

So full path below :
(labels sorted out on insertions)

1 del	5 ins
2 del	6 ins
3 del	7 ins
4 del	8 ins

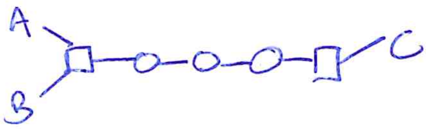
Before 5 ins :





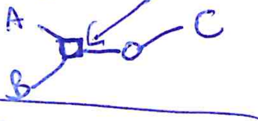
valid tree where no node insertion can take place except on ▨

how to go to



?

After 5 ins, we get e.g.



and then we keep on ins here

distance d :

① non-negativity: $\forall (x,y)$, $d(x,y) \geq 0$

② identity : $d(x,y)=0 \iff x=y$

③ symmetry : $\forall (x,y)$, $d(x,y)=d(y,x)$

④ triangular ineq : $\forall (x,y,z)$ $d(x,z) \leq d(x,y) + d(y,z)$

⓪ definition on all the set of interest (here T_ℓ) :

$\forall (x,y) \in T_\ell$, $d(x,y)$ is defined

The mere existence of a transformation path for any two trees in $\mathcal{T}_\mathcal{L}$ has not been demonstrated by the authors, and indeed such universal existence does not hold when we accept trees with internal degrees 2.

$Del(T,x,y)$ results in deleting 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. This definition is equivalent to the more classical formulation in terms of edit operations on edges. Formally, let T and T' be two trees on the same leafset $\mathcal{L}$. The *Robinson-Foulds* or *Edit distance* (Robinson and Foulds, 1981) ~~$RF(T,T')$~~ between T and T' is the length of a shortest path of node edit operations transforming T into T'. This distance measure, equivalently defined as the symmetrical difference between the bipartitions of the two trees in case of unrooted trees, or the symmetrical difference between the clades of the two trees in case of rooted trees, has been shown to be a metric.

add "as per definition 2"

which ones?

~~In the case of rooted trees~~, the RF distance is defined as the symmetric difference between the clades of the two trees.

size of the

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. The only thing that can make bipartitions and clades differ in number is rooting into a bad edge. In that case, the same bipartition, corresponding to the two edges adjacent to the root, would be counted twice. Given two rooted trees, their RF distance can then be deduced from the RF distance of the "unrooted version" of the two trees by applying Lemma 1 in Briand *et al.* (2020).

In this paper, we focus on unrooted trees, thus avoiding the special case of the root. Therefore, from now on, all trees are considered unrooted.

MUST say here that we only consider as trees those in which all internal nodes have degree ≥ 3, which is a clear limitation

then it's confusing to keep on reading the words "child" and "children" below.

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 works 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 child 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 child of a non-binary node y, and moving $Z \subsetneq Ch(y)$ such that $|Z| \geq 2$, to be the children 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 $\lambda \in \Lambda$.* different from the existing one

These operations are illustrated in Figure 1.

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 $LRF(T,T')$ the length of a shortest path of labeled node edit operations transforming T into T' (or vice versa). The two following lemma 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).

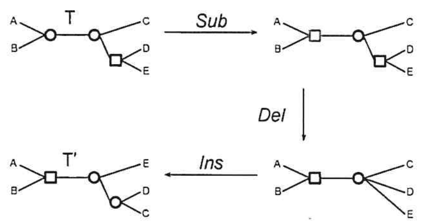


Fig. 1: The transformation of a tree T into a tree T' depicting the three edit operations on nodes. From top to bottom: node label substitution (leading to the red label), node deletion (the parent of D and E) and node insertion (the parent of D and C).

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 length of a shortest path of node edit operations transforming T into T' defines a distance on $\mathcal{T}_\mathcal{L}$.*

In particular, ~~considering~~ minimal paths, the triang. ineq. follows

Proof. The non-negative and identity 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.

NOT QUITE: they don't come with the same constraints: I can go from A–□–○–B to A–□–B, but I can't insert in A–□–B, so can't go back.

Finally, we prove the triangular inequality condition: for three trees T, T' and T'', to transform T into T', we may take any path of edit operations from T to T'', followed by any path of edit operations from T'' to T'. It follows that $LRF(T,T') \leq LRF(T,T'') + LRF(T'',T')$. □

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

Proof. Let l be the only label of Λ. Let $\mathcal{P}$ be a path of node edit operations transforming the unlabeled version of T into the unlabeled version of T', such that $|\mathcal{P}| = RF(T,T')$. Labeling by l each inserted node leads to a corresponding path of labeled node edit operations transforming T into T', and thus $LRF(T,T') \leq RF(T,T')$.

Conversely, Let $\mathcal{P}$ be a path labeled node edit operations transforming T into T', such that $|\mathcal{P}| = LRF(T,T')$. As a single label exists, node substitutions are not defined, and thus $\mathcal{P}$ is restricted to a set of node insertion and deletion transforming T into T', and thus *a fortiori* the unlabeled version of T into the unlabeled version of T'. Thus $RF(T,T') \leq LRF(T,T')$, which completes the proof. □

A previous extension of RF to labeled trees, based on edit operations on edges rather than on nodes, was introduced in Briand *et al.* (2020). 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 child of x and assigning to y the label of x;
- Edge contraction $Cont(T,\{x,y\})$ similar to a node deletion $Del(T,y,x)$ deleting y, but only defined if $\lambda(x) = \lambda(y)$;
- Node flip $Flip(x,\lambda)$ assigning the label λ to x.

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

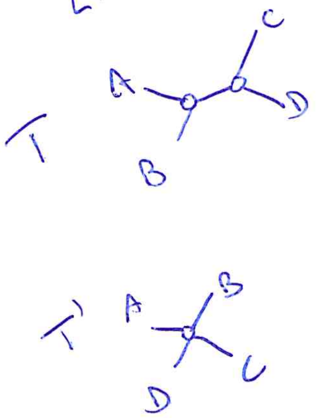
The following lemma makes the link between LRF and $ELRF$.

ok, because RF defined in terms of α and α^{-1} operations

why would that be necessarily possible in the first place? If the existence of such a path has been proven in Briand & al 2020, it must be said here. Again, not possible if one internal node of degree 2.

RF(T,T') = 1

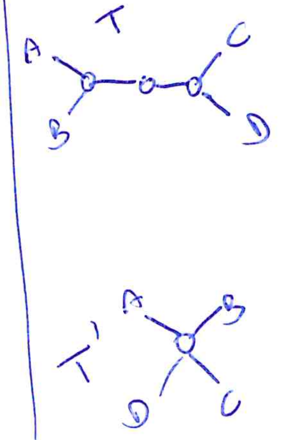
LRF(T,T') = 1



⚠

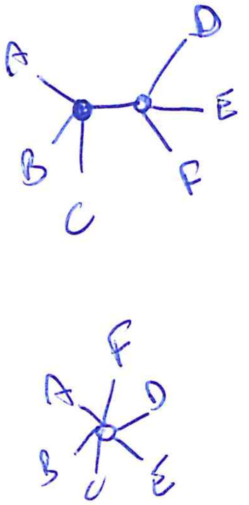
RF(T,T') = 1

LRF(T,T') = 2



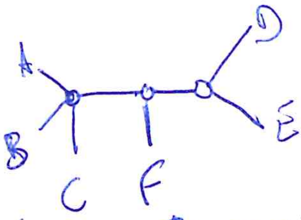
RF(T,T') = 1

LRF(T,T') = 1

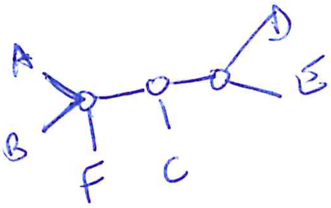


RF(T,T') = 2

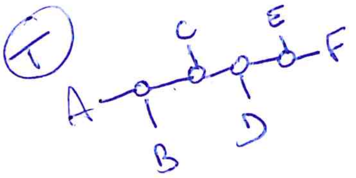
LRF(T,T') = 2



{ABC|DEF, ABCF|DE}

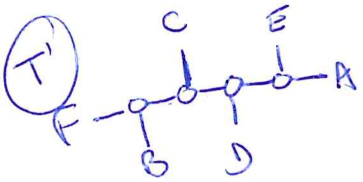


{ABF|CDE, ABCF|DE}



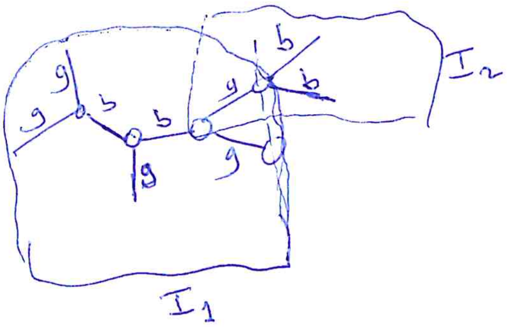
→ after 3 del: star

↓ then 3 ins



RF(T,T') = 6

LRF(T,T') = 6



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 length $|\mathcal{P}'| = |\mathcal{P}| = ELRF(T, T')$ transforming T into T'. And thus $LRF(T, T') \leq ELRF(T, T')$. □

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

3.1 Reduction to Islands

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

While a good edge e of 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 edges may be grouped into pairs of corresponding *islands* (called maximum bad subtrees in Briand *et al.* (2020)), as defined bellow.

the islands "delineate" the problem ✓

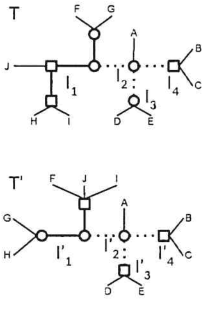
Definition 3 (Islands). *An island of T is a maximum subtree (i.e. a subtree with a maximum number of edges) I of T such that I contains no internal edge which is a good edge of T, and all terminal edges of I are good edges of T. The size of I, denoted* $\epsilon(I)$*, is its number of internal edges.*

defined w.r.t. some alternate tree T'

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 a ~~star~~ tree (if $\epsilon(I) = 0$). Moreover, a tree T is "partitioned" into its set $\{I_1, I_2, \cdots I_n\}$ of islands in the sense that $\{V(I_1), V(I_2), \cdots V(I_n)\}$ is a partition of $V(T)$. 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.

NO. True if one considers internal nodes only.

this is OK



G, H, I ✓

Fig. 2: 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 partition 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 belongs to its two adjacent islands

Finally, the following lemma from Briand *et al.* (2020) 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* (good) *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 in Briand 2020

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 2 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

because they share the same subset of

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

✓

As described in Briand *et al.* (2020), 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.

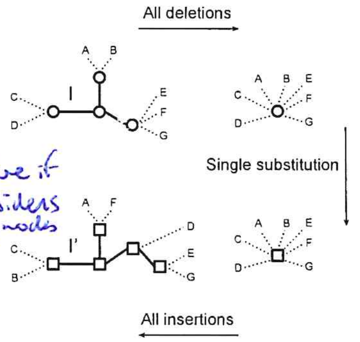


Fig. 3: An optimal sequence of edit operations for the island pair (I, I').

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 2 or the pair (I, I') in Figure 3 are label-disjoint.

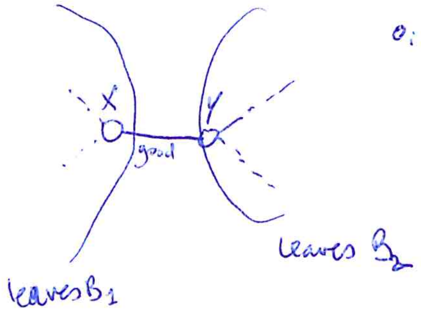
×

×

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

sequence

⚠ Where is the guarantee that at that stage, in my hypothetical T_{k-1}, z and x will be neighbours? That's not straightforward...



o_i deletes x and joins its ~~[illegible]~~ children from B_1 to $y \in B_2$

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 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 3 depicts an example of such tree editing for a label disjoint island pair.

The following lemma shows 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 d 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, 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 no 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~~ star nodes can't be deleted, 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. □

* or adding *

that's more because all internal edges in an island are bad therefore need to be removed

The following lemma shows that good edge deletions can be avoided in a minimal edit path. Consequently, island merging can also be avoided, which will then allow us ~~considering~~ to consider each pair of islands separately.

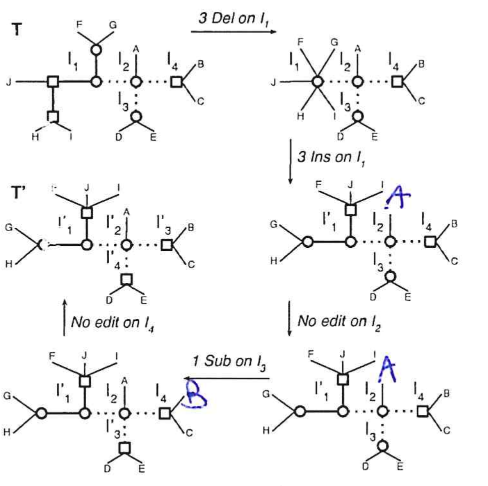


Fig. 4: A path $\mathcal{P}$ transforming T into T' of the form $\mathcal{P}_1.\mathcal{P}_2.\mathcal{P}_3.\mathcal{P}_4$, each $\mathcal{P}_i$ being a shortest path for the island pair (I_i, I'_i). Here $|\mathcal{P}_1| = 6$, $|\mathcal{P}_2| = 0$, $|\mathcal{P}_3| = 1$, and $|\mathcal{P}_4| = 0$.

Lemma 6. *Let T and T' be two trees of $\mathcal{T}_{\mathcal{L}}$. There exists a shortest path of edit operations transforming T into T' involving no deletion of a good edge of T.*

Proof. Let $\mathcal{P} = (o_1, o_2, \cdots o_p)$ be a path transforming T into T'. Let o_i be the leftmost operation of the form $o_i = Del(T, x, y)$ were $e = \{x, y\}$ is a good edge of T. We denote by $\{B_{\cancel{x}1}, B_{\cancel{y}2}\}$ with $B_1 = L(T_x)$ and $B_2 = L(T_y)$ the bipartition of $\mathcal{L}$ corresponding to e. As $\{B_1, B_2\}$ is also a bipartition in T', there should exist a smallest $j > i$ such that the operation o_j is a node insertion operation recreating this bipartition. Let T_{i-1} be the tree obtained after performing the sequence of operations $(o_1, \cdots, o_{i-1})$ on T, and T_j be the tree obtained from T_{i-1} after performing the sequence of operations $\mathcal{P}[i, j] = (o_i, o_{i+1}, \cdots, o_{j-1}, o_j)$. Now let $\mathcal{P}'[i, j] = (o'_{i+1}, \cdots, o'_{j-1})$ be the sequence of operations obtained from $\mathcal{P}[i, j]$ as follows: (1) Remove the two operations o_i and o_j; (2) For each k, $i + 1 \leq k \leq j - 1$, if o_k does not affect node y or if it is a node substitution, o'_k is simply o_k; (3) if $o_k = Del(T, z, y)$, then replace it by the operation $o'_k = Del(T, z, x)$ if $z \in B_1$ or by the operation $o'_k = Del(T, z, y)$ if $z \in B_2$; (4) if $o_k = Del(T, y, z)$, then replace it by the operation $o'_k = Del(T, x, z)$ if $z \in B_1$ and rename z as x, or replace it by the operation $o'_k = Del(T, y, z)$ if $z \in B_2$ and rename z as y. This sequence of operations then leads to the tree T'_j, which is the same as T_j except possibly the two labels of x and y, which can be corrected by at most two additional substitutions. Therefore, we can substitute the subpath $\mathcal{P}[i, j]$ by a subpath of at most the same number of operations that do not involve deleting the good edge e.

means keeping o'_k OK

that's P_k

T_{i-1} : before deleting good edge

T_j : after recreating that good edge

T_{k-1} (also in all other occurrences nearby)

B_1 is an "alphabet" subset of $\mathcal{L}$ but not a set of vertices

a tr

It suffices then to proceed in the same way with the next leftmost good edge deletion of $\mathcal{P}$, and so on, until no good edge deletion remains. □

We are now ready to prove the equality leading to the efficient computation of the LRF distance of two trees (see Figure 4 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 deletion of good edges. As islands can only share good edges, and good edges are never deleted by any operation of $\mathcal{P}$, islands are never merged during the process of transforming T into T', and thus $\mathcal{P}$ can be reordered in the form $\mathcal{P}_1.\mathcal{P}_2. \cdots .\mathcal{P}_n$ where each $\mathcal{P}_i$, $1 \leq i \leq n$, is a path of edit operations transforming I_i into I'_i. Each $\mathcal{P}_i$ ~~should be~~ is a shortest path from I_i to I'_i as otherwise it ~~can~~ could 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 island 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 at a logical level (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 $n = |\mathcal{L}|$.

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

⚠ The proof of Lemma 6 is quite difficult to follow and it leaves the reader with the impression that weaknesses there exist that are not addressed.

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 can be done in time $\mathcal{O}(n)$ (Day, 1985).

Next the algorithm identifies and characterises the islands of T_1 and T_2 (line~~s~~ 4 and 5). This is performed by a traversal of each tree in pre-order and in doing so ~~identifying~~ the islands, which are separated by good edges, keeping track of the number of internal nodes, the labels of the internal nodes of the islands, and the nodes associated with each island. Each tree traversal is done in time $\mathcal{O}(n)$.

, identifies

The next steps requires pairing islands of T_1 and T_2 by iterating over ~~the~~ good edges ($\mathcal{O}(n)$). Line 8 first retrieves, for both input trees, the two islands ~~delimited~~ by the current good edge, then it proceeds by pairing one island from T_1 to its matching island from T_2, and then by pairing the two ?? remaining islands from each tree. Using the node-to-island map computed earlier, the retrieval of the two island pairs associated with a good edge can be done in constant time.

non-trivial, i.e. internal

in

overlapping

For each of the matching island pairs, at lines 9 and 14, the algorithm checks whether each island pair has already been visited in a previous iteration of the loop (the same island can be visited from multiple good edges). If not, the current distance is implemented by adding $\epsilon(I_1)+\epsilon(I_2)$.

Algorithm 1 LRF(T_1, T_2)

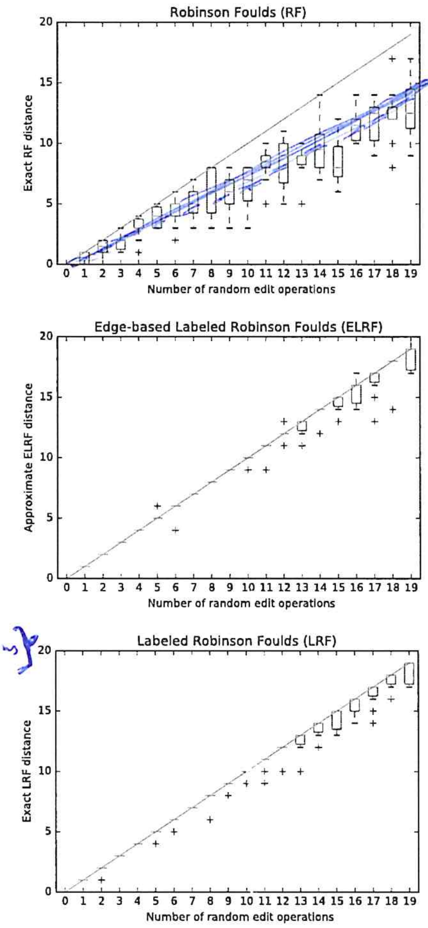
```
1: $bipartitions_1 = getBiparitions(T_1)$;
2: $bipartitions_2 = getBiparitions(T_2)$;
3: $goodEdges = bipartitions_1 \cap bipartitions_2$; \{trivial bipartitions\}
4: $islands_1 = getIslands(T_1, goodEdges)$;
5: $islands_2 = getIslands(T_2, goodEdges)$;
6: $distance = 0$;
7: for $i \in goodEdges$:
8:     $((x_1, y_1), (x_2, y_2)) = islandPair(i, islands_1, islands_2)$;
9:     if $x_1.visited == False$:
10:        $distance += x_1.\epsilon + y_1.\epsilon$;
11:        if $x_1.labels \cap y_1.labels == \emptyset$:
12:            $distance += 1$;
13:        $x_1.visited = True$
14:    if $x_2.visited == False$:
15:        $distance += x_2.\epsilon + y_2.\epsilon$;
16:        if $x_2.labels \cap y_2.labels == \emptyset$:
17:            $distance += 1$;
18:        $x_2.visited = True$
19: if $goodEdges == \emptyset$ :
20:     $distance += islands_1[0].\epsilon + islands_2[0].\epsilon$
21:     if $islands_1[0].labels \cap islands_2[0].labels == \emptyset$:
22:         $distance += 1$;
23: return $distance$;
```

unclear what these are. You should explain in the text

The for-loop ends with lines 11-12 and 16-17 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 19-22 are needed to handle the special case where there is no good edge between T_1 and T_2, for instance if T_1 or T_2 is a star. In such a case, there is only one island per tree, which is matching.

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



← also show the regression line with equation y=0.7 here, since RF insensitive to the node substitution operations

Fig. 5: 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 (Briand *et al.* (2020); 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 benefits from denser taxon sampling?

5.1 Empirical comparison of LRF with RF and $ELRF$

We retrieved the labeled tree associated with human gene NOX4 from Ensembl release 99 (Yates *et al.*, 2020), 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

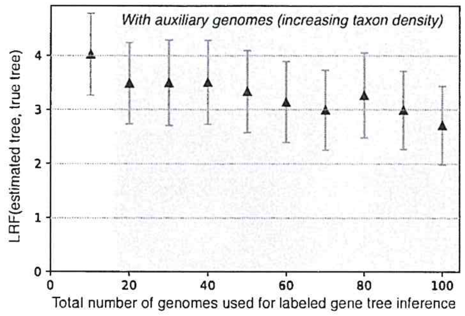


Fig. 6: Denser taxon sampling decreases labeled tree estimation error: labeled gene trees reconstructed with an increasing number of auxiliary genomes (i.e. obtained by including the additional genomes during tree inference and labeling, followed by pruning) have a smaller LRF distance to the true trees. Error bars depict 95% confidence intervals around the mean.

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 deletion 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. 5). As expected, the conventional RF distance returns the smallest values because it ignores labels. 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 (Nabhan and Sarkar, 2011), we are not aware of any previous study on labeled gene tree inference.

First, using ALF (Dalquen *et al.*, 2012), 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 sequences translated from homologous genes using MAFFT L-INS-i (Katoh and Standley, 2013), inferred phylogenetic trees from the alignments using FastTree (Price *et al.*, 2010), and annotated their nodes using the species overlap algorithm (van der Heijden *et al.*, 2007) as implemented in the ETE3 python library (Huerta-Cepas *et al.*, 2016). 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. 6). 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 (Briand *et al.*, 2020). 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 and reducing to the conventional Robinson Foulds distance in the presence of trees with only one type of label. Finally, we showed that the new distance is computable for an arbitrary number of label types associated with internal nodes of the tree.

Our experimental results illustrate the utility of computing tree distances taking labels into account, as the conventional RF distance is blind to label changes. 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" (Abramson and Moser, 1970): 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