

A Metric for Phylogenetic Trees Based on Matching

Yu Lin, Vaibhav Rajan, and Bernard M.E. Moret

Laboratory for Computational Biology and Bioinformatics,
Swiss Federal Institute of Technology (EPFL),
EPFL-IC-LCBB, INJ 230, Station 14, CH-1015 Lausanne, Switzerland
{yu.lin, vaibhav.rajan, bernard.moret}@epfl.ch

Abstract. Comparing two or more phylogenetic trees is a fundamental task in computational biology. The simplest outcome of such a comparison is a pairwise measure of similarity, dissimilarity, or distance. A large number of such measures have been proposed, but so far all suffer from problems varying from computational cost to lack of robustness; many can be shown to behave unexpectedly under certain plausible inputs. For instance, similarity measures based on maximum agreement are too strict, while measures based on the elimination of rogue taxa work poorly when the proportion of rogue taxa is significant; distance measures based on edit distances under simple tree operations (such as nearest-neighbor interchange or subtree pruning and regrafting) are NP-hard; and the widely used Robinson-Foulds distance is poorly distributed and thus affords little discrimination, while also lacking robustness in the face of very small changes—reattaching a single leaf elsewhere in a tree of any size can instantly maximize the distance.

In this paper, we introduce an entirely new pairwise distance measure, based on matching, for phylogenetic trees. We prove that our measure induces a metric on the space of trees, show how to compute it in low polynomial time, verify through statistical testing that it is robust, and finally note that it does not exhibit unexpected behavior under the same inputs that cause problems with other measures. We also illustrate its usefulness in clustering trees, demonstrating significant improvements in the quality of hierarchical clustering as compared to the same collections of trees clustered using the Robinson-Foulds distance.

1 Introduction

Comparing two or more phylogenetic trees is a fundamental task in computational biology. The simplest outcome of such a comparison is a pairwise measure of similarity, dissimilarity, or distance. A large number of such measures have been proposed, but so far all suffer from problems varying from computational cost to lack of robustness; many can be shown to behave unexpectedly under certain plausible inputs. For instance, similarity measures based on maximum agreement are too strict, while measures based on the elimination of rogue taxa work poorly when the proportion of rogue taxa is significant; distance measures based on edit distances under simple tree operations (such as nearest-neighbor interchange or subtree pruning and regrafting) are NP-hard; and the widely used Robinson-Foulds distance is poorly distributed and thus affords little discrimination, while also lacking robustness in the face of very small changes—reattaching a single leaf elsewhere in a tree of any size can instantly maximize the distance.

In this paper, we introduce an entirely new pairwise distance measure, based on matching, for phylogenetic trees. We prove that our measure induces a metric on the space of trees, show how to compute it in low polynomial time, verify through statistical testing that it is robust, and finally note that it does not exhibit unexpected behavior under the same inputs that cause problems with other measures. Our matching metric can be viewed as a weighted extension of the Robinson-Foulds distance, but can also be interpreted in the context of tree editing, thus bridging two types of tree-to-tree measures.

We illustrate the use of our tree metric in clustering trees; we obtain significant improvements in the quality of hierarchical clustering as compared to the same collections of trees clustered using the Robinson-Foulds distance.

2 Background

2.1 Similarity, Editing, and Distance

Phylogenetic trees are leaf-labelled trees, most often unrooted. Perhaps the simplest way to quantify the similarity of a set of phylogenetic trees is to determine the smallest collection of leaves that, when removed, induce the same tree (on the remaining leaves) from each tree in the set. Such an induced tree is called the *Maximum Agreement SubTree (MAST)*. Several variations have been proposed on this theme, all seeking to identify a tree structure that is common, in exact or approximate form, to all trees in the given set. For a pair of trees, most such measures are fairly easy to compute. Trees can also be transformed through various operations that disconnect and reconnect sub-pieces; given any collection of such operations, and assuming that the operations are sufficiently powerful to enable us to transform any tree on n leaves into any other tree on n leaves, we can define an *edit distance* between two trees as the smallest number of allowed operations that will transform one tree into the other. Computing such edit distances is typically NP-hard, however, nor is it clear which set of operations should be used in the characterization. Finally, we can focus on the characteristics of two trees to determine the number of differences and thus induce a distance measure based on outcomes rather than on transformations. The *Robinson-Foulds (RF) distance*, the most commonly used distance measure for trees, counts the number of edges (or, equivalently, bipartitions of the leaves) present in one tree, but not the other; it can be computed in linear time. We now look at each of these three approaches in turn.

2.2 Tree Similarity Measures

The MAST problem has been well studied [5,9,12]. While the general problem of finding the MAST of three or more trees is NP-hard [2], it can be solved in $O(n \log n)$ time for two binary trees [18]. Since requiring exact agreement may prove too demanding and lead to poor results, several authors proposed variations on this formulation, among them the *maximum information subtree (MIST)* [3] and the *maximum information subtree consensus (MISC)* [15], variations that are more robust than MAST in the presence of “rogue” taxa (taxa whose placement in the tree is unclear and highly variable). These methods work well in the presence of a small number of rogue taxa, but poorly (both in terms of running time and of quality of results) when rogues are numerous; they also work only on sizeable collections of trees, not on pairs of trees.

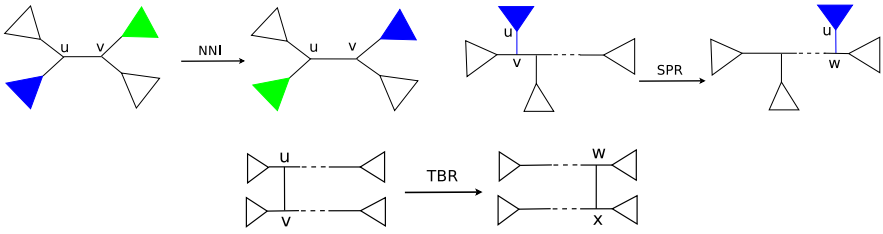


Fig. 1. NNI, SPR and TBR operations

2.3 Tree Editing

Editing operations are commonly used to explore tree space in phylogenetic inference, but also for comparing phylogenetic trees. We briefly describe the three most common operations, in increasing order of generality.

Nearest Neighbor Interchange (NNI). Let $e = \{u, v\}$ be an internal edge of a tree T and S_u and S_v be the set of subtrees connected to u and v respectively. A single NNI operation interchanges two subtrees across e : it disconnects one of the subtrees from S_u and connects it to vertex v , then disconnects one of the subtrees from S_v and connects it to vertex u , as illustrated in Fig. 1.

Subtree Prune and Regraft (SPR). An SPR operation disconnects a subtree from the larger tree by removing some edge $\{u, v\}$; the pruned subtree has vertex u , while the larger tree has vertex v . If the larger tree was binary, then v now has degree 2 and is eliminated by merging its two incident edges. Then the subtree is reconnected to the larger tree by creating a new vertex w on some edge of the larger tree and connecting it to the pruned subtree by a new edge $\{u, w\}$, as illustrated in Fig. 1. The *Leaf Prune and Regraft (LPR)* operation is the simplified version in which the subtree pruned always consists of a single leaf.

Tree Bisection and Reconnection (TBR). Let $e = \{u, v\}$ be an internal edge of a tree T and let C_1 and C_2 be the components of the tree formed by removing e and (if the tree was binary) suppressing vertices u and v . Form tree T' by choosing one edge in C_1 and adding a vertex w along that edge, choosing one edge in C_2 and adding a vertex x along that edge, and finally adding the edge $\{w, x\}$, as illustrated in Fig. 1. (If any of the components is just a single vertex, then the newly added edge is attached to the vertex.)

Any tree operation can be used to define an edit distance between trees: the minimum number of such operations needed to transform one tree into the other. Regrettably, computing the edit distance for each of the above three operations is NP-hard [1,6,11]. The NNI edit distance between two trees is $O(n \log n)$ [13] and can be approximated within a ratio of $O(\log n)$ [6]. The edit distances between two trees for SPR and TBR are $O(n)$ [1] and there is a 3-approximation algorithm to compute the TBR edit distance [20]. The LPR edit distance between two trees on n leaves is just n minus the number of leaves in the MAST of those two trees and so can be computed in polynomial time for two binary trees.

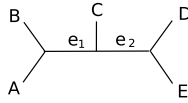


Fig. 2. An unrooted tree with 5 leaves

2.4 The Robinson-Foulds Distance

The Robinson-Foulds (RF) distance [16] is by far the most widely used measure of dissimilarity between trees. One of its main advantages is its independence from any model of tree editing: it does not infer any series of editing operations, but relies only on the current characteristics of the two trees.

Every internal edge e in a leaf-labeled tree T defines a non-trivial bipartition π_e on the leaves, and hence the tree T is uniquely represented by the set of bipartitions $\Gamma(T) = \{\pi_e \mid e \in E(T)\}$, where $E(T)$ is the set of internal edges in T . For example the unrooted tree in Fig. 2 is represented by two nontrivial bipartitions $\{AB|CDE, ABC|DE\}$ induced by edges e_1 and e_2 , respectively. Given two unrooted leaf-labeled trees T_1 and T_2 on the same set of leaf labels, the Robinson-Foulds (RF) distance between them is the normalized count of the bipartitions induced by one tree and not the other, that is,

$$\mathcal{D}_{RF}(T_1, T_2) = \frac{1}{2}((|\Gamma(T_1) - \Gamma(T_2)|) + (|\Gamma(T_2) - \Gamma(T_1)|)).$$

Since there are at most $n - 3$ non-trivial bipartitions in a tree on n leaves, the largest possible RF distance between two trees is $n - 3$. The RF distance between two trees can be computed in linear time [7], while the RF distance matrix for a collection of trees can be computed in sublinear time [14]. However, the RF distance is overly sensitive to some small changes in the tree. For example, just moving a leaf at the end of a caterpillar tree (a single spine to which all leaves are attached) to the other end will create a tree with the maximum possible RF distance to the original tree, yet this change takes a single LPR operation. The RF distance between two random binary trees has a very skewed distribution [17,4] in which most values equal $n - 3$ (also see section 4.1 for details).

3 Our Matching Distance

A tree T is uniquely represented by the set of bipartitions $\Gamma(T) = \{\pi_e \mid e \in E(T)\}$, where $E(T)$ is the set of internal edges in T . Given two trees, T_1 and T_2 on the same set of leaf labels, we define a complete weighted bipartite graph $B(X, Y, E)$ with $X = \Gamma(T_1)$ and $Y = \Gamma(T_2)$, that is, every bipartition is represented by a vertex in B . We denote this graph by $B(T_1, T_2)$. An edge (u, v) has weight 0 if the bipartitions $u \in \Gamma(T_1)$ and $v \in \Gamma(T_2)$ are the same, otherwise it has weight 1. We can then rephrase the RF distance between T_1 and T_2 as the weight of the minimum-weight matching in $B(T_1, T_2)$.

The binary weighting scheme does not make full use of the information in the bipartitions. Each bipartition π_e can be represented by a binary vector V_e of length n , where n is the number of leaves in T_1 (or T_2). For any leaf i , we set $V_e[i] = 1$ if leaf i and leaf 1

are on the same side of the bipartition π_e and set $V_e[i] = 0$ otherwise. We set the weight of each edge $e = \{u, v\}$ in $B(T_1, T_2)$ (where vertices u, v in B represent internal edges in T_1 and T_2 respectively) to:

$$W(u, v) = \min\{\mathcal{D}_H(V_u, V_v), \mathcal{D}_H(V_u, \overline{V}_v)\},$$

where \mathcal{D}_H is the Hamming distance between the two vectors and \overline{V} , the complement vector of V , is equal to $I - V$. The matching distance $\mathcal{D}_M(T_1, T_2)$ between trees T_1 and T_2 is the weight of the minimum-weight matching in $B(T_1, T_2)$ with the weighting scheme W . This definition is a natural choice since the Hamming distance between the two bipartitions represents the minimum number of leaves that must be moved in order to transform one into the other.

The minimum-weight matching problem can be solved in cubic time [8]. If the input weights are integers and the value of each weight is not greater than the number of leaves (as is the case for our matching problem), the running time of the algorithm can be improved to $O(n^{5/2} \log(n))$ by cost scaling and blocking flow techniques [10].

3.1 Basic Properties

First, we show that our distance measure is well defined: it is indeed a metric.

Lemma 1. *The matching distance \mathcal{D}_M on binary leaf-labeled trees is a metric.*

For any binary trees T_i, T_j and T_k on n labeled leaves, we have

1. $\mathcal{D}_M(T_i, T_j) \geq 0$.
2. $\mathcal{D}_M(T_i, T_j) = 0$ if and only if $T_i = T_j$.
3. $\mathcal{D}_M(T_i, T_j) = \mathcal{D}_M(T_j, T_i)$.
4. $\mathcal{D}_M(T_i, T_j) + \mathcal{D}_M(T_j, T_k) \geq \mathcal{D}_M(T_i, T_k)$.

Proof. Properties 1, 2 and 3 follow directly from the definition of the matching distance. We prove Property 4. Assume $M_{i,j}$ and $M_{j,k}$ are the minimum-weight matchings in $B(T_i, T_j)$ and $B(T_j, T_k)$. Construct a matching $M_{i,k} = \{(u, w) \mid (u, v) \in M_{i,j} \wedge (v, w) \in M_{j,k}\}$ in $B(T_i, T_k)$. Since $\mathcal{D}_M(T_i, T_k)$ is the minimum-weight matching in $B(T_i, T_k)$, we have

$$\begin{aligned} \mathcal{D}_M(T_i, T_k) &\leq \sum_{(u,w) \in M_{i,k}} W(u, w) \\ &\leq \sum_{(u,v) \in M_{i,j}, (v,w) \in M_{j,k}} (W(u, v) + W(v, w)) \\ &= \sum_{(u,v) \in M_{i,j}} W(u, v) + \sum_{(v,w) \in M_{j,k}} W(v, w) \\ &= \mathcal{D}_M(T_i, T_j) + \mathcal{D}_M(T_j, T_k). \end{aligned}$$

Next we investigate extremal properties of our matching distance.

Definition 1. *Let $T(n)$ be the space of all binary trees on n labeled leaves. The diameter (δ) of $T(n)$ with respect to a distance metric \mathcal{D} on $T(n)$ is defined as*

$$\delta(T(n), \mathcal{D}) = \max\{\mathcal{D}(T_1, T_2) \mid T_1, T_2 \in T(n)\}.$$

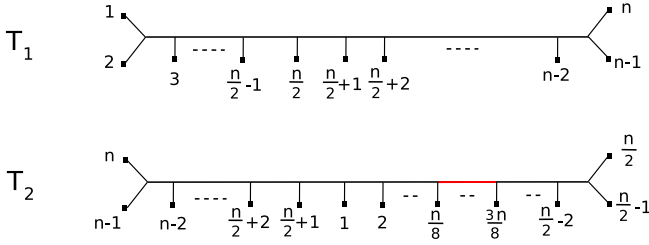


Fig. 3. Example for two trees on n leaves with Matching distance $\Theta(n^2)$

Theorem 1

$$\begin{aligned} \delta(T(n), \mathcal{D}_{RF}) &= n - 3, \\ \delta(T(n), \mathcal{D}_M) &= \Theta(n^2) \end{aligned}$$

Proof. We prove the bounds on the diameter by explicitly constructing two trees T_1 and T_2 . For the RF distance, choose T_1 and T_2 to be two caterpillar trees with different cherries, then no bipartition can appear both in $\Gamma(T_1)$ and $\Gamma(T_2)$, thus $(n - 3)$ mismatches result in an RF distance of $(n - 3)$. For the matching distance, construct two caterpillar trees T_1 and T_2 as shown in Fig. 3. The leaves in T_1 are ordered as $(1, \dots, n)$, and the leaves in T_2 are ordered as $(n, \dots, n/2 + 1, 1, 2, \dots, n/2)$. It is easy to verify (by case analysis) that each bipartition corresponding to an internal edge along the path between leaf $n/8$ and leaf $3n/8$ in T_2 (marked in red) is at least $n/8$ away from every bipartition in T_1 . Since there are $n/4$ such bipartitions in T_2 , any matching between $\Gamma(T_1)$ and $\Gamma(T_2)$ will have a weight at least $(n/4) * (n/8) = \Omega(n^2)$. The upper bound is trivial.

3.2 Sensitivity to Tree Editing

We now study the change in the distance measures caused by a single tree editing operation. Let $\phi(T)$ be the set of trees derived by applying operation ϕ to a tree T , where ϕ can be one of NNI, SPR, TBR, LPR, or *Leaf Label Exchange (LLI)*, this last an operation that does not alter the tree structure, but simply exchanges the labels of two leaves.

Definition 2. *The gradient of a tree rearrangement operation ϕ with respect to a distance metric \mathcal{D} on $T(n)$ is defined as*

$$\mathcal{G}(T(n), \mathcal{D}, \phi) = \max\{\mathcal{D}(T_1, T_2) | T_1, T_2 \in T(n), T_2 \in \phi(T_1)\}.$$

Theorem 2

$$\begin{aligned} \mathcal{G}(T(n), \mathcal{D}_{RF}, NNI) &= 1, \\ \mathcal{G}(T(n), \mathcal{D}_M, NNI) &= \Theta(n). \end{aligned}$$

Proof. Let T_2 be the tree obtained by applying one NNI operation on T_1 . Every NNI operation changes only one bipartition in $\Gamma(T_1)$ into a new one in $\Gamma(T_2)$ (induced by the

internal edge which is selected). Thus $\mathcal{G}(T(n), \mathcal{D}_{RF}, NNI) = 1$. Since $\Gamma(T_1)$ and $\Gamma(T_2)$ share $n - 4$ bipartitions, we can construct a matching $M_{1,2}$ in $B(T_1, T_2)$ that contains $n - 4$ matched pairs with weight zero and 1 matched pair with weight at most n . The sum of the weights for $M_{1,2}$ is upper bounded by n , and hence $\mathcal{D}_M(T_1, T_2) \leq n$. Let $e = (u, v)$ be an internal edge in T_1 connecting four rooted subtrees $\{S_1, S_2, S_3, S_4\}$ where S_1 and S_2 are attached to u and S_3 and S_4 are attached to v . Assume each of the four subtrees contains $n/4$ leaves and one NNI operation interchanges S_2 and S_3 . The newly created bipartition by NNI in T_2 is now at least $\Theta(n)$ distance away from all possible bipartitions in T_1 . So any matching in $B(T_1, T_2)$ will have weight at least $\Theta(n)$. From the upper and lower bounds, we have $\mathcal{G}(T(n), \mathcal{D}_M, NNI) = \Theta(n)$.

Theorem 3

$$\begin{aligned} \mathcal{G}(T(n), \mathcal{D}_{RF}, LPR) &= n - 3 \\ \mathcal{G}(T(n), \mathcal{D}_M, LPR) &= \Theta(n). \end{aligned}$$

Proof. The bound for $\mathcal{G}(T(n), \mathcal{D}_{RF}, LPR)$ is derived by applying LPR to a caterpillar tree T_1 , where one leaf at one end of the tree is transposed to the other end of the tree. Let T_2 be the tree obtained by applying one LPR operation on T_1 . T_2 shares no bipartitions with the tree T_1 and the RF distance between them is $n - 3$. The matching distance between T_1 and T_2 is $\Omega(n)$ since each pair of bipartitions from $\Gamma(T_1)$ and $\Gamma(T_2)$ contributes at least 1 to the matching weight and there are $n - 3$ pairs. Because every LPR operation only affects two internal edges in $\Gamma(T_1)$ (we remove an internal edge while pruning and create a new internal edge while regrafting), there are $n - 5$ internal edges left untouched and shared by T_1 and T_2 . We can construct a matching $M_{1,2}$ in $B(T_1, T_2)$ that contains $n - 5$ matched pairs corresponding to the shared edges and another 2 matched pairs. For each matched pair for the shared edges, the weight is at most 1 since the corresponding bipartitions can only differ at the pruned leaf. For the other 2 matched pairs, the contribution to the total weight is at most $O(n)$. The weight for this matching $M_{1,2}$ is thus bounded by $O(n)$. From the upper and lower bounds, we have $\mathcal{G}(T(n), \mathcal{D}_M, LPR) = \Theta(n)$.

Theorem 4

$$\begin{aligned} \mathcal{G}(T(n), \mathcal{D}_{RF}, SPR) &= n - 3, \\ \mathcal{G}(T(n), \mathcal{D}_M, SPR) &= \Theta(n^2). \end{aligned}$$

Proof. The bound for $\mathcal{G}(T(n), \mathcal{D}_{RF}, SPR)$ follows from Theorem 3 since LPR is a special case of SPR and $(n - 3)$ is already the maximum change in RF distance. The bound for $\mathcal{G}(T(n), \mathcal{D}_M, SPR)$ is obtained from the trees in Fig. 3, where one SPR operation on T_1 results in T_2 and $\mathcal{D}_M(T_1, T_2) = \Theta(n^2)$.

Theorem 5

$$\begin{aligned} \mathcal{G}(T(n), \mathcal{D}_{RF}, TBR) &= n - 3, \\ \mathcal{G}(T(n), \mathcal{D}_M, TBR) &= \Theta(n^2). \end{aligned}$$

Proof. The results follow directly from Theorem 4 since SPR is a special case of TBR and both gradients have trivial upper bounds.

Theorem 6

$$\begin{aligned}\mathcal{G}(T(n), \mathcal{D}_{RF}, LLI) &= n - 3, \\ \mathcal{G}(T(n), \mathcal{D}_M, LLI) &= \Theta(n).\end{aligned}$$

Proof. The bound for $\mathcal{G}(T(n), \mathcal{D}_{RF}, LLI)$ is derived by applying LLI to a caterpillar tree T_1 , where the labels of two leaves at two ends of the tree are interchanged. Let T_2 be the tree obtained by applying one LLI operation on T_1 . T_2 shares no bipartitions with the tree T_1 , and the RF distance between them is $n - 3$. The matching distance between T_1 and T_2 is $\Omega(n)$ since each pair of bipartitions from $\Gamma(T_1)$ and $\Gamma(T_2)$ contributes at least 1 to the matching weight and there are $n - 3$ pairs. Because every LLI operation only affects two leaves in T_1 and T_2 , all $n - 5$ internal edges are left untouched. We can construct a matching $M_{1,2}$ in $B(T_1, T_2)$ that contains those $n - 3$ matched pairs corresponding to the shared edges. For each matched pair for the shared edges, the weight is at most 2 since the corresponding bipartitions can differ at not more than two leaves. The weight for this matching $M_{1,2}$ is thus bounded by $O(n)$. From the upper and lower bounds, we have $\mathcal{G}(T(n), \mathcal{D}_M, LLI) = \Theta(n)$.

The ratio of the gradient to the diameter is an indication of the sensitivity of the distance measure. Our theorems indicate that the matching distance has the same asymptotic sensitivity as the RF distance with respect to NNI, SPR, and TBR, but is more sensitive than the RF distance with respect to LPR and LLI.

4 Experimental Results

The previous section gave extremal properties of our matching distance, but its main advantages are best seen by comparing its distribution of values to that of the RF distance. We have not derived an exact formula for the distribution, but present experimental results that show that our matching distance on random binary trees yields a distribution with a fairly broad bell curve, in sharp contrast to the highly skewed distribution of the RF distance.

4.1 Distribution of the Tree Distance Metrics

We first study the distribution of RF and matching distances by sampling pairs of random trees generated in two different ways. The first, uniformly sampled binary trees, are generated by the randomized leaf attachment process [17]), and the second, birth-death trees, are generated by a uniform, time-homogeneous birth-death process (birth rate = 0.1, death rate = 0). Fig. 4 shows the distribution of RF and matching distances for 100,000 pairs of uniformly sampled binary trees on 100 and 1,000 leaves each and birth-death trees on 100 leaves. The range of values for each distance is divided into 100 intervals and each point on the x axis represents an interval. Compared to RF (a very skewed distribution as shown in the figure and in [4]), our matching distance offers a larger range and is more broadly distributed, and thus also more discriminating.

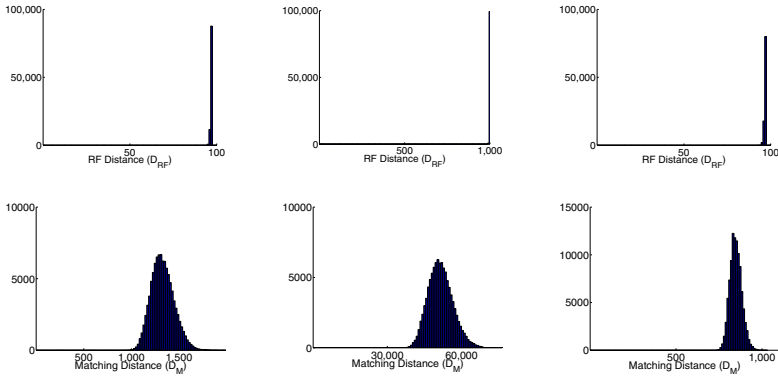


Fig. 4. Distribution of pairwise RF (above) and matching (below) distances between uniformly sampled binary trees on 100 leaves (left), on 1,000 leaves (middle), and between birth-death trees on 100 leaves (right)

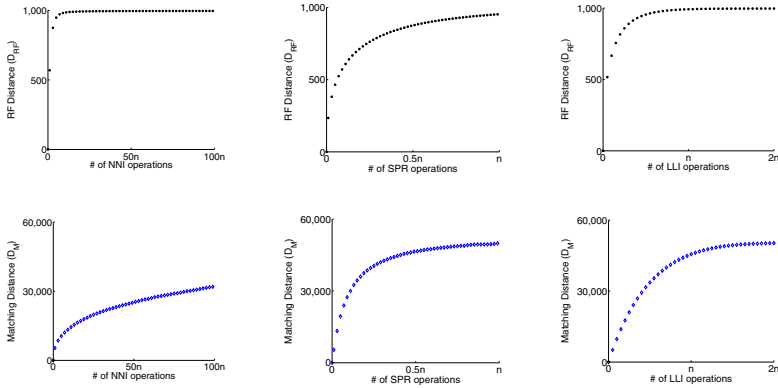


Fig. 5. RF (above) and matching (below) distances as a function of the number of NNI operations (left), SPR operations (middle), and LLI operations (right) for trees on 1,000 leaves ($n = 1,000$)

4.2 Tree Distance Metrics Under Tree Editing Operations

We study the behavior of both RF and matching metrics under various tree editing operations. For each operation, we study the change in the distance after successive applications of the operation. From the distributions of the two distance metrics seen in the previous section we expect the RF distance to saturate faster and the matching distance to have a better correlation with the number of tree rearrangement operations which is indeed the case. Note that such a comparison with the edit distance (with respect to the rearrangement operations) is infeasible since it is NP-hard to compute the distances.

Our experiments start with 1,000 uniformly generated binary trees on 1,000 leaves each. We summarize the average pairwise RF and matching distances between the trees and the original as a function of the number of operations applied. Fig. 5 shows RF

and matching distances as a function of the number of NNI operations. While the RF distance reaches saturation after 10,000 ($10n$) operations ($n = 1,000$), our matching distance still shows an increasing trend; indeed, the average matching distance ($\sim 30,000$) after 100,000 ($100n$) operations is still far from the average matching distance ($\sim 50,000$) between two randomly selected binary trees on 1,000 leaves (as seen in Fig. 4). Similar results are shown in Fig. 5 for SPR and LLI operations.

5 Clustering Trees: An Application

In this section we provide a proof-of-concept study of the usefulness of the matching distance in clustering phylogenetic trees.

Phylogenetic analyses such as maximum-parsimony or maximum-likelihood analyses often produce many (possibly thousands) of candidate trees that are nearly optimal with respect to the defined objective function. To obtain a biologically relevant tree, postprocessing of these candidate trees is essential. Consensus tree methods are frequently used to extract the common structure from the candidate trees and summarize the output; however these methods often lose information and are sensitive to outliers. A different approach divides the set of candidate trees into several subsets using clustering methods, each cluster being characterized by its own consensus tree [19]. The authors of that approach demonstrate an improvement over traditional consensus methods by obtaining better resolved output trees and by providing details of the distribution of the candidate trees.

The efficacy of clustering relies on the dissimilarity measure used. We conducted a preliminary test on RF and matching metrics as dissimilarity measures in clustering. We generated 100 datasets, each of 200 random binary trees. The trees in each dataset were generated by a two-step process. We first sampled two binary trees on k leaves ($k < 100$) and used them as two different skeletons. Then from each of the two skeletons, a set of 100 trees is generated by adding the rest of the $(n - k)$ leaves one by one. To add a new leaf, an edge in the current tree is selected uniformly at random and the new leaf is attached to that edge. We vary k from 40 ($0.4n$) to 90 ($0.9n$). The 200 trees in each dataset is given as input to the clustering algorithm to check if the algorithm can distinguish the trees in the two clusters. Trees generated from the same skeleton are considered to be in the same cluster.

Notice that a MAST-based distance metric can easily distinguish the input trees into the correct cluster. We deliberately choose this experimental setup to provide a test case for the matching distance even in those settings where a MAST-based distance will perform better than RF.

We apply a standard hierarchical clustering approach (recommended for phylogenetic postprocessing in [19]) to the pairwise distance matrices generated by RF and matching distances. The similarity between clusters C_1 and C_2 is measured by the following three linkage criteria:

1. Complete linkage: $\max\{\mathcal{D}(a, b) \mid a \in C_1, b \in C_2\}$.
2. Single linkage: $\min\{\mathcal{D}(a, b) \mid a \in C_1, b \in C_2\}$.
3. Average linkage: $\frac{1}{|C_1||C_2|} \sum_{a \in C_1, b \in C_2} \mathcal{D}(a, b)$.

Table 1. Error rates for the clustering test

	0.4n		0.5n		0.6n		0.7n	
	\mathcal{D}_{RF}	\mathcal{D}_M	\mathcal{D}_{RF}	\mathcal{D}_M	\mathcal{D}_{RF}	\mathcal{D}_M	\mathcal{D}_{RF}	\mathcal{D}_M
Complete linkage	70%	0%	71%	0%	38%	0%	3%	0%
Single linkage	22%	8%	23%	0%	0%	0%	0%	0%
Average linkage	49%	0%	3%	0%	0%	0%	0%	0%

A run of the algorithm on a particular dataset is considered to err if it is unable to place every tree generated from the same skeleton in one cluster. We present the error rate obtained from 100 such datasets for each parameter in Table 1. For values of k higher than $0.7n$ both distance measures perform equally well, but, as expected, the matching distance has significantly better performance over a large range of input parameters.

6 Conclusion

We have introduced a new tree metric for phylogenetic analysis. This metric can be computed efficiently, in contrast to various edit distances, and offers better discrimination than the standard Robinson-Foulds distance, thanks to a much broader and less biased distribution of distance values. We have given extremal results as well as experimental results to characterize this new metric. Finally, we have demonstrated the use of this metric in clustering trees with an agglomerative hierarchical clustering method, where using our metric considerably improved over using the Robinson-Foulds metric.

Acknowledgement

We thank Nicholas D. Pattengale for many helpful discussions.

References

1. Allen, B.L., Steel, M.: Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics* 5(1), 1–15 (2001)
2. Amir, A., Keselman, D.: Maximum agreement subtree in a set of evolutionary trees: Metrics and efficient algorithms. *SIAM J. Computing* 26(6), 1656–1669 (1997)
3. Bryant, D.: Hunting for trees, building trees and comparing trees: Theory and method in phylogenetic analysis. PhD thesis, University of Canterbury (1997)
4. Bryant, D., Steel, M.: Computing the distribution of a tree metric. *ACM/IEEE Trans. on Comput. Biology and Bioinformatics* 6(3), 420–426 (2009)
5. Cole, R., Farach-Colton, M., Hariharan, R., Przytycka, T., Thorup, M.: An $O(n \log n)$ algorithm for the maximum agreement subtree problem for binary trees. *SIAM J. Computing* 30(5), 1385–1404 (2000)
6. DasGupta, B., He, X., Jiang, T., Li, M., Tromp, J., Zhang, L.: On distances between phylogenetic trees. In: *Proc. 8th ACM/SIAM Symp. Discrete Algs. (SODA 1997)*, pp. 427–436 (1997)
7. Day, W.H.E.: Optimal algorithms for comparing trees with labeled leaves. *J. Classification* 2(1), 7–28 (1985)

8. Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM* 19(2), 248–264 (1972)
9. Farach, M., Przytycka, T.M., Thorup, M.: On the agreement of many trees. *Inf. Process. Lett.* 55(6), 297–301 (1995)
10. Gabow, H.N., Tarjan, R.E.: Faster scaling algorithms for network problems. *SIAM J. Computing* 18(5), 1013–1036 (1989)
11. Hickey, G., Dehne, F., Rau-Chaplin, A., Blouin, C.: SPR distance computation of unrooted trees. *Evol. Bioinform. Online* 4, 17–27 (2008)
12. Kao, M.Y.: Tree contractions and evolutionary trees. *SIAM J. Computing* 27(6), 1592–1616 (1998)
13. Li, M., Tromp, J., Zhang, L.: On the nearest-neighbour interchange distance between evolutionary trees. *J. Theor. Biol.* 182(4), 463–467 (1996)
14. Pattengale, N.D., Gottlieb, E.J., Moret, B.M.E.: Efficiently computing the Robinson-Foulds metric. *J. Comput. Biol.* 14(6), 724–735 (2007)
15. Pattengale, N.D., Swenson, K.M., Moret, B.M.E.: Uncovering hidden phylogenetic consensus. In: Borodovsky, M., Gogarten, J.P., Przytycka, T.M., Rajasekaran, S. (eds.) *ISBRA 2010*. LNCS, vol. 6053, pp. 128–139. Springer, Heidelberg (2010)
16. Robinson, D.R., Foulds, L.R.: Comparison of phylogenetic trees. *Mathematical Biosciences* 53, 131–147 (1981)
17. Steel, M., Penny, D.: Distributions of tree comparison metrics—some new results. *Syst. Biol.* 42(2), 126–141 (1993)
18. Steel, M., Warnow, T.: Kaikoura tree theorems: computing maximum agreement subtree problem. *Information Processing Letters* 48, 77–82 (1993)
19. Stockham, C., Wang, L.-S., Warnow, T.: Statistically-based postprocessing of phylogenetic analysis using clustering. In: *Proc. 10th Conf. Intelligent Systems for Mol. Biol. (ISMB 2002)*. Bioinformatics, vol. 18, pp. S285–S293. Oxford U. Press, Oxford (2002)
20. Whidden, C., Zeh, N.: A unifying view on approximation and fpt of agreement forests. In: Bücher, P., Moret, B.M.E. (eds.) *WABI 2006*. LNCS (LNBI), vol. 4175, pp. 390–402. Springer, Heidelberg (2006)