Novel XBWT-based Distance Measures for Labeled Trees

Danilo G. Dolce[1,†], Sabrina Mantaci[1,†], Giuseppe Romana[1,*,†], Giovanna Rosone[2,†] and Marinella Sciortino[1,†]

[1]*University of Palermo, Italy*
[2]*University of Pisa, Italy*

Abstract

Comparing labeled trees has applications in various domains, particularly in the study of cancer phylogenies.

In this paper, we address the problem of comparing fully labeled unordered trees, focusing on their structural and label similarities. We define a novel class of distance measures by exploiting the eXtended Burrows-Wheeler Transform (XBWT), an extension to labeled trees of the well-known Burrows-Wheeler Transform. The XBWT, introduced in [Ferragina et al., FOCS 2005], produces a linearization of the tree that is both compressible and efficiently searchable. We extend the definition of this linearization to pairs of trees, and we produce a partition based on the prefixes of a given length $k \geq 1$ of the parent-to-root paths of the nodes.

We define, for any $k \geq 1$, the distance measure d_k by applying the Jaccard distance to each element of this partition. We prove that all the measures d_k are pseudometrics, i.e. they assume non-negative values, are zero when applied to identical trees, are symmetric, and satisfy the triangle inequality. These measures become metrics when all the labels within each tree are distinct.

Furthermore, we show that these distances are sensitive to some operations on trees, such as the removal and insertion of subtrees, swapping of subtrees, and label swapping.

Finally, to show the effectiveness of our approach, we have analyzed experimentally the behavior of the distances when operations on trees are applied to a randomly generated fully labeled tree. Here, we present the results obtained in the case $k = 1$.

Keywords

Burrows-Wheeler Transform, XBWT, labeled tree, distance measure

1. Introduction

Trees are fundamental and well-studied combinatorial structures in computer science since their structure can encode, in a natural way, hierarchical relations in many domains.

Comparing trees is a key problem that arises in various fields like computational biology, structured text databases, image analysis, automated theorem proving, and compiler optimization, where it can be crucial to have effective distance measures able to capture different types

ICTCS'24: Italian Conference on Theoretical Computer Science, September 11–13, 2024, Torino, Italy
*Corresponding author.
†These authors contributed equally.
dolcedanilo1995@gmail.com (D. G. Dolce); sabrina.mantaci@unipa.it (S. Mantaci); giuseppe.romana01@unipa.it (G. Romana); giovanna.rosone@unipi.it (G. Rosone); marinella.sciortino@unipa.it (M. Sciortino)
0000-0002-9200-0520 (S. Mantaci); 0000-0002-3489-0684 (G. Romana); 0000-0001-5075-1214 (G. Rosone); 0000-0001-6928-0168 (M. Sciortino)

of operations or transformations on the trees used to model data. A survey on the methods for comparing labeled trees based on simple local operations of deleting, inserting, and relabeling nodes can be found in [1]. In Bioinformatics, the trees are also used to model cancer phylogenies. In this context comparing trees typically means analyzing and comparing the genetic mutations and progression of cancer cells over time. In fact, according to the clonal theory of cancer [2], each node in these trees is labeled to represent a distinct genetic mutation or a group of mutations, and the edges represent the evolutionary pathways that these mutations have taken. The comparison between these trees aims to understand the evolutionary history of the cancer, identify common pathways of mutation, and potentially uncover patterns that could lead to better treatment strategies [3, 4, 5]. Several distance measures have been recently introduced with the aim to compare the topology of the trees and the mutations involved (see [6, 7, 8, 9, 10] and references therein). Other recent approaches are based on metaheuristics [11].

Motivated by the application in the study of cancer phylogeny, in this paper we focus our attention on the comparison between fully labeled trees, i.e. assuming that each node, whether internal or a leaf, has a label over a finite alphabet. However, the methodology we present in this paper can also be extended to the case of multi-labeled trees.

Here we address the problem of comparing unordered labeled trees by exploiting a linearization of the trees produced by the eXtended Burrows-Wheeler Transform (XBWT) [12, 13]. The XBWT is an elegant transformation that extends to trees the functionalities of the well-known Burrows-Wheeler Transform (denoted as BWT [14]), which is instead defined on strings and has been introduced in the context of data compression. The XBWT is applied to a labeled tree and emits, in addition to a permutation of the labels of the tree (tree linearization), a sequence of bits that makes the transformation reversible. This output is compressible and efficiently searchable. This is particularly useful for applications in Bioinformatics and XML document processing.

In this paper, we are not interested in the aspects related to compression and indexing. For this reason, here we will not use the full output of the XBWT, but only the tree linearization it produces. More in detail, we extend to a pair of trees the linearization computed by XBWT with the aim of measuring how the labels of the nodes of the two trees are mixed in the output of this transformation. We therefore define a novel class of distances d_k between two trees through a partition of the linearization induced by the lexicographically sorted prefixes of length $k \geq 1$ of the parent-to-root paths of the nodes of the tree, eventually extended to length k with an appropriate special character. To define these measures, the Jaccard distance is used in each element of such partition. We prove that such measures are pseudometrics. Note that the measures d_k become metrics when all the labels within each tree are distinct.

Furthermore, we study the sensitivity of the d_k measures with respect to some operations on trees, such as the insertion or deletion of a subtree, subtree swapping, and the exchange of labels between two nodes.

Finally, to show the robustness and effectiveness of our method, we have also analyzed the behavior of the d_k measures on simulated datasets obtained by applying a sequence of operations on randomly generated trees. Here we present the results obtained for the case $k = 1$ and when trees with distinct labels are considered.

In this paper, we do not focus on implementation aspects, but rather on methodological issues. The problems related to space and time complexity will be addressed in a subsequent full version

of the paper. To our knowledge, this approach is innovative compared to the measures used in the literature for labeled tree comparison. The idea of comparing two combinatorial structures by measuring how they mix within a common structure has been used in the context of string comparison through the use of an extension to a collection of sequences of the Burrows-Wheeler Transform [15] but with different output partitioning strategies [16, 17]. Such an extension has been largely applied for comparing biological sequences (see [18] and references therein). Moreover, it has been recently used in [19] to reconstruct the phylogenetic tree of a collection of biological sequences. We believe that the methodology introduced in this paper can also provide new insights in the context of string comparison.

2. Preliminaries

Let $\Sigma = \{a_1, a_2, \ldots, a_\sigma\}$ be a finite ordered alphabet with $a_1 < a_2 < \ldots < a_\sigma$, where $<$ denotes the standard lexicographic order. We denote by Σ^* the set of words over Σ. Given a finite word $w \in \Sigma^*$, let n be the length of w, denoted $|w|$. We also denote by $w[i]$ the i-th letter in w for any $1 \leq i \leq n$, therefore $w = w[1]w[2] \ldots w[n]$, and we denote by $w[i, j]$ the substring $w[i]w[i + 1] \cdots w[j]$. A *prefix* is a substring of the form $w[1, i]$ for some i, and a *suffix* one of the form $w[i, |w|]$. Given two strings w and v, we denote by $lcp(w, v)$ the length of the *longest common prefix* (LCP) of w and v, i.e., $lcp(w, v) = \max\{i \mid w[1, i] = v[1, i]\}$.

A rooted tree $T = (V, E)$, with V set of nodes and $E \subseteq V \times V$ set of edges, is a directed connected acyclic graph where: 1. all the nodes have one in-edge, except the root that has no in-edges; 2. all nodes have zero (*leaves*) or more (*internal nodes*) out-edges. The *size* of a tree T, denoted by $|T|$, is equal to the number $|V|$ of its nodes. Given a tree T, we denote by $L(T)$ the set of its leaves. If there is an edge from node u to node v, then u is the *parent* of v and v is the *child* of u. A *path* in the tree is a sequence of nodes $v_1, v_2, \ldots v_n$ where v_i is parent of v_{i-1} for all $1 \leq i \leq n - 1$. If $i < j$ then v_i is *ancestor* of v_j, and v_j is *descendent* of v_i. The *depth* of a node is the length of the path from its parent to the root. The depth of the root is 0. For a given $k > 0$, we denote by $T^{\leq k}$ the subtree of T from the root up to the nodes at depth k. Two nodes that are children of the same node are called *siblings*. The tree T is an *ordered* tree if a left-to-right order among siblings in T is given, otherwise it is *unordered*. A tree $T = (V, E)$ is *labeled* over the alphabet Σ if it is defined a labeling function $\ell \colon V \to \Sigma$ that associates a letter from Σ to each node of T. For each node $\mathbf{x}$ of a labeled tree, we denote by $\pi(\mathbf{x})$ the string obtained as the concatenation of the labels in the path from the parent node of $\mathbf{x}$ to the root of the tree. Let us denote by $S_\pi(T)$ the multiset of all the strings $\pi(\mathbf{x})$ for every node $\mathbf{x} \in T$. If T is an ordered tree, S_π is populated with the parent-to-root string paths of the tree nodes visited in pre-order.

The eXtended Burrows-Wheeler Transform (XBWT) of an ordered labeled tree T, denoted by $\mathrm{xbw}(T)$, linearizes the tree with a string, denoted as $S_\alpha(T)$, obtained as a concatenation of the labels of all nodes, ordered according to the lexicographical sorting of their parent-to-root string paths in $S_\pi(T)$.

In this paper, we will consider labeled trees where either all the nodes have different labels, or nodes with equal labels are allowed, but only if they are not siblings and appear in different root-to-leaf paths. Moreover, for a technical reason, we add a child to each leaf, labeled with a

special symbol $\$ \notin \Sigma$. We exclude these nodes to count the size $|T|$. Observe that for each tree T extended in this way, the set $L(T)$ consists solely of nodes labeled with \$. We denote by $\mathcal{A}_\Sigma$ the set of all such trees.

Let X and Y be two sets. The Jaccard distance D_J between X and Y is defined from the Jaccard coefficient J of similarity for X and Y, as follows:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}, \quad D_j(X,Y) = 1 - J(X,Y) = \frac{|X \cup Y| - |X \cap Y|}{|X \cup Y|}.$$

We further assume that whenever $X = Y = \emptyset$, $J(X,Y) = 1$, and $D_J(X,Y) = 0$. Observe that the measure D_J is a metric. Obviously, $0 \leq D_J(X,Y) \leq 1$ and $D_J(X,Y) = 0$ iff $X = Y$, and $D_J(X,Y) = 1$ iff $X \cap Y = \emptyset$ and $X \cup Y \neq \emptyset$.

3. Linearization of Pairs of Trees via XBWT

In this section, we aim to define a linearization for pairs of ordered trees using an XBWT-based approach. Specifically, we define $\mathrm{xbw}(T_1, T_2)$ as the string over the alphabet $\Sigma \times \{1, 2\}$ defined in the following. Note that such a linearization can be defined for every pair of trees. However, motivated by practical applications, we assume here that each tree contains distinct labels or that it may contain repeated labels but only in different root-to-leaf paths and not for nodes with the same parent. Let us denote by $S_\pi(T_1, T_2)$ the multiset of all the strings $\pi(\mathbf{x})$ for every node $\mathbf{x}$ in the trees T_1 and T_2. Let us denote by $S_\alpha(T_1, T_2)$ the string obtained by concatenating the labels of all the nodes of T_1 and T_2 ordered according to the lexicographical sorting of the paths in $S_\pi(T_1, T_2)$.

The output $\mathrm{xbw}(T_1, T_2)$ is a sequence of pairs $(\ell(\mathbf{x}), t)$, where $\ell(\mathbf{x}) \in S_\alpha$ and t is the flag assuming value 1 or 2 if $\mathbf{x}$ comes from T_1 or T_2, respectively. However, for simplicity of exposition, in the figures and tables, we replace the flags with the full names of the trees considered.

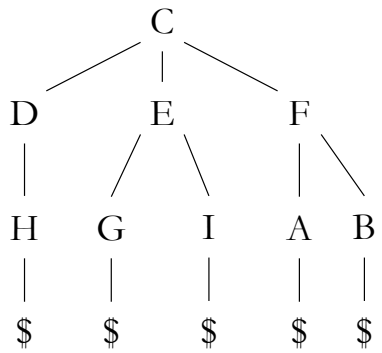
We enrich the definition of XBWT of the two trees T_1 and T_2 with the LCP array, defined as $LCP[1] = 0$ and $LCP[i] = lcp(S_\pi[i], S_\pi[i-1])$ for any $1 < i < |T_1|+|T_2|+|L(T_1)|+|L(T_2)|$.

Example 1. *Let T_1 and T_2 be the pair of trees depicted in Fig. 1a and in Fig. 1b, respectively. The output of $\mathrm{xbw}(T_1, T_2)$ is represented in the table showed in Fig. 1c. We remark that, by construction, the first two lines in the table contain the roots of T_1 and T_2, respectively. The last column in the table contains the values of the LCP array.*

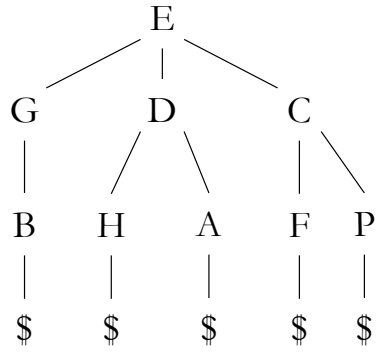
Note that the parent-to-root string paths of two sibling nodes are equal. If the trees are ordered, the relative order of such paths is determined by the left-to-right order of the nodes. However, the distance measures we will introduce in the next section consider the labels of such nodes as a set, thus their relative order is not relevant. Then, they could be applied to unordered trees as well.

4. A new Class of Distance Measures between Trees

In this section, we introduce a new measure for comparing two unordered trees, T_1 and T_2, and we will prove that this measure is a pseudometric. To compute this measure, for a given



(a) Tree T_1



(b) Tree T_2

Index	Flag	S_α	S_π	LCP
1	T_1	C	ϵ	0
2	T_2	E	ϵ	0
3	T_2	$	ADE	0
4	T_1	$	AFC	1
5	T_1	$	BFC	0
6	T_2	$	BGE	1
7	T_1	D	C	0
8	T_1	E	C	1
9	T_1	F	C	1
10	T_2	F	CE	1
11	T_2	P	CE	2
12	T_1	H	DC	0
13	T_2	H	DE	1
14	T_2	A	DE	2
15	T_2	G	E	0
16	T_2	D	E	1
17	T_2	C	E	1
18	T_1	G	EC	1
19	T_1	I	EC	2
20	T_1	A	FC	0
21	T_1	B	FC	2
22	T_2	$	FCE	2
23	T_2	B	GE	0
24	T_1	$	GEC	2
25	T_1	$	HDC	0
26	T_2	$	HDE	2
27	T_1	$	IEC	0
28	T_2	$	PCE	0

(c) $\mathrm{xbw}(T_1, T_2)$ and LCP array

Figure 1: Given the trees T_1 and T_2 shown in (a) and (b), the output of $\mathrm{xbw}(T_1, T_2)$ is shown in (c) and consists of the two columns S_α and $Flag$. The LCP array is contained in the last column of the table.

$k > 0$ we map each word in S_π of length $< k$ into a new word over the right-padded alphabet $\Sigma^k_\# = \bigcup_{k' \in [0,k]} \left\{ \Sigma^{k'} \cdot \{\#\}^{k-k'} \right\}$. To improve readability, we omit k whenever it is clear from the context.

In defining the measure, we refer to a partition of $S_\alpha(T_1, T_2)$ according to the values of the LCP array, defined as follows.

Definition 1. LCP-based partition of order k. *Given a positive integer k, let us consider the list of strings $S^{(k)}_\pi(T_1, T_2)$ obtained by extending all the strings in $S_\pi(T_1, T_2)$ to the length k with the $\#$ fill character, where $\#$ is smaller than all the characters in $\Sigma \cup \{\$\}$. Let us denote by LCP_k the LCP array for $S^{(k)}_\pi(T_1, T_2)$. We denote by $\mathcal{P}_k$ the partition of the interval $[1, |T_1| + |T_2| + |L(T_1)| + |L(T_2)|]$ obtained as union of the following intervals:*

- $\bigcup_{1 \le i \le j < n} \{[i, j] \mid LCP_k[i] < k,\ LCP_k[j+1] < k,\ LCP_k[t] \ge k, \forall t \in [i+1, j]\}$;

- $[i, |T_1| + |T_2| + |L(T_1)| + |L(T_2)|] \mid LCP_k[i] < k,\ LCP_k[t] \geq k, \forall t \in [i+1, |T_1| + |T_2| + |L(T_1)| + |L(T_2)|]$.

Note that each interval $[i, j]$ in the partition $\mathcal{P}_k$ can be uniquely associated with a word in Σ^k, which is a k-length prefix of the longest common prefix of the strings, possibly extended to length k with the padding character $\#$, in the list $S_\pi(T_1, T_2)$ contained in the interval $[i, j]$.

We denote by $\lambda([i, j])$ such a word, and the indexes i and j are denoted by $f(\lambda[i, j])$ and $l(\lambda[i, j])$, respectively. For each pair of trees T_1 and T_2, we denote by $\Lambda(T_1, T_2) = \{\lambda[i, j] \mid [i, j] \in \mathcal{P}_k\}$, that is the set of k-length words from $\Sigma^k_\#$ that appear as prefix of some word in $S^{(k)}_\pi(T_1, T_2)$. Moreover, for $t \in \{1, 2\}$, we denote by $S_t([i, j]) = \{c \in \Sigma \cup \{\$\} \mid (c, t) \in \text{xbw}(T_1, T_2)[i, j]\}$, that is the set of labels in $S_\alpha[i, j]$ coming from the tree T_t. Analogously, for each tree and word $w \in \Sigma^k_\#$, we define the set $\Gamma^{(k)}_T(w) = \{c \in \Sigma \cup \{\$\} \mid c \cdot w \in S^{(k)}_\pi(T)\}$, that is the set of letters c in $\text{xbw}(T)[i, j]$ such that $\lambda[i, j] = w$.

Definition 2. *The distance measure d_k is defined as:*

$$d_k(T_1, T_2) = \sum_{[i,j] \in \mathcal{P}_k} D_J(S_1[i, j], S_2[i, j]),$$

where D_J is the Jaccard distance.

Example 2. *Let us consider the two trees T_1 and T_2 depicted in Figure 2. The partition of $S_\alpha(T_1, T_2)$ induced by $\mathcal{P}_k$ (with $k = 1$ and $k = 2$) is shown in Table 1. By applying the Jaccard distance D_J to each element of the partition,*

$$d_1(T_1, T_2) = \sum_{[i,j] \in \mathcal{P}_1} D_J(S_1[i, j], S_2[i, j]) = 0 + 0 + 0 + 0 + 0 + 0 = 0,$$

and

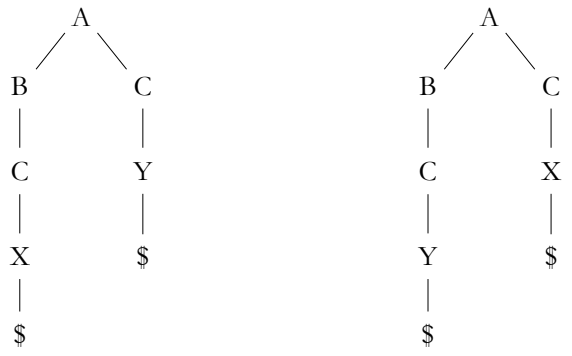
$$d_2(T_1, T_2) = \sum_{[i,j] \in \mathcal{P}_2} D_J(S_1[i, j], S_2[i, j]) = 0 + 0 + 0 + 1 + 1 + 0 + 0 = 2.$$

Proposition 1. *The distance measure d_k is a pseudometric on $\mathcal{A}_\Sigma$. In fact, the following statements hold:*

1. *for each pair of trees T_1, T_2, $d_k(T_1, T_2) \geq 0$*
2. *$d_k(T, T) = 0$ for every tree T;*
3. *for each pair of trees T_1, T_2, $d_k(T_1, T_2) = d_k(T_2, T_1)$;*
4. *for each triplets of trees T_1, T_2, T_3, $d_k(T_1, T_2) \leq d_k(T_1, T_3) + d_k(T_2, T_3)$.*

Proof.

1. The statement follows from the definition.
2. From the definition, we have that $d_k(T, T) = 0$ since $D_J(S_1[i, j], S_2[i, j]) = 0$ for all $[i, j] \in \mathcal{P}_k$.



(a) T_1 (b) T_2

Figure 2: Two labeled trees with the nodes X and Y swapped. The tree on the right T_2 (b) is obtained from the tree on the left T_1 (a) by swapping the subtrees rooted in X and Y, respectively.

Index	Flag	S_α	S_π	LCP	$S_\pi^{(1)}$	LCP_1	$S_\pi^{(2)}$	LCP_2
1	T_1	A	ϵ	0	#	0	##	0
2	T_2	A	ϵ	0	#	1	##	2
3	T_1	B	A	0	**A**	0	**A**#	0
4	T_2	B	A	1	**A**	1	**A**#	2
5	T_1	C	A	1	**A**	1	**A**#	2
6	T_2	C	A	1	**A**	1	**A**#	2
7	T_1	C	BA	0	**B**A	0	**BA**	0
8	T_2	C	BA	2	**B**A	2	**BA**	2
9	T_1	Y	CA	0	**C**A	0	**CA**	0
10	T_2	X	CA	2	**C**A	2	**CA**	2
11	T_1	X	CBA	1	**C**BA	1	**CB**A	1
12	T_2	Y	CBA	3	**C**BA	3	**CB**A	3
13	T_2	$	XCA	0	**X**CA	0	**XC**A	0
14	T_1	$	XCBA	2	**X**CBA	2	**XC**BA	2
15	T_1	$	YCA	0	**Y**CA	0	**YC**A	0
16	T_2	$	YCBA	2	**Y**CBA	2	**YC**BA	2

Table 1
The table shows how the partition $\mathcal{P}_k$ relative to the two trees depicted in Fig. 2 can vary as k changes. Specifically, the partitions $\mathcal{P}_1$ and $\mathcal{P}_2$ are illustrated. In the columns $S_\pi^{(1)}$ and $S_\pi^{(2)}$ we highlight in bold the letters of the prefixes which define the partitions $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively. Partition $\mathcal{P}_2$ contains the same intervals as $\mathcal{P}_1$, except for two intervals obtained as a refinement of the interval $[9, 12]$ from $\mathcal{P}_1$. This refinement is indicated with a dashed line. The table also shows the output S_α of XBWT applied to the two trees and LCP arrays.

3. The statement follows from the fact that the Jaccard distance is a metric and then the symmetric property holds.

4. Since the Jaccard distance is a metric and the triangle inequality holds, for each $w \in \Sigma^k$

holds that

$$\begin{aligned}
d_k(T_1, T_2) &= \sum_{[i,j]\in\mathcal{P}_k} D_j(S_1[i,j], S_2[i,j]) \\
&= \sum_{w\in\Lambda(T_1,T_2)} D_J(\Gamma^{(k)}_{T_1}(w), \Gamma^{(k)}_{T_2}(w)) \\
&= \sum_{w\in\Sigma^k_\#} D_J(\Gamma^{(k)}_{T_1}(w), \Gamma^{(k)}_{T_2}(w)) \\
&\leq \sum_{w\in\Sigma^k_\#} \Big(D_J(\Gamma^{(k)}_{T_1}(w), \Gamma^{(k)}_{T_3}(w)) + D_J(\Gamma^{(k)}_{T_2}(w), \Gamma^{(k)}_{T_3}(w)) \Big) \\
&= \sum_{w\in\Sigma^k_\#} D_J(\Gamma^{(k)}_{T_1}(w), \Gamma^{(k)}_{T_3}(w)) + \sum_{w\in\Sigma^k_\#} D_J(\Gamma^{(k)}_{T_2}(w), \Gamma^{(k)}_{T_3}(w)) \\
&= \sum_{w\in\Lambda(T_1,T_3)} D_J(\Gamma^{(k)}_{T_1}(w), \Gamma^{(k)}_{T_3}(w)) + \sum_{w\in\Lambda(T_2,T_3)} D_J(\Gamma^{(k)}_{T_2}(w), \Gamma^{(k)}_{T_3}(w)) \\
&= d_k(T_1, T_3) + d_k(T_2, T_3).
\end{aligned}$$

This concludes the proof. □

Remark 1. *Note that, in general, d_k is not a metric. In fact, for a given k there could exist pairs of distinct trees T_1 and T_2 with $d_k(T_1T_2) = 0$, i.e. d_k is not able to distinguish T_1 and T_2. Despite this, one can always find a $k' > k$ such that $d_{k'}(T_1T_2) > 0$, i.e. $d_{k'}$ makes T_1 and T_2 distinguishable. This fact is shown in Fig. 2, where the trees (a) and (b) are indistinguishable for d_1, i.e. their d_1 distance is 0, but they become distinguishable when d_2 distance is applied.*

By using a similar argument as in the proof of Proposition 1, it is possible to prove the following corollary.

Corollary 1. *Let $\mathcal{T}$ be the set of all unordered trees whose nodes are labeled by distinct symbols. Then d_k is a metric over $\mathcal{T}$ for each $k \geq 1$, i.e. for each pair of trees $T_1, T_2 \in \mathcal{T}$, $d_k(T_1, T_2) = 0$ if and only if $T_1 = T_2$.*

What we have stated in the previous remark can be more generally formalized in the following proposition.

Proposition 2. *Given two trees T_1 and T_2 in $\mathcal{A}_\Sigma$, $d_k(T_1, T_2) \leq d_{k+1}(T_1, T_2)$, for each $k \geq 1$.*

Note that if we consider the set $\mathcal{A}_\Sigma$ of all labeled trees, the d_k measures can be used to define a class of dissimilarity measures normalized with respect to the number of words in $\Sigma^k_\#$, i.e. $1 + \sum_{i\in[1,k]} |\Sigma^i|$.

5. Sensitivity to Operations on Trees

In this section, we evaluate how the distance d_k changes when a swap of subtrees, label exchanges, insertion, or removal of nodes are applied to a tree. Other operations will be considered in the full version of the paper.

Proposition 3. *Let T_1 and T_2 be two unordered trees such that T_2 is obtained from T_1, by swapping two disjoint subtrees $T_{\mathbf{v}_1}$ and $T_{\mathbf{v}_2}$, rooted in the nodes $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively. Then, $d_1(T_1, T_2) \leq 2$, and $d_k(T_1, T_2) \leq 2(|T_{\mathbf{v}_1}^{\leq k-2}| + |T_{\mathbf{v}_2}^{\leq k-2}|) + 2$ for all $k > 1$, where $|T_{\mathbf{v}_1}^{\leq k-1}|$ and $|T_{\mathbf{v}_2}^{\leq k-1}|$ denote the number of nodes at depth at most $k-1$ in the subtrees $T_{\mathbf{v}_1}$ and $T_{\mathbf{v}_2}$, respectively.*

Proof. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the two nodes in T_1 that are the roots of the subtrees swapped to obtain T_2, and let $\mathcal{P}_k$ be the LCP-based partition of order k for T_1 and T_2. Let us denote by $\overline{\pi(\mathbf{v}_1)}$ and $\overline{\pi(\mathbf{v}_2)}$ the parent-to–root string path of $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively, both possibly padded up to length k by using the character $\#$. The upper bound for the distance $d_k(T_1, T_2)$ is obtained when all the labels in T_1 are distinct (and therefore in T_2 as well) and if each of the two nodes is the only child of their respective parent. In this case, for all $k > 0$, $D_J(S^{(1)}[f(\overline{\pi(\mathbf{v}_1)}), l(\overline{\pi(\mathbf{v}_1)})], S^{(2)}[f(\overline{\pi(\mathbf{v}_1)}), l(\overline{\pi(\mathbf{v}_1)})]) = 1$ and $D_J(S^{(1)}[f(\overline{\pi(\mathbf{v}_2)}), l(\overline{\pi(\mathbf{v}_2)})], S^{(2)}[f(\overline{\pi(\mathbf{v}_2)}), l(\overline{\pi(\mathbf{v}_2)})]) = 1$. Moreover, for all $k > 1$, each node $\mathbf{z}$ in the subtrees $T_{\mathbf{v}_1}$ and $T_{\mathbf{v}_2}$ at depth at most $k-2$ from the root increases by 2 the distance d_k. All the other intervals in the partition $\mathcal{P}_k$ give a zero contribution to the distance. Then, the thesis follows. □

The following proposition provides an evaluation of the distance between two trees when one is obtained from the other by removing or inserting an entire subtree.

Proposition 4. *Let T_1 and T_2 be two unordered trees such that T_2 is obtained by removing from T_1 a subtree $T_{\mathbf{x}} \neq T_1$ with root $\mathbf{x}$. Then, $d_k(T_1, T_2) \leq |T_{\mathbf{x}}| + 1$.*

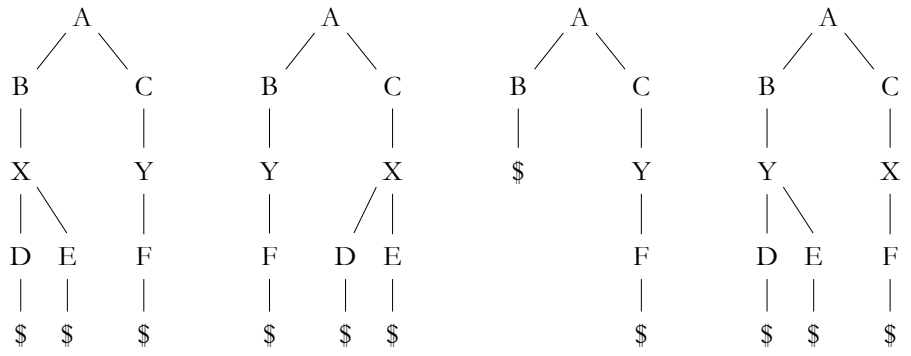
Proof. Let $T_{\mathbf{x}}$ be the subtree rooted in $\mathbf{x}$ removed from T_1 to obtain T_2. Let $\mathcal{P}_k$ be the LCP-based partition of order k for T_1 and T_2. Let us denote by $\overline{\pi(\mathbf{x})}$ the parent-to–root string path of $\mathbf{x}$ possibly padded up to length k by using the character $\#$.

The upper bound for the distance $d_k(T_1, T_2)$ is obtained when $\mathbf{x}$ has no sibling nodes and $\overline{\pi(\mathbf{x})}$ does not have any common k-length prefix with other parent-to-root string paths. In this case, the parent of $\mathbf{x}$ and each node of $T_{\mathbf{x}}$, leaves excluded, increases by 1 the distance d_k. □

The following proposition considers the operation of swapping the labels of two nodes. Note that the label swap does not involve any descendants of the nodes we are considering.

Proposition 5. *Let T_1 and T_2 be two unordered trees such that T_2 is obtained from T_1, by swapping the label of the nodes $\mathbf{v}_1$ and $\mathbf{v}_2$. Let us denote by $T_{\mathbf{v}_1}$ and $T_{\mathbf{v}_2}$ the subtrees rooted in the nodes $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively. Then, $d_1(T_1, T_2) \leq 4$, and $d_k(T_1, T_2) \leq 2(|T_{\mathbf{v}_1}^{\leq k-1}| + |T_{\mathbf{v}_2}^{\leq k-1}|) + 2$ for all $k > 1$, where $|T_{\mathbf{v}_1}^{\leq k-1}|$ and $|T_{\mathbf{v}_2}^{\leq k-1}|$ denote the number of nodes at depth at most $k-1$ in the subtrees $T_{\mathbf{v}_1}$ and $T_{\mathbf{v}_2}$, respectively.*

Proof. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be the two nodes in T_1 whose labels are swapped in T_1 to obtain T_2. This means that in T_2 we can find the subtrees $T'_{\mathbf{v}_1}$ and $T'_{\mathbf{v}_2}$ obtained by swapping the roots of the subtrees $T_{\mathbf{v}_1}$ and $T_{\mathbf{v}_2}$ in T_1 respectively. Let $\mathcal{P}_k$ be the LCP-based partition of order k for T_1 and T_2. Let us denote by $\overline{\pi(\mathbf{v}_1)}$ and $\overline{\pi(\mathbf{v}_2)}$ the parent–to–root string path of $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively, both possibly padded up to length k by using the character $\#$. In order to obtain the upper bound for the distance $d_k(T_1, T_2)$ we assume that all the labels in T_1 are distinct (and therefore in T_2 as



(a) T_1 (b) T_2 (c) T_3 (d) T_4

Figure 3: The trees T_2, T_3, and T_4 are obtained from T_1 by applying respectively a swap of the subtrees rooted in X and Y, removing the subtree rooted in X, and swapping the labels X and Y.

well), each of the two nodes is the only child of their respective parent and the subtrees $T_{\mathbf{v}_1}$ and $T_{\mathbf{v}_2}$ have distinct labels. In this case, $D_J(S^{(1)}[f(\overline{\pi(\mathbf{v}_1)}), l(\overline{\pi(\mathbf{v}_1)})], S^{(2)}[f(\overline{\pi(\mathbf{v}_1)}), l(\overline{\pi(\mathbf{v}_1)})]) = 1$, and symmetrically $D_J(S^{(1)}[f(\overline{\pi(\mathbf{v}_2)}), l(\overline{\pi(\mathbf{v}_2)})], S^{(2)}[f(\overline{\pi(\mathbf{v}_2)}), l(\pi(\mathbf{v}_2))]) = 1$. For $k = 1$, observe that $D_J(S^{(1)}[f(\ell(\mathbf{v}_1)), l(\ell(\mathbf{v}_1))], S^{(2)}[f(\ell(\mathbf{v}_1)), l(\ell(\mathbf{v}_1))]) = 1$, and equivalently $D_J(S^{(1)}[f(\ell(\mathbf{v}_2)), l(\ell(\mathbf{v}_2))], S^{(2)}[f(\ell(\mathbf{v}_2)), l(\ell(\mathbf{v}_2))]) = 1$. On the other hand, for all $k > 1$, each node $\mathbf{z}$ in the subtrees $T_{\mathbf{v}_1}$ and $T_{\mathbf{v}_2}$ (equivalently in $T'_{\mathbf{v}_1}$ and $T'_{\mathbf{v}_2}$) at depth at most $k-2$ from the root increases by 2 the distance d_k. All the other intervals in the partition $\mathcal{P}_k$ give a zero contribution to the distance. Then, the thesis follows. □

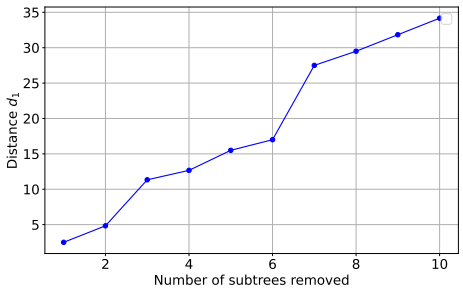
	Flag	S_α	S_π	$S_\pi^{(2)}$	LCP_2	D_J
1	T_1	A	ε	##	0	0
2	T_2	A	ε	##	2	
3	T_1	B	A	**A**#	0	0
4	T_2	B	A	**A**#	2	
5	T_1	C	A	**A**#	2	
6	T_2	C	A	**A**#	2	
7	T_1	X	BA	**BA**	0	1
8	T_2	Y	BA	**BA**	2	
9	T_1	Y	CA	**CA**	0	1
10	T_2	X	CA	**CA**	2	
11	T_1	$	DXBA	**DX**BA	0	0
12	T_2	$	DXCA	**DX**CA	2	
13	T_1	$	EXBA	**EX**BA	0	0
14	T_2	$	EXCA	**EX**CA	2	
15	T_2	$	FYBA	**FY**BA	0	0
16	T_1	$	FYCA	**FY**CA	2	
17	T_1	D	XBA	**XB**A	0	1
18	T_1	E	XBA	**XB**A	3	
19	T_2	D	XCA	**XC**A	1	1
20	T_2	E	XCA	**XC**A	3	
21	T_2	F	YBA	**YB**A	0	1
22	T_1	F	YCA	**YC**A	1	1

	Flag	S_α	S_π	$S_\pi^{(2)}$	LCP_2	D_J
1	T_1	A	ε	##	0	0
2	T_3	A	ε	##	2	
3	T_1	B	A	**A**#	0	0
4	T_3	B	A	**A**#	2	
5	T_1	C	A	**A**#	2	
6	T_3	C	A	**A**#	2	
7	T_1	X	BA	**BA**	0	1
8	T_3	$	BA	**BA**	2	
9	T_1	Y	CA	**CA**	0	0
10	T_3	Y	CA	**CA**	2	
11	T_1	$	DXBA	**DX**BA	0	1
12	T_1	$	EXBA	**EX**BA	0	1
13	T_1	$	FYCA	**FY**CA	0	0
14	T_3	$	FYCA	**FY**CA	2	
15	T_1	D	XBA	**XB**A	0	1
16	T_1	E	XBA	**XB**A	3	
17	T_1	F	YCA	**YC**A	0	0
18	T_3	F	YCA	**YC**A	3	

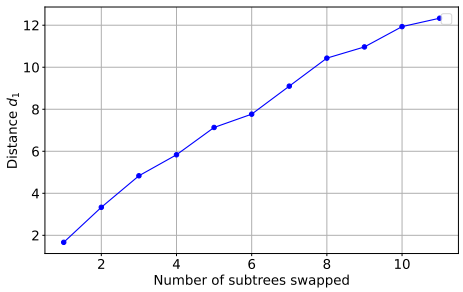
	Flag	S_α	S_π	$S_\pi^{(2)}$	LCP_2	D_J
1	T_1	A	ε	##	0	0
2	T_4	A	ε	##	2	
3	T_1	B	A	**A**#	0	0
4	T_4	B	A	**A**#	2	
5	T_1	C	A	**A**#	2	
6	T_4	C	A	**A**#	2	
7	T_1	X	BA	**BA**	0	1
8	T_4	Y	BA	**BA**	2	
9	T_1	Y	CA	**CA**	0	1
10	T_4	X	CA	**CA**	2	
11	T_1	$	DXBA	**DX**BA	0	1
12	T_4	$	DYBA	**DY**BA	1	1
13	T_1	$	EXBA	**EX**BA	0	1
14	T_4	$	EYBA	**EY**BA	1	1
15	T_4	$	FXCA	**FX**CA	0	1
16	T_1	$	FYCA	**FY**CA	1	1
17	T_1	D	XBA	**XB**A	0	1
18	T_1	E	XBA	**XB**A	3	
19	T_4	F	XCA	**XC**A	1	1
20	T_4	D	YBA	**YB**A	0	1
21	T_4	E	YBA	**YB**A	2	
22	T_1	F	YCA	**YC**A	1	1

Table 2
The tables show the phases of computation of $d_2(T_1, T_2) = 6$, $d_2(T_1, T_3) = 4$, $d_2(T_1, T_4) = 12$, where T_1, T_2, T_3, T_4 are depicted in Figure 3.

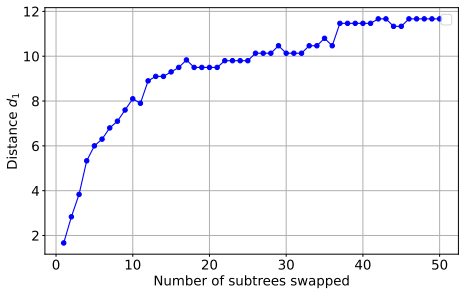
In Fig. 3 and Table 2 is displayed a worst–case example for each of the cases described in Propositions 3, 4, and 5, showing that the three upper–bounds are tight.



(a) Subtree removals



(b) Subtree swap



(c) Subtree swap with repetitions

Figure 4: Behaviour of the distance d_1 when different operations on trees are applied to a randomly generated fully labeled tree with 26 nodes with distinct labels.

We have analyzed the behavior of the d_k measures on simulated data, by evaluating the distance values after the application of perturbations, which consist of subtree removal and subtree swapping operations, on randomly generated trees. We conclude this section by showing in Fig. 4a the results obtained by considering the values of the d_1 measure after applying a maximum of 10 subtree removals, chosen randomly, on randomly generated trees having distinct labels and such that each internal node has at most three children. We also show the behavior of the d_1 measure when the operation applied on a randomly generated tree is the subtree swapping. In Fig. 4b, we consider the case where each subtree can be swapped at most once, and in Fig. 4c, successive swapping operations on the same subtree are allowed.

A more extensive description of the experiments will be presented in a subsequent extended version of the paper.

6. Conclusions and Further Work

In this paper, we introduced a new class of distances for unordered fully labeled trees. Theoretical results, as well as preliminary experimental analyses on simulated data, show that these measures can effectively capture some operations on trees such as removal and insertion of subtrees, subtree swapping, and label swapping. These distances are defined using an LCP-based partition of a linearization of trees, defined by a generalization of the XBWT. We have proven that the

measures in this class are pseudometrics and become metrics when trees having distinct labels are considered. In the general case in which repeated labels are allowed, we have observed that for any given collection of trees, there exists an integer k for which d_k is a metric over the dataset. It would be interesting to experimentally determine for any given dataset of trees the smallest value of k for which d_k is a metric.

We have focused on combinatorial aspects related to the extension of XBWT to compare pairs of trees. The algorithmic issues related to the efficiency of computing these measures, as well as the use of this transformation for finding common subtrees, will be explored in the full paper.

Our preliminary experimental evaluation shows that our method is able to capture structural differences and similarities between unordered trees, with significant possible implications for computational biology, XML data processing, and hierarchical clustering. We intend to evaluate the behavior of the d_k measures concerning a more comprehensive set of tree operations, as well as to test these measures on real datasets for the study of cancer phylogenies. To this end, we plan to extend the methodology introduced in this paper to the more general case of multi-labeled trees, by using the Jaccard distance defined on multisets, and compare our approach with others existing in the literature.

Acknowledgments

Sabrina Mantaci is partially supported by INdAM-GNCS Project 2024- CUP E53C23001670001 ("Proprietà combinatorie e distanze basate su parole da evitare"). Giuseppe Romana, Giovanna Rosone, and Marinella Sciortino are partially supported by MUR PRIN project no. 2022YRB97K - "PINC" (Pangenome INformatiCs: From Theory to Applications), and partially funded by the INdAM-GNCS Project CUP E53C23001670001 ("Compressione, indicizzazione, analisi e confronto di dati biologici"). Giovanna Rosone is partially supported by PAN-HUB T4-AN-07 ("Hub multidisciplinare e interregionale di ricerca e sperimentazione clinica per il contrasto alle pandemie e all'antibiotico resistenza"), CUP I53C22001300001. Sabrina Mantaci and Marinella Sciortino are partially supported by the project "ACoMPA – Algorithmic and Combinatorial Methods for Pangenome Analysis" (CUP B73C24001050001) funded by the NextGeneration EU programme PNRR ECS00000017 Tuscany Health Ecosystem (Spoke 6).

References

[1] P. Bille, A survey on tree edit distance and related problems, Theoretical Computer Science 337 (2005) 217–239. doi:https://doi.org/10.1016/j.tcs.2004.12.030.

[2] P. C. Nowell, The clonal evolution of tumor cell populations: Acquired genetic lability permits stepwise selection of variant sublines and underlies tumor progression., Science 194 (1976) 23–28. doi:10.1126/science.959840.

[3] N. Beerenwinkel, C. D. Greenman, J. Lagergren, Computational cancer biology: An evolutionary perspective, PLOS Computational Biology 12 (2016) 1–12. doi:10.1371/journal.pcbi.1004717.

[4] M. Llabrés, F. Rosselló, G. Valiente, A generalized Robinson-Foulds distance for clonal trees, mutation trees, and phylogenetic trees and networks, in: BCB '20: 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, Virtual Event, USA, September 21-24, 2020, ACM, 2020, pp. 13:1–13:10. doi:10.1145/3388440.3412479.

[5] R. Schwartz, A. A. Schäffer, The evolution of tumour phylogenetics: principles and practice, Nature Reviews Genetics 18 (2017) 213–229. doi:10.1038/nrg.2016.170.

[6] G. Bernardini, P. Bonizzoni, G. Della Vedova, M. Patterson, A Rearrangement Distance for Fully-Labelled Trees, in: 30th Annual Symposium on Combinatorial Pattern Matching (CPM 2019), volume 128 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2019, pp. 28:1–28:15. doi:10.4230/LIPIcs.CPM.2019.28.

[7] Z. DiNardo, K. Tomlinson, A. Ritz, L. Oesper, Distance measures for tumor evolutionary trees, Bioinformatics 36 (2019) 2090–2097. doi:10.1093/bioinformatics/btz869.

[8] S. Ciccolella, G. Bernardini, L. Denti, P. Bonizzoni, M. Previtali, G. Della Vedova, Triplet-based similarity score for fully multilabeled trees with poly-occurring labels, Bioinformatics 37 (2020) 178–184. doi:10.1093/bioinformatics/btaa676.

[9] G. Bernardini, P. Bonizzoni, P. Gawrychowski, On Two Measures of Distance Between Fully-Labelled Trees, in: I. L. Gørtz, O. Weimann (Eds.), 31st Annual Symposium on Combinatorial Pattern Matching (CPM 2020), volume 161 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, pp. 6:1–6:16. doi:10.4230/LIPIcs.CPM.2020.6.

[10] N. Karpov, S. Malikic, M. K. Rahman, S. C. Sahinalp, A multi-labeled tree dissimilarity measure for comparing "clonal trees" of tumor progression, Algorithms for Molecular Biology 14 (2019) 17. doi:10.1186/s13015-019-0152-9.

[11] S. Ciccolella, G. Della Vedova, V. Filipović, M. Soto Gomez, Three metaheuristic approaches for tumor phylogeny inference: An experimental comparison, Algorithms 16 (2023). doi:10.3390/a16070333.

[12] P. Ferragina, F. Luccio, G. Manzini, S. Muthukrishnan, Structuring labeled trees for optimal succinctness, and beyond, in: 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05), 2005, pp. 184–193. doi:10.1109/SFCS.2005.69.

[13] P. Ferragina, F. Luccio, G. Manzini, S. Muthukrishnan, Compressing and indexing labeled trees, with applications, J. ACM 57 (2009). doi:10.1145/1613676.1613680.

[14] M. Burrows, D. J. Wheeler, A Block Sorting data Compression Algorithm, Technical Report, DIGITAL System Research Center, 1994.

[15] S. Mantaci, A. Restivo, G. Rosone, M. Sciortino, An extension of the Burrows-Wheeler Transform, Theoret. Comput. Sci. 387 (2007) 298–312. doi:10.1016/j.tcs.2007.07.014.

[16] S. Mantaci, A. Restivo, G. Rosone, M. Sciortino, A new combinatorial approach to sequence comparison, Theory Comput. Syst. 42 (2008) 411–429. doi:10.1007/s00224-007-9078-6.

[17] S. Mantaci, A. Restivo, M. Sciortino, Distance measures for biological sequences: Some recent approaches, Int. J. Approx. Reason. 47 (2008) 109–124. doi:10.1016/J.IJAR.2007.03.011.

[18] V. Guerrini, F. A. Louza, G. Rosone, Metagenomic analysis through the extended Burrows-Wheeler transform, BMC Bioinform. 21-S (2020) 299. doi:10.1186/S12859-020-03628-W.

[19] V. Guerrini, A. Conte, R. Grossi, G. Liti, G. Rosone, L. Tattini, phyBWT2: phylogeny reconstruction via eBWT positional clustering, Algorithms Mol. Biol. 18 (2023) 11. doi:10.1186/S13015-023-00232-4.