

Alignment of Cyclically Ordered Trees*

Takuya Yoshino¹ and Kouichi Hirata²

¹Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology,
Kawazu 680-4, Iizuka 820-8502, Japan

²Department of Artificial Intelligence, Kyushu Institute of Technology, Kawazu 680-4, Iizuka 820-8502, Japan

Keywords: Alignment, Alignment Distance, Segmental Alignment Distance, Cyclically Ordered Tree, Tai Mapping.

Abstract: In this paper, as unordered trees preserving the adjacency among siblings, we introduce the following three kinds of a *cyclically ordered tree*, that is, a *biordered tree* that allows both a left-to-right and a right-to-left order among siblings, a *cyclic-ordered tree* that allows cyclic order among siblings in a left-to-right direction and a *cyclic-biordered tree* that allows cyclic order among siblings in both left-to-right and right-to-left directions. Then, we design the algorithms to compute the *alignment distance* and the *segmental alignment distance* between biordered trees in $O(n^2D^2)$ time and ones between cyclic-ordered trees and cyclic-biordered trees in $O(n^2D^4)$ time, where n is the maximum number of nodes and D is the maximum degree in two given trees.

1 INTRODUCTION

Comparing tree-structured data is one of the important tasks for many research areas such as pattern recognition, natural language processing, machine learning, data mining, bioinformatics, and so on. In these researches, the tree-structured data are well regarded as *rooted labeled trees* (*trees*, for short). Also a tree is *ordered* if the left-to-right order among siblings is fixed and *unordered* otherwise.

An *edit distance* (Tai, 1979) is one of the standard distance measures between trees. The edit distance is formulated as the minimum cost to transform from a tree to another tree by applying *edit operations* of a *substitution*, a *deletion* and an *insertion* to trees.

It is known that the edit distance is closely related to a *Tai mapping* (Tai, 1979). The minimum cost of Tai mappings coincides with the edit distance (Tai, 1979). Then, whereas the problem of computing the edit distance between ordered trees is tractable (Demaine et al., 2009), one between unordered trees is MAX SNP-hard (Zhang and Jiang, 1994). This MAX SNP-hardness holds even if both trees are binary (Hirata et al., 2011).

An *alignment distance* is an alternative distance measure between trees introduced by (Jiang et al.,

1995) and applied to comparing RNA secondary structures in bioinformatics (Höchsmann et al., 2003; Schiermer and Giegerich, 2013; Shapiro and Zhang, 1990; Zhang, 1998). The alignment distance is formulated as the minimum cost of possible *alignments* (as trees) obtained by first inserting nodes labeled with spaces into two trees such that the resulting trees have the same structure and then overlaying them. In operational, the alignment distance is an edit distance such that every insertion precedes to deletions.

Kuboyama (Kuboyama, 2007) has first formulated an *alignable mapping* as a variation of the Tai mapping. Then, he has shown that the alignment distance coincides with the minimum cost of alignable mappings and the alignable mapping coincides with a less-constrained mapping (Lu et al., 2001). As same as the edit distance, whereas the problem of computing the alignment distance between ordered trees is tractable, one between unordered trees is MAX SNP-hard (Jiang et al., 1995). On the other hand, this problem becomes tractable if the degrees of unordered trees are bounded (Jiang et al., 1995).

In the above results of computing distances, we deal with either ordered or unordered trees. Note that unordered trees allow all of the permutations among siblings. On the other hand, several applications require to allow just some permutations, not all of the permutations, among siblings. For example, when representing graphs with cyclic compounds such as monosaccharides in glycans (Hizukuri et al.,

*This work is partially supported by Grant-in-Aid for Scientific Research 24240021, 24300060, 25540137, 26280085 and 26370281 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

2005) and molecules in molecular graphs (Horváth et al., 2010) as trees, the adjacency of nodes in the compounds is represented as the adjacency among siblings in the tree representation. Also, when comparing or modeling RNA secondary structures as trees (Höchsmann et al., 2003; Schiermer and Giegerich, 2013; Shapiro and Zhang, 1990; Zhang, 1998), the base pairs in nucleotides are connected with preserving the adjacency among siblings.

Hence, as unordered trees preserving the adjacency among siblings, in this paper, we formulate the following three kinds of a *cyclically ordered tree*. Let v_1, \dots, v_n be siblings from left to right. Then, we say that a tree is *biordered* if it allows two orders v_1, \dots, v_n and v_n, \dots, v_1 . Also we say that a tree is *cyclic-ordered* if it allows a cyclic order $v_i, \dots, v_n, v_1, \dots, v_{i-1}$ for every i ($1 \leq i \leq n$). Furthermore, we say that a tree is *cyclic-biordered* if it allows cyclic orders $v_i, \dots, v_n, v_1, \dots, v_{i-1}$ and $v_i, \dots, v_1, v_n, \dots, v_{i-1}$ for every i ($1 \leq i \leq n$).

Since an unordered binary tree is always cyclically ordered, the problem of computing the edit distance, the segmental distance (Kan et al., 2014) and the bottom-up distance (Valiente, 2001; Kuboyama, 2007) between cyclically ordered trees is also MAX SNP-hard (Hirata et al., 2011; Yamamoto et al., 2014). On the other hand, the problems of computing the isolated-subtree (or constrained) distance (Zhang, 1995; Zhang, 1996), the accordant (or Lu's) distance (Lu, 1979; Kuboyama, 2007), the LCA-preserving (or degree-2) distance (Zhang et al., 1996) and the top-down (or degree-1) distance (Selkow, 1977; Chawathe, 1999) between unordered trees are tractable, so are the problems of computing these distances between cyclically ordered trees.

In this paper, we focus on the alignment distance and a *segmental alignment* distance, which is an alignment distance to preserve the parent-children relationship as possible (Yoshino and Hirata, 2013), between cyclically ordered trees, because the problems of computing both distances are tractable if the degrees of unordered trees are bounded. Note that the algorithms to compute all of the above tractable variations of the edit distance between unordered trees contain the maximum weighted bipartite matching algorithm (Yamamoto et al., 2014; Zhang et al., 1996) or originally the minimum cost maximum flow algorithm (Wang et al., 2003; Zhang, 1996).

On the other hand, in this paper, by directly extending the recurrences to compute the alignment distance between ordered trees (Jiang et al., 1995), we first design the algorithms to compute the alignment distance between biordered trees in $O(n^2D^2)$ time,

where n is the maximum number of nodes and D is the maximum degree in two given trees. This time complexity is same as one between ordered trees (Jiang et al., 1995). Also we design the algorithms to compute the alignment distance between cyclic-ordered and cyclic-biordered trees in $O(n^2D^4)$ time.

Next, by using the same strategy of (Kan et al., 2014) to compute a top-down distance for every pair of nodes in given two cyclically ordered trees in advance, we design the algorithm to compute the segmental alignment distance between cyclically ordered trees with the same time complexity as above.

Finally, we give experimental results for the alignment distance between biordered trees comparing with the edit distance between ordered trees, by using N-glycan data provided from KEGG (Kyoto Encyclopedia of Genes and Genomes, <http://www.kegg.jp/>).

2 PRELIMINARIES

A *tree* is a connected graph without cycles. For a tree $T = (V, E)$, we denote V and E by $V(T)$ and $E(T)$, respectively. Also the *size* of T is $|V|$ and denoted by $|T|$. We sometime denote $v \in V(T)$ by $v \in T$. We denote an empty tree by \emptyset .

A *rooted tree* is a tree with one node r chosen as its *root*. We denote the root of a rooted tree T by $r(T)$. For each node v in a rooted tree with the root r , let $UP_r(v)$ be the unique path from v to r . The *parent* of $v (\neq r)$, which we denote by $par(v)$, is its adjacent node on $UP_r(v)$ and the *ancestors* of $v (\neq r)$ are the nodes on $UP_r(v) - \{v\}$. We denote the set of all ancestors of v by $anc(v)$. We say that u is a *child* of v if v is the parent of u . The set of children of v is denoted by $ch(v)$. We call the number of children of v the *degree* of v and denote it by $d(v)$, that is, $d(v) = |ch(v)|$. Also we define $d(T) = \max\{d(v) \mid v \in T\}$ and call it the *degree* of T .

In this paper, we use the ancestor orders $<$ and \leq , that is, $u < v$ if v is an ancestor of u and $u \leq v$ if $u < v$ or $u = v$. We say that w is the *least common ancestor* of u and v , denoted by $u \sqcup v$, if $u \leq w$, $v \leq w$ and there exists no w' such that $w' \leq w$, $u \leq w'$ and $v \leq w'$. A (*complete*) *subtree* of $T = (V, E)$ rooted by v , denoted by $T[v]$, is a tree $T' = (V', E')$ such that $r(T') = v$, $V' = \{u \in V \mid u \leq v\}$ and $E' = \{(u, w) \in E \mid u, w \in V'\}$.

We say that a rooted tree is *labeled* if each node is assigned a symbol from a fixed finite alphabet Σ . For a node v , we denote the label of v by $l(v)$, and sometimes identify v with $l(v)$. Also let $\varepsilon \notin \Sigma$ denote a special *blank* symbol and define $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$.

We say that a rooted tree is *ordered* if a left-to-

right order among siblings is fixed; *unordered* otherwise. In particular, for nodes u and v in an ordered tree, u is *to the left of* v , denoted by $u \preceq v$, if $pre(u) \leq pre(v)$ and $post(u) \leq post(v)$ for the pre-order number pre and the postorder number $post$.

Furthermore, in this paper, we introduce *cyclically ordered trees* by using the following functions $\sigma_{p,n}^+(i)$ and $\sigma_{p,n}^-(i)$ for $1 \leq i, p \leq n$.

$$\begin{aligned}\sigma_{p,n}^+(i) &= ((i + p - 1) \bmod n) + 1, \\ \sigma_{p,n}^-(i) &= ((n - i - p + 1) \bmod n) + 1.\end{aligned}$$

Definition 1 (Cyclically Ordered Trees). Let T be a tree and suppose that v_1, \dots, v_n are the children of $v \in T$ from left to right.

1. We say that T is *biordered* if T allows the orders of both v_1, \dots, v_n and v_n, \dots, v_1 .
2. We say that T is *cyclic-ordered* if T allows the orders $v_{\sigma_{p,n}^+(1)}, \dots, v_{\sigma_{p,n}^+(n)}$ for every $1 \leq p \leq n$.
3. We say that T is *cyclic-biordered* if T allows the orders $v_{\sigma_{p,n}^+(1)}, \dots, v_{\sigma_{p,n}^+(n)}$ and $v_{\sigma_{p,n}^-(1)}, \dots, v_{\sigma_{p,n}^-(n)}$ for every $1 \leq p \leq n$.

Sometimes we use the scripts o, b, c, cb, u , and the notation of $\pi \in \{o, b, c, cb, u\}$, which we call a π -tree.

It is obvious that the cyclically ordered trees are an extension of ordered trees and a restriction of unordered trees. The number of orders among siblings of a node v in ordered trees, biordered trees, cyclic-ordered trees, cyclic-biordered trees and unordered trees is $1, 2, d(v), 2d(v)$ and $d(v)!$, respectively.

Next, we introduce the alignment distance (Jiang et al., 1995). Here, for $\pi \in \{o, b, c, cb, u\}$, we call an isomorphism for π -trees a π -isomorphism.

Definition 2 (Alignment (Jiang et al., 1995)). Let T_1 and T_2 be trees and $\pi \in \{o, b, c, cb, u\}$. An *alignment* between T_1 and T_2 is a tree \mathcal{T} obtained by the following two steps.

1. Insert new nodes labeled by ε into T_1 and T_2 such that the resulting trees T'_1 and T'_2 are π -isomorphic with ignoring labels and $l(\phi(v)) \neq \varepsilon$ whenever $l(v) = \varepsilon$ for a π -isomorphism ϕ between T'_1 and T'_2 and every node $v \in T'_1$.
2. Set \mathcal{T} to an obtained tree T'_1 by relabeling a label $l(v)$ for every node $v \in T'_1$ with $(l(v), l(\phi(v)))$. (Note that $(\varepsilon, \varepsilon) \notin \mathcal{T}$.)

Let $\mathcal{A}^\pi(T_1, T_2)$ denote the set of all possible alignments between T_1 and T_2 .

We define a *cost function* $\gamma: (\Sigma_\varepsilon \times \Sigma_\varepsilon - \{(\varepsilon, \varepsilon)\}) \mapsto \mathbf{R}^+$ on pairs of labels. We constrain γ to be a *metric*, that is, $\gamma(l_1, l_2) \geq 0$, $\gamma(l_1, l_1) = 0$, $\gamma(l_1, l_2) = \gamma(l_2, l_1)$ and $\gamma(l_1, l_3) \leq \gamma(l_1, l_2) + \gamma(l_2, l_3)$. In particular, we sometimes use a *unit cost function* such that $\gamma(l_1, l_2) =$

1 if $l_1 \neq l_2$. The *cost* of an alignment \mathcal{T} , denoted by $\gamma(\mathcal{T})$, is the sum of the costs of all labels in \mathcal{T} .

Definition 3 (Alignment Distance (Jiang et al., 1995)). Let $\pi \in \{o, b, c, cb, u\}$. Then, the *alignment distance* between T_1 and T_2 is defined as the minimum cost $\gamma(\mathcal{T})$ for every alignment $\mathcal{T} \in \mathcal{A}^\pi(T_1, T_2)$. Also we call an alignment with the minimum cost an *optimal alignment*.

Example 1. Consider ordered trees T_1 and T_2 in Figure 1 (left). Then, \mathcal{T} in Figure 1 (right) is the optimal alignment between T_1 and T_2 . Under the unit cost function γ , since $\gamma(\mathcal{T}) = 4$, the alignment distance between T_1 and T_2 is 4.

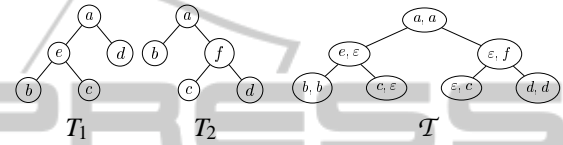


Figure 1: Ordered trees T_1 and T_2 (left) and the optimal alignment $\mathcal{T} \in \mathcal{A}^o(T_1, T_2)$ (right) in Example 1.

3 MAPPING AND DISTANCE

In this section, we introduce a *Tai mapping* and its variations, and then the distance as the minimum cost of all the mappings.

Definition 4 (Tai Mapping (Tai, 1979)). Let T_1 and T_2 be trees and $M \subseteq V(T_1) \times V(T_2)$.

1. We say that a triple (M, T_1, T_2) is an *ordered Tai mapping* from T_1 to T_2 , denoted by $M \in \mathcal{M}_{\text{Tai}}^o(T_1, T_2)$, if every pair (u_1, v_1) and (u_2, v_2) in M satisfies the following conditions.
 - (i) $u_1 = u_2$ iff $v_1 = v_2$ (one-to-one condition).
 - (ii) $u_1 \leq u_2$ iff $v_1 \leq v_2$ (ancestor condition).
 - (iii) $u_1 \preceq u_2$ iff $v_1 \preceq v_2$ (sibling condition).
2. We say that a triple (M, T_1, T_2) is an *unordered Tai mapping* from T_1 to T_2 , denoted by $M \in \mathcal{M}_{\text{Tai}}^u(T_1, T_2)$, if M satisfies the conditions (i) and (ii).

In the following, let $u_1, u_2, u_3, u_4 \in ch(u)$ and $v_1, v_2, v_3, v_4 \in ch(v)$.

3. We say that a triple (M, T_1, T_2) is a *biordered Tai mapping* from T_1 to T_2 , denoted by $M \in \mathcal{M}_{\text{Tai}}^b(T_1, T_2)$, if M satisfies the above conditions (i) and (ii) and the following condition (iv).
 - (iv) For every $u \in T_1$ and $v \in T_2$ such that $(u_1, v_1), (u_2, v_2), (u_3, v_3) \in M$, one of the following statements holds.
 1. $u_1 \preceq u_2 \preceq u_3$ iff $v_1 \preceq v_2 \preceq v_3$.

2. $u_1 \preceq u_2 \preceq u_3$ iff $v_3 \preceq v_2 \preceq v_1$.
4. We say that a triple (M, T_1, T_2) is a *cyclic-ordered Tai mapping* from T_1 to T_2 , denoted by $M \in \mathcal{M}_{\text{TAI}}^c(T_1, T_2)$, if M satisfies the above conditions (i) and (ii) and the following condition (v).
 - (v) For every $u \in T_1$ and $v \in T_2$ such that $(u_1, v_1), (u_2, v_2), (u_3, v_3) \in M$, one of the following statements holds.
 1. $u_1 \preceq u_2 \preceq u_3$ iff $v_1 \preceq v_2 \preceq v_3$.
 2. $u_1 \preceq u_2 \preceq u_3$ iff $v_2 \preceq v_3 \preceq v_1$.
 3. $u_1 \preceq u_2 \preceq u_3$ iff $v_3 \preceq v_1 \preceq v_2$.
5. We say that a triple (M, T_1, T_2) is a *cyclic-biordered Tai mapping* from T_1 to T_2 , denoted by $M \in \mathcal{M}_{\text{TAI}}^{cb}(T_1, T_2)$, if M satisfies the above conditions (i) and (ii) and the following condition (vi).
 - (vi) For every $u \in T_1$ and $v \in T_2$ such that $(u_1, v_1), (u_2, v_2), (u_3, v_3), (u_4, v_4) \in M$, one of the following statements holds.
 1. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_1 \preceq v_2 \preceq v_3 \preceq v_4$.
 2. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_2 \preceq v_3 \preceq v_4 \preceq v_1$.
 3. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_3 \preceq v_4 \preceq v_1 \preceq v_2$.
 4. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_4 \preceq v_1 \preceq v_2 \preceq v_3$.
 5. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_4 \preceq v_3 \preceq v_2 \preceq v_1$.
 6. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_3 \preceq v_2 \preceq v_1 \preceq v_4$.
 7. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_2 \preceq v_1 \preceq v_4 \preceq v_3$.
 8. $u_1 \preceq u_2 \preceq u_3 \preceq u_4$ iff $v_1 \preceq v_4 \preceq v_3 \preceq v_2$.

We will use M instead of (M, T_1, T_2) simply.

Since a less-constrained mapping (Lu et al., 2001) coincides with an alignable mapping (Kuboyama, 2007) characterizing the alignment, we formulate the alignable mapping as the less-constrained mapping.

Definition 5 (Variations of Tai Mapping). Let T_1 and T_2 be trees, $\pi \in \{o, b, c, cb, u\}$ and $M \in \mathcal{M}_{\text{TAI}}^\pi(T_1, T_2)$. Here, we denote $M - \{(r(T_1), r(T_2))\}$ by M^- .

1. We say that M is an *alignable mapping* (Kuboyama, 2007) (or a *less-constrained mapping* (Lu et al., 2001)), denoted by $M \in \mathcal{M}_{\text{ALN}}^\pi(T_1, T_2)$, if M satisfies the following condition.

$$\forall (u_1, v_1), (u_2, v_2), (u_3, v_3) \in M \\ \left(u_1 \sqcup u_2 < u_1 \sqcup u_3 \implies v_2 \sqcup v_3 = v_1 \sqcup v_3 \right).$$

2. We say that M is a *segmental mapping* (Kan et al., 2014), denoted by $M \in \mathcal{M}_{\text{SG}}^\pi(T_1, T_2)$, if M satisfies the following condition.

$$\forall (u, v) \in M^- \\ \left(\exists (u', v') \in M \left((u' \in \text{anc}(u)) \wedge (v' \in \text{anc}(v)) \right) \right) \\ \implies \left((\text{par}(u), \text{par}(v)) \in M \right).$$

3. We say that M is a *segmental alignable mapping* (Yoshino and Hirata, 2013), denoted by $M \in \mathcal{M}_{\text{SGALN}}^\pi(T_1, T_2)$, if $M \in \mathcal{M}_{\text{SG}}^\pi(T_1, T_2) \cap \mathcal{M}_{\text{ALN}}^\pi(T_1, T_2)$.
4. We say that M is a *top-down mapping* (Selkow, 1977; Chawathe, 1999) (or a *degree-1 mapping*), denoted by $M \in \mathcal{M}_{\text{TOP}}^\pi(T_1, T_2)$, if M satisfies the following condition.

$$\forall (u, v) \in M^- \left((\text{par}(u), \text{par}(v)) \in M \right).$$

Let M be a mapping from T_1 to T_2 . Let I and J be the sets of nodes in T_1 and T_2 but not in M . Then, the cost $\gamma(M)$ of M is given as follows.

$$\gamma(M) = \sum_{(u,v) \in M} \gamma(u, v) + \sum_{u \in I} \gamma(u, \varepsilon) + \sum_{v \in J} \gamma(\varepsilon, v).$$

Definition 6 (Variations of Edit Distance). For every $A \in \{\text{TAI}, \text{ALN}, \text{SGALN}, \text{TOP}\}$ and $\pi \in \{o, b, c, cb, u\}$, we define the distance $\tau_A^\pi(T_1, T_2)$ as follows.

$$\tau_A^\pi(T_1, T_2) = \min\{\gamma(M) \mid M \in \mathcal{M}_A^\pi(T_1, T_2)\}.$$

Theorem 1. Let T_1 and T_2 be trees and $\pi \in \{o, b, c, cb, u\}$.

1. $\tau_{\text{TAI}}^\pi(T_1, T_2)$ coincides with the edit distance (Tai, 1979).
2. $\tau_{\text{ALN}}^\pi(T_1, T_2)$ coincides with the alignment distance (Kuboyama, 2007).

Theorem 2. Let T_1 and T_2 be trees such that $n = |T_1| \geq |T_2| = m$ and $D = \max\{d(T_1), d(T_2)\}$.

1. We can compute $\tau_{\text{TAI}}^o(T_1, T_2)$ in $O(nm^2(1 + \log \frac{n}{m})) = O(n^3)$ time. On the other hand, the problem of computing $\tau_{\text{TAI}}^u(T_1, T_2)$ is MAX SNP-hard, even if T_1 and T_2 are binary (Demaine et al., 2009; Zhang and Jiang, 1994; Hirata et al., 2011).
2. We can compute $\tau_{\text{ALN}}^o(T_1, T_2)$ and $\tau_{\text{SGALN}}^o(T_1, T_2)$ in $O(nmD^2)$ time. On the other hand, the problem of computing $\tau_{\text{ALN}}^u(T_1, T_2)$ and $\tau_{\text{SGALN}}^u(T_1, T_2)$ is MAX SNP-hard, but it is tractable if the degrees of T_1 and T_2 are bounded (Jiang et al., 1995; Yoshino and Hirata, 2013).

Proposition 1 (cf. (Kuboyama, 2007; Yoshino and Hirata, 2013)). Let T_1 and T_2 be trees and $\pi \in \{o, b, c, cb, u\}$. Also suppose that a cost function is a metric. Then, $\tau_{\text{TAI}}^\pi(T_1, T_2)$ and $\tau_{\text{TOP}}^\pi(T_1, T_2)$ are metrics, whereas neither $\tau_{\text{ALN}}^\pi(T_1, T_2)$ nor $\tau_{\text{SGALN}}^\pi(T_1, T_2)$ is a metric.

Proposition 2. Let T_1 and T_2 be trees. For $A \in \{\text{TAI}, \text{ALN}, \text{SGALN}, \text{TOP}\}$ and $\pi \in \{o, b, c, cb, u\}$, the following statements hold.

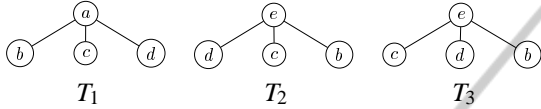
1. $\tau_A^o(T_1, T_2) \leq \tau_A^{cb}(T_1, T_2) \leq \tau_A^b(T_1, T_2) \leq \tau_A^c(T_1, T_2)$.
2. $\tau_A^u(T_1, T_2) \leq \tau_A^{cb}(T_1, T_2) \leq \tau_A^c(T_1, T_2) \leq \tau_A^o(T_1, T_2)$.

$$3. \tau_{\text{TAI}}^{\pi}(T_1, T_2) \leq \tau_{\text{ALN}}^{\pi}(T_1, T_2) \leq \tau_{\text{SGALN}}^{\pi}(T_1, T_2) \leq \tau_{\text{TOP}}^{\pi}(T_1, T_2).$$

Proposition 3. For $A \in \{\text{TAI}, \text{ALN}, \text{SGALN}, \text{TOP}\}$, there exist trees T_1 and T_2 satisfying each of the following conditions.

1. $\tau_A^c(T_1, T_2) < \tau_A^b(T_1, T_2)$.
2. $\tau_A^b(T_1, T_2) < \tau_A^c(T_1, T_2)$.

Proof. Consider the following trees T_1 , T_2 and T_3 .



Under the unit cost function, Statement 1 follows that $\tau_{\text{TOP}}^b(T_1, T_2) = 1 < 3 = \tau_{\text{TOP}}^c(T_1, T_2)$ and Statement 2 follows that $\tau_{\text{TOP}}^c(T_1, T_3) = 1 < 3 = \tau_{\text{TOP}}^b(T_1, T_2)$. \square

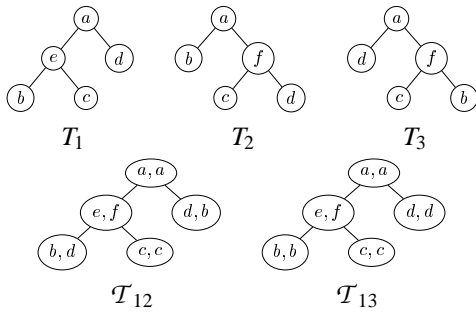
Proposition 4. Let T_1 and T_2 be trees and $A \in \{\text{TAI}, \text{ALN}, \text{SGALN}, \text{TOP}\}$.

1. If $\max\{d(T_1), d(T_2)\} \leq 1$, then it holds that $\tau_A^o(T_1, T_2) = \tau_A^b(T_1, T_2) = \tau_A^c(T_1, T_2) = \tau_A^{cb}(T_1, T_2) = \tau_A^u(T_1, T_2)$.
2. If $\max\{d(T_1), d(T_2)\} \leq 2$, then it holds that $\tau_A^b(T_1, T_2) = \tau_A^c(T_1, T_2) = \tau_A^{cb}(T_1, T_2) = \tau_A^u(T_1, T_2)$.
3. If $\max\{d(T_1), d(T_2)\} \leq 3$, then it holds that $\tau_A^{cb}(T_1, T_2) = \tau_A^u(T_1, T_2)$.

Proposition 5. For $\pi \in \{b, c, cb\}$, there exist trees T_1 and T_2 satisfying each of the following conditions.

1. $\tau_{\text{TAI}}^o(T_1, T_2) < \tau_{\text{ALN}}^{\pi}(T_1, T_2)$.
2. $\tau_{\text{ALN}}^{\pi}(T_1, T_2) < \tau_{\text{TAI}}^o(T_1, T_2)$.

Proof. Consider the following trees T_1 , T_2 and T_3 and suppose that a cost function is the unit cost function.



1. It is obvious that $\tau_{\text{TAI}}^o(T_1, T_2) = 2$. On the other hand, since the alignment \mathcal{T}_{12} is an optimal alignment between T_1 and T_2 for cyclically ordered trees, it holds that $\tau_{\text{ALN}}^b(T_1, T_2) = \tau_{\text{ALN}}^c(T_1, T_2) = \tau_{\text{ALN}}^{cb}(T_1, T_2) = 3$. Note that $\tau_{\text{ALN}}^o(T_1, T_2) = 4$ (Jiang et al., 1995).

2. Since the alignment \mathcal{T}_{13} is an optimal alignment between T_1 and T_3 for cyclically ordered

trees, it holds that $\tau_{\text{ALN}}^b(T_1, T_3) = \tau_{\text{ALN}}^c(T_1, T_3) = \tau_{\text{ALN}}^{cb}(T_1, T_3) = 1$. On the other hand, it is obvious that $\tau_{\text{TAI}}^o(T_1, T_3) = 4$. Note that $\tau_{\text{ALN}}^o(T_1, T_2) = 5$. \square

4 ALGORITHMS

In this section, we identify a node with its post-order number. Also let $n = |T_1|$, $m = |T_2|$ (and suppose that $n \geq m$), $d = \min\{d(T_1), d(T_2)\}$ and $D = \max\{d(T_1), d(T_2)\}$.

A(n ordered) *forest* is a sequence $[T_1, \dots, T_n]$ of trees. For a tree T and a node $i \in T$, $T(i)$ is a forest obtained by deleting the root i in $T[i]$. For nodes $i \in T_1$ and $j \in T_2$, let the children of i and j be i_1, \dots, i_s and j_1, \dots, j_t . That is, it holds that $d(i) = s$ and $d(j) = t$. Also, for trees T_1 and T_2 , we denote the forests $T_1(i) = [T_1[i_1], \dots, T_1[i_s]]$ and $T_2(j) = [T_2[j_1], \dots, T_2[j_t]]$ by $F_1(i_1, i_s)$ and $F_2(j_1, j_t)$.

For $A \in \{\text{ALN}, \text{SGALN}, \text{TOP}\}$ and $\pi \in \{o, b, c, cb\}$, the recurrences in Figure 2 compute the distance τ_A^{π} and the forest distance δ_A^{π} when containing an empty tree or forest. Also Figure 3 illustrates the common recurrences Γ_A^{π} and Δ_A^{π} to compute τ_A^{π} and δ_A^{π} .

$$\begin{aligned} \delta_A^{\pi}(\emptyset, \emptyset) &= 0, \\ \tau_A^{\pi}(T_1[i], \emptyset) &= \delta_A^{\pi}(T_1(i), \emptyset) + \gamma(i, \varepsilon), \\ \tau_A^{\pi}(\emptyset, T_2[j]) &= \delta_A^{\pi}(\emptyset, T_2(j)) + \gamma(\varepsilon, j), \\ \delta_A^{\pi}(T_1(i), \emptyset) &= \sum_{k=1}^s \tau_A^{\pi}(T_1[i_k], \emptyset), \\ \delta_A^{\pi}(\emptyset, T_2(j)) &= \sum_{k=1}^t \tau_A^{\pi}(\emptyset, T_2[j_k]). \end{aligned}$$

Figure 2: The basic recurrences of computing $\tau_A^{\pi}(T_1, T_2)$.

$$\begin{aligned} \Gamma_A^{\pi}(T_1[i], T_2[j]) &= \min \left\{ \begin{aligned} &\tau_A^{\pi}(T_1[i], \emptyset) + \min_{T \in \mathcal{T}_1(i)} \{ \tau_A^{\pi}(T, T_2[j]) - \tau_A^{\pi}(T, \emptyset) \}, \\ &\tau_A^{\pi}(\emptyset, T_2[j]) + \min_{T \in \mathcal{T}_2(j)} \{ \tau_A^{\pi}(T_1[i], T) - \tau_A^{\pi}(\emptyset, T) \} \end{aligned} \right\}, \\ \Delta_A^{\pi}(F_1(i_1, i_s), F_2(j_1, j_t)) &= \min \left\{ \begin{aligned} &\delta_A^{\pi}(F_1(i_1, i_{s-1}), F_2(j_1, j_t)) + \tau_A^{\pi}(T_1[i_s], \emptyset), \\ &\delta_A^{\pi}(F_1(i_1, i_s), F_2(j_1, j_{t-1})) + \tau_A^{\pi}(\emptyset, T_2[j_t]), \\ &\delta_A^{\pi}(F_1(i_1, i_{s-1}), F_2(j_1, j_{t-1})) + \tau_A^{\pi}(T_1[i_s], T_2[j_t]) \end{aligned} \right\}. \end{aligned}$$

Figure 3: The common recurrences $\Gamma_A^{\pi}(T_1, T_2)$ and $\Delta_A^{\pi}(F_1, F_2)$.

Let $T_1(i) = [T_1[i_1], \dots, T_1[i_s]]$ and $T_2(j) = [T_2[j_1], \dots, T_2[j_t]]$. Also let $1 \leq p \leq s$ and $1 \leq q \leq t$. Then, we denote the forests $[T_1[i_{\sigma_{p,s}^+(1)}], \dots, T_1[i_{\sigma_{p,s}^+(s)}]]$ and $[T_2[j_{\sigma_{q,t}^+(1)}], \dots, T_2[j_{\sigma_{q,t}^+(t)}]]$ by $T_1^p(i)$ and $T_2^q(j)$. Also we denote the forests $[T_1[i_{\sigma_{p,s}^-(1)}], \dots, T_1[i_{\sigma_{p,s}^-(s)}]]$ and $[T_2[j_{\sigma_{q,t}^-(1)}], \dots, T_2[j_{\sigma_{q,t}^-(t)}]]$ by $T_1^{-p}(i)$ and $T_2^{-q}(j)$. It is obvious that $T_1(i) = T_1^1(i)$ and $T_2(j) = T_2^1(j)$.

Furthermore, the values of p and q in $T_1^p(i)$, $T_1^p(i_s)$, $T_2^q(j)$ and $T_2^q(j_t)$ are (1) $p = q = 1$ if $\pi = o$, (2) $p = \pm 1$ and $q = \pm 1$ if $\pi = b$, (3) $1 \leq p \leq s$ and $1 \leq q \leq t$ if $\pi = c$ and (4) $1 \leq p \leq s$, $-s \leq p \leq -1$, $1 \leq q \leq t$ and $-t \leq q \leq -1$ if $\pi = cb$. Hence, we prepare the following sets: (1) $o(s) = o(t) = \{1\}$, (2) $b(s) = b(t) = \{-1, 1\}$, (3) $c(s) = \{1, \dots, s\}$, $c(t) = \{1, \dots, t\}$, and (4) $cb(s) = \{-s, \dots, -1, 1, \dots, s\}$, $cb(t) = \{-t, \dots, -1, 1, \dots, t\}$. We refer these sets to $\pi(s)$ and $\pi(t)$ for $\pi \in \{o, b, c, cb\}$.

Then, by introducing the sets $\pi(s)$ and $\pi(t)$ into the recurrences in (Jiang et al., 1995), we design the recurrences of computing $\tau_{\text{ALN}}^\pi(T_1, T_2)$ between cyclically ordered trees T_1 and T_2 as Figure 4.

$$\begin{aligned} & \tau_{\text{ALN}}^\pi(T_1[i], T_2[j]) \\ &= \min \left\{ \begin{array}{l} \min_{p \in \pi(s), q \in \pi(t)} \left\{ \delta_{\text{ALN}}^\pi(T_1^p(i), T_2^q(j)) + \gamma(i, j) \right\}, \\ \Gamma_{\text{ALN}}^\pi(T_1[i], T_2[j]) \end{array} \right\}, \\ \delta_{\text{ALN}}^\pi(F_1(i_1, i_s), F_2(j_1, j_t)) &= \min \left\{ \begin{array}{l} \Delta_{\text{ALN}}^\pi(F_1(i_1, i_s), F_2(j_1, j_t)), \\ \left. \begin{array}{l} \gamma(i_s, \varepsilon) \\ + \min_{1 \leq k < t, p \in \pi(s)} \left\{ \delta_{\text{ALN}}^\pi(F_1(i_1, i_{s-1}), F_2(j_1, j_{k-1})) \right\} \\ + \delta_{\text{ALN}}^\pi(T_1^p(i_s), F_2(j_k, j_t)) \right\}, \\ \gamma(\varepsilon, j_t) \\ + \min_{1 \leq k < s, q \in \pi(t)} \left\{ \delta_{\text{ALN}}^\pi(F_1(i_1, i_{k-1}), F_2(j_1, j_{t-1})) \right\} \\ + \delta_{\text{ALN}}^\pi(F_1(i_k, i_s), T_2^q(j_t)) \right\} \end{array} \right\}. \end{array} \right. \end{aligned}$$

Figure 4: The recurrences of computing $\tau_{\text{ALN}}^\pi(T_1, T_2)$ between cyclically ordered trees.

Theorem 3. *The recurrences in Figure 4 are correct to compute $\tau_{\text{ALN}}^\pi(T_1, T_2)$ between cyclically ordered trees T_1 and T_2 for $\pi \in \{b, c, cb\}$.*

Proof. In the proof of (Jiang et al., 1995) showing that the recurrences of computing $\tau_{\text{ALN}}^o(T_1, T_2)$ is correct, the formulas and the cases of an optimal alignment tree or forest \mathcal{T} are presented as follows.

1. The formula $\delta_{\text{ALN}}^o(T_1(i), T_2(j)) + \gamma(i, j)$ is corresponding to the case that (i, j) is a label in an optimal alignment tree \mathcal{T} of $T_1[i]$ and $T_2[j]$, and \mathcal{T} contains the alignment of $T_1(i)$ and $T_2(j)$.
2. The formula $\gamma(i_s, \varepsilon) + \min_{1 \leq k < t} \left\{ \delta_{\text{ALN}}^o(F_1(i_1, i_{s-1}), F_2(j_1, j_{k-1})) \right\} + \delta_{\text{ALN}}^o(T_1(i_s), F_2(j_k, j_t))$ is corresponding to the case that (i_s, ε) is a label in an optimal alignment forest \mathcal{T} of $F_1(i_1, i_s)$ and $F_2(j_1, j_t)$, and \mathcal{T} contains the alignment of $T_1(i_s)$ and $F_2(j_k, j_t)$ for $1 \leq k < t$.
3. The formula $\gamma(\varepsilon, j_t) + \min_{1 \leq k < s} \left\{ \delta_{\text{ALN}}^o(F_1(i_1, i_{k-1}), F_2(j_1, j_{t-1})) \right\} + \delta_{\text{ALN}}^o(F_1(i_k, i_s), T_2(j_t))$ is corresponding to the case that (ε, j_t) is a label in an optimal alignment forest \mathcal{T} of $F_1(i_1, i_s)$

and $F_2(j_1, j_t)$, and \mathcal{T} contains the alignment of $F_1(i_k, i_s)$ and $T_2(j_t)$ for $1 \leq k < s$.

In just above three formulas, an optimal alignment \mathcal{T} contains and expands the siblings of some node in T_1 or T_2 (or both). When extending from $\tau_{\text{ALN}}^o(T_1, T_2)$ to $\tau_{\text{ALN}}^\pi(T_1, T_2)$, it is sufficient to deal with more than two orders in the above three formulas, instead of one left-to-right order, and then to replace $T_1(i)$, $T_2(j)$, $T_1(i_s)$ and $T_2(j_t)$ with $T_1^p(i)$, $T_2^q(j)$, $T_1^p(i_s)$ and $T_2^q(j_t)$ for $p \in \pi(s)$ and $q \in \pi(t)$. Hence, by replacing the formulas in the above statements 1, 2 and 3 with the first formula in $\tau_{\text{ALN}}^\pi(F_1(i_1, i_s), F_2(j_1, j_t))$ and the second and the third formulas in $\delta_{\text{ALN}}^\pi(F_1(i_1, i_s), F_2(j_1, j_t))$ in Figure 4, we can compute $\tau_{\text{ALN}}^\pi(T_1, T_2)$ correctly. \square

Theorem 4. *We can compute $\tau_{\text{ALN}}^b(T_1, T_2)$ in $O(nmD^2)$ time. Also we can compute $\tau_{\text{ALN}}^c(T_1, T_2)$ and $\tau_{\text{ALN}}^{cb}(T_1, T_2)$ in $O(nmD^3)$ time.*

Proof. In Figure 4, the number of recurrences in τ_{ALN}^o is 3 and one in δ_{ALN}^o is 5; the number of recurrences in τ_{ALN}^b is 6 and one in δ_{ALN}^b is 7; the number of recurrences in τ_{ALN}^c is $d(i)d(j) + 2$ and one in δ_{ALN}^c is $d(i) + d(j) + 3$; the number of recurrences in τ_{ALN}^{cb} is $4d(i)d(j) + 2$ and one in δ_{ALN}^{cb} is $2d(i) + 2d(j) + 3$.

According to the proof of (Jiang et al., 1995), for $s = d(i)$ and $t = d(j)$, we can compute $\delta_{\text{ALN}}^o(F_1(i_{s'}, i_s), F_2(j_{t'}, j_t))$ in $O((s - s') \times (t - t') \times ((s - s') + (t - t'))) = O(d(i)d(j)(d(i) + d(j)))$ time. Then, we can compute $\delta_{\text{ALN}}^b(F_1(i_{s'}, i_s), F_2(j_{t'}, j_t))$ in $O(d(i)d(j)(d(i) + d(j)))$ time. So the running time of computing $\tau_{\text{ALN}}^b(T_1[i], T_2[j])$ for each $(i, j) \in T_1 \times T_2$ is $O(d(i)d(j)(d(i) + d(j))d(i) + d(i)d(j)(d(i) + d(j))d(j)) = O(d(i)d(j)(d(i) + d(j))^2)$. Hence, the running time of computing $\tau_{\text{ALN}}^b(T_1, T_2)$ is:

$$\begin{aligned} & \sum_{i=1}^{|T_1|} \sum_{j=1}^{|T_2|} O\left(d(i)d(j)(d(i) + d(j))^2\right) \\ & \leq \sum_{i=1}^{|T_1|} \sum_{j=1}^{|T_2|} O\left(d(i)d(j)(d(T_1) + d(T_2))^2\right) \\ & \leq O\left((d(T_1) + d(T_2))^2 \times \sum_{i=1}^{|T_1|} d(i) \times \sum_{j=1}^{|T_2|} d(j)\right) \\ & \leq O(|T_1| \times |T_2| \times (d(T_1) + d(T_2))^2) \\ & = O(nmD^2). \end{aligned}$$

Also, by focusing on the number of recurrences in Figure 4, we can compute $\delta_{\text{ALN}}^c(F_1(i_{s'}, i_s), F_2(j_{t'}, j_t))$ and $\delta_{\text{ALN}}^{cb}(F_1(i_{s'}, i_s), F_2(j_{t'}, j_t))$ in $O((s - s')d(j) \times (t - t')d(i) \times ((s - s') + (t - t'))) = O(d(i)^2d(j)^2(d(i) + d(j)))$ time. So the running time of computing $\tau_{\text{ALN}}^c(T_1[i], T_2[j])$ and $\tau_{\text{ALN}}^{cb}(T_1[i], T_2[j])$ for each $(i, j) \in T_1 \times T_2$ is

$O(d(i)^2 d(j)^2 (d(i) + d(j)) d(i) + d(i)^2 d(j)^2 (d(i) + d(j)) d(j) + d(i) d(j)) = O(d(i)^2 d(j)^2 (d(i) + d(j))^2)$, where the last formula $d(i)d(j)$ is corresponding to the time complexity of computing the first recurrence in $\tau_{\text{ALN}}^\pi(T_1[i], T_2[j])$ in Figure 4. Hence, the running time of computing $\tau_{\text{ALN}}^c(T_1, T_2)$ and $\tau_{\text{ALN}}^{cb}(T_1, T_2)$ is:

$$\begin{aligned}
 & \sum_{i=1}^{|T_1|} \sum_{j=1}^{|T_2|} O\left(d(i)^2 d(j)^2 (d(i) + d(j))^2\right) \\
 & \leq \sum_{i=1}^{|T_1|} \sum_{j=1}^{|T_2|} O\left(\begin{array}{c} d(i)d(j)d(T_1)d(T_2) \\ \times (d(T_1) + d(T_2))^2 \end{array}\right) \\
 & \leq O\left(\begin{array}{c} d(T_1)d(T_2)(d(T_1) + d(T_2))^2 \\ \times \sum_{i=1}^{|T_1|} d(i) \times \sum_{j=1}^{|T_2|} d(j) \end{array}\right) \\
 & \leq O(|T_1| \times |T_2| \times d(T_1)d(T_2)(d(T_1) + d(T_2))^2) \\
 & = O(nmdD^3). \quad \square
 \end{aligned}$$

Next, we design the algorithm to compute the segmental alignment distance $\tau_{\text{SGALN}}^\pi(T_1, T_2)$ for $\pi \in \{b, c, cb\}$. Here, we adopt the same strategy of (Kan et al., 2014) to compute $\tau_{\text{TOP}}^\pi(T_1[i], T_2[j])$ between $T_1[i]$ and $T_2[j]$ for every pair $(i, j) \in T_1 \times T_2$ ($1 \leq i \leq n, 1 \leq j \leq m$) in advance.

Then, Figure 5 illustrates the recurrences of computing $\tau_{\text{SGALN}}^\pi(T_1, T_2)$ for cyclically ordered trees. Here, $\delta_{\text{SGALN}}^\pi(F_1, F_2)$ is same as $\delta_{\text{ALN}}^\pi(F_1, F_2)$ by replacing the subscript ALN with SGALN.

$$\begin{aligned}
 & \tau_{\text{TOP}}^\pi(T_1[i], T_2[j]) \\
 & = \min_{p \in \pi(s), q \in \pi(s)} \{ \delta_{\text{TOP}}^\pi(T_1^p(i), T_2^q(j)) + \gamma(i, j) \}, \\
 & \delta_{\text{TOP}}^\pi(F_1(i_1, i_s), F_2(j_1, j_t)) = \Delta_{\text{TOP}}^\pi(F_1(i_1, i_s), F_2(j_1, j_t)). \\
 & \tau_{\text{SGALN}}^\pi(T_1[i], T_2[j]) \\
 & = \min \left\{ \begin{array}{l} \tau_{\text{TOP}}^\pi(T_1[i], T_2[j]) \\ \dots \text{use the value computed in advance,} \\ \min_{p \in \pi(s), q \in \pi(t)} \{ \delta_{\text{SGALN}}^b(T_1^p(i), T_2^q(j)) + \gamma(i, j) \}, \\ \Gamma_{\text{SGALN}}^\pi(T_1[i], T_2[j]) \\ \delta_{\text{SGALN}}^\pi(F_1(i_1, i_s), F_2(j_1, j_t)) \\ \dots \text{same as } \delta_{\text{ALN}}^\pi \text{ with replacing ALN with SGALN.} \end{array} \right\},
 \end{aligned}$$

Figure 5: The recurrence of computing $\tau_{\text{SGALN}}^\pi(T_1, T_2)$ between cyclically ordered trees.

Theorem 5. *The recurrences in Figure 5 are correct to compute $\tau_{\text{SGALN}}^b(T_1, T_2)$ in $O(nmD^2)$ time, and $\tau_{\text{SGALN}}^c(T_1, T_2)$ and $\tau_{\text{SGALN}}^{cb}(T_1, T_2)$ in $O(nmdD^3)$ time.*

Proof. The correctness follows from Theorem 3 and (Yoshino and Hirata, 2013). Since the number of recurrences in τ_{TOP}^b in Figure 5 is $O(1)$ and one in τ_{TOP}^c and τ_{TOP}^{cb} is $O(D)$, we can compute $\tau_{\text{TOP}}^b(T_1[i], T_2[j])$ in $O(nm)$ time and compute $\tau_{\text{TOP}}^c(T_1[i], T_2[j])$ and $\tau_{\text{TOP}}^{cb}(T_1[i], T_2[j])$ in $O(nmdD)$ time for every pair

$(i, j) \in T_1 \times T_2$. Hence, by Theorem 4, the running time of computing $\tau_{\text{SGALN}}^b(T_1, T_2)$ is $O(nm) + O(nmD^2) = O(nmD^2)$. Also, the running time of computing $\tau_{\text{SGALN}}^c(T_1, T_2)$ and $\tau_{\text{SGALN}}^{cb}(T_1, T_2)$ is $O(nmdD) + O(nmdD^3) = O(nmdD^3)$. \square

5 EXPERIMENTAL RESULTS

In this section, we give experimental results for τ_{ALN}^b comparing with τ_{TAI}^o , by using N-glycan data provided from KEGG. Here, the number of N-glycan data is 2142, the average number of nodes is 11.09, the average number of labels is 5.43 and the average depth and degree are 5.38 and 2.07, respectively.

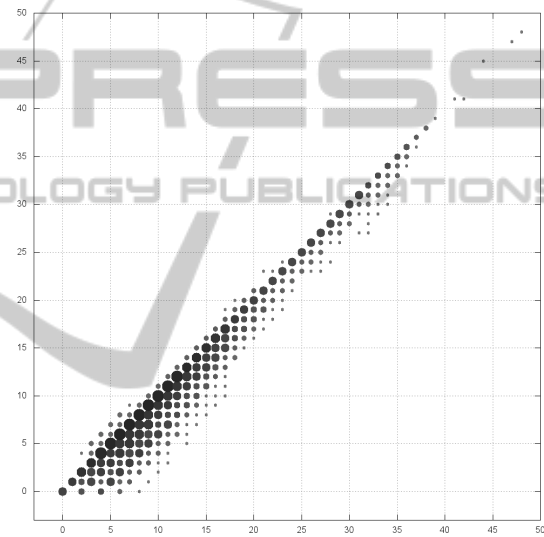


Figure 6: The correlation diagrams to the edit distance τ_{TAI}^o of τ_{ALN}^b for N-glycan data.

Figures 6 illustrates the correlation diagrams to τ_{TAI}^o of τ_{ALN}^b for all the 2293011 pairs of N-glycan data. The plots in Figures 6 are the ratio (%) of the pairs of trees whose value of τ_{ALN}^b is given as the y-axis to the value of τ_{TAI}^o given in the x-axis.

Figures 6 shows that, for N-glycan data, whereas τ_{ALN}^b tends to be smaller than τ_{TAI}^o , we can observe the pairs that τ_{ALN}^b is greater than τ_{TAI}^o as Proposition 5.

Table 1 represents the number of pairs comparing τ_{ALN}^b with τ_{TAI}^o in all the pairs of N-glycan data.

Table 1: The number of pairs comparing τ_{ALN}^b with τ_{TAI}^o .

case	#pairs
$\tau_{\text{ALN}}^b > \tau_{\text{TAI}}^o$	675
$\tau_{\text{ALN}}^b = \tau_{\text{TAI}}^o$	1193559
$\tau_{\text{ALN}}^b < \tau_{\text{TAI}}^o$	298777

Hence, we conclude that, $\tau_{ALN}^b \leq \tau_{TAI}^o$ for almost pairs of N-glycan data; Only 675 pairs (about 0.029%) satisfies that $\tau_{ALN}^b > \tau_{TAI}^o$. This result implies that τ_{ALN}^b (and τ_{ALN}^{cb}) is possible to be a good approximation of τ_{TAI}^u for N-glycan data.

6 CONCLUSION

In this paper, we have formulated biordered, cyclic-ordered and cyclic-biordered trees as cyclically ordered trees, and then designed the algorithms to compute $\tau_{ALN}^b(T_1, T_2)$ and $\tau_{SGALN}^b(T_1, T_2)$ in $O(nmD^2)$ time and to compute $\tau_{ALN}^\pi(T_1, T_2)$ and $\tau_{SGALN}^\pi(T_1, T_2)$ ($\pi \in \{c, cb\}$) in $O(nmD^3)$ time. Finally, we have given the experimental results of computing τ_{ALN}^b comparing with τ_{TAI}^o by using N-glycan data.

It is a future work to implement the algorithms to compute τ_{ALN}^c , τ_{ALN}^{cb} and τ_{SGALN}^π ($\pi \in \{b, c, cb\}$), and apply τ_{ALN}^π and τ_{SGALN}^π to real data such as glycans (Hizukuri et al., 2005) or molecular graphs (Horváth et al., 2010). Also, it is a future work to apply cyclically ordered trees to compare RNA secondary structures (Höchsmann et al., 2003; Schiermer and Giegerich, 2013; Shapiro and Zhang, 1990; Zhang, 1998).

As the comparison with τ_{TAI}^u , it is a future work to investigate how τ_{ALN}^π ($\pi \in \{b, c, cb\}$) is a good approximation of τ_{TAI}^u and to compare τ_{ALN}^π with tractable variations of τ_{TAI}^u such as the isolated-subtree distance (Zhang, 1996) and the LCA-preserving distance (Zhang et al., 1996). Also, it is a future work to solve whether or not the problem of computing τ_{ALN}^u is tractable if the number of permutations among siblings is bounded by some polynomial with respect to degrees.

REFERENCES

- Chawathe, S. S. (1999). Comparing hierarchical data in external memory. In *Proc. VLDB'99*, pages 90–101.
- Demaine, E. D., Mozes, S., Rossman, B., and Weimann, O. (2009). An optimal decomposition algorithm for tree edit distance. *ACM Trans. Algo.*, 6.
- Hirata, K., Yamamoto, Y., and Kuboyama, T. (2011). Improved MAX SNP-hard results for finding an edit distance between unordered trees. In *Proc. CPM'11 (LNCS 6661)*, pages 402–415.
- Hizukuri, Y., Yamanishi, T., Nakamura, O., Yagi, F., Goto, S., and Kanehisa, M. (2005). Extraction of leukemia specific glycan motifs in humans by computational glycomics. *Carbohydrate Research*, 340:2270–2278.
- Höchsmann, M., Töller, T., Giegerich, R., and Kurtz, S. (2003). Local similarity in RNA secondary structures. In *Proc. CSB'03*, pages 159–168.
- Horváth, T., Ramon, J., and Wrobel, S. (2010). Frequent subgraph mining in outerplanar graphs. *Data Min. Knowl. Disc.*, 21:472–508.
- Jiang, T., Wang, L., and Zhang, K. (1995). Alignment of trees – an alternative to tree edit. *Theoret. Comput. Sci.*, 143:137–148.
- Kan, T., Higuchi, S., and Hirata, K. (2014). Segmental mapping and distance for rooted ordered labeled trees. *Fundamenta Informaticae*, 132:1–23.
- Kuboyama, T. (2007). *Matching and learning in trees*. Ph.D thesis, University of Tokyo.
- Lu, C. L., Su, Z.-Y., and Yang, C. Y. (2001). A new measure of edit distance between labeled trees. In *Proc. COCOON'01 (LNCS 2108)*, pages 338–348.
- Lu, S.-Y. (1979). A tree-to-tree distance and its application to cluster analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1:219–224.
- Schiermer, S. and Giegerich, R. (2013). Forest alignment with affine gaps and anchors, applied in RNA structure comparison. *Theoret. Comput. Sci.*, 483:51–67.
- Selkow, S. M. (1977). The tree-to-tree editing problem. *Inform. Process. Lett.*, 6:184–186.
- Shapiro, B. A. and Zhang, K. (1990). Comparing multiple RNA secondary structures using tree comparison. *Comp. Appl. Biosci.*, 6:309–318.
- Tai, K.-C. (1979). The tree-to-tree correction problem. *J. ACM*, 26:422–433.
- Valiente, G. (2001). An efficient bottom-up distance between trees. In *Proc. SPIRE'01*, pages 212–219.
- Wang, Y., DeWitt, D. J., and Cai, J.-Y. (2003). X-Diff: An effective change detection algorithm for XML documents. In *Proc. ICDE'03*, pages 519–530.
- Yamamoto, Y., Hirata, K., and Kuboyama, T. (2014). Tractable and intractable variations of unordered tree edit distance. *Internat. J. Found. Comput. Sci.*, 25:307–329.
- Yoshino, T. and Hirata, K. (2013). Hierarchy of segmental and alignable mapping for rooted labeled trees. In *Proc. DDS'13*, pages 62–69.
- Zhang, K. (1995). Algorithms for the constrained editing distance between ordered labeled trees and related problems. *Pattern Recog.*, 28:463–474.
- Zhang, K. (1996). A constrained edit distance between unordered labeled trees. *Algorithmica*, 15:205–222.
- Zhang, K. (1998). Computing similarity between RNA secondary structures. In *Proc. IEEE Internat. Joint Symp. Intell. Sys.*, pages 126–132.
- Zhang, K. and Jiang, T. (1994). Some MAX SNP-hard results concerning unordered labeled trees. *Inform. Process. Lett.*, 49:249–254.
- Zhang, K., Wang, J., and Shasha, D. (1996). On the editing distance between undirected acyclic graphs. *Internat. J. Found. Comput. Sci.*, 7:43–58.