Supplementary material for "Comparative Analysis of Merge Trees using Local Tree Edit Distance"

Raghavendra Sridharamurthy, *Student Member, IEEE*, and Vijay Natarajan, *Member, IEEE*

Abstract—This document presents additional material supporting the paper "Comparative Analysis of Merge Trees using Local Tree Edit Distance". We recall definitions and descriptions of mappings between trees within the scope of tree edit distances (content from Sridharamurthy et al. [1, Supplementary Material]). We describe the detailed algorithm to compute LMTED, and present the pseudocode. We also provide additional results of the application of LMTED to symmetry detection in various datasets.

1 TREE EDIT DISTANCE MAPPINGS AND ILLUSTRATIONS [1]

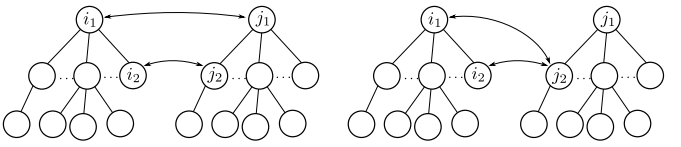
The LMTED is based on tree mappings, the same as MTED. In this section, for easy reference and completeness, we restate the properties of tree edit distance mappings, both unconstrained and constrained. We also include examples to understand their properties. The description and images in this section is from the paper describing MTED [1, Supplementary Material]. The illustrations are on simple examples of trees with equal number of nodes and that are similar to each other. Further, only the relevant mappings between node-pairs are highlighted in the figures.

1.1 Unconstrained tree edit distance mappings

The unconstrained edit distance mappings satisfy the following properties [2]. A triple (M_e, T_1, T_2) defines the *edit distance mapping* from T_1 to T_2, where each pair $(i_1, j_1), (i_2, j_2) \in M_e$ satisfies the following properties:

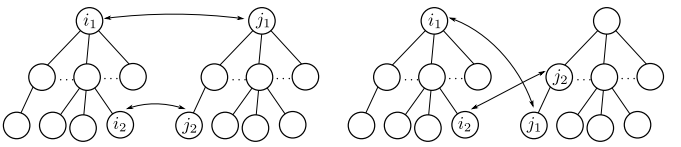
1) $i_1 = i_2$ if and only if $j_1 = j_2$ (one-to-one)
2) i_1 is an ancestor of i_2 if and only if j_1 is an ancestor of j_2 (ancestor ordering).

Figures 1 and 2 illustrate these properties using a small example. The mapping in Figure 2(b) is one-to-one but does not satisfy the ancestor preservation property, i_1 is ancestor of i_2 but j_1 is child of j_2.



(a) one-one mapping (b) violates one-one mapping

Fig. 1. Unconstrained tree edit distance mappings satisfying the one-to-one mapping property. (a) A mapping that satisfies the property. (b) A mapping that violates the property. Image source: Figure 1 from [1, Supp. Material]



(a) ancestor preserving mapping (b) ancestor order not preserved

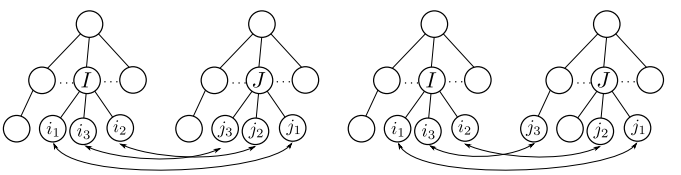
Fig. 2. Unconstrained tree edit distance mappings satisfying the ancestor preservation property. (a) A mapping that satisfies the property. (b) A mapping that violates the property. Image source: Figure 2 from [1, Supp. Material]

1.2 Constrained tree edit distance mappings

The constrained edit distance mappings satisfy the following properties [2]. A triple (M_c, T_1, T_2) is called a *constrained edit distance mapping* if,

1) (M_c, T_1, T_2) is an edit distance mapping, and
2) Given three pairs $(i_1, j_1), (i_2, j_2), (i_3, j_3) \in M_c$, the *least common ancestor* $lca(i_1, i_2)$ is a proper ancestor of i_3 if and only if $lca(j_1, j_2)$ is a proper ancestor of j_3.

Figure 3 illustrates an important property required for a mapping to be constrained, namely disjoint subtrees map to disjoint subtrees. Figure 3(b) illustrates a mapping that satisfies the properties of unconstrained tree edit distance mapping but is not a constrained tree edit distance mapping. The node i_3 is a descendant (immediate descendant in this case) of the $lca(i_1, i_2) = I$ but j_3 is not a descendant of the $lca(j_1, j_2) = J$.



(a) disjoint subtrees map to disjoint subtrees (b) disjoint subtrees do not map to disjoint subtrees

Fig. 3. Constrained tree edit distance mappings satisfying the disjoint subtree mapping property. (a) A mapping that satisfies the property. (b) A mapping that violates the property. Image source: Figure 3 from [1, Supp. Material]

These mappings are meaningful in the context of merge trees and form the basis of both MTED and LMTED with suitable modifications and appropriate cost models.

2 LMTED ALGORITHM

In this section, we describe the LMTED algorithm along with the pseudocode for the sake of completeness. The algorithm is based on Zhang [2] with suitable modifications to locally compare merge trees. Algorithm 1 computes the LMTED. It is a dynamic programming (DP) based algorithm that follows from the recurrences described in Section 3.3 of the paper and using the cost model defined in Section 3.2, which in turn is based on the truncated persistence defined in Section 3.1 of the paper. The properties of LMTED is discussed in Section 4.2 of the paper.

The notation is the same as in the paper. The DP tables are named D_c and D'. Similarly, γ denotes the original cost model and γ' denotes the truncated cost model. Line 2 initializes the distances between two empty trees to 0. The loops spanning lines $3-8$ and $9-14$ fill the table entries for both D_c and D' corresponding to the distances between the empty tree and all trees and forests. Note that lines $6,7$ and $12,13$ are new additions compared to the MTED algorithm, which depend on values from both D_c and D'. The nested loops spanning lines $15-26$ fill the entries that correspond to distances between non-empty forests and trees. Again, lines $19-24$ are additions to the MTED algorithm. To avoid clutter, the expressions $min_{F_j}, min_{F_i}, min_{T_2}, min_{T_1}$ are written separately, though they are part of the expressions calculating D' in lines $23,24$. Though the expressions look complicated, if we substitute D' with D_c, γ' with γ, and M'_r with M_r in the RHS of the expressions in lines $23,24$ we get the original MTED expressions which are in lines $17,18$. The entry $D_c(T_1[m], T_2[n])$ in the table with $m = |T_1|$ and $n = |T_2|$ corresponds to the final result for MTED. In case of LMTED, if we are interested in the distance between the pair of subtrees rooted at i and j, then the distance is given by

$$\text{LMTED}(i,j) = D'(i,j) + \Gamma(i_u \longrightarrow j_u). \tag{1}$$

$\Gamma(i_u \longrightarrow j_u)$ denotes the relabel cost computed using the truncated persistence values of i_u and j_u. The algorithm computes the distance in

$$O(|T_1| \times |T_2| \times (deg(T_1) + deg(T_2)) \times log_2(deg(T_1) + deg(T_2)))$$

time in the worst case. The analysis is as described by Zhang [2].

3 SYMMETRY DETECTION

In this section, we provide additional experimental results to demonstrate the utility of LMTED towards symmetry detection and provide additional evidence for the claims in Section 6.2. Finding symmetric structures in scalar fields is a very important problem [3]–[5]. We use CryoEM data from EMDB [6], which contains 3D electron microscopy density data of macromolecules, subcellular structures, and viruses. We first compute the merge tree, simplify the tree using a small persistence threshold, and consider all possible subtrees. We also ensure the subtrees are modified so that they satisfy merge tree properties. Since the distances are computed in the modified DP for all subproblems, the distances between these subtrees are already computed and recorded. The refinement described in Section 5.1 reduces the number of pairs of subtrees that are compared.

We have chosen two examples – EMDB 1603 (12 Angstrom resolution cryo-electron microscopy reconstruction of a recombinant active ribonucleoprotein particle of influenza virus) to show how the symmetric regions are found without any matrix reordering and EMDB 1897 (AMP-Activated Protein Kinase) to illustrate the case where reordering might be required.

The volume rendering of EMDB 1603 is shown in Figure 4(a) The modified DP is calculated for the merge tree of EMDB 1603, which has pairs of subtrees marked based on refinement criteria to get the distance matrix DM as shown in Figure 4(b). The empty regions in the DM corresponds to pairs of subtrees which are not being compared as they are eliminated by the refinement steps discussed in Section 5.1 of the paper. Consider the submatrices highlighted, these correspond to set of regions in the data which are symmetric. For clarity, we have shown the submatrices and the corresponding set of regions in Figures 5(a), 5(b), 5(c), 5(d). We can observe that we are able to detect multiple set of symmetric regions in different scales. Note that the set of regions corresponding to 4×4 submatrix given by $122,125$ is detected even though it belongs to the noisy regions outside the molecule because of its large size. Since the method prioritizes larger regions, the submatrix occurs at the bottom right.

A volume rendering of EMDB 1897 is shown in Figure 6(a). The distance matrix DM is shown in Figure 6(b). We observe that the symmetric regions do not appear as submatrices. It is difficult to visually inspect the matrix and detect the submatrices (unlike EMDB 1654 discussed in Section 6.2 of the paper). Matrix reordering techniques (leaf-reordering [7]) are applied on the DM to obtain the matrix shown in Figure 6(c). After reordering, we observe that the symmetric regions appear together. The highlighted submatrices correspond to a set of symmetric regions in the data. For clarity, we have shown the submatrices and the corresponding set of regions in Figure 7. We observe multiple sets of symmetric regions at different length scales.

REFERENCES

[1] R. Sridharamurthy, T. B. Masood, A. Kamakshidasan, and V. Natarajan, "Edit distance between merge trees," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 3, pp. 1518–1531, 2020.

[2] K. Zhang, "A Constrained Edit Distance Between Unordered Labeled Trees," *Algorithmica*, vol. 15, pp. 205–222, 1996.

[3] D. M. Thomas and V. Natarajan, "Symmetry in scalar field topology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2035–2044, 2011.

[4] ——, "Detecting symmetry in scalar fields using augmented extremum graphs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2663–2672, 2013.

[5] ——, "Multiscale symmetry detection in scalar fields by clustering contours," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 12, pp. 2427–2436, 2014.

[6] "Protein data bank in Europe," https://www.ebi.ac.uk/pdbe/emdb/, 2021, accessed: 20-03-2021.

[7] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete, "Matrix reordering methods for table and network visualization," *Computer Graphics Forum*, vol. 35, no. 3, pp. 693–716, 2016.

Algorithm 1: LocalTreeEditDistance (LMTED) [2]

Data: Merge trees T_1, T_2.

Result: $D'(T_1[i], T_2[j])$ and $D_c(T_1[i], T_2[j])$, where $1 \le i \le |T_1|$, $1 \le j \le |T_2|$

begin

$D_c(\theta,\theta) = 0, D'(\theta,\theta) = 0$

for $i = 1$ **to** $|T_1|$ **do**

$D_c(F_1[i],\theta) = \sum_{k=1}^{n_i} D_c(T_1[i_k],\theta)$

$D_c(T_1[i],\theta) = D_c(F_1[i],\theta) + \gamma(i \longrightarrow \lambda)$

$D'(F_1[i],\theta) = \sum_{k=1,k\neq u_i}^{n_i} D_c(T_1[i_k],\theta) + D'(T_1[i_{u_i}],\theta)$

$D'(T_1[i],\theta) = D'(F_1[i],\theta) + \gamma'(i \longrightarrow \lambda)$

end

for $j = 1$ **to** $|T_2|$ **do**

$D_c(\theta,F_2[j]) = \sum_{k=1}^{n_j} D_c(\theta,T_2[j_k])$

$D_c(\theta,T_2[j]) = D_c(\theta,T_2[j]) + \gamma(\lambda \longrightarrow j)$

$D'(\theta,F_2[j]) = \sum_{k=1,k\neq u_j}^{n_j} D_c(\theta,T_2[j_k]) + D'(\theta,T_2[j_{u_j}])$

$D'(\theta,T_2[j]) = D'(\theta,F_2[j]) + \gamma'(\lambda \longrightarrow j)$

end

for $i = 1$ **to** $|T_1|$ **do**

for $j = 1$ **to** $|T_2|$ **do**

$$D_c(F_1[i],F_2[j]) = \min \begin{cases} D_c(\theta,F_2[j]) + \min\limits_{1\le t\le n_j} \{D_c(F_1[i],F_2[j_t]) - D_c(\theta,F_2[j_t])\}, \\ D_c(F_1[i],\theta) + \min\limits_{1\le s\le n_i} \{D_c(F_1[i_s],F_2[j]) - D_c(F_1[i_s],\theta)\}, \\ \min\limits_{MM(i,j)} \gamma(MM(i,j)). \end{cases}$$

$$D_c(T_1[i],T_2[j]) = \min \begin{cases} D_(\theta,T_2[j]) + \min\limits_{1\le t\le n_j} \{D_c(T_1[i],T_2[j_t]) - D_c(\theta,T_2[j_t])\}, \\ D_c(T_1[i],\theta) + \min\limits_{1\le s\le n_i} \{D_c(T_1[i_s],T_2[j]) - D_c(T_1[i_s],\theta)\}, \\ D_c(F_1[i],F_2[j]) + \gamma(i \longrightarrow j). \end{cases}$$

$$min_{F_j} = \min \begin{cases} \min\limits_{1\le t\le n_j, t\neq u_j} \{D_c(F_1[i],F_2[j_t]) - D_c(\theta,F_2[j_t])\}, \\ \{D'(F_1[i],F_2[j_{u_j}]) - D'(\theta,F_2[j_{u_j}])\} \end{cases}$$

$$min_{F_i} = \min \begin{cases} \min\limits_{1\le s\le n_i, t\neq u_i} \{D_c(F_1[i_s],F_2[j]) - D_c(F_1[i_s],\theta)\}, \\ \{D'(F_1[i_{u_i}],F_2[j]) - D'(F_1[i_{u_i}],\theta)\} \end{cases}$$

$$min_{T_2} = \min \begin{cases} \min\limits_{1\le t\le n_j, t\neq u_j} \{D_c(T_1[i],T_2[j_t]) - D_c(\theta,T_2[j_t])\}, \\ \{D'(T_1[i],T_2[j_{u_j}]) - D'(\theta,T_2[j_{u_j}])\} \end{cases}$$

$$min_{T_1} = \min \begin{cases} \min\limits_{1\le s\le n_i, t\neq u_i} \{D_c(T_1[i_s],T_2[j]) - D_c(T_1[i_s],\theta)\}, \\ \{D'(T_1[i_{u_i}],T_2[j]) - D'(T_1[i_{u_i}],\theta)\} \end{cases}$$

$$D'(F_1[i],F_2[j]) = \min \begin{cases} D'(\theta,F_2[j]) + min_{F_j}, \\ D'(F_1[i],\theta) + min_{F_i}, \\ \min\limits_{M'_r(i,j)} \gamma'(M'_r(i,j)) \end{cases}$$

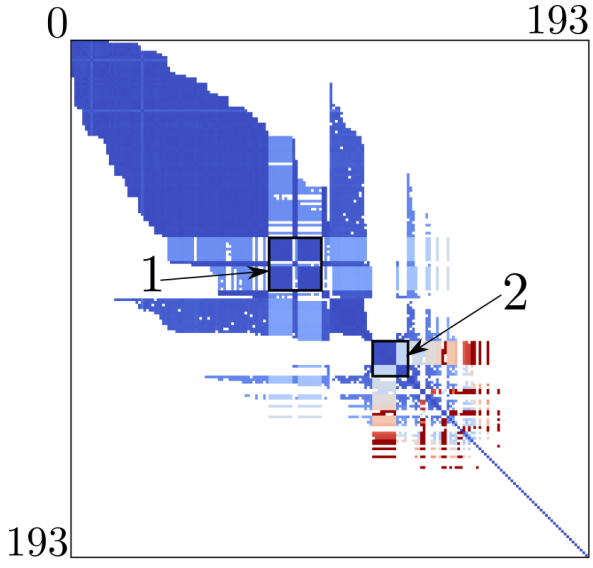
$$D'(T_1[i],T_2[j]) = \min \begin{cases} D'(\theta,T_2[j]) + min_{T_2}, \\ D'(T_1[i],\theta) + min_{T_1}, \\ D'(F_1[i],F_2[j]) + \gamma'(i \longrightarrow j). \end{cases}$$

end

end

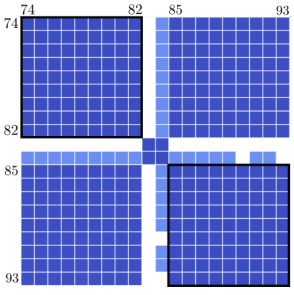
end

(a) Volume rendering of EMDB 1603

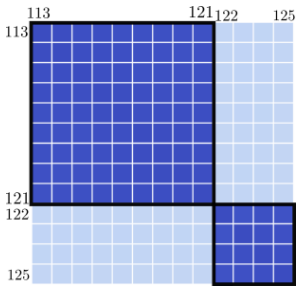


(b) Distance Matrix DM with highlighted submatrices 1 and 2

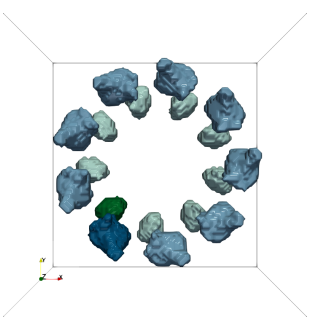
Fig. 4. LMTED values in the DM are shown using a blue-red colormap (0 0.1)



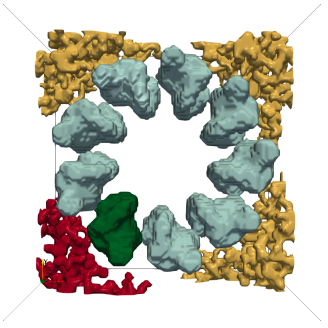
(a) Submatrix 1



(b) Submatrix 2



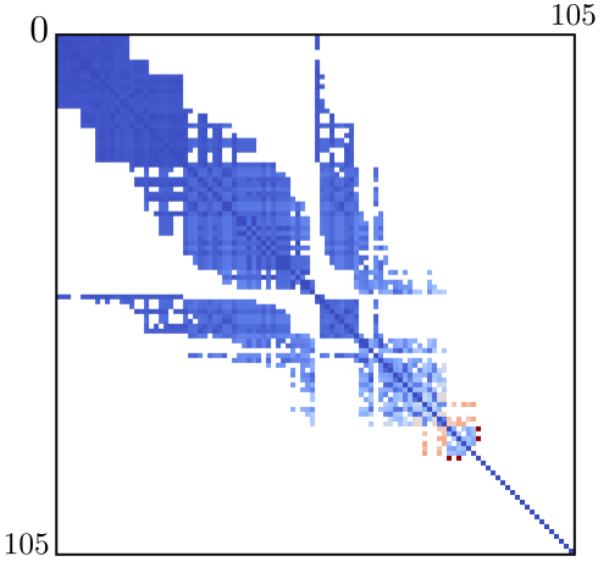
(c) regions 74…82 and 85…93



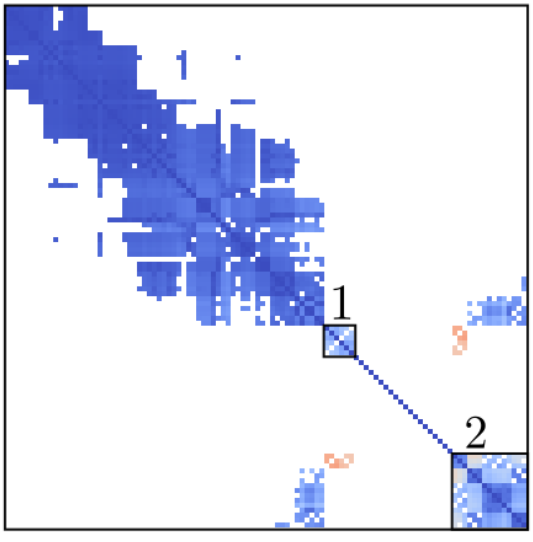
(d) regions 113…121 and 122…125

Fig. 5. Regions and the highlighted submatrices. Each of the submatrices highlighted in Figure (a) and (b) with the corresponding sets of symmetric regions (c), (d). In (c) two representative regions are shown in dark blue and dark green respectively along with regions which are symmetric to these two colored with lighter shade of blue and green. In (d) two representative regions are shown in dark green and red respectively along with regions which are symmetric to these two colored with lighter shade of green and orange.

(a) Volume rendering of EMDB 1897

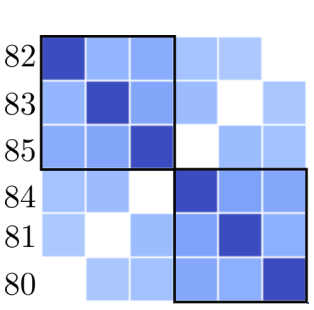


(b) Distance Matrix DM

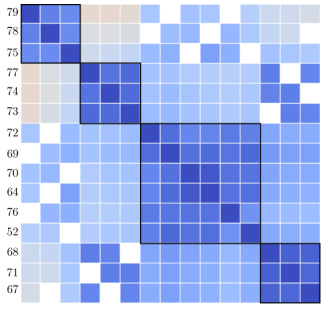


(c) Reordered Distance Matrix DM with highlighted submatrices 1 and 2

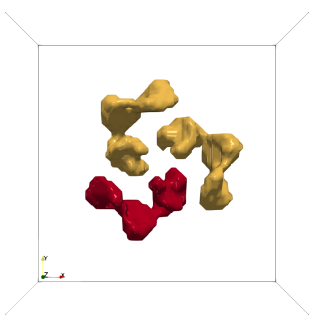
Fig. 6. LMTED values in the DM are shown using a blue-red colormap (0 — 0.2)



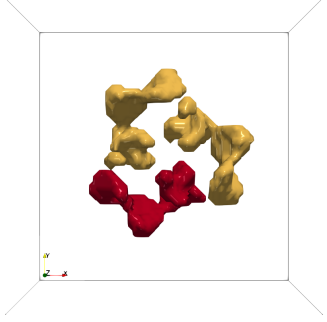
(a) Submatrix 1



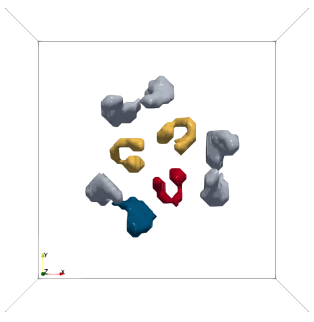
(b) Submatrix 2



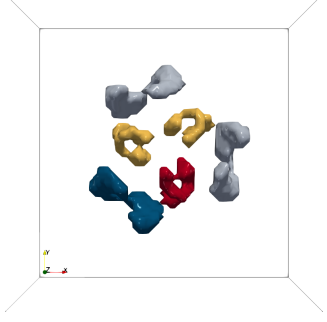
(c) regions 80…84



(d) regions 82…85



(e) regions 52…72 and 67…68



(f) regions 73…77 and 75…79

Fig. 7. Regions and the highlighted submatrices. Figure (a) and (b) are zoomed in versions of the submatrices 1 and 2 highlighted in 6(c). Each of the submatrices highlighted in Figure (a) and (b) with the corresponding sets of symmetric regions (c), (d), (e), (f). In (c), (d) a representative region is colored in red and the symmetric regions colored in yellow. In (e), (f) two representative regions are shown in red and blue respectively along with region symmetric as yellow and grey.