# Compact Set Neighboring Relation and Its Application in the Evaluating the Evolution Tree

submitted to

Workshop on Algorithms and Computatonal Molecular Biology

Jia-Ming Chang [*]    ChuanYi Tang [†]    Chao-Ling Chang [‡]

**Abstract** For evaluating the quality of a evolution tree reconstructed by computer, the neighboring relations consist with original distances are very important. However, the neighboring relations are not equally important. In this paper, from a relative point of view, we propose an estimating criterion that neighboring relations with respect to compact sets, those neighboring relations are more important.We estimate the evolution tree with considering whether those relations are consist or not. In the last part of the paper, we estimate some famous programs of constructing evolution tree with using the criterion on real data.

**Key Words:** computational biology , compact set , evolution tree , heuristic algorithm

[*] Department of Computer Science National Tsing Hua University , Hsinchu, Taiwan, R.O.C. E-mail:jmchang@cs.nthu.edu.tw

[†] Department of Computer Science National Tsing Hua University, Hsinchu, Taiwan, R.O.C. Eamil: cytang@cs.nthu.edu.tw
TEL:(03)5715131-1070

[‡] Department of Computer Science National Tsing Hua University, Hsinchu, Taiwan, R.O.C.
TEL:(03)5715131-3596

# 1 INTRODUCTION

The purposes for studying phylogenetics include (1) resconstructing the correct genealogical ties between species and (2) estimating the time when a divergence occurs between species from a common ancestor. These usually can be done by constructing trees, whose leaves represent present-day species and whose interior nodes represent hypothesized ancestors. Trees of this kind are called *evolution trees.* When an evolution tree is constructed from a distance matrix, the distances should be properly reflected. Usually, the criterion is that all of the distances in the tree reflect the original distances among species. That is, when two species are close to each other in the distance matrix, they should be close to each other in the evolution tree. The following object functions give us different evolution trees, and each criterion corresponds to a distinct evolution tree problem. We use $\mathcal{D}_{\mathcal{T}}(i,j)$ to denote the path length from specie $i$ to specie $j$ in the evolution tree $\mathcal{T}$ and $\mathcal{D}(i,j)$ to denote the distance between species $i$ and $j$ in the input distance matrix $\mathcal{D}$.

1. Minimax

   In a minimax evolution tree, the maximum of $\mathcal{D}_{\mathcal{T}}(i,j)-\mathcal{D}(i,j)$ is minimized.

2. Minisum

   In a minisum evolution tree, the total sum of all pairs of distances among leaf nodes is minimized.

3. Minisize

   In a minisize evolution tree, the total length of the tree is minimized.

These *absolute* object functions are insufficient for constructing evolution trees. That is, new evolutionary relations between species defined by a evolution tree with good minisum may conflict with those defined by the original distance matrix $\mathcal{D}$. Or, there are two evolution trees with the same minisize, yet it is hard to determine which one is better? From the output result, we cannot measure the details of the topology of a tree.

So we propose an objective function, compact set neighboring relation , and use it to work out the *preserved neighboring ratio.* We describe a new optimization probelm below and its hardness is still open.

**Maximum Preserved Neighboring Ratio**

INSTANCE: A distance matrix D over label set S.

QUESTION: Finding the evolution tree T labeled by S with maximum preserved ratio.

We introduce the definition of neighboring relation, compact set, and compact set neighboring relation in section 2.In section 3 we present our algorithm used to generate the compact set neighboring relations and the calculation of preserved neighboring ratio. In section 4 we show the experimental result.

# 2 Preliminaries

## 2.1 Neighboring Relation

Given a set $S$ of $n$ sequences, we use a symmetric $n \times n$ matrix $D$ to denote the distance matrix of $S$ and $D(i,j)$ to denote the distance between sequence $i$ and $j$ in $D$. That is, all the elements in $D$ are nonnegative, $D(i,i) = 0$ and $D(i,j) = D(j,i)$ for any $i, j \in S$. $D$ is *metric* if the distances obey the triangle inequality, i.e. $D(i,j) + D(j,k) > D(i,k)$ for any $i, j, k \in S$.

For any sequences $i, j, k \in S$, let $((i,j),k)$ denote the relation $D(i,j) \leq \min\{D(i,k), D(j,k)\}$. For any sequences $i, j$ in a tree $\mathcal{T}$, we use $\mathrm{LCA}(i,j)$ to denote the least common ancestor of $i$ and $j$ in $\mathcal{T}$. For any sequences $i, j$ and $k$, there is a *nerighboring relation* in $\mathcal{T}$ if and only if the relations $((i,j),k)$ and $(\mathrm{LCA}(i,j) < \mathrm{LCA}(i,k) = \mathrm{LCA}(j,k))$ exist.

## 2.2 Compact Sets

Given a set $S = \{S_1, S_2, \cdots, S_N\}$ of $N$ sequences, we use $D(S_i, S_j)$ to denote the distance between $S_i$ and $S_j$ in the distance matrix $D$. We also can represent this instance by a connected undirected graph $G = (V, E)$, such that $V$ is $S$ and each edge $(S_i, S_j)$ in $E$ is associated with a weight $D(S_i, S_j)$. For any subset $C$ of $S$, $C$ is called a *compact set* if the distance between elements in $C$ and elements not in $C$ is larger than the longest distance in $C$, i.e., $\min\{D(S_i, S_j) | S_i \in C, S_j \in S \setminus C\} > \max\{D(S_i, S_j) | S_i \in C, S_j \in C\}$. In other words, the sequences of a compact set are closer to each other than to other sequences. By definition, $S$ is a compact set and each set consisting of single sequence is also a compact set (i.e., these compact sets are trivial). Consider the distnace matrix of Figure 1 as an example. It is not hard to see that there are three nontrivial compact sets $\{S_1, S_6\}, \{S_1, S_2, S_6\}$ and $\{S_3, S_4, S_5\}$ among the sequences.

The relation among the compact sets can be represented in tree with using the following property:

LEMMA 2.1 ([1]). *Let $A$ and $B$ be two different compact sets of $S$. If $A \cap B \neq \emptyset$, then either $A \subset B$ or $B \subset A$.*

| $D$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ |
|---|---|---|---|---|---|---|
| $S_1$ | 0 | 10 | 16 | 18 | 13 | 8 |
| $S_2$ |  | 0 | 14 | 17 | 15 | 9 |
| $S_3$ |  |  | 0 | 9 | 10 | 12 |
| $S_4$ |  |  |  | 0 | 9 | 19 |
| $S_5$ |  |  |  |  | 0 | 11 |
| $S_6$ |  |  |  |  |  | 0 |

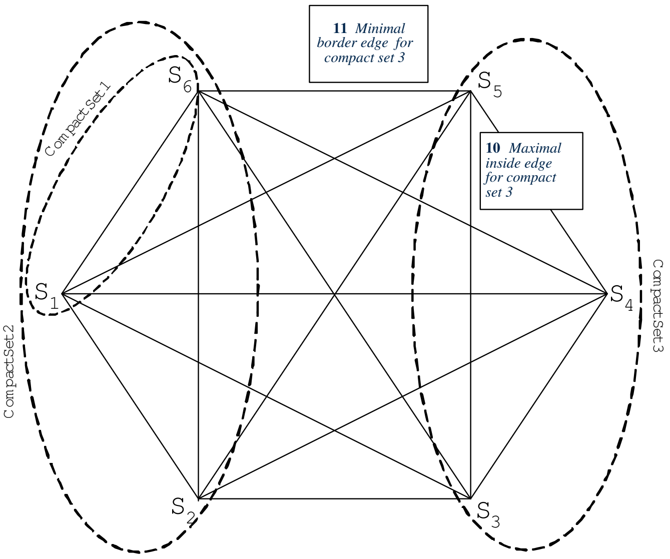
11 Minimal border edge for compact set 3
10 Maximal inside edge for compact set 3
Compact Set 1
Compact Set 2
Compact Set 3
S6
S5
S1
S4
S2
S3

Figure 1. A distance matrix $D$ for 5 sequences and its compact set tree with three nontrivial compact sets $\{S_1, S_6\}, \{S_1, S_2, S_6\}$ and $\{S_3, S_4, S_5\}$ .

Given a compact set $A$ of $S$ such that $A \neq S$, we consider the smallest size compact set $C_A$ that properly conatins $A$. According to Lemma 2.1, $C_A$ is unique. Since, if two different compact sets of $S$ have non-empty intersection. For example, they both contain the compact set $A$. According to Lemma 2.1, one properly contains the other, so the smallest size compact set containing $A$ is unique. We represent the containment relationship of these compact sets by a graph $\mathcal{T}_\mathcal{C} = (V_{\mathcal{T}_\mathcal{C}}, E_{\mathcal{T}_\mathcal{C}})$, such that $V_{\mathcal{T}_\mathcal{C}}$ is the set of all compact sets and each edge $(A, C_A)$ in $E_{\mathcal{T}_\mathcal{C}}$ links a compact set $A$ to the set $C_A$. Clearly, $\mathcal{T}_\mathcal{C}$ is a rooted tree, in which the root represents the compact set $S$, each of its leaves represents a compact set of single sequence of $S$, and each internal node represents the union of the compact sets represented by its children. Given $\mathcal{T}_\mathcal{C}$, the compact sets can be found by travesing $\mathcal{T}_\mathcal{C}$ in postorder and unioning the compact set represented by the children to obtain their parent's compact set. Here, we call $\mathcal{T}_\mathcal{C}$ as a *compact set tree* of $S$ (see Figure **??** for example). For example, we can represnet the compact set in Fingure **??** as $\{\{S_3, S4_,S5\}, \{\{S_1, S_6\}, S_2\}\}$.

## 2.3 Compact Set Neighboring Relation

Let $\mathcal{C}$ be the sets of all compact sets of $S$. For any three species $S_i, S_j, S_k \in S$, if there is a compact set $C$ of $\mathcal{C}$ such that $S_i, S_j \in C$ and $S_k \notin C$, then we say

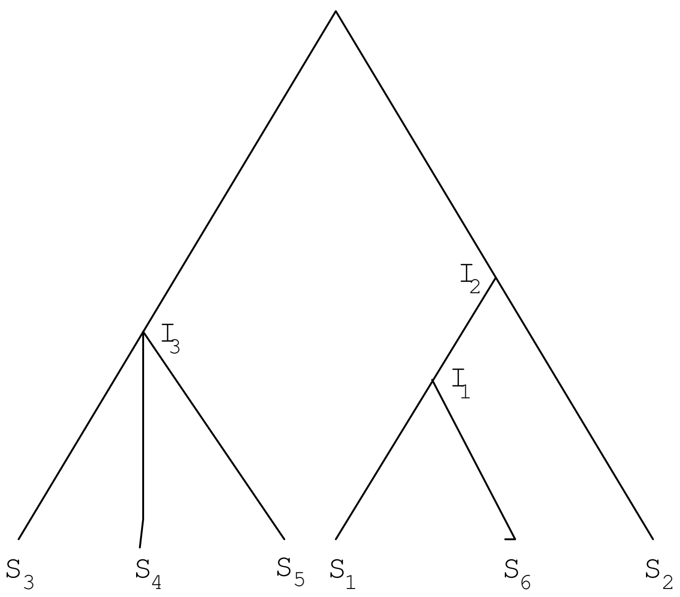

Figure 2. The compact set tree $\mathcal{T}_{\mathcal{C}}$ of above distcne matrx $D$ with three internal node $I_1$, $I_2$, and $I_3$ representing compact sets $\{S_1, S_6\}, \{S_1, S_2, S_6\}, and \{S_3, S_4, S_5\}$ respectively.

that these three sequences have the *compact set neighboring relation* , and denote it by $((S_i, S_j), S_k) \in \mathcal{N}_{\mathcal{C}}$. Let $\mathcal{R}$ be the sets of all three sequences in $S$ possessing the neighboring relations with respect to $\mathcal{C}$, i.e., $\mathcal{R} = \{((S_i, S_j), S_k) | ((S_i, S_j), S_k) \in \mathcal{N}_{\mathcal{C}}\}$.

# 3 The Implementation

Before we find all compact set neighboring relations, we should know all the compact sets. we use Kim's algorithm[2] to find all compact sets, then start our algorithm.Our algorithm is composed of two stages:

1. Generate the neghboring relations $\mathcal{R}$ of $\mathcal{C}$ and $\mathcal{R}_{\mathcal{T}}$ from $\mathcal{D}_{\mathcal{T}}$.
2. Calculate the set-difference of $\mathcal{R}$ and $\mathcal{R}_{\mathcal{T}}$.

## 3.1 Stage 1

When given a $\mathcal{C}$, the bellow algorithm generates the total *compact set neighboring relations.*

**Algorithm 1:** Compact Set Neighboring Relations

```
Input: Given $\mathcal{C}$ be a compact set of $S = \{S_1, S_2, \cdots, S_n\}$ represented as
       brackets.
Output: The total compact set neighboring relations.
begin
    NR-STACK =NULL ;
    PARESE-CS($S$) *Construct a compact set tree $\mathcal{T}_{\mathcal{C}}$ from $\mathcal{C}$ ;
    for each child $v$ of root($\mathcal{T}_{\mathcal{C}}$) do
        NERCOM ($S, v$) ;
end
```

Let $\mathcal{C}_v$ be the compact set belonging to the node $v$ of the compact set tree $\mathcal{T}_{\mathcal{C}}$. We now describe the subroutine NERCOM $(s, v)$:

```
Procedure PARESE-CS($s$)
begin
    repeat
        switch $\mathcal{W}$ =getc(s) do
            case {
                Push $\mathcal{W}$ into STACK ;
            case Species
                Push $\mathcal{W}$ into STACK ;
            case }
                1. Pop the species from STACK until the top of STACK is {.
                2. Pop {.
                3. Create a new internal node $I$ and assign the species poped in step1 as
                   children.
                4. Push $I$ into STACK.
    until Reading to the end of $s$;
end
```

```
Procedure NERCOM($s, v$)
begin
    $C_{out} = s \setminus C_v$ ;
    NR-STACK $\Leftarrow ((S_i, S_j), S_k)$, for all $S_i, S_j \in \mathcal{C}_v andS_k \in C_{out}$ ;
    for each child $u$ of $v$ and $u$ isn't a leave do
        NERCOM $(\mathcal{C}_v, u)$ ;
end
```

For example, the input compact set is represented as $\{\{S_3, S4, S5\}, \{\{S_1, S_6\}, S_2\}\}$.

Its compact set tree $\mathcal{T_C}$ can be obtained after running step 1, like shown in Figure ??. The root node of $\mathcal{T_C}$ contains of two children $I_2$ and $I_3$ so two subroutines NERCOM $(S, I_2)$ and NERCOM $(S, I_3)$ are called respectively. In subroutine NERCOM $(S, I_2)$, it generates neighboring relation listed in Table 1. Because internal node $I_2$ contains of a child, $I_1$, another subroutine NERCOM $(C_{v_2}, I_1)$ is called. Neighboring

| $I_2$ | $I_1$ | $I_3$ |
|---|---|---|
| $((S_1,S_2),S_3)$ | $((S_1,S_6),S_2)$ | $((S_3,S_4),S_1)$ |
| $((S_1,S_2),S_4)$ | | $((S_3,S_4),S_2)$ |
| $((S_1,S_2),S_5)$ | | $((S_3,S_4),S_6)$ |
| $((S_1,S_6),S_3)$ | | $((S_3,S_5),S_1)$ |
| $((S_1,S_6),S_4)$ | | $((S_3,S_5),S_2)$ |
| $((S_1,S_6),S_5)$ | | $((S_3,S_5),S_6)$ |
| $((S_2,S_6),S_3)$ | | $((S_4,S_5),S_1)$ |
| $((S_2,S_6),S_4)$ | | $((S_4,S_5),S_2)$ |
| $((S_2,S_6),S_5)$ | | $((S_4,S_5),S_6)$ |

Table 1. The neighboring relation in internal nodes $I_2, I_1$ and $I_3$.

relation generated in this subroutine is also listed in Table 1. We note that it does not generate duplicate neighboring relation been generated in its parent. In subroutine NERCOM $(S, I_3)$, it is the same. The respective $C_v$ and $C_{out}$ of each internal node are shown in Figure 3.

There are $n$ species poped and pushed in the procedure PARESE-CS. Because each couple brackets merges at least one species, there are at most $O(n)$ couple brackets poped and pushed in PARESE-CS. The number of internal node poped and pushed is also at most $O(n)$. Conbining above analysis, we have the cost of step 1 is $O(n)$. In the procedure NERCOM, it generates every neighboring relation only once. Therefor, denoting by $p$ the number of compact set neighborong relations, we have that the total work of NERCOM is $O(p)$. Conbining this two results, we have

Theorem 3.1. *The total neighboring relation with respect to $\mathcal{C}$ can be generated in time $O(n+p)$.*

## 3.2 Stage 2

Given a topology of $\mathcal{T}$, we can list a linear inequality satisfying that tha path length between leaves $i$ and $j$ on $\mathcal{T}$ is larger than or equal to $D(i,j)$, for any two $i,j \in V$. Then, we can get $D_{\mathcal{T}}$ by solving the linear porgramming minimizing the tree size. For any three sequences $i,j,k \in V$, if $D_{\mathcal{T}}(i,j) \leq \min\{D_{\mathcal{T}}(i,k), D_{\mathcal{T}}(j,k)\}$, then we say that these three sequences have the *neighboring relation* with respect

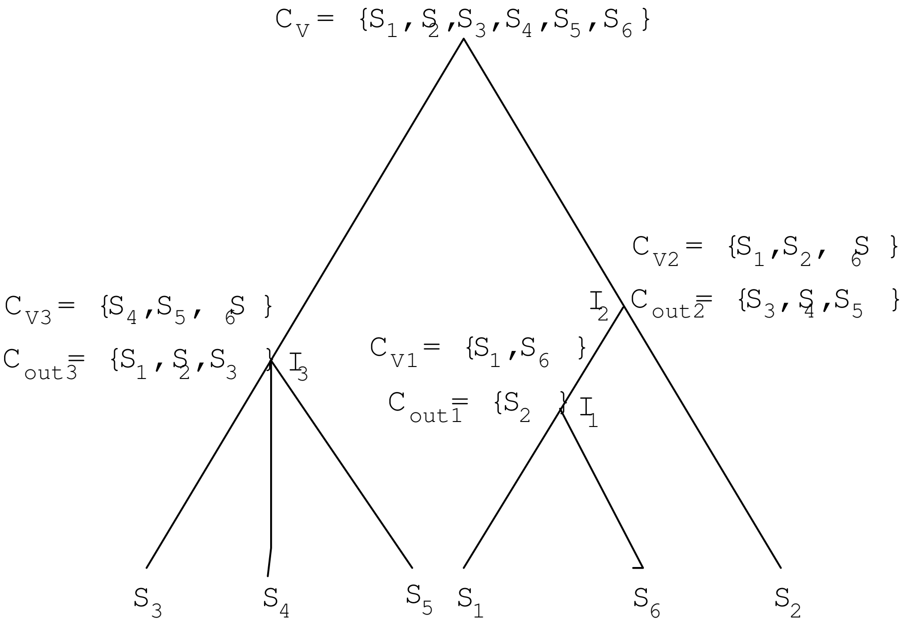

Figure 3. The respective $C_v$ and $C_{out}$ of each internal node.

to $\mathcal{T}$, and denote it by $((i,j),k) \in \mathcal{N}_{\mathcal{T}}$. Let $R_T$ be the sets of all three sequences in $V$ possessing the neighboring relation with respect to $\mathcal{T}$. Then we define the *preserved neighboring ratio* of $T$ to be $\mathcal{P}(\mathcal{T}) = \frac{|\mathcal{R} \cap \mathcal{R}_{\mathcal{T}}|}{|\mathcal{R}|}$.

# 4 Exprimental Results

We choose the two algorithms NJA and GA with some real data, then calculate their preserved neighboring ratio.

The data sets we use in the expriment are as follow and their $r$ are smaller than 30.

1. The first data set, denoted by PROTEIN12, consists of 12 protein sequences with length 80–160 residues.

2. The second data set, denoted by DNA28, consists of 28 DNA sequences of fruit flies with length 800–900 nucleotides.

3. The third,fourth, fifth and sixth date set, denoted by HUMAN34-1, HUMAN34-2, HUMAN34-3 and HUMAN34-4 respectively, each of them consists of 34 DNA sequences of human mitochondria with length 660–690 nucleotides.

4. The other 5 data sets are selected from the BAliBASE [4, 5] benchmark alignment database, denoted by 1vln(4), 1ycc(4), 2abk(4), 1pysA(5) and 2cba(5) with length 118–236, 105–190, 211–344, and 234–328, respectively, where the number in the brackets denotes which reference sets this data set belogs to. 1vln(4) and 2cba(5) consist of 14 and 8 sequences, respectively. However, we have to remove some sequences from the original data sets such that the program can successfully output result. 1ycc(8) has 8 sequenes, which is the same as BAliBASE except sequence 1etp; 2abk(4) has 5 sequenes, which is the same as BAliBASE except sequence 1MPGA; 1pysA(5) has 7 sequenes, which is the same as BAliBASE except sequence 1aszB, 1adjA and 1lylA.

Our experimental flow chat is shown as Figure 4.

Table 2 shows the preserved neighboring ratios of $T_{Kru}$ and NJ. In this case, the trees almost have the equal number of outperformance.

| Data | NJA | GA |
|---|---|---|
| PROTEIN12 | 41.35% | 41.35% |
| DNA28 | 59.57% | 73.48% |
| HUMAN34-1 | 80.50% | 90.50% |
| HUMAN34-2 | 79.66% | 82.63% |
| HUMAN34-3 | 65.34% | 85.96% |
| HUMAN34-4 | 82.70% | 81.45% |
| 1vln(4) | 38.37% | 53.49% |
| 1ycc(4) | 52.94% | 23.53% |
| 2abk(4) | 60.00% | 30.00% |
| 1pysA(5) | 65.52% | 48.28% |
| 2cba(5) | 42.50% | 32.50% |

Table 2. The results of preserved compact-set ratio.

# 5 CONCLUSIONS AND FUTURE WORK

From the topology viewpoint, we propose a new measurment of preserved neighboring ratio to evalute the quality of evolution tree. When there are two evolution trees with the same or similar tree size, we may chioce the one with the larger preserved neighboring ratio. In the following, we propose some open problems concerning our compact set evalutation.

1. Can we classify the nerighboring relations with respect to compact set according to the height of least common ancestor ?
2. When there are few non-trivial compact sets, can we relax compact set restrictions? For example, any inside is less than border edge multiplying a constant number.

# References

[1] E. Dekel, J. Hu, and W. Ouyang. An optimal algorithm for finding compact sets. *Information Processing Letters*, 44:285–289, 1992.

[2] S.K. Kim. A note on finding compact sets in graphs represented by an adjacency list. *Information Processing Letters*, 57:335–338, 1996.

[3] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.

[4] J.D. Thompson, F. Plewinak, and O. Poch. Balibase: a benchmark alignment database for the evolution of multiple sequence alignments. *Bioinformatics*, 15:87–88, 1999.

[5] J.D. Thompson, F. Plewinak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.*, 27:2682–2690, 1999.