



CBTH: A New Algorithm for Maximum Rooted Triplets Consistency Problem

Hadi Poormohammadi*, Mohsen Sardari Zarchi

Department of Computer Engineering, Meybod University, Meybod, Iran

*Corresponding author: Hadi Poormohammadi, Department of Computer Engineering, Meybod University, Meybod, Iran; Tel: +983533212453;
E-mail: pormohamadi@meybod.ac.ir

Background: Phylogenetics is a branch of bioinformatics that studies and models the evolutionary relationships between currently living species. A phylogenetic tree is the simplest possible model in which leaves are distinctly labeled by species. Rooted triplets are one of the most important inputs for constructing rooted phylogenetic trees. A rooted triplet is the simplest possible rooted tree that contains information and explains the biological relation between three species.

Objectives: The problem of constructing a rooted phylogenetic tree that contains the maximum number of input triplets is a maximization problem and is known as the maximum rooted triplets consistency (MRTC) problem. MRTC problem is an NP-hard problem, so there is no any polynomial exact solution for it. The goal is to introduce a new efficient method to solve MRTC.

Materials and Methods: In this research, a new algorithm called CBTH, is introduced innovatively for MRTC problem with the goal to improve the consistency of input rooted triplets with the final rooted phylogenetic tree.

Results: In order to show the efficiency of CBTH, the CBTH is compared with TRH on biological data. According to our knowledge, TRH is one of the best methods for MRTC problem on rooted triplets that are obtained from biological data. The Experimental results show that CBTH outperforms TRH based on rooted triplet consistency parameter in the same time order.

Conclusions: The introduced method (CBTH) solve MRTC problem with high performance without increasing time complexity compared to the other state of the art algorithms.

Keywords: Biological sequence; Consistency; Height function; MRTC problem; Rooted phylogenetic tree; Rooted triplet

1. Background

Biological sequences are usually used to analyze the relation between species or taxa. This relation can be obtained by finding a genome sequence which is common among all set of taxa (given input). Phylogenetics is a field in bioinformatics that studies and models the evolutionary relation between species (1,2). In the basic phylogenetics model of evolutionary process, we usually assume that all given taxa are evolved from a common ancestor and the biological differences between them are because of the environmental effects and mutation occurrence in the evolutionary process. Therefore the tree structures are used basically to model the species relations (1). In the tree models leaves represent the given set of taxa and the internal nodes show the ancestor of the given species.

In the simplest case, the input for obtaining a tree model is in the form of biological sequences. Based on these sequences, different methods are available to build the evolutionary tree models. Furthermore, some methods were introduced to build a tree by using a species distance matrix. The distance matrix is obtained from the biological sequences processing and by considering some standard criterions (1). In this case, the evolutionary tree is weighted. The optimum structure is a tree in which the sum of edge weights between each two leaves is equals its corresponding value in the distance matrix. Since, in the process of obtaining the distance matrix, various criterions are considered; by changing these criterions, different distance matrix can be obtained. Therefore, some new methods are proposed to convert biological sequences

into standard and robust inputs for constructing the evolutionary trees (1).

In order to have a reliable tree model, in the new approach, a small tree is built for each set of three or four taxa, which is called rooted triplet or quartet, respectively. A rooted triplet is a rooted binary tree with three leaves and a quartet is an unrooted binary tree with four leaves. In a rooted triplet or quartet, leaves are distinctly labeled by a given set of taxa (1). **Figures 1 A and 1B** show an example of a rooted triplet and a quartet. These rooted triplets or quartets are used as the input for constructing a standard model. The goal is finding a tree or network structure in which all inputs (rooted triplets or quartets) are preserved. In this paper, we are interested in rooted triplets.

There are different standard methods for converting sequences information into rooted triplets. Maximum Parsimony (MP) and Maximum Likelihood (ML) are two well-known methods (3,4). Recently TCD method was introduced for constructing rooted triplets from a distance matrix (5). In some circumstances, rooted triplets are obtained directly from laboratory results (6). For a given set of triplets τ , building a rooted tree T in which each rooted triplet is an induced subgraph of T , is one of the important challenge in phylogenetics (1). A triplet t is an induced subgraph of a rooted tree T if it

is obtained from T by removing some edges and then contracting each path of length $l \geq 2$ to an edge (1). Here, when T is obtained, each member of τ is formally called consistent with T (1).

Aho et al. introduced BUILD algorithm to construct a rooted tree consistent with a given set of rooted triplets (7). Their algorithm indicates if there is a rooted tree or not in polynomial time. If the BUILD response is positive, it constructs a unique rooted tree in time order $O(mn)$ where n and m are the number of taxa and the number of input rooted triplets, respectively. Henzinger et al. improved BUILD to time order $Min\{O(n + mn^{\frac{1}{2}}), O(m + n^2 \log n)\}$ (8). Later, Jansson et al. reduced the time order to $Min\{O(n + m \log^2 n), O(m + n^2 \log n)\}$ (9). If BUILD cannot find a tree, the problem is finding a rooted phylogenetic tree that contains the maximum number of rooted triplets that is an optimization problem.

In this problem, the goal is removing minimum information (rooted triplets) to build a rooted tree based on the remaining rooted triplets. This problem is called Maximum Rooted Triplets Consistency, the MRTC problem for short. MRTC is known as an NP-hard problem which receives more attention recently (10,11,12,13). Regarding BUILD algorithm, Gasieniec et al. proposed One-Leaf-Split and Min-Cut-Split algorithms (13). One-Leaf-Split guarantees that at

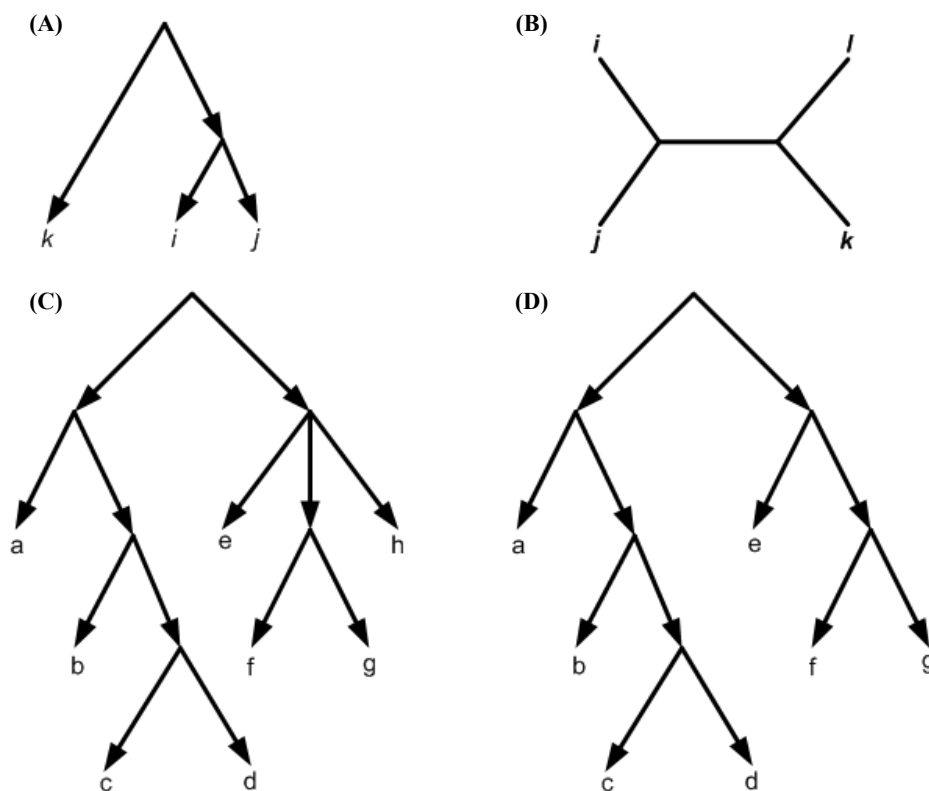


Figure 1: (A) A rooted triplet with leaves $\{i, j, k\}$. (B) A quartet with leaves $\{i, j, k, l\}$. (C) A rooted phylogenetic tree. (D) A rooted binary tree.

least one third of rooted triplets are consistent with the resulting rooted phylogenetic tree. The resulting tree is not reliable since in the worst case two third of information is removed. Therefore, practically One-Leaf-Split is not efficient and is used only to show a lower bound for MRTC problem. Min-Cut-Split algorithm follows BUILD procedure. The difference is when BUILD stops. In this situation, Min-Cut-Split algorithm removes some rooted triplets, which let it to continue the process to build a tree. In fact, Min-Cut-Split algorithm result is far away the optimum solution and is just proposed to precede BUILD algorithm when it halts (13). Wu introduced an algorithm for MRTC problem in time order $O((m+n^2)3^n)$ and memory order $O(2^n)$ (14). Although this algorithm gives an optimum solution, its running time is exponential respect to n . Therefore, by increasing the number of taxa (n), it needs unlimited time to give an output. Additionally, Wu proposed BPF algorithm for this problem in time order $O(mn^3)$ (14). This algorithm uses a bottom-up approach, which performs well on the randomly generated data and on average outperforms the previous existing methods. Maemura et al. introduced BPF method in time order $O(mn^3)$ as an improvement of BPF (15). Jahangiri et al. introduced FastTree and BPFTR for MRTC problem (16). BPFTR is an extended version of BPF that gives results with higher consistency compared to BPF in time order $O(mn^3)$. FastTree execution time is $O(m + \alpha(n)n^2)$ where $\alpha(n) < 4$ for any practical input size n (16). Therefore, FastTree running time is lower than BPFTR but its consistency is worse than BPFTR (16). In addition, Poormohammadi introduced TRH algorithm based on the height function idea in time order $O(mn^2 + n^3 \log n)$ (4). On average TRH triplets consistency outperforms BPFTR and BPFTR on the sets of rooted triplets that are obtained from biological sequences. In addition, TRH running time is better, compared to BPFTR and BPFTR on average (4). The disadvantage of TRH is that it employs a random method to convert a non-binary rooted tree into a binary rooted tree in its final steps.

2. Objective

In order to overcome this challenge, we propose a heuristic method to obtain a binary tree in which more rooted triples are covered. The heuristic method uses nine different score functions, which makes the algorithm more efficient without increasing the time complexity.

The structure of this paper is as follows. In section 3, the definitions and notations are explained. Then proposed

algorithm is introduced. Section 4 describes our results. Finally, in section 5, the results are discussed and the conclusion is presented.

3. Materials and Methods

3.1 Definitions

Definition 1: Let X be a set of taxa. A rooted phylogenetic tree (tree for short) on X is a rooted tree in which the root has indegree=0 and outdegree>1. The leaves' labels of the tree represent the input set of taxa. The internal nodes have indegree=1 and outdegree>1. **Figure 1C** shows an example of a tree. A rooted binary tree is a rooted tree in which all internal nodes have outdegree=2. For example, **Figure 1D** shows a rooted binary tree.

Definition 2: A rooted triplet (triplet for short) is a rooted binary tree with three leaves. In order to show a triplet with leaves i and j in one side as a cherry and k in the other side, the symbol $ij | k$ is used (**Fig. 1A**).

Definition 3: A triplet $ij | k$ is consistent with a tree T and, vice versa if T contains distinct nodes u and v where there are paths $u \rightarrow i, u \rightarrow j, v \rightarrow k, v \rightarrow u$. These paths intersect each other at most in their origin and destination nodes and have no edges in common. Let τ be a set of triplets. τ is consistent with T if each member of τ is consistent with T . For example, **Figure 2A** shows a tree that is consistent with the set of triplets $\{mj|i,kl|i,im|k,ij|k\}$.

Definition 4: Let T be a tree and τ be a set of triplets. The set of all triplets consistent with T and the set of leaves of T are defined by $\tau(T)$ and L_T , respectively. The set of leaves corresponds to τ is defined by $l(\tau) = \cup_{t \in \tau} L_t$. A τ is a set of triplets on X if $L(\tau) = X$. For example $\tau = \{mj|i,kl|i,im|k,ij|k\}$ is a set of triplets on $X = \{i, j, k, l, m\}$.

Definition 5: A rooted tree T' is a binarization of a rooted tree T if T' is a binary tree and $L_{T'} = L_T$ and the set of triplets that is consistent with T is a subset of the set of triplets that is consistent with T' (5). **Figure 2. B-C** shows an example of a tree and one of its binarization.

Definition 6: The directed graph related to τ is defined by G_τ where $V(G_\tau) = \{\{i, j\} : i, j \in L(\tau), i \neq j\}$ and $E(G_\tau) = \{(ij, ik) : ij|k \in \tau\} \cup \{(ij, jk)\}$. For simplicity ij is used instead $\{i, j\}$.

Definition 7: let $\binom{X}{2}$ denotes the set of all subsets with size=2 of X . A height function h on X is in the form of $h: \binom{X}{2} \rightarrow \mathbb{N}$.

Definition 8: let G_τ be a directed acyclic graph and the length of its longest directed path be l_{G_τ} . Since G_τ is acyclic it contains the nodes with $outdegree=0$. In order to obtain h_{G_τ} , the height function related to G_τ ,

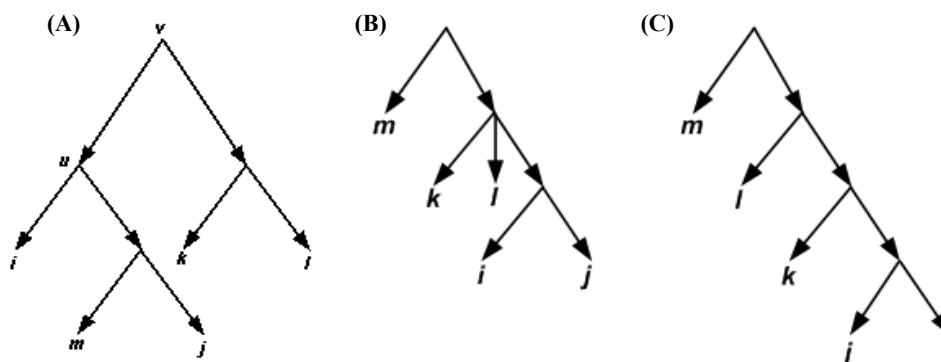


Figure 2: (A) A tree which is consistent with the set of triplets $\{mj|i,kl|i,im|k,ij|k\}$. (B) A tree T. (C) A binarization of T.

the following process is performed. Assign $v=l_{G_\tau}+1$ to the nodes with *outdegree*=0 and remove them. Set $v=v-1$ and perform the following process recursively on the resulting graph. This process will continue until all nodes are removed. At the end all nodes gets a value $1 \leq v \leq l_{G_\tau}+1$. For each pair $i, j \in L(\tau)$, the $h_{G_\tau}(i, j)$ is the value (v) assigned to the node $ij \in V(G_\tau)$ (5). If τ is consistent with a tree then G_τ is acyclic and therefore h_{G_τ} exists (5).

3.2. Preliminary Methods

Let X be set of taxa and τ be a set of triplets on X . Aho graph $AG(\tau) = (V, E)$ related to τ is a graph with $V = X$ and the edge condition: two nodes i and j are connected iff $\exists k \in X$ such that $ij|k \in \tau$ (7). BUILD is an algorithm that constructs a tree consistent with τ based on Aho graph if such a tree exists. Build algorithm starts with assigning root and then determines its descendant in a top-down approach (7). HBUILD algorithm is a height based version of BUILD (5,7). These two algorithms are explained in reminder of this section.

3.2.1. BUILD Algorithm

The goal of BUILD is construct a rooted tree consisted with τ (if exists). If $|X| \leq 2$, BUILD returns a rooted tree with at most two leaves. Otherwise, BUILD computes Aho Graph, $AG(\tau)$. If $AG(\tau)$ itself is a connected component, BUILD stops i.e. there is not any tree consistent with τ . Otherwise, a tree exists and is constructed in what follows. For each set of nodes in a connected component U of $AG(\tau)$, the set $\tau|_U$ is computed. The set $\tau|_U$ shows the triplets of τ that all their leaves are in U . Then recursively the rooted subtree $T(\tau|_U)$ that is consistent with $\tau|_U$ is computed. A node r is considered as the root of a tree T and each subtree is connected to r via an edge. The tree T is consistent with τ (7).

3.2.2. HBUILD

Let $h()$ be a height function on X . The weighted complete graph (G, h) with $V(G) = X$ is defined. Here, the $h(i, j)$ shows the weight of the edge $\{i, j\}$. In HBUILD, firstly the edges with maximum weights are removed. If by removing the maximum edges from a connected component, the graph remains connected, HBUILD stops. Otherwise this process continues until each connected component contains only one node. At the end and by reversing the steps of removing the edges a unique tree in which the leaves are labeled by X is constructed in polynomial time (5).

3.3. The proposed Algorithm

As mentioned before TRH is an efficient height based method for MRTC problem; but using a random binarization is its main drawback. Therefore in this paper an intellectual algorithm based on TRH is proposed for solving MRTC problem. In the proposed algorithm, we innovatively used a heuristic method instead of random binarization. Since our algorithm is based on the height function and binarization, we called it CBTH (Constructing Binary Tree with Height).

The CBTH input is a set of triplets τ . If there is a tree consistent with τ then CBTH constructs a tree based on HBUILD algorithm (5). If there is no tree consistent with τ the following steps are performed in the proposed algorithm. Firstly, a height function based on G_τ is obtained. If G_τ is acyclic, h_{G_τ} is obtained. Else minimum number of edges are removed from G_τ to obtain an acyclic graph G'_τ , and then the $h_{G'_\tau}$ is achieved. Removing minimum number of edges to obtain G'_τ is a type of Minimum Feedback Arc Set (MFAS) problem. MFAS is an optimization problem and is known as a NP-hard problem (17). Different methods are available for MFAS problem. In this paper we use GR (17) algorithm for MFAS. For simplicity without loss of generality, we

use G_τ and h_{G_τ} instead of G'_τ and $h_{G'_\tau}$, respectively. After obtaining G_τ and h_{G_τ} , the proposed algorithm constructs a rooted phylogenetic tree T , and then makes it binary using a heuristic method. In order to build T , the maximum edges are removed from the graph (G_τ, h_{G_τ}) similarly to HBUILD (5) algorithm. If in a step of removing the maximum edges from a connected component C , the resulting graph C' remains connected, it concludes that the set of triplets are not consistent with a tree (5). Then the following process is applied to disconnect C' to achieve the top-down structure of T . The maximum edges are removed until the resulting becomes disconnected. Let in the resulting graph the number of connected components is c . If $c > 2$, this process is replaced with the Min-Cut (18) algorithm. Min-Cut is an algorithm that removes minimum number of edges from a connected component to obtain a disconnected graph with exactly two components. The aim of using Min-Cut algorithm in some steps is to achieve a structure similar to the binary tree structure. The removing edge process continues until each connected component contains only one node. After this process, by reversing the edge removing steps, the components are merged to build T in a bottom-up procedure.

The final step of our proposed algorithm is binarization process. In this process we use a heuristic algorithm with considering three parameters w, p, t that were defined in BPF (14) algorithm. For a set of triplets τ , let V_i and V_j be two subsets of $L(\tau)$ such that $V_i \cap V_j = \emptyset$. These parameters are defined as follows (14):

- (i) $W(V_i, V_j) = \{v_i v_j \mid v \in \tau \text{ such that } v_i \in V_i, v_j \in V_j \text{ and } v \notin V_i \cup V_j\}$ and $w = |W|$.
- (ii) $P(V_i, V_j) = \{v_i v_j \mid v \in \tau \text{ or } v_j v_i \mid v_i \in V_i, v_j \in V_j \text{ and } v \notin V_i \cup V_j\}$ and $p = |P|$.
- (iii) $T(V_i, V_j) = \{v_i v_j \mid v \in \tau \text{ such that } v_i \in V_i, v_j \in V_j\}$ and $t = |T|$.

By using the idea of e-score used in BPF (14), and the above three parameters, nine different measures are defined to find the best configuration

for the binary tree. The nine measures are:

$$t, w, w-p, \frac{w}{w+p}, \frac{w}{t}, \frac{w-p}{w+p}, \frac{w-p}{t}, w-p + \frac{w}{t}, \frac{w-p}{t} + \frac{w}{w+p}$$

If the tree T is binary there is nothing to do. Else there is at least a node x that has x_1, x_2, \dots, x_k children where $k > 2$. The descendent of x is a subtree that should be replaced by a binary tree. Let V_i be the set of all leaves of T that are the descendent of $x_i, 1 \leq i \leq k$. Obviously $V_i \cap V_j = \emptyset, 1 \leq i, j \leq k, i \neq j$. Choose a measure from nine defined measures. Based on the measure, the two sets V_i and $V_j, 1 \leq i, j \leq k, i \neq j$ are selected and merged in such a way that the value of the measure is maximum. This process continues until only two sets remains. By applying this process, the subtree with root x is replaced with a binary tree. In CBTH, the process of subtree binarization is started from the leaves and will continue in a bottom-up approach. **Figures 2B and 2C** show an example of binarization.

The process of binarization is repeated for each measure. So we have nine different binary trees. Each of this nine tree is a binarization of T . The best tree among these nine trees according to triplet consistency is reported as the output of CBTH algorithm.

4. Results

Generally the inputs of MRTC problem are real data i.e. sets of triplets that are achieved from biological sequences (4,5,19). Hence, in the first step of the experiment, 2000 sets of biological sequence are generated using TREEVOLVE (3). The size of each set is 15. Then triplets are generated from the sequences by using Maximum Likelihood (ML) criterion (3). Threshold is a criterion to obtain reliable triplets (5). The threshold is in the range $[0,1]$. By increasing the threshold value, the number of triplets decreases. To evaluate the proposed algorithm, the CBTH results are compared with TRH results. TRH is selected because it is the best method on average for MRTC problem on the sets of real data. The results are shown in **Table 1**. In addition, CBTH is compared with TRH on the randomly generated sets of triplets. For this purpose, the two parameters n and m are used, where n is the number of taxa and m is the number of triplets. For

Table 1: The CBTH and TRH experimental results on 2000 sets of triplets and for different thresholds

Threshold	0	0.2	0.4	0.6	0.8	0.9
Percent of trees that TRH outperforms CBTH	0.7 %	0.6 %	0.7 %	0.6%	0.5 %	0.5 %
Percent of trees that CBTH outperforms TRH	24 %	24 %	20 %	23 %	19 %	17 %
Average percent of triplets consistency for TRH	81 %	83 %	84 %	85 %	89 %	90 %
Average percent of triplets consistency for CBTH	84 %	85 %	85 %	88 %	90 %	91 %

Table 2: The CBTH and TRH experimental results on 2000 sets of triplets for $n = 15$ and $m = 50, 100, 200, 300$.

Number of triplets	50	100	200	300
Percent of trees that TRH outperforms CBTH	0.6 %	0.4 %	0.4 %	0.5 %
Percent of trees that CBTH outperforms TRH	26 %	22 %	21 %	17 %
Average percent of triplets consistency for TRH	59 %	51 %	43 %	39 %
Average percent of triplets consistency for CBTH	64 %	53 %	49 %	43 %

$n = 15$ and $m = 50, 100, 200, 300$, randomly 2000 samples are generated i.e. for each case 500 samples are generated. The results are shown in **Table 2**.

The results on the real and randomly generated triplets show that on average CBTH outperforms TRH.

In order to study the CBTH time complexity, let for a set of triplets τ , $|\tau| = m$ and $L(\tau) = n$. Firstly G_τ is obtained in time order $O(m)$. If G_τ is not acyclic, GR algorithm is applied in time order $O(|edges|)$ which is equivalent to $O(m)$. Then the nodes with *outdegree* = 0 are recognized and topological sort is performed on the new acyclic graph in time order $O(|nodes| + |edges|)$ or equivalently $O(m + n^2)$. Then the height function $h()$ is assigned in time order $O(n^2)$. In the next step, the graph (G, h) is constructed in time order $O(n^2)$. So the time complexity of the above steps is $O(m + n^2)$. Now the edges removing steps is done. Removing the maximum edges from each connected component is done in time order $O(m)$. Also in each step, the number of connected components should be compared with the previous step by using the DFS (Depth First Search) algorithm in time order $O(n)$. So the runtime is $O(mn)$. The total runtime for n nodes is $O(mn^2)$. Also Min-Cut algorithm is performed in time order $O(mn + n^2 \log n)$ and totally for at most n nodes in time order $O(mn^2 + n^3 \log n)$. So the time order of maximum edges removing step is $O(mn^2 + n^3 \log n)$. Also the binarization process is done in time order less than $O(mn^2 + n^3 \log n)$. So the CBTH time order is $O(mn^2 + n^3 \log n)$.

5. Discussion

In this paper, we focused on MRTC problem. According to our best knowledge, TRH is the best method for MRTC problem on the sets of triplets that are obtained from biological sequences i.e. real data. However, using a random binarization is the main disadvantage of TRH. Therefore, in this paper, CBTH algorithm based on TRH was proposed. In the proposed algorithm, we innovatively used a heuristic method instead of random binarization. Then we compared the CBTH results with the TRH results on the triplets that are obtained from biological sequences and randomly generated triplets. The results show that CBTH outperforms TRH.

In details, the results on 2000 generated sets of real data show that for different thresholds and in the best situation, at most in 0.7% of cases TRH results outperforms CBTH results; while for these data in at least 17% and at most 24% of cases CBTH outperforms TRH. Also in all cases the percent of triplet consistency for CBTH results are better than TRH results. Note that the time complexity of both methods is the same and is $O(mn^2 + n^3 \log n)$ (4). So by considering the main two parameters for evaluating an algorithm for MRTC problem i.e. time complexity and triplet consistency, CBTH is the best method for MRTC problem on the triplets that are obtained from biological sequences and by using Maximum Likelihood (ML) criterion.

Also on the randomly generated triplets we compared the CBTH and TRH results. To achieve this purpose for $n = 15$ and $m = 50, 100, 200, 300$ where n is the number of taxa and m is the number of triplets, and for each case we generated 500 sets of triplets i.e. totally 2000 sets of triplets were randomly generated. The results show that in the best situation, at most in 0.6% of cases TRH results outperforms CBTH results; while in all cases and in at least 17% of cases the CBTH results is better than TRH results. Also like real data, in all cases the percent of triplet consistency for CBTH results are better than TRH results. Generally, the results show that CBTH outperform TRH.

Acknowledgment

The authors would like to thank Mehdi Kargar for his useful comments.

Conflict of Interest

All authors declare that there is no conflict of interest.

Authors' Contribution

The authors contributed to this work equally.

References

- Felsenstein J. Inferring phylogenies. *Sunderland, MA: Sinauer associates*. 2004.
- Bouckaert R, Lemey P, Dunn M, Greenhill SJ, Alekseyenko AV, Drummond AJ, et al. Mapping the origins and expansion of the Indo-European language family. *Science*. 2012; **337**(6097):957-60. doi: 10.1126/science.1219669.

3. Van-Iersel L, Kelk S. Constructing the simplest possible phylogenetic network from triplets. *Algorithmica*. 2011; **60**(2):207-235. doi: 10.1007/s00453-009-9333-0
4. Poormohammadi H. A New Heuristic Algorithm for MRTC Problem. *J Emerg Trends Comput Inform Sci*. 2012;**3**(7):1037-1041.
5. Poormohammadi H, Eslahchi CH, Tusserkani R. TripNet: A method for Constructing Rooted Phylogenetic Networks from Rooted Triplets. *PLoS One*. 2014; **9**(9):p.e106531. doi: 10.1371/journal.pone.0106531.
6. Kannan S, Lawler E, Warnow T. Determining the evolutionary tree using experiments. *J. Algorithms*. 1996;**21**(1):26-50. doi: 10.1006/jagm.1996.0035.
7. Aho AV, Sagiv Y, Szymanski TG, Ullman JD. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *Siam J Comput*. 1981;**10**(3):405-421. doi: 0097-5397/81/1003-0001 \$01.00/0.
8. Henzinger MR, King V, Warnow T. Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. *Algorithmica*. 1999;**24**(1):1-13. doi: 10.1007/PL00009268.
9. Jansson J, Ng JH, Sadakane K, Sung WK. Rooted maximum agreement supertrees. *Algorithmica*. 2005;**43**(4):293-307. doi: 10.1007/s00453-004-1147-5.
10. Bryant D. Building trees, hunting for trees, and comparing trees-theory and methods in phylogenetic analysis. *Ph.D. thesis, University of Canterbury, New Zealand*. 1997.
11. Jansson J. On the complexity of inferring rooted evolutionary trees. *Electron Note Discret Math*. 2001; **7**:1-4. Doi: 10.1016/S1571-0653(04)00222-7.
12. Byrka J, Gawrychowski P, Huber KT, Kelk S. Worst-case optimal approximation algorithms for maximizing triplet consistency within phylogenetic networks. *J Discrete Algorithms*. 2010;**8**(1):65-75. doi: 10.1016/j.jda.2009.01.004.
13. Gasieniec L, Jansson J, Lingas A, Ostlin A. On the complexity of constructing evolutionary trees. *JCO*. 1999;**3**(2-3):183-197. doi: 10.1023/A:1009833626004.
14. Wu BY. Constructing the maximum consensus tree from rooted triples. *J Comb Optim*. 2004;**8**(1):29-39. doi:10.1023/B:JOCO.0000021936.04215.68.
15. Maemura K, Jansson J, Ono H, Sadakane K, Yamashita M. Approximation algorithms for constructing evolutionary trees from rooted triplets. 10th Korea-Japan joint workshop on algorithms and computation (WAAC). 2007: 56-63.
16. Jahangiri S, Hashemi SN, Poormohammadi H. New heuristics for rooted triplet consistency. *Algorithms*. 2013;**6**(3):396-406. doi: 10.3390/a6030396.
17. Eades P, Lin X, Smyth WF. A fast and effective heuristic for the feedback arc set problem. *Inform Process Lett*. 1993;**47**(6):319-323. doi: 10.1016/0020-0190(93)90079-O.
18. Stoer M, Wagner F. A simple min-cut algorithm. *J ACM*. 1997;**44**(4):585 - 591.
19. Poormohammadi H, Zarchi M S, Ghaneai H. NCHB: A method for constructing rooted phylogenetic networks from rooted triplets based on height function and binarization. *J Theor Biol*. 2020;**489**:110-144. doi:10.1016/j.jtbi.2019.110144.