Do Triplets Have Enough Information to Construct the Multi-Labeled Phylogenetic Tree?

Reza Hassanzadeh[1], **Changiz Eslahchi**[2]*, **Wing-Kin Sung**[3,4]

1 Department of Mathematics, Shahid Beheshti University, G.C., Tehran, Iran, **2** Department of Computer Science, Shahid Beheshti University, G.C., Tehran, Iran, **3** School of Computing, National University of Singapore, Singapore, Singapore, **4** Genome Institute of Singapore, Singapore, Singapore

Abstract

The evolutionary history of certain species such as polyploids are modeled by a generalization of phylogenetic trees called multi-labeled phylogenetic trees, or MUL trees for short. One problem that relates to inferring a MUL tree is how to construct the smallest possible MUL tree that is consistent with a given set of rooted triplets, or SMRT problem for short. This problem is NP-hard. There is one algorithm for the SMRT problem which is exact and runs in $O(7^n)$ time, where n is the number of taxa. In this paper, we show that the SMRT does not seem to be an appropriate solution from the biological point of view. Indeed, we present a heuristic algorithm named MTRT for this problem and execute it on some real and simulated datasets. The results of MTRT show that triplets alone cannot provide enough information to infer the true MUL tree. So, it is inappropriate to infer a MUL tree using triplet information alone and considering the minimum number of duplications. Finally, we introduce some new problems which are more suitable from the biological point of view.

Citation: Hassanzadeh R, Eslahchi C, Sung W-K (2014) Do Triplets Have Enough Information to Construct the Multi-Labeled Phylogenetic Tree? PLoS ONE 9(7): e103622. doi:10.1371/journal.pone.0103622

Editor: M. Sohel Rahman, Bangladesh University of Engineering and Technology, Bangladesh

Received December 1, 2013; **Accepted** July 4, 2014; **Published** July 31, 2014

Funding: These authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

* Email: ch-eslahchi@sbu.ac.ir

Introduction

MUL trees are rooted phylogenetic trees where some leaves are labeled by the same taxa. They find applications in the study of the evolution of polyploids. The other applications of MUL trees include molecular systematics, biogeography, the study of host-parasite cospeciation and computer science [8,11,15,18–20,22]. In this paper we focus on rooted binary MUL trees. Several algorithms for constructing MUL trees from various datasets are introduced. Examples include building consensus MUL trees [6,14,15], constructing a phylogenetic network from a MUL tree [10] and transforming a collection of MUL trees into a collection of evolutionary trees [23]. One of the problems in the field of inferring MUL trees is to construct a smallest possible MUL tree consistent with a given set of rooted triplets, or SMRT problem for short. It is proved that SMRT is an NP-hard problem [9]. Up to now, a number of algorithms for inferring a phylogenetic tree or network from a set of triplets are presented [1,4,12,13,24–26]. However, there is only one algorithm for constructing a smallest possible MUL tree from a set of triplets [9]. This algorithm is exact and runs in $O(7^n)$ time where n is the number of taxa. Here, we present the MTRT algorithm which is a heuristic method for the SMRT problem. MTRT is based on Aho et al.'s algorithm presented in [1]. Aho et al.'s algorithm is a top-down algorithm that constructs a rooted tree consistent with a given set of triplets, if such a tree exists. In the MTRT algorithm, we modify the Aho et al.'s algorithm to construct a MUL tree with the minimum number of duplications that is consistent with a given set of triplets. The duplication in a MUL tree is defined in the next section. We tested the performance of the MTRT algorithm on more than 400 biological and simulated datasets and showed that MTRT is efficient and can often find the optimal answer in practice. Furthermore, we showed that minimizing the number of duplications may not be an appropriate criterion for inferring a MUL tree.

Preliminaries

A rooted triplet, or triplet for short, is a binary rooted tree on three distinct taxa. A triplet on three taxa i, j and k is denoted by $ij|k$ if the lowest common ancestor of i and j is a proper descendant of that of i and k, or j and k. Let $\Re$ be a set of triplets on a taxa set L. For any subset L' of L, the set of all triplets $ij|k \in \Re$ for which $i, j, k \in L'$ is called the set of triplets induced by L' and is denoted by $\Re|_{L'}$. We also set $\Re(L, L') := \{ab|c \in \Re|_L : \ either\ a, b \in L'\ or\ c \in L'\}$. A triplet $ij|k$ and a MUL tree M are said to be consistent if $ij|k$ is an embedded subtree of M. We say that a MUL tree M and a given set $\Re$ of rooted triplets are consistent if every triplet in $\Re$ is consistent with M. The set $\Re(M)$ of all triplets consistent with M is called the triplet encoding of M. The following definitions are taken from [9]:

For any MUL tree M, denote the set of all leaf labels that occur in M by $L(M)$. For any leaf label $x \in L(M)$, the number of duplications of x is equal to the number of occurrences of x in M minus 1. The number of leaf duplications in M, denoted by $d(M)$, is the total number of duplications of all leaf labels in $L(M)$. Define $m(M)$ as the number of leaves in M. Then, $d(M) = m(M) - |L(M)|$. Now, we consider the following problem, called the smallest MUL tree from rooted triplets problem, or SMRT for short:

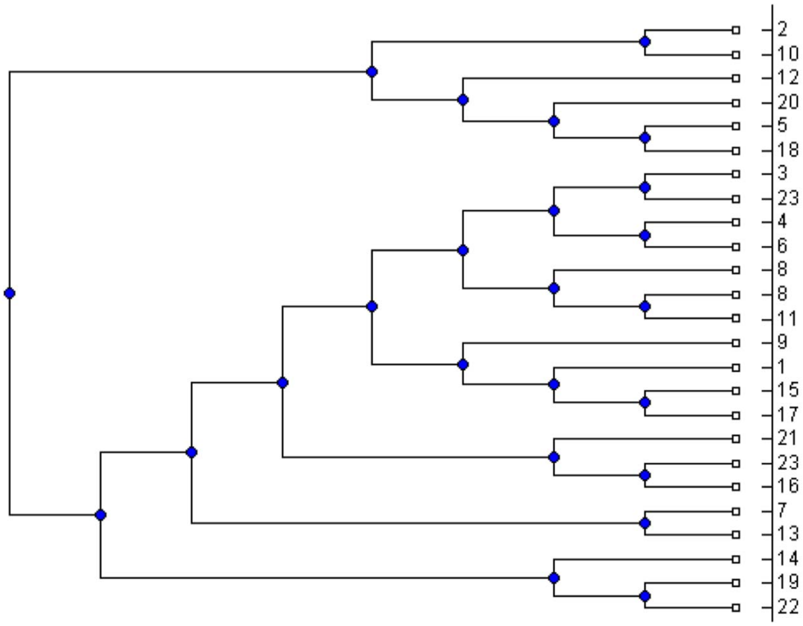


Figure 1. An original MUL tree used to test the MTRT algorithm.
doi:10.1371/journal.pone.0103622.g001

SMRT problem. Given a set $\Re$ of rooted triplets over a leaf label set L, output a MUL tree M with $L(M)=L$ which is consistent with $\Re$ and minimizes $d(M)$.

Results

Simulation data

In this section, we report the results of our simulation study. For all data, the MTRT algorithm was run on a laptop with a 1.8 GHz Dual Core processor and 1GB RAM. MTRT is implemented in MATLAB. To test the performance of the algorithm, we simulated 400 MUL trees by *Mesquite* program [16]. This program can simulate and analyze gene trees from multiple populations. Three components must be established in Mesquite to do this:

1. A block of taxa representing the gene sequences.
2. A block of taxa representing the species (or populations).
3. A taxa association block, which is a special block of information that indicates how the taxa representing genes are associated with the taxa representing species.

Once these three components are established, Mesquite simulates gene trees by a coalescent process. The simulation starts at each extant population. Within each, the ancestry of the gene copies contained (as specified by the Taxa Association) is simulated by coalescence, going backward in time until the simulation arrives at the previous population (species) divergence. Mesquite makes this reconstruction under one assumption: that the only process occurring is gene duplication or extinction. Thus, the reconstruction reconciles the gene tree into the population tree so as to minimize the depths of gene tree divergences, which also minimizes gene duplication or extinction events, see [16] for more details.

Now we describe the procedure of simulating MUL trees. Suppose the gene tree GT produced by Mesquite has n taxa. We considered the number of taxa for the species tree ST associated with GT between $n/2$ and n. Then, we randomly indicated how the taxa representing genes are associated with the taxa representing species to obtain a taxa association block. After the simulation of the gene tree, to obtain a MUL tree, we replaced each gene by the species that belong to it. In all simulations, we considered n between 5 and 50. For each simulated MUL tree, we extracted all its triplets and applied the MTRT algorithm on the triplet set. The results show that in 42 percent of the datasets, MTRT produces a MUL tree which has less number of duplications than that of the original MUL tree. In only 10 percent of the datasets, the number of duplications for the output MUL tree of MTRT is greater than that of the original MUL tree. For the remaining 48 percent, the number of duplications for both MUL trees are the same. Hence, in 90 percent of the datasets, the algorithm MTRT constructs a MUL tree that has less or equal number of duplications than that of the original MUL tree. The minimum, maximum and average running times of the algorithm on 400 simulation datasets are 0.017, 40.36 and 9.1 seconds respectively. Figure 1 shows a simulated MUL tree. The output of the MTRT for the triplet set extracted from this MUL tree is given in Figure 2. The output MUL tree has one duplication while the original MUL tree has two duplications. We also compare MTRT with the exact algorithm presented in [9]. Since the exact algorithm requires exponential time and space, we can only run this algorithm on 100 small datasets which have 5–10 taxa. In 86 datasets, the MUL trees produced by both MTRT and exact algorithm have the same duplications. This shows that MTRT in many cases produces the smallest MUL trees for the triplet sets. For further study, we analysed the results of the exact algorithm. We found that, in 56 datasets, the exact algorithm produces a MUL tree which has less number of duplications than that of the original MUL tree.

Real data

To test the performance of the MTRT on real biological datasets, we applied MTRT on three datasets. The first and second datasets containing high-polyploid North American and Hawaiian violets [17]. All major morphological groups occurring

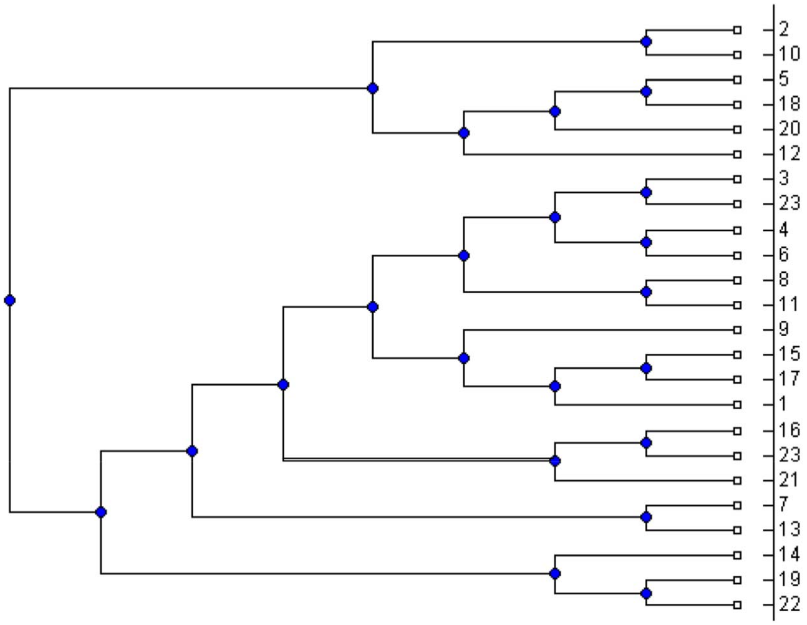


Figure 2. The obtained MUL tree by applying MTRT on the triplets extracted from the MUL tree shown in Figure 1.
doi:10.1371/journal.pone.0103622.g002

in North America were sampled. All sequence were aligned with MUSCLE [7] and phylogenies were constructed using maximum likelihood. The third dataset containing the flowering plant genus Silene (Caryophyllaceae) was published in [21]. The gene trees in [21] are reconstructed using standard techniques in phylogenetic analysis from regions of the nuclear RNA polymerase gene family, two concatenated chloroplast regions and one nuclear ribosomal region, see [10] for more details. For each original MUL tree, we extracted all triplets and then apply MTRT on these triplets. In all cases, MTRT constructs a MUL tree which has less number of duplications than that of the original MUL tree. The original MUL trees for first and second datasets have 13 and 20 duplications, whereas the MUL trees produced by MTRT have 11 and 18 duplications respectively. Due to limitations of space, the MUL trees associated with one of the data are shown. Figure 3 and Figure 4 show the original MUL tree and the MUL tree constructed by MTRT for the triplet set extracted from the original MUL tree respectively. The original MUL tree for third dataset has 7 duplications, whereas the MUL tree produced by MTRT has 5 duplications. Figure 5 and Figure 6 show the original MUL tree and the MUL tree constructed by MTRT respectively. The labels represent Silene species, namely, S. ajanensis (A), S. uralensis (U), S. involucrata (I), S. sorensenis (S), S. ostenfeldii (O), S. zawadskii (Z), S. linnaeana (L), S. uralensis (Mongolia) (UM), S. samojedora (SAM), S. villosula (V), S. sachalinensis (SAC) and S. tolmatchevii (T).

Reconstruction accuracy

For a phylogeny reconstruction algorithm, if a certain tree or network is used to obtain the input data, the algorithm should return exactly this tree or network. This is an important property for reconstructing phylogenies and known as the consistency principle. In the previous section, we observed that, for half of the simulated datasets and two real datasets, the number of duplications for input and output MUL trees are different. Further investigation showed that although some output MUL trees differ from input MUL trees, the outputs are consistent with all triplets corresponding to input MUL trees. In addition, we observed that some output MUL trees have more triplets than the corresponding input MUL trees. These observations show that inferring a MUL tree by minimizing the number of duplications may not properly detect biological properties and evolutionary relationships. So, there is a deficiency in the SMRT problem from a biological point of view. For further analysis, we used a concept which has already been defined for a tree called the rooted triplet distance to compare the output MUL trees with the input MUL trees [5].

Definition 1. The rooted triplet distance between two rooted phylogenetic trees T_1 and T_2 on taxa set X is defined as

$$TD(T_1, T_2) = \frac{1}{2}|\Re(T_1)\Delta\Re(T_2)|,$$

where Δ is the symmetric difference between two sets. For example, for the two MUL trees M_1 and M_1' shown in Figure 7a and Figure 7b respectively, M_1' is consistent with all triplets in M_1 and has less duplication than M_1. Since M_1' satisfies an extra triplet 23|1 which is not contained in $\Re(M_1)$, so $TD(M_1, M_1') = 0.5$. It shows that it is possible to present an algorithm satisfying all conditions of SMRT problem but does not return the correct MUL tree, that is, it does not satisfy the consistency principle of phylogeny reconstruction algorithms. Now, consider another two examples: MUL trees M_2 and M_2' shown in Figure 7c and Figure 7d respectively. These MUL trees have the same number of duplications and $\Re(M_2) = \Re(M_2')$, that is, $TD(M_2, M_2') = 0$. But these are different MUL trees because they have different duplication leaves and have different clusters. This situation happened because in a MUL tree, a triplet may occur several times. For example, the triplet 12|3 occurred three times in the MUL tree shown in Figure 8. This phenomenon exactly occurred in Figure 7c and Figure 7d. For instance, the triplet 12|4 occurred in M_2 once whereas it occurred twice in M_2'.

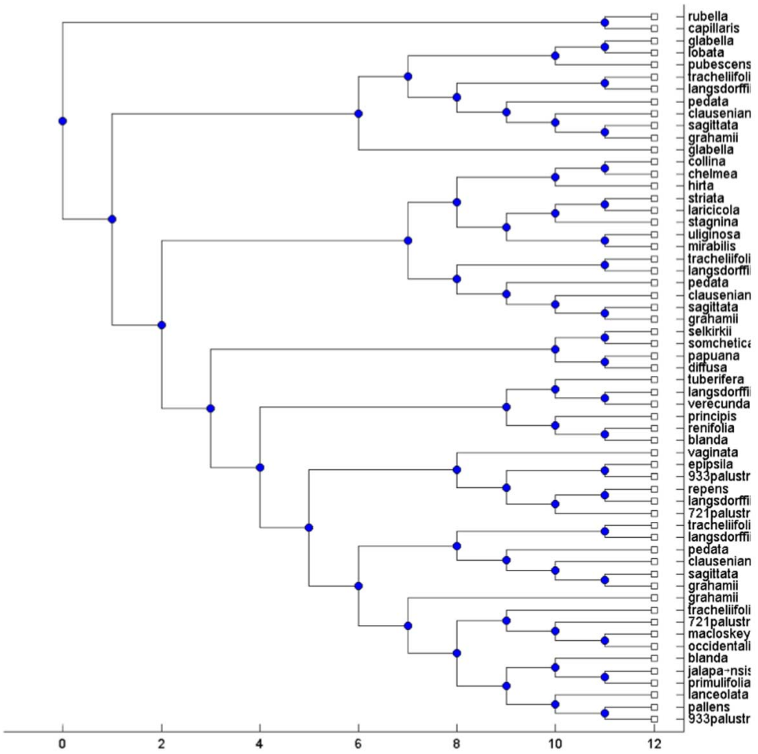


Figure 3. An original MUL tree on violet species with 20 duplications.
doi:10.1371/journal.pone.0103622.g003

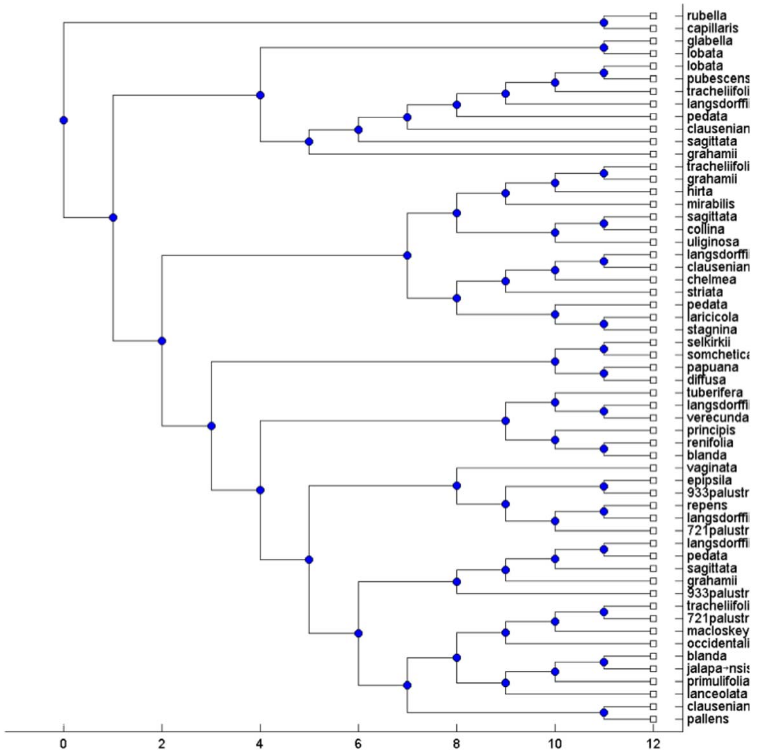


Figure 4. The obtained MUL tree by applying MTRT on the triplets extracted from the MUL tree shown in Figure 3. This MUL tree has 18 duplications.
doi:10.1371/journal.pone.0103622.g004

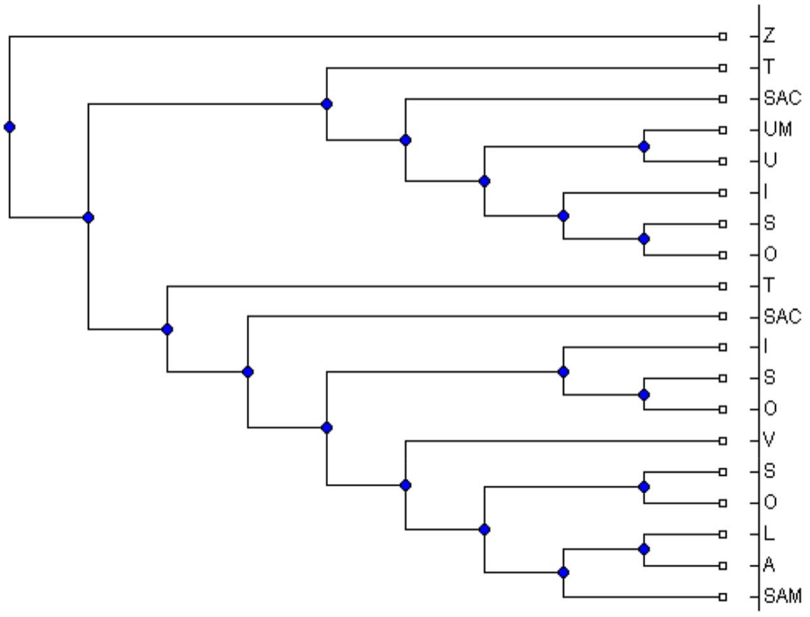


Figure 5. An original MUL tree on flowering plants with 7 duplications.
doi:10.1371/journal.pone.0103622.g005

Hence, the rooted triplet distance introduced in Def. 1 does not properly show the distance between two MUL trees.

A multiset is defined as a 2-tuple (Y, m) where Y is some set and m is a function from Y to the positive natural numbers N. The set Y is called the underlying set of elements. For each $y \in Y$, the multiplicity $m(y)$ is denoted to be the number of occurrences of y. The symmetric difference between two multisets (Y_1, m_1) and (Y_2, m_2) is denoted by $(Y_1, m_1)\Delta_m(Y_2, m_2) := \{x \in Y_1 \bigcup Y_2 : m(x) \neq 0\}$, where

$$m(x) = \begin{cases} |m_1(x) - m_2(x)| & x \in \quad Y_1 \bigcap Y_2 \\ m_1(x) & x \in Y_1 - Y_2 \\ m_2(x) & x \in Y_2 - Y_1 \end{cases}$$

We also define the size of a multiset (Y, m) as $|(Y, m)| := \sum_{y \in Y} m(y)$. For example, consider two multisets $\{1, 1, 1, 2, 3, 3, 4\}$ and $\{1, 1, 2, 2, 2, 3, 3, 5, 5\}$. The symmetric

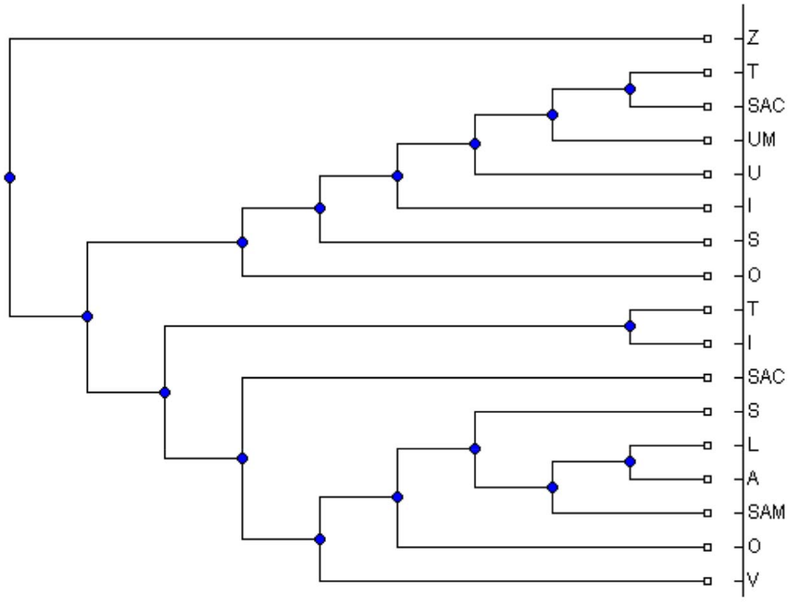


Figure 6. The obtained MUL tree by applying MTRT on the triplets extracted from the MUL tree shown in Figure 5. This MUL tree has 5 duplications.
doi:10.1371/journal.pone.0103622.g006

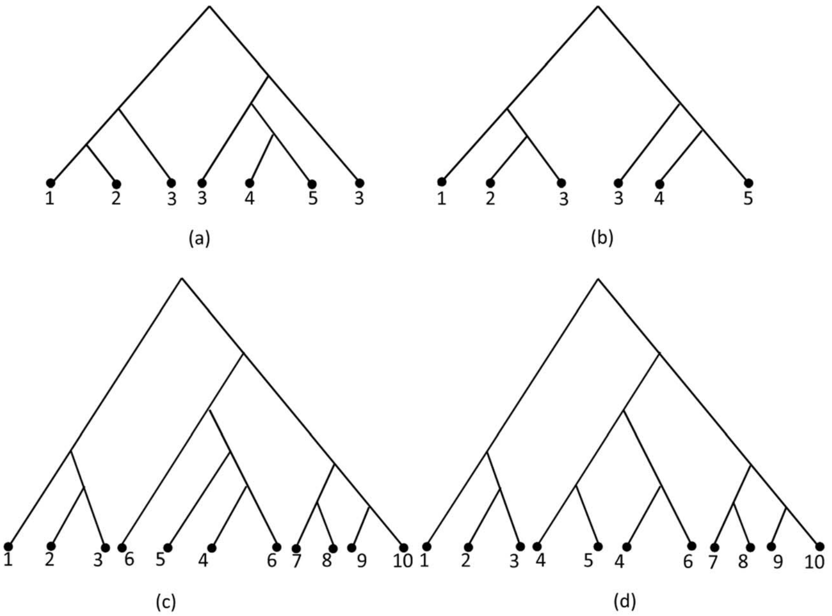


Figure 7. Comparing MUL trees using triplet distance. (a) The MUL tree M_1, (b) The MUL tree $M_1^{'}$ is consistent with $\Re(M_1)$. The MUL tree $M_1^{'}$ has less duplication than M_1 and is consistent with the triplet 23|1 which is not contained in $\Re(M_1)$. So, $TD(M_1, M_1^{'})=0.5$, (c) The MUL tree M_2, (d) The MUL tree $M_2^{'}$ is consistent with $\Re(M_2)$. The MUL trees M_1 and $M_2^{'}$ have the same number of duplications and $TD(M_2, M_2^{'})=0$.
doi:10.1371/journal.pone.0103622.g007

difference between these sets and its size are $\{1, 2, 2, 4, 5, 5\}$ and 6 respectively. For a MUL tree M, let $\Re'(M)=(\Re(M), m)$ be the triplet encoding multiset of M. It means that if a triplet is seen in the MUL tree k times, then $\Re'(M)$ contains this triplet k times. We define the new triplet distance between two MUL trees as follows:

Definition 2.

(a) The rooted triplet distance between two rooted phylogenetic MUL trees M_1 and M_2 on taxa set X is defined as

$$TD_M(M_1, M_2)=|\Re'(M_1)\Delta_m\Re'(M_2)|.$$

(b) The rooted triplet distance between a rooted phylogenetic MUL tree M' and a multiset of triplets $\Re$ on taxa set X is defined as

$$TD_M(M', \Re)=|\Re'(M')\Delta_m\Re|.$$

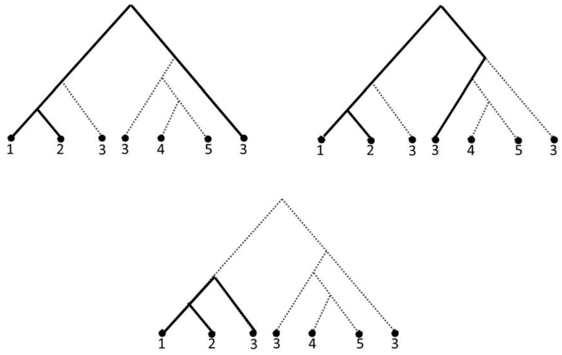


Figure 8. A MUL tree which has three different triplets 12|3.
doi:10.1371/journal.pone.0103622.g008

(c) The rooted triplet distance between two multisets of triplets $\Re_1$ and $\Re_2$ on taxa set X is defined as

$$TD_M(\Re_1, \Re_2)=|\Re_1\Delta_m\Re_2|.$$

Using the new rooted triplet distance $TD_M()$ defined in Def. 2, the distance between MUL trees M_2 and $M_2^{'}$ shown in Figure 7 equals $TD_M(M_2, M_2^{'})=56$. Note that a MUL tree is not uniquely defined by its multiset of triplets. For example, two MUL trees shown in Figure 9 have the same multiset of triplets. However, it seems that for most of the MUL trees specially for large MUL trees, it is true that two MUL trees are isomorphic if they have new triplet distance $TD_M()$ equal to 0. To show this, we computed the triplet distance $TD()$ and new triplet distance $TD_M()$ for all simulated and real datasets. The results of simulated datasets are shown in Table 1. Suppose M_{in} is a MUL tree and M_{out} is the result of applying MTRT algorithm on $\Re(M_{in})$. We define $RD(M_{in}, M_{out}):=d(M_{out})-d(M_{in})$. We classify the simulated datasets into 5 classes:

- $A:=\{datasets: TD(M_{in}, M_{out})=0\}$,
- $B:=\{datasets: RD(M_{in}, M_{out})<0\}$,

- $C := \{datasets : RD(M_{in}, M_{out}) = 0\}$,
- $D := \{datasets : RD(M_{in}, M_{out}) > 0\}$,
- $E := \{datasets : TD_M(M_{in}, M_{out}) = 0\}$.

Table 1 shows the intersection of above sets. For example, in 100 datasets, MTRT produces a MUL tree which has less duplication than that of the input MUL tree and the corresponding triplet distance is 0. In 74 datasets, the output and input MUL trees have the same number of duplications and the new distance between them is 0. We studied these 74 datasets and found that their corresponding output and input MUL trees are exactly the same. We also examined the exact algorithm on 100 datasets mentioned in Results section. The results show that in 56 datasets, the exact algorithm produces MUL trees which have less number of duplications than that of the original MUL tree. For the remaining datasets, the number of duplications for both MUL trees are the same. This shows that for more than fifty percent of the cases, the MUL tree produced by the exact algorithm is different from the input MUL tree. We also obtained the $TD()$ and $TD_M()$ for real datasets. For the first real data, $TD()$ is 98, that is, the output MUL tree has 196 triplets which are not contained in input triplet set. $TD_M()$ for this data is 2573. For second real data, $TD()$ is 76.5, that is, the output MUL tree has 153 triplets which are not contained in input triplet set. $TD_M()$ for this data is 6151. For third data, $TD()$ and $TD_M()$ are 2 and 255 respectively. These numbers and Table 1 show that in many cases the SMRT problem and its conditions do not satisfy the consistency principle. Hence in many cases, the algorithms based on SMRT fail to produce the exact MUL tree.

Discussion and Future Works

In this paper, we presented a heuristic algorithm MTRT for the SMRT problem. MTRT is implemented in MATLAB and is available at http://bs.ipm.ir/softwares/MTRT/. The goal of the algorithm is to construct a minimal MUL tree that is consistent with the input set of triplets and minimizes the number of its duplications. Note that a phylogenetic network can be associated to a MUL tree [14]. Therefore, it seems that constructing the smallest MUL tree from a set of triplets could be an alternative method for the problem of constructing a phylogenetic network with minimum reticulation from a set of triplets. To test the performance of the MTRT, we applied it on 400 simulated MUL trees and three real datasets. For each simulated and real MUL tree, we extracted all its triplets and applied the MTRT algorithm on the triplet set. We have shown that in most cases, the MTRT works well and has an acceptable running time. In only 10 percent of the datasets, the number of duplications for the output MUL tree of MTRT is greater than that of the original MUL tree. We also compared MTRT with exact algorithm. To do this, we executed the exact algorithm on 100 datasets. We showed that, in 86 datasets, the MUL trees produced by both MTRT and exact algorithm have the same duplications. We found that for more than 50 percent of the cases, the exact algorithm produces an output which is different from the input. It shows that the SMRT problem does not satisfy the consistency principle. So, having the set of triplets consistent to a MUL tree is not enough to infer that MUL tree. Furthermore, considering the minimum number of duplications to reconstruct a MUL tree that is consistent with a given set of triplets is not appropriate to infer the correct MUL tree. Therefore, from a biological point of view, there is a deficiency in the SMRT problem. Equivalently, the problem of constructing a phylogenetic network with minimum reticulation from a set of triplets is not consistent with the consistency principle of phylogeny reconstruction methods. It is necessary to consider other conditions to obtain proper MUL trees or phylogenetic networks. We extended the definition of triplet distance $TD()$ and introduced a new triplet distance $TD_M()$. For all datasets, we compared the output MUL tree with original MUL tree by $TD_M()$. For all datasets with $TD_M() = 0$, we showed that the output and original MUL trees are the same. According to these observations, we propose the following problem, called MUL tree from a multiset of rooted triplets with minimum triplet distance, or mMTd for short:

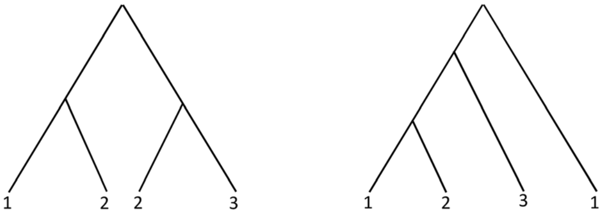


Figure 9. Two different MUL trees with tha same multiset of triplets $\{12|3, 23|1\}$.
doi:10.1371/journal.pone.0103622.g009

mMTd problem. Given a multiset $\Re$ of rooted triplets over a leaf label set L, output a MUL tree M which minimizes $TD_M(M, \Re)$.

Note that the maximum rooted triplets consistency problem, or MRTC for short [4], is a special case of mMTd problem. A natural question is how a multiset can be generated from biological data? For example, in the study of area cladograms, suppose a set of triplets is produced and we are interested to replace organisms by area names. Or in the other field, suppose we want to replace parasites by their host. Thus, a multiset of triplets may be derived from a great variety of biological processes.

We can simply extend the definition of the new triplet distance to a phylogenetic network. Hence, the other problem can be defined as follows, called Network from a multiset of rooted triplets with minimum triplet distance, or nMTd for short:

nMTd problem. Given a multiset $\Re$ of rooted triplets over a leaf label set L, output a network N which minimizes $TD_M(N,\Re)$.

Materials and Methods

This section describes a heuristic method MTRT that aims to solve the SMRT problem. We first define the concept of a separating set in a graph. Consider a graph $G = (V, E)$. The subgraph $G[U]$ induced by $U \subset V$ has a vertex set U and an induced edge set $E|_U$ that consists of all edges in G whose both endpoints lie in U. Suppose G is a connected graph. The set $S \subset V$ is called a separator, or a separating set, of G if $G[V \backslash S]$ is disconnected. Now, let $\Re$ denotes a given set of triplets over a leaf label set L. MTRT tries to build a MUL tree M which is consistent with $\Re$ and its leaf duplications $d(M)$ is as small as possible. MTRT is based on Aho et al.'s algorithm [1]. The Auxiliary graph, denoted by $AG(\Re)$, is required, which is a graph corresponding to $\Re$ with vertex set L and edge set E such that:

Table 1. The results of MTRT algorithm on simulated datasets.

$\lvert A\rvert=95$	$\lvert A\cap B\rvert=50$	$\lvert A\cap C\rvert=45$	$\lvert A\cap D\rvert=0$	$\lvert A\cap E\rvert=37$
$\lvert B\rvert=168$	$\lvert B\cap A\rvert=100$	$\lvert B\cap C\rvert=0$	$\lvert B\cap D\rvert=0$	$\lvert B\cap E\rvert=0$
$\lvert C\rvert=192$	$\lvert C\cap A\rvert=90$	$\lvert C\cap B\rvert=0$	$\lvert C\cap D\rvert=0$	$\lvert C\cap E\rvert=74$
$\lvert D\rvert=40$	$\lvert D\cap A\rvert=0$	$\lvert D\cap B\rvert=0$	$\lvert D\cap C\rvert=0$	$\lvert D\cap E\rvert=0$
$\lvert E\rvert=74$	$\lvert E\cap A\rvert=74$	$\lvert E\cap B\rvert=0$	$\lvert E\cap C\rvert=74$	$\lvert E\cap D\rvert=0$

In the table, $A:=\{datasets: TD(M_{in}, M_{out})=0\}$, $B:=\{datasets: RD(M_{in}, M_{out})<0\}$, $C:=\{datasets: RD(M_{in}, M_{out})=0\}$, $D:=\{datasets: RD(M_{in}, M_{out})>0\}$ and $E:=\{datasets: TD_M(M_{in}, M_{out})=0\}$ where M_{in} is a MUL tree and M_{out} is the result of applying MTRT algorithm on $\Re(M_{in})$.

doi:10.1371/journal.pone.0103622.t001

$$\forall a, b \in L: \quad e=\{a, b\} \in E \Leftrightarrow \exists c \in L \; s.t \; ab|c \in \Re.$$

In general, the algorithm MTRT does the following steps. $AG(\Re)$ is computed first. If $AG(\Re)$ is disconnected, then the set L is partitioned into two non-empty sets A and B such that the set of vertices in each connected component of $AG(\Re)$ is a subset of either A or B. Now, the triplet sets $\Re_A$ and $\Re_B$ are computed. We set $\Re_A := \Re|_A$ and $\Re_B := \Re|_B$. If $AG(\Re)$ is connected, then MTRT tries to find the minimum separating set S and classifies the connected components of $AG[L\backslash S]$ into two non-empty sets A' and B'. It is well known that finding the all minimum-size separators is an NP-hard problem [3]. To find a minimum separator, we use AllMinSep algorithm [2]. AllMinSep computes the set of all minimal separators of a graph G in time $O(n^3|\theta|)$ where $|\theta|$ is the number of all minimal separators. AllMinSep first produces an initial set of minimal separators θ. Then for each $\varphi \in \theta$, a family of other minimal separators is generated and added to θ. This procedure is done until all minimal separators are obtained, see [2] for more details. Since the number of all minimal separators can be exponential and we do not need all the minimal separators, so we use the AllMinSep with a small change to make it a greedy algorithm. Suppose the initial set of minimal separators θ has been obtained and m is the size of the smallest separator in θ. Then for each $\varphi \in \theta$, a family of other minimal separators θ' is generated. Now, the separator $\varphi' \in \theta'$ is added to θ if $|\varphi'| \leq m$.

Let S be a separator computed by AllMinSep and the connected components of $AG[L\backslash S]$ are classified in two non-empty sets A' and B'. We set $A = A' \bigcup S$ and $B = B' \bigcup S$. The triplet sets corresponding to A and B are considered as follows:

$$\Re_A := \Re|_A - \Re(A, S) \quad and \quad \Re_B := \Re|_B - \Re(B, S).$$

Now, the algorithm recursively handles sets A and B with triplet sets $\Re_A$ and $\Re_B$ respectively. Let the MUL trees constructed by MTRT for the sets A and B are M_A and M_B respectively. We report the MUL tree $MT_{\{A,B\}}$ formed by connect M_A and M_B with the same root. For the case that $AG(\Re)$ is connected, we define $\Re_A$ and $\Re_B$ in such a way because the members of S are repeated on both sides of the root. So, the set $\{ab|c \in \Re: \; either \; a, b \in S \; or \; c \in S\}$ is consistent with the $MT_{\{A,B\}}$ and it is unnecessary to consider this set. It is obvious that the output MUL tree of the algorithm is consistent with $\Re$. We now illustrate the steps of the algorithm MTRT by an example.

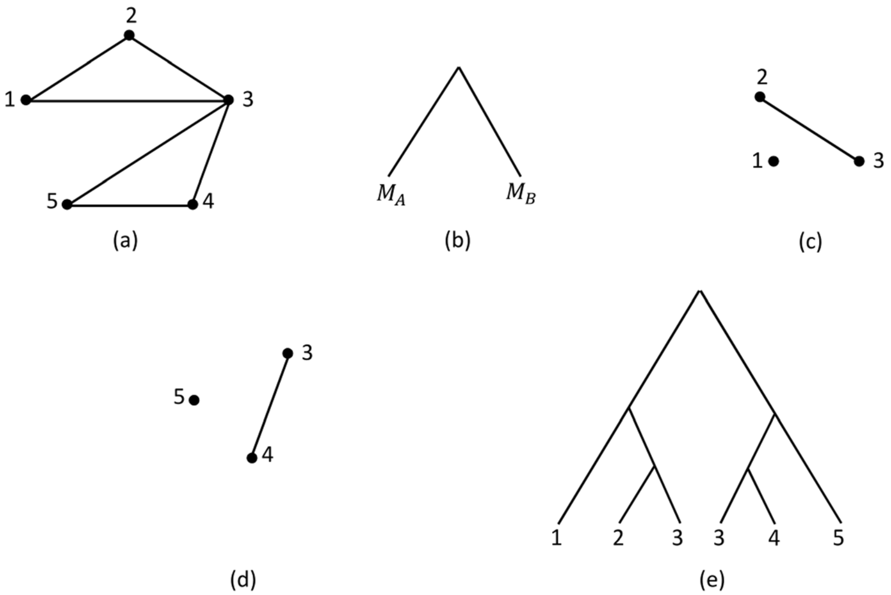
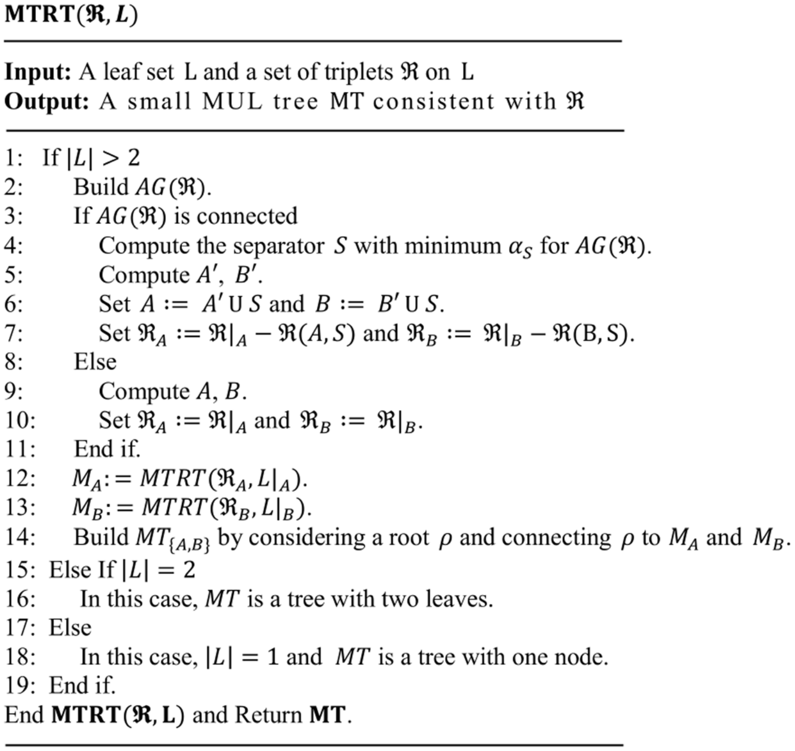


Figure 10. Steps of MTRT. (a) The auxiliary graph corresponding to $\Re = \{12|3, 13|4, 23|1, 34|1, 35|2, 45|1, 45|3\}$, (b) $MT_{\{A,B\}}$, (c) The auxiliary graph $AG(\Re_A)$, (d) The auxiliary graph $AG(\Re_B)$, (e) A smallest MUL tree produced by MTRT algorithm.

doi:10.1371/journal.pone.0103622.g010



```
MTRT($\Re, L$)

Input: A leaf set L and a set of triplets $\Re$ on L
Output: A small MUL tree MT consistent with $\Re$

If $|L| > 2$
    Build $AG(\Re)$.
    If $AG(\Re)$ is connected
        Compute the separator $S$ with minimum $\alpha_S$ for $AG(\Re)$.
        Compute $A'$, $B'$.
        Set $A := A' \cup S$ and $B := B' \cup S$.
        Set $\Re_A := \Re|_A - \Re(A,S)$ and $\Re_B := \Re|_B - \Re(B,S)$.
    Else
        Compute $A$, $B$.
        Set $\Re_A := \Re|_A$ and $\Re_B := \Re|_B$.
    End if.
    $M_A := MTRT(\Re_A, L|_A)$.
    $M_B := MTRT(\Re_B, L|_B)$.
    Build $MT_{\{A,B\}}$ by considering a root $\rho$ and connecting $\rho$ to $M_A$ and $M_B$.
Else If $|L| = 2$
    In this case, $MT$ is a tree with two leaves.
Else
    In this case, $|L| = 1$ and $MT$ is a tree with one node.
End if.
End MTRT($\Re$, L) and Return MT.
```

Figure 11. Pseudocode of the MTRT algorithm.
doi:10.1371/journal.pone.0103622.g011

Let $L = \{1, 2, 3, 4, 5\}$ and $\Re = \{12|3, 13|4, 23|1, 34|1, 35|2, 34|5, 45|1, 45|2\}$ be the set of triplets over L. The auxiliary graph corresponding to $\Re$ is shown in Figure 10a. The set $S = \{3\}$ is the minimum separator of $AG(\Re)$. Hence, $A = \{1, 2, 3\}$ and $B = \{3, 4, 5\}$. $MT_{\{A, B\}}$ is shown in Figure 10b. The induced triplet sets for A and B are $\Re|_A = \{12|3, 23|1\}$ and $\Re|_B = \{34|5\}$ respectively. Now, $R(A, S) = \{12|3\}$ is removed from $\Re|_A$ to obtain $\Re_A$. So, $\Re_A = \{23|1\}$ and $\Re_B = \{34|5\}$. The auxiliary graphs $AG(\Re_A)$ and $AG(\Re_B)$ are shown in Figure 10c and Figure 10d respectively. Finally, the MUL tree produced by MTRT algorithm is shown in Figure 10e.

We now describe two cases that may occur in some steps of the algorithm:

Case 1. It is possible at some steps of the algorithm, for a leaf label set U, $\Re_U = \emptyset$. In this case, the triplets of an arbitrary tree on U is considered as $\Re_U$. For instance, let $\Re = \{12|3, 13|5, 23|4, 34|1, 35|2, 34|5, 45|1, 45|2\}$. The separator of $AG(\Re)$ is $S = \{3\}$. So, $A = \{1, 2, 3\}$, $B = \{3, 4, 5\}$, $\Re|_A = \{12|3\}$ and $\Re|_B = \{34|5\}$ and consequently, $\Re_A = \emptyset$ and $\Re_B = \{34|5\}$. Now, an arbitrary triplet set consistent with a tree on leaf label set A is considered as $\Re_A$, for example $\Re_A := \{23|1\}$. If the algorithm runs to the end, the MUL tree shown in Figure 10e is produced.

Case 2. There are more than one minimum separating set. In this case, MTRT chooses a separator S with minimum α_S, where

$$\alpha_S = 2(|\Re_A| + |\Re_B|) + ||\Re_A| - |\Re_B||.$$

If $\Re$ has more triplets, then the probability of having more duplications is high. The first part of α_S help to reduce the number of duplications and the second part of α_S help to produce a MUL tree which is relatively balanced. Since minimizing the number of triplets is more important, we give bigger weight (2, by default) for the first part. The pseudocode of the MTRT algorithm is detailed in Figure 11.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. They are also grateful to Thomas Marcussen for providing real datasets.

Author Contributions

Conceived and designed the experiments: RH CE. Performed the experiments: RH. Analyzed the data: RH. Wrote the paper: RH CE WKS.

References

1. Aho AV, Sagiv Y, Szymanski TG, Ullman JD (1981) Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. SIAM Journal on Computing 10(3): 405–421.
2. Berry A, Bordat JP, Cogis O (2000) Generating all the minimal separators of a graph. International Journal of Foundations of Computer Science 11(3): 397–403.
3. Bui TN, Jones C (1992) Finding good approximate vertex and edge partitions is NP-hard. Inf. Pro cess. Lett. 42: 153–159.
4. Byrka J, Guillemot S, Jansson J (2010) New results on optimizing rooted triplets consistency. Discrete Applied Mathematics 158(11): 1136–1147.
5. Critchlow DE, Pearl DK, Qian C (1996) The triples distance for rooted bifurcating phylogenetic trees. Systematic Biology 45(3): 323–334.
6. Cui Y, Jansson J, Sung WK (2012) Polynomial-Time Algorithms for Building a Consensus MUL-Tree. Journal of Computational Biology 19(9): 1073–1088.
7. Edgar RC (2006) MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic acids research 32(5): 1792–1797.

8. Ganapathy G, Goodson B, Jansen R, Le HS, Ramachandran V, et al. (2006) Pattern identification in biogeography. IEEE/ACM Transactions on Computational Biology and Bioinformatics 3(4): 334–346.
9. Guillemot S, Jansson J, Sung WK (2011) Computing a smallest multilabeled phylogenetic tree from rooted triplets. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8(4): 1141–1147.
10. Huber KT, Oxelman B, Lott M, Moulton V (2006) Reconstructing the evolutionary history of polyploids from multilabeled trees. Molecular Biology and Evolution 23(9): 1784–1791.
11. Huber KT, Spillner A, Suchecki R, Moulton V (2011) Metrics on Multilabeled Trees: Interrelationships and Diameter Bounds. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8(4): 1029–1040.
12. Jansson J, Nguyen NB, Sung WK (2006) Algorithms for Combining Rooted Triplets into a Galled Phylogenetic Network. Bioinformatics 18: 337–338.
13. Jansson J, Sung WK (2006) Inferring a Level-1 Phylogenetic Network from a Dense Set of Rooted Triplets. Theoretical Computer Science 363: 60–68.
14. Lott M, Spillner A, Huber KT, Moulton V (2009) PADRE: a package for analyzing and displaying reticulate evolution. Bioinformatics 25(9): 1199–1200.
15. Lott M, Spillner A, Huber KT, Petri A, Oxelman B, et al. (2009) Inferring polyploid phylogenies from multiply-labeled gene trees. BMC Evolutionary Biology 9: 216.
16. Maddison WP, Maddison DR (2011) Mesquite: a modular system for analysis. http://mesquiteproject.org, Version 2.75.
17. Marcussen T, Jakobsen KS, Danihelka J, Ballard HE, Blaxland K, et al. (2012) Inferring species networks from gene trees in high-polyploid North American and Hawaiian violets (Viola, Violaceae). Systematic biology 61(1): 107–126.
18. Nelson G, Platnick N (1981) Systematics and Biogeography: Cladistics and Vicariance. Columbia University Press.
19. Page RDM (1994) Maps between trees and cladistic analysis of historical associations among genes, organisms, and areas. Systematic Biology 43(1): 58–77.
20. Page RDM (1993) Parasites, phylogeny and cospeciation. International Journal for Parasitology 23: 499–506.
21. Popp M, Erixon P, Eggens F, Oxelman B (2005) Origin and evolution of a circumpolar polyploid species complex in Silene (Caryophyllaceae) inferred from low copy nuclear RNA polymerase introns, rDNA, and chloroplast DNA. Systematic botany 30(2): 302–313.
22. Scornavacca C, Berry V, Ranwez V (2011) Building species trees from larger parts of phylogenomic databases. Information and Computation 209(3): 590–605.
23. Scornavacca C, Berry V, Ranwez V (2009) From gene trees to species trees through a supertree approach. Language and Automata Theory and Applications: 702–714.
24. To TH, Habib M (2009) Level-k phylogenetic networks are constructable from a dense triplet set in polynomial time. In Combinatorial Pattern Matching: Proceeding of the 20th Annual Symposium Combinatorial Pattern Matching (CPM), 5577(LNCS): 275–278.
25. Van Iersel L, Keijsper J, Kelk S, Stougie L, Hagen F, et al. (2009) Constructing level-2 phylogenetic networks from triplets. IEEE/ACM Transactions on Computational Biology and Bioinformatics 6(4): 667–681.
26. Van Iersel L, Kelk S (2011) Constructing the simplest possible phylogenetic network from triplets. Algorithmica 60: 207–235.