Hindawi Publishing Corporation
BioMed Research International
Volume 2016, Article ID 4236858, 7 pages
http://dx.doi.org/10.1155/2016/4236858

Research Article

Constructing Phylogenetic Networks Based on the Isomorphism of Datasets

Juan Wang,[1] Zhibin Zhang,[1] and Yanjuan Li[2]

[1]*School of Computer Science, Inner Mongolia University, Hohhot 010021, China*
[2]*Department of Information and Computer Engineering, Northeast Forestry University, Harbin 150040, China*

Correspondence should be addressed to Juan Wang; wangjuanangle@hit.edu.cn

Received 31 May 2016; Accepted 30 June 2016

Academic Editor: Yungang Xu

Constructing rooted phylogenetic networks from rooted phylogenetic trees has become an important problem in molecular evolution. So far, many methods have been presented in this area, in which most efficient methods are based on the incompatible graph, such as the CASS, the LNETWORK, and the BIMLR. This paper will research the commonness of the methods based on the incompatible graph, the relationship between incompatible graph and the phylogenetic network, and the topologies of incompatible graphs. We can find out all the simplest datasets for a topology G and construct a network for every dataset. For any one dataset $\mathscr{C}$, we can compute a network from the network representing the simplest dataset which is isomorphic to $\mathscr{C}$. This process will save more time for the algorithms when constructing networks.

1. Introduction

The evolutionary history of species is usually represented as a (rooted) phylogenetic tree, in which one species has only one parent. Actually, the evolution of species has caused reticulate events such as hybridizations, horizontal gene transfers, and recombinations [1–5], so species may have more than one parent. Then, the phylogenetic trees cannot describe well the evolutionary history of species. However, phylogenetic networks can represent the reticulate events, and they are a generalization of phylogenetic trees. Phylogenetic networks can also represent the conflicting evolution information that may be from different datasets or different trees [6–9].

Phylogenetic networks can be classified into unrooted [10–12] and rooted networks [4, 13–19]. An unrooted phylogenetic network is an unrooted graph whose leaves are bijectively labelled by the taxa. A rooted phylogenetic network is a rooted directed acyclic graph (DAG for short) whose leaves are bijectively labelled by taxa [20–22]. The rooted phylogenetic networks have been studied widely for representing the evolution of taxa, as evolution of species is inherently directed. The paper will study relevant properties of the rooted phylogenetic networks constructed from the rooted trees.

The algorithms constructing rooted phylogenetic networks from rooted phylogenetic trees are mainly classified into three types: the cluster network [17] based on the Hasse diagram; the galled network [16] based on the seed-growing algorithm; the Cass [23], the Lnetwork [24], and the BIMLR [25] based on the decomposition property of networks. In particular, the third type of methods (Cass, Lnetwork, and BIMLR) can construct more precise networks than the other methods. In the following, unless otherwise specified, we refer to rooted phylogenetic networks as networks.

Let $\mathscr{X}$ be a set of taxa. A proper subset of $\mathscr{X}$ (except for both $\emptyset$ and $\mathscr{X}$) is called a cluster. A cluster C is trivial if $|C| = 1$; otherwise, it is nontrivial. Let T be a rooted phylogenetic tree on $\mathscr{X}$; if there is an edge $e = (u, v)$ in T such that the set of taxa which are descendants of v equals C, we say that T represents C. Figure 1 shows two rooted phylogenetic trees T_1 and T_2 and all nontrivial clusters represented by T_1 and T_2. Here, all trivial clusters are not listed. Given a network N and a cluster C, when just connecting one incoming edge and disconnecting all other incoming edges for each reticulate node (i.e., its incoming edges >1), if there is a tree edge $e = (u, v)$ (i.e., incoming edge of $v \leq 1$) in N such that the set of taxa which are descendants of v equals C, we say that N represents C in the softwired sense. On the other hand, if

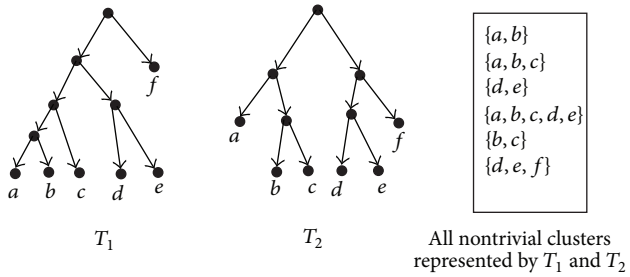

Figure 1: Two rooted phylogenetic trees T_1 and T_2.

there is a tree edge $e = (u, v)$ in N such that the set of taxa which are descendants of v equals C, we say that N represents C in the hardwired sense.

The abovementioned three types of methods constructing networks are based on clusters; that is, they first compute all of the clusters represented by input trees and then construct a network representing all clusters in the softwired sense. In this process, the third type of methods (Cass, Lnetwork, and BIMLR) will recur to the incompatibility graph (will be discussed in the following). This paper will discuss the relationship between the incompatibility graphs and the constructed networks.

2. Preliminaries

A rooted phylogenetic network $N = (V, E)$ on $\mathcal{X}$ is a rooted DAG, and its leaves are bijectively labelled as $\mathcal{X}$. The indegree of a node $v \in V$ is denoted by $\text{indeg}(v)$. A node v with $\text{indeg}(v) \geq 2$ is called a reticulate node, a node v with $\text{indeg}(v) \leq 1$ is called a tree node, and, specially, the tree node with indegree 0 is the root node. The reticulation number in a network $N = (V, E)$ is $\sum_{\text{indeg}(v)>0}(\text{indeg}(v) - 1) = |E| - |V| + 1$.

Given a set of taxa $\mathcal{X}$, two clusters C_1 and C_2 on $\mathcal{X}$ are called compatible, if they are disjoint or one contains the other; that is, $C_1 \cap C_2 = \emptyset$ or $C_1 \subseteq C_2$ or $C_2 \subseteq C_1$; otherwise, they are incompatible. Obviously, a trivial cluster and any one cluster are compatible. Given two incompatible clusters C_1 and C_2, $C_1 \cap C_2$ is called the incompatible taxa with respect to C_1 and C_2. A set of clusters $\mathscr{C}$ on $\mathcal{X}$ is called compatible, if $\mathscr{C}$ is pairwise compatible; otherwise, it is incompatible. For a set of clusters $\mathscr{C}$, its incompatibility graph $\text{IG}(\mathscr{C}) = (V, E)$ is an undirected graph with node set $V = C$ and edge set E, where an edge connects two incompatible clusters.

Given a cluster set $\mathscr{C}$ on $\mathcal{X}$ and a subset S of $\mathcal{X}$, the result of removing all elements in $\mathcal{X} \setminus S$ from each cluster in $\mathscr{C}$ is called the restriction of $\mathscr{C}$ to S, denoted by $\mathscr{C}|_S$. If S (where $|S| > 1$) and any one cluster $C \in \mathscr{C}$ are compatible and $\mathscr{C}|_S$ is also compatible, then we say that S is an ST-set (Strict Tree Set) with respect to $\mathscr{C}$. If there are no other ST-sets containing S except itself, we say that S is maximal. For a maximal ST-set S, there is a subtree constructed by the set of clusters $\{C \mid C \in \mathscr{C}, C \subset S\} \cup S$.

For each maximal ST-set S with respect to $\mathscr{C}$, after collapsing it into a single taxon S, the result set is denoted as $\text{Collapse}(\mathscr{C})$. For example, $\mathscr{C} = \{\{1, 2\}, \{1, 2, 3\}, \{3, 4\}\}, \{1, 2\}$ is the only maximal ST-set; then, $\text{Collapse}(\mathscr{C}) = \{\{3, 4\}, \{\{1, 2\}, 3\}\}$. Then, the taxa of $\text{Collapse}(\mathscr{C})$ are $\{\{1, 2\}, 3, 4\}$, denoted as $\mathcal{X}(\text{Collapse}(\mathscr{C}))$. A set of clusters $\mathscr{C}$ is called the simplest if it has no maximal ST-set with respect to $\mathscr{C}$.

Let $\mathscr{C}$ be a set of clusters on $\mathcal{X}$ and let N be a network representing $\mathscr{C}$. Usually, a tree edge in N can represent more than one cluster in $\mathscr{C}$ and a cluster in $\mathscr{C}$ can be represented by more than one tree edge in N. A mapping ϵ is defined from $\mathscr{C}$ to the set of tree edges of N, such that $\epsilon(C)$ is a tree edge of N that represents C for any one cluster $C \in \mathscr{C}$. A network N is decomposable with respect to $\mathscr{C}$ if there exists a mapping $\epsilon : \mathscr{C} \to E'$ (E' is the set of tree edges of N) such that

(i) for any two clusters $C_1, C_2 \in \mathscr{C}$, C_1 and C_2 lie in the same connected component of the incompatibility graph $\text{IG}(\mathscr{C})$ if and only if two tree edges $\epsilon(C_1)$ and $\epsilon(C_2)$ are contained in the same biconnected component of N.

Then, we also say that the network N has the decomposition property. The decomposition property makes the network constructed by an appropriate divide-and-conquer (DC for short) strategy; that is, it first constructs a subnetwork for each one connected component of the incompatibility graph and then merges all subnetworks into a whole network. Then, the constructed network is called DC network, and the algorithms are called DC algorithms. The paper [23] has proven the DC networks satisfying the decomposition property.

Given a set of clusters $\mathscr{C}$, the DC algorithms first compute the incompatibility graph $\text{IG}(\mathscr{C})$ and then compute the subnetwork for the result set after collapsing each one maximal ST-set into one taxon for each biconnected component of $\text{IG}(\mathscr{C})$; next, "decollapse," that is, replace each leaf labelled by a maximal ST-set by a maximal subtree, and finally integrate those subnetworks into a final network. The paper [25] has proven that there exists a DC network N for any one set of clusters $\mathscr{C}$. Figure 2 shows the construction process of the DC algorithms for the set of clusters in Figure 1, in which constructing subnetwork for each one connected component (i.e., Step 2) is crucial.

The Cass, the Lnetwork, and the BIMLR algorithms are the DC algorithms, which can construct the networks with fewer reticulations than other algorithms. The networks constructed by the BIMLR and the Lnetwork have fewer redundant clusters except for the input clusters than other available methods. When constructing phylogenetic networks, the BIMLR and the Lnetwork are faster than the Cass, and the constructed networks are more stable, that is, the difference between constructed networks for the same dataset when different input orders are used is smaller than the Cass. Figure 3 shows three networks constructed by the Cass for the same dataset with different input orders, while BIMLR and Lnetwork can construct only one network N_1 for the dataset with different input orders [25].

3. Topologies of Incompatibility Graphs

Definition 1. Two networks $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$ on $\mathcal{X}$ are isomorphic if and only if there exists a bijection H from V_1 to V_2 such that

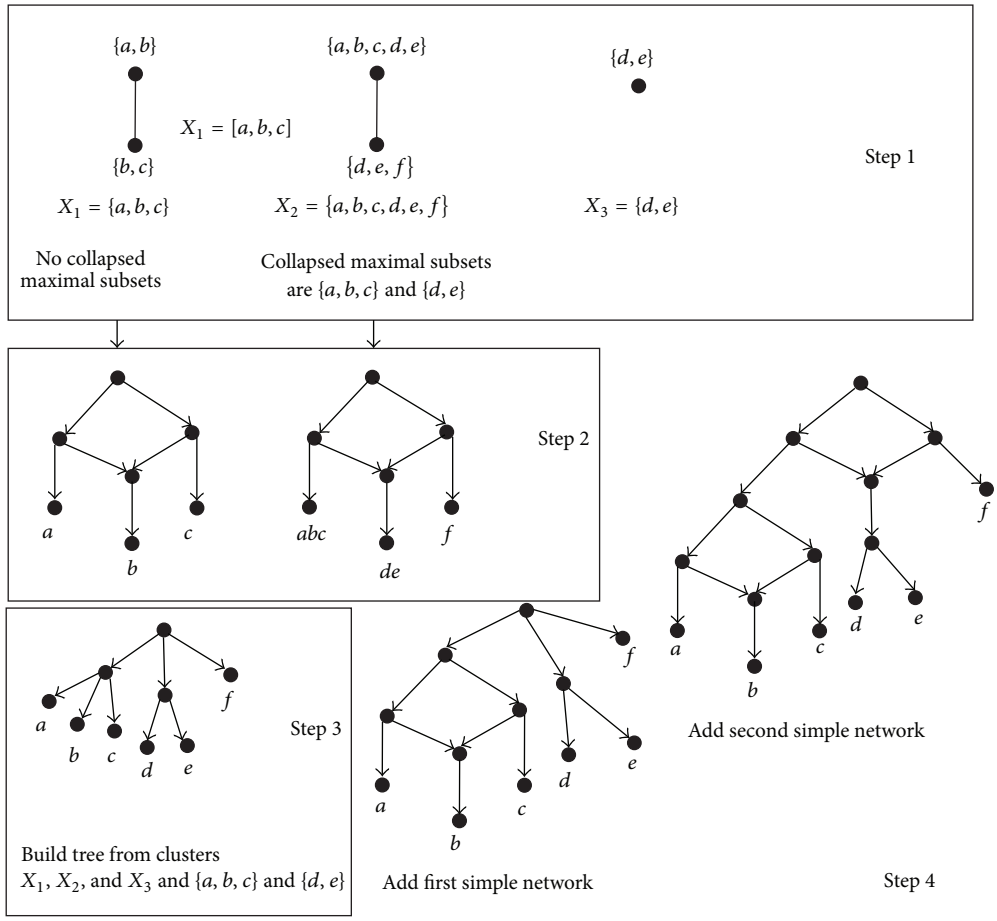

Figure 2: A network constructed by the DC algorithms for the set of clusters in Figure 1.

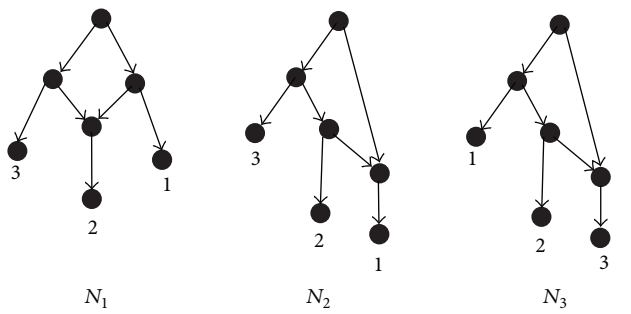

Figure 3: All networks constructed by the Cass for the set of clusters $\mathscr{C} = \{\{1, 2\}, \{2, 3\}\}$.

(i) (u, v) is an edge in E_1 if and only if $(H(u), H(v))$ is an edge in E_2;

(ii) the label of w is equal to the label of $H(w)$ for any one leaf $w \in V_1$.

Given two sets of clusters $\mathscr{C}_1$ on $\mathscr{X}_1$ and $\mathscr{C}_2$ on $\mathscr{X}_2$, let $\mathscr{C}_1'$ and $\mathscr{C}_2'$ be the results after collapsing all maximal ST-sets of $\mathscr{C}_1$ and $\mathscr{C}_2$, respectively, $\mathscr{C}_1'$ on $\mathscr{X}_1'$ and $\mathscr{C}_2'$ on $\mathscr{X}_2'$.

Definition 2. $\mathscr{C}_1$ and $\mathscr{C}_2$ are isomorphic, if and only if there is a bijection G from $\mathscr{X}_1'$ to $\mathscr{X}_2'$ such that

(i) a and b are in the same cluster $C_1 \in \mathscr{C}_1'$ if and only if $G(a)$ and $G(b)$ are in the same cluster $C_2 \in \mathscr{C}_2'$.

By Definition 2, we have that the isomorphism of the cluster sets is an equivalence relation; that is, it is reflexive, symmetric, and transitive.

Lemma 3. *Given a DC network N representing the set of clusters $\mathscr{C}$, then any one maximal ST-set with respect to $\mathscr{C}$ is a maximal subtree in N.*

Proof. From the constructing process of DC networks, this conclusion is obvious. □

Lemma 4. *Let $\mathscr{C}_1$ and $\mathscr{C}_2$ be two sets of clusters on $\mathscr{X}_1$ and $\mathscr{X}_2$, respectively. $\mathscr{C}_1$ and $\mathscr{C}_2$ are isomorphic. There exists a DC network N_1 representing $\mathscr{C}_1$ if and only if there exists a DC network N_2 representing $\mathscr{C}_2$.*

Proof. There must exist a DC network N_1 for $\mathscr{C}_1$. Given a tree edge $e = (u, v)$, the subtree of the root v in N_1 is a maximal subtree if and only if the set of taxa S is a maximal ST-set with

respect to $\mathscr{C}_1$, where the taxa in S are labels of leaves which are descendants of v. Replace each maximal subtree of N_1 by a node, and then denote the result network as N_1'. Obviously, N_1' represents the set of clusters $\mathscr{C}_1'$. From Definition 2, there exists a bijection G from $\mathscr{X}_1'$ to $\mathscr{X}_2'$ such that a and b are in the same cluster $C_1 \in \mathscr{C}_1'$ if and only if $G(a)$ and $G(b)$ are in the same cluster $C_2 \in \mathscr{C}_2'$.

Then, we can obtain a network N_2' from N_1' by replacing each one taxon a in $\mathscr{X}_1'$ by $G(a)$ in $\mathscr{X}_2'$. Obviously, N_2' represents $\mathscr{C}_2'$. Finally, we replace each leaf labelled by a maximal ST-set with respect to $\mathscr{C}_2$ in N_2' by a maximal subtree, and the result network is denoted as N_2 which represents $\mathscr{C}_2$. □

For two isomorphic sets of clusters $\mathscr{C}_1$ and $\mathscr{C}_2$, let N_1 be a DC network representing $\mathscr{C}_1$. Lemma 4 tells us that there is a DC network N_2 representing $\mathscr{C}_2$, which can be obtained from N_1.

Lemma 5. *Let $\mathfrak{C} = \{\mathscr{C} \mid \mathscr{C}$ is a set of clusters$\}$, where $IG(\mathscr{C})$ is a biconnected component with two nodes. Then, any one element $\mathscr{C}$ in $\mathfrak{C}$ is isomorphic to $\mathscr{C}_0 = \{\{1,2\},\{2,3\}\}$.*

Proof. Any one element $\mathscr{C} \in \mathfrak{C}$ has two incompatible clusters. Let $\mathscr{C}_1 = \{C_{11}, C_{12}\}$ and $\mathscr{C}_2 = \{C_{21}, C_{22}\}$ be two sets of clusters in $\mathfrak{C}$, where C_{11} and C_{12} are incompatible and C_{21} and C_{22} are incompatible. Let $A_1 = C_{11} \cap C_{12}$ be the incompatible taxa with respect to C_{11} and C_{12}, and let $A_2 = C_{21} \cap C_{22}$ be the incompatible taxa with respect to C_{21} and C_{22}. Let $B_{11} = C_{11} \setminus A_1$, $B_{12} = C_{12} \setminus A_1$, $B_{21} = C_{21} \setminus A_2$, and $B_{22} = C_{22} \setminus A_2$; then, $\mathscr{C}_1 = \{\{B_{11}, A_1\}, \{B_{12}, A_1\}\}$ and $\mathscr{C}_2 = \{\{B_{21}, A_2\}, \{B_{22}, A_2\}\}$.

Each one of B_{11}, A_1, B_{12}, B_{21}, A_2, and B_{22} is a maximal ST-set if it contains more than one taxon; then, we can collapse it into one taxon which is also denoted by itself. Denote the set of clusters after collapsing all maximal ST-sets as $\mathscr{C}_1'$ and $\mathscr{C}_2'$. Obviously, there is a bijection G from $\mathscr{X}_1' = \{B_{11}, A_1, B_{12}\}$ to $\mathscr{X}_2' = \{B_{21}, A_2, B_{22}\}$, and any two taxa $a, b \in \mathscr{X}_1'$ are in the same cluster in $\mathscr{C}_1'$ if and only if $G(a)$ and $G(b)$ are in the same cluster in $\mathscr{C}_2'$. Hence, $\mathscr{C}_1$ and $\mathscr{C}_2$ are isomorphic. Accordingly, any one set of clusters $\mathscr{C} \in \mathfrak{C}$ is isomorphic to $\mathscr{C}_0 = \{\{1,2\},\{2,3\}\}$ because $\mathscr{C}_0 \in \mathfrak{C}$. □

For a cluster set $\mathscr{C}$, there may be several cluster sets isomorphic to $\mathscr{C}$, but the simplest set of clusters isomorphic to $\mathscr{C}$ is only one, denoted as $\mathscr{C}_0$. Let N_0 be the DC network representing $\mathscr{C}_0$. Then, we can obtain a DC network representing $\mathscr{C}$ from N_0. Lemmas 4 and 5 show there is a DC network for any one set of clusters whose incompatible graph is a biconnected component with two nodes, and it is obtained from the network N_0 (see Figure 3) representing $\mathscr{C}_0$.

Lemma 6. *Let $\mathfrak{C} = \{\mathscr{C} \mid \mathscr{C}$ is a set of clusters$\}$, where $IG(\mathscr{C})$ is a linear biconnected component with three nodes (see Figure 4). Let $\mathscr{C}_1 = \{\{1,3\},\{1,2\},\{1,3,4\}\}$, $\mathscr{C}_2 = \{\{1,3\}, \{1,2,4\},\{1,2,3\}\}$, $\mathscr{C}_3 = \{\{1,2\},\{2,3\},\{3,4\}\}$, and $\mathscr{C}_4 = \{\{1,2\}, \{2,3,5\},\{3,4\}\}$. Then, any one set of clusters $\mathscr{C}$ ($\mathscr{C} \in \mathfrak{C}$) is isomorphic to one of $\mathscr{C}_1$, $\mathscr{C}_2$, $\mathscr{C}_3$, and $\mathscr{C}_4$.*

Proof. Figure 4 shows the topology of the linear biconnected component with three nodes. $\mathscr{C}_i$ is the simplest set of clusters,

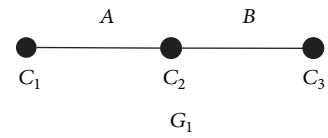

Figure 4: The topology of the linear biconnected component with three nodes.

and its incompatible graph is the topology in Figure 4. Next, we will prove that $\mathscr{C}_i$ ($1 \le i \le 4$) are all simplest sets of clusters for the topology in Figure 4.

Any one set of clusters in $\mathfrak{C}$ has three clusters denoted as C_1, C_2, and C_3. Let A be the incompatible taxa with respect to C_1 and C_2, and let B be the incompatible taxa with respect to C_2 and C_3; then A and B have the following cases: (i) $A = B$; (ii) $A \subset B$; (iii) $B \subset A$; (iv) $A \cap B = \emptyset$; (v) $A \cap B \neq \emptyset$, $A \not\subseteq B$ and $B \not\subseteq A$.

(i) $A = B$. Since there is no edge between C_1 and C_3, C_1 and C_3 are compatible; that is, $C_1 \cap C_3 = \emptyset$, or $C_1 \subseteq C_3$, or $C_3 \subseteq C_1$. Because $A \subseteq C_1$ and $A \subseteq C_3$, we have that $C_1 \cap C_3 \neq \emptyset$. Therefore, $C_1 \subseteq C_3$ or $C_3 \subseteq C_1$. Then, we have the simplest set of clusters $\mathscr{C}_1 = \{\{1,3\},\{1,2\},\{1,3,4\}\}$, and any one set of clusters in this case is isomorphic to $\mathscr{C}_1$.

(ii) $A \subset B$. Assume that $B = \{A, B_0\}$. It is similar to the case (i), and we have that $C_1 \subseteq C_3$. Then, the simplest set of clusters is $\mathscr{C}_2 = \{\{1,3\},\{1,2,4\},\{1,2,3\}\}$, and any one set of clusters in this case is isomorphic to $\mathscr{C}_1$.

(iii) $B \subset A$. This case is similar to case (ii). The sets of clusters are in case (ii) if and only if they are in case (iii). Hence, any one set of clusters in case (iii) and $\mathscr{C}_2$ are isomorphic.

(iv) $A \cap B = \emptyset$. Then, $C_1 \cap C_3 = \emptyset$. We have that $|A| = 1$ and $|B| = 1$ in the simplest set of clusters, since they can be collapsed if $|A| \ge 2$ or $|B| \ge 2$. Assume that $C_1 = \{A, B_1\}$ and $C_3 = \{B, B_2\}$. We have that $|B_1| = 1$ and $|B_2| = 1$ in the simplest set of clusters, since they can be collapsed if $|B_1| \ge 2$ or $|B_2| \ge 2$. Then, $|C_1| = 2$ and $|C_3| = 2$ in the simplest set of clusters. $\mathscr{C}_3 = \{\{1,2\},\{2,3\},\{3,4\}\}$ and $\mathscr{C}_4 = \{\{1,2\},\{2,3,5\},\{3,4\}\}$ are the simplest sets of clusters in this case. Therefore, any one set of clusters in this case is isomorphic to $\mathscr{C}_3$ or $\mathscr{C}_4$.

(v) $A \cap B \neq \emptyset$, $A \not\subseteq B$ and $B \not\subseteq A$. Let $A = \{A_0, A_1\}$ and $B = \{A_1, B_0\}$, where A_0, A_1, and B_0 are not empty. We have $\{A_0, A_1, B_0\} \subseteq C_2$, and $C_1 \subseteq C_3$ or $C_3 \subseteq C_1$. If $C_1 \subseteq C_3$, then $A_1 \subseteq C_3$. So $A_1 \subseteq B$, which contradicts the case that $A \not\subseteq B$. Similarly, we can get the contradiction when $C_3 \subseteq C_1$. Thus, there exists no set of clusters in this case. □

Figure 5 shows the DC networks for the simplest sets of clusters $\mathscr{C}_1$, $\mathscr{C}_2$, $\mathscr{C}_3$, and $\mathscr{C}_4$, respectively.

Lemma 7. *Let $\mathfrak{C} = \{\mathscr{C} \mid \mathscr{C}$ is a set of clusters$\}$, where $IG(\mathscr{C})$ is a nonlinear biconnected component with three nodes*

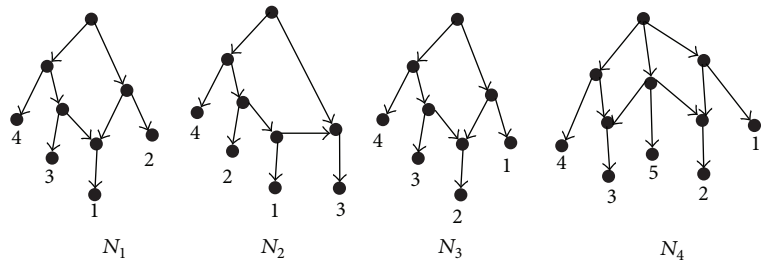

Figure 5: The DC networks for all simplest cluster sets whose incompatible graphs are topologies in Figure 4.

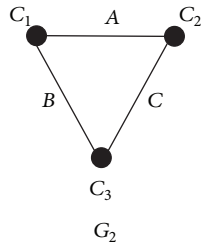

Figure 6: The topology of the nonlinear biconnected component with three nodes.

(see Figure 6). Let $\mathscr{C}_1 = \{\{1,3\},\{1,2,4\},\{1,2,5\}\}$, $\mathscr{C}_2 = \{\{1,2\},\{1,3\},\{1,4\}\}$, $\mathscr{C}_3 = \{\{1,2,4\},\{1,3,5\},\{1,2,3\}\}$, $\mathscr{C}_4 = \{\{1,2,4\},\{1,3,5\},\{1,2,3,6\}\}$, $\mathscr{C}_5 = \{\{1,2\},\{2,3\},\{1,3\}\}$, $\mathscr{C}_6 = \{\{1,2,4\},\{1,3\},\{2,3\}\}$, $\mathscr{C}_7 = \{\{1,2,4\},\{1,3,5\},\{2,3\}\}$, $\mathscr{C}_8 = \{\{1,2,4\},\{1,3,5\},\{2,3,6\}\}$, $\mathscr{C}_9 = \{\{1,2,3\},\{1,2,4\},\{1,3,4\}\}$, $\mathscr{C}_{10} = \{\{1,2,3,5\},\{1,2,4\},\{1,3,4\}\}$, $\mathscr{C}_{11} = \{\{1,2,3,5\},\{1,2,4,6\},\{1,3,4\}\}$, *and* $\mathscr{C}_{12} = \{\{1,2,3,5\},\{1,2,4,6\},\{1,3,4,7\}\}$. *Then, any one set of clusters in* $\mathfrak{C}$ *is isomorphic to one of* $\mathscr{C}_i$ $(1 \le i \le 12)$.

Proof. Figure 6 shows the topology of the nonlinear biconnected component with three nodes. Here, C_1, C_2, and C_3 are the clusters, and A, B, and C are the incompatible taxa corresponding to them. All cases are as follows: (i) $A = B$; then, $A \subseteq C$ or $A = C$; (ii) $A \subset B$; then, $A \subset C$, and $C \cap B = A$; (iii) $A \cap B = \emptyset$; then, $A \cap C = \emptyset$ and $B \cap C = \emptyset$; (iv) $A \cap B \neq \emptyset$, $A \not\subseteq B$, $B \not\subseteq A$; then, $A \cap C \neq \emptyset$ and $B \cap C \neq \emptyset$.

(i) $A = B$. If $A \subseteq C$, then $A \subseteq C_1$, $C \subseteq C_2$, and $C \subseteq C_3$. We have $|A| = 1$ in the simplest set of clusters; otherwise, A can be collapsed into one taxon. Similarly, we have $|C| = 2$ in the simplest set of clusters. Let $A = \{1\}$ and $C = \{1,2\}$; then, we can obtain the only simplest set of clusters $\mathscr{C}_1 = \{\{1,3\},\{1,2,4\},\{1,2,5\}\}$. Any one set of clusters meeting this case will be isomorphic to $\mathscr{C}_1$.

If $A = C$, then $A = B = C$. There is $|A| = 1$ in the simplest set of clusters; otherwise, A can be collapsed into one taxon. Let $A = B = C = \{1\}$; then, we can obtain the only simplest set of clusters $\mathscr{C}_2 = \{\{1,2\},\{1,3\},\{1,4\}\}$. Any one set of clusters in this case will be isomorphic to $\mathscr{C}_2$.

(ii) $A \subset B$, $A \subset C$, *and* $C \cap B = A$. Then, we can obtain the simplest sets of clusters $\mathscr{C}_3 = \{\{1,2,4\},\{1,3,5\},\{1,2,3\}\}$ and $\mathscr{C}_4 = \{\{1,2,4\},\{1,3,5\},\{1,2,3,6\}\}$. Any one set of clusters in this case will be isomorphic to $\mathscr{C}_3$ or $\mathscr{C}_4$.

(iii) $A \cap B = \emptyset$; *then,* $A \cap C = \emptyset$ *and* $B \cap C = \emptyset$. Then, we can obtain the simplest sets of clusters $\mathscr{C}_5 = \{\{1,2\},\{2,3\},\{1,3\}\}$ and $\mathscr{C}_6 = \{\{1,2,4\},\{1,3\},\{2,3\}\}$ and $\mathscr{C}_7 = \{\{1,2,4\},\{1,3,5\},\{2,3\}\}$ and $\mathscr{C}_8 = \{\{1,2,4\},\{1,3,5\},\{2,3,6\}\}$. Any one set of clusters in this case will be isomorphic to one of $\mathscr{C}_5$, $\mathscr{C}_6$, $\mathscr{C}_7$, and $\mathscr{C}_8$.

(iv) $A \cap B \neq \emptyset$, $A \not\subseteq B$, $B \not\subseteq A$; *then,* $A \cap C \neq \emptyset$ *and* $B \cap C \neq \emptyset$. Let $A \cap B = A_0$; then, $A \cap C = A_0$ and $B \cap C = A_0$. We have $|A_0| = 1$ in the simplest set of clusters; otherwise, A_0 can be collapsed into one taxon. Let $A_0 = \{1\}$. Then, $A = \{1,2\}$, $B = \{1,3\}$, and $C = \{1,4\}$. For the first case, we can obtain the simplest sets of clusters $\mathscr{C}_9 = \{\{1,2,3\},\{1,2,4\},\{1,3,4\}\}$ and $\mathscr{C}_{10} = \{\{1,2,3,5\},\{1,2,4\},\{1,3,4\}\}$ and $\mathscr{C}_{11} = \{\{1,2,3,5\},\{1,2,4,6\},\{1,3,4\}\}$ and $\mathscr{C}_{12} = \{\{1,2,3,5\},\{1,2,4,6\},\{1,3,4,7\}\}$. Any one set of clusters in this case will be isomorphic to one of them. □

Figure 7 shows the DC networks for the simplest sets of clusters $\mathscr{C}_i$ $(1 \le i \le 12)$, respectively. Lemmas 5, 6, and 7 compute all simplest sets of clusters, whose incompatible graphs are the biconnected components with two nodes or three nodes. Figures 6 and 7 show the DC networks constructed by the BIMLR algorithm for all simplest sets of clusters; then, the DC network for a set of clusters $\mathscr{C}$ can be obtained from the DC network representing the simplest set of clusters which is isomorphic to $\mathscr{C}$; that is, it does not need to be constructed once again. This conclusion is very important to the construction of networks.

4. Conclusion

This paper computes all simplest sets of clusters for the topologies of incompatible graph with two nodes and three nodes. We can construct the DC networks for those simplest sets of clusters and save them. When constructing DC networks for any one set of clusters $\mathscr{C}$, algorithms only need to read the DC network N_0 of the simplest set of clusters isomorphic to $\mathscr{C}$ and then compute the DC network for $\mathscr{C}$ from N_0 by replacing labels of leaves in N_0 by the taxa in $\mathscr{C}$, which will save more time for the algorithms.

We will compute the simplest sets of clusters for more topologies of incompatible graph in the future.

Competing Interests

The authors declare that they have no competing interests.

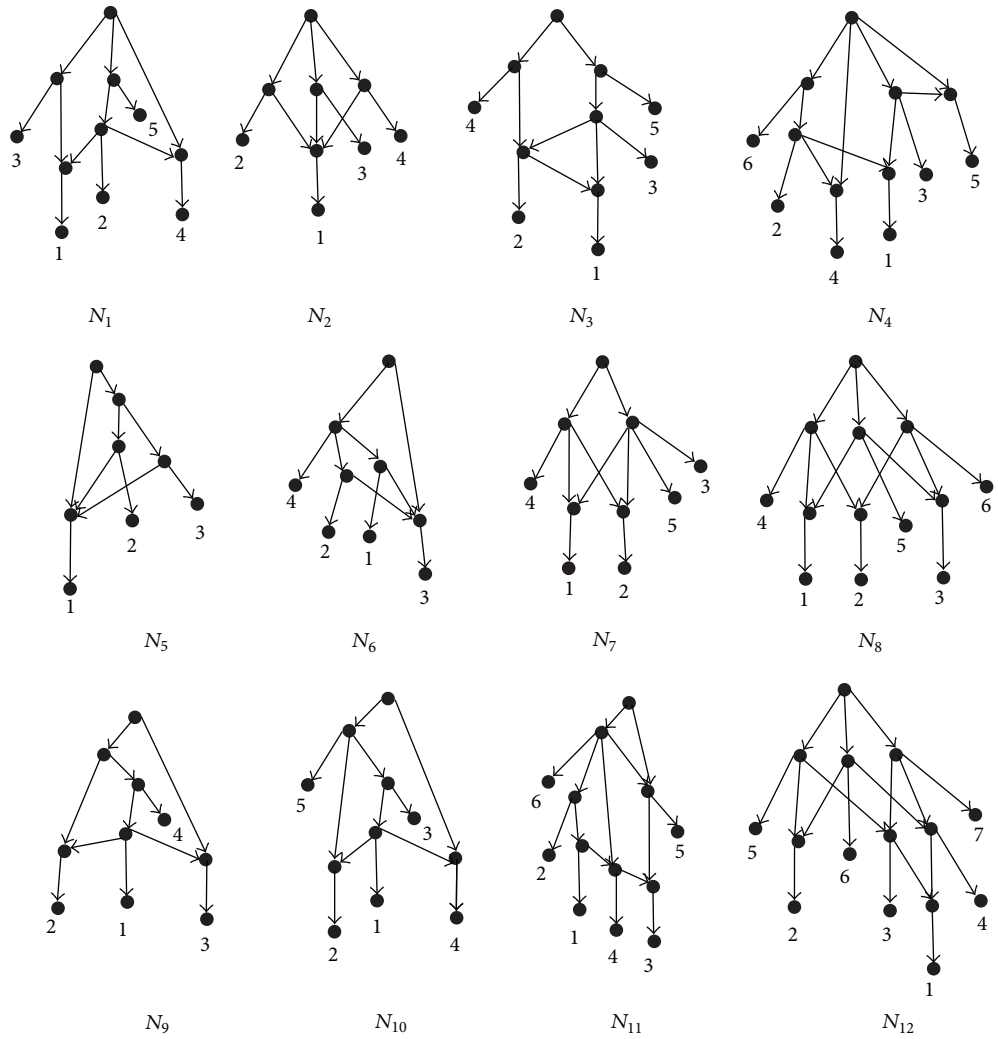

Figure 7: The DC networks for all simplest cluster sets whose incompatible graphs are topologies in Figure 6.

Acknowledgments

The work was supported by the Natural Science Foundation of Inner Mongolia Province of China (2015BS0601) and the National Natural Science Foundation of China (61300098, 31360289).

References

[1] Q. Zou, Q. Hu, M. Guo, and G. Wang, "Halign: fast multiple similar DNA/RNA sequence alignment based on the centre star strategy," *Bioinformatics*, vol. 31, no. 15, pp. 2475–2481, 2015.

[2] D. Mrozek, M. Brozek, and B. Małysiak-Mrozek, "Parallel implementation of 3D protein structure similarity searches using a GPU and the CUDA," *Journal of Molecular Modeling*, vol. 20, no. 2, pp. 1–17, 2014.

[3] D. Gusfield, D. Hickerson, and S. Eddhu, "An efficiently computed lower bound on the number of recombinations in phylogenetic networks: theory and empirical study," *Discrete Applied Mathematics*, vol. 155, no. 6-7, pp. 806–830, 2007.

[4] Q. Zou, X.-B. Li, W.-R. Jiang, Z.-Y. Lin, G.-L. Li, and K. Chen, "Survey of MapReduce frame operation in bioinformatics," *Briefings in Bioinformatics*, vol. 15, no. 4, pp. 637–647, 2014.

[5] Y. Liu, X. Zeng, Z. He, and Q. Zou, "Inferring microRNA-disease associations by random walk on a heterogeneous network with multiple data sources," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2016.

[6] D. H. Huson, R. Rupp, and C. Scornavacca, *Phylogenetic Networks: Concepts, Algorithms and Applications*, Cambridge University Press, New York, NY, USA, 2011.

[7] Q. Zou, Q. Hu, M. Guo, and G. Wang, "HAlign: fast multiple similar DNA/RNA sequence alignment based on the centre star strategy," *Bioinformatics*, vol. 31, no. 15, pp. 2475–2481, 2015.

[8] J. Wang, M.-Z. Guo, and L. L. Xing, "FastJoin, an improved neighbor-joining algorithm," *Genetics and Molecular Research*, vol. 11, no. 3, pp. 1909–1922, 2012.

[9] Q. Zou, J. Zeng, L. Cao, and R. Ji, "A novel features ranking metric with application to scalable visual and bioinformatics data classification," *Neurocomputing*, vol. 173, pp. 346–354, 2016.

[10] Q. Zou, J. Li, L. Song, X. Zeng, and G. Wang, "Similarity computation strategies in the microRNA-disease network: a survey," *Briefings in Functional Genomics*, vol. 15, no. 1, pp. 55–64, 2016.

[11] D. Bryant and V. Moulton, "Neighbor-net: an agglomerative method for the construction of phylogenetic networks," *Molecular Biology and Evolution*, vol. 21, no. 2, pp. 255–265, 2004.

[12] D. H. Huson, T. Dezulian, T. Klopper, and M. A. Steel, "Phylogenetic super-networks from partial trees," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 1, no. 4, pp. 151–158, 2004.

[13] D. H. Huson and T. H. Kloepper, "Computing recombination networks from binary sequences," *Bioinformatics*, vol. 21, supplement 2, pp. ii159–ii165, 2005.

[14] L. van Iersel, S. Kelk, R. Rupp, and D. Huson, "Phylogenetic networks do not need to be complex: using fewer reticulations to represent conflicting clusters," *Bioinformatics*, vol. 26, no. 12, pp. i124–i131, 2010.

[15] Y. Wu, "Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees," *Bioinformatics*, vol. 26, no. 12, Article ID btq198, pp. i140–i148, 2010.

[16] D. H. Huson, R. Rupp, V. Berry, P. Gambette, and C. Paul, "Computing galled networks from real data," *Bioinformatics*, vol. 25, no. 12, pp. i85–i93, 2009.

[17] D. H. Huson and R. Rupp, "Summarizing multiple gene trees using cluster networks," in *Algorithms in Bioinformatics*, K. A. Crandall and J. Lagergren, Eds., vol. 5251, pp. 296–305, Springer, New York, NY, USA, 2008.

[18] L. van Iersel, J. Keijsper, S. Kelk, L. Stougie, F. Hagen, and T. Boekhout, "Constructing level-2 phylogenetic networks from triplets, computational Biology and Bioinformatics," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 4, pp. 667–681, 2009.

[19] L. van Iersel and S. Kelk, "Constructing the simplest possible phylogenetic network from triplets," *Algorithmica*, vol. 60, no. 2, pp. 207–235, 2011.

[20] J. Wang, "A new algorithm to construct phylogenetic networks from trees," *Genetics and Molecular Research*, vol. 13, no. 1, pp. 1456–1464, 2014.

[21] D. Mrozek, *High-Performance Computational Solutions in Protein Bioinformatics*, Springer Publishing Company, Incorporated, 2014.

[22] D. Mrozek, P. Gosk, and B. Małysiak-Mrozek, "Scaling Ab initio predictions of 3D protein structures in microsoft azure cloud," *Journal of Grid Computing*, vol. 13, no. 4, pp. 561–585, 2015.

[23] L. Van Iersel, S. Kelk, R. Rupp, and D. Huson, "Phylogenetic networks do not need to be complex: using fewer reticulations to represent conflicting clusters," *Bioinformatics*, vol. 26, no. 12, Article ID btq202, pp. i124–i131, 2010.

[24] J. Wang, M. Guo, X. Liu et al., "Lnetwork: an efficient and effective method for constructing phylogenetic networks," *Bioinformatics*, vol. 29, no. 18, pp. 2269–2276, 2013.

[25] J. Wang, M. Guo, L. Xing, K. Che, X. Liu, and C. Wang, "BIMLR: a method for constructing rooted phylogenetic networks from rooted phylogenetic trees," *Gene*, vol. 527, no. 1, pp. 344–351, 2013.