RESEARCH ARTICLE

A Survey of Methods for Constructing Rooted Phylogenetic Networks

Juan Wang*

School of Computer Science, Inner Mongolia University, Hohhot, China

* wangjuanangle@hit.edu.cn

Abstract

Rooted phylogenetic networks are primarily used to represent conflicting evolutionary information and describe the reticulate evolutionary events in phylogeny. So far a lot of methods have been presented for constructing rooted phylogenetic networks, of which the methods based on the decomposition property of networks and by means of the incompatible graph (such as the CASS, the LNETWORK and the BIMLR) are more efficient than other available methods. The paper will discuss and compare these methods by both the practical and artificial datasets, in the aspect of the running time of the methods and the effective of constructed phylogenetic networks. The results show that the LNETWORK can construct much simper networks than the others.

OPEN ACCESS

Citation: Wang J (2016) A Survey of Methods for Constructing Rooted Phylogenetic Networks. PLoS ONE 11(11): e0165834. doi:10.1371/journal.pone.0165834

Editor: Andreas Hejnol, Principal Investigator, NORWAY

Received: July 6, 2016

Accepted: October 18, 2016

Published: November 2, 2016

Data Availability Statement: Data are available from the Dryad repository with the following DOI: doi:10.5061/dryad.c3g8t.

Funding: The author(s) received no specific funding for this work.

Competing Interests: The author has declared that no competing interests exist.

Introduction

The evolutionary history of species is traditionally denoted as a rooted phylogenetic tree. Each tree represents certain evolutionary information of the species (denoted as cluster, will be discussed in the following) [1–4]. When rooted phylogenetic trees are constructed by different methods or from different datasets, all of the evolutionary information represented by these tree are often conflicting. The conflicting evolutionary information cannot be expressed as a phylogenetic tree. However, phylogenetic network can represent the conflicting evolutionary information, which is a generalization of phylogenetic tree. It can also describe the evolution involving significant amounts of reticulate events such as recombinations, hybridizations, and horizontal gene transfers [5–9].

Phylogenetic networks can be divided into unrooted [10–15] and rooted networks [16–26]. Unrooted phylogenetic networks are mainly used to visualize conflicting evolutionary information. Rooted phylogenetic networks not only can represent conflicting evolutionary information implied phylogenetic trees, but also can describe the reticulate evolutionary events that species occurred during evolution [27]. There is a large body of research on rooted phylogenetic networks. Devising appropriate algorithms for constructing rooted phylogenetic networks from rooted phylogenetic trees has become an important field of research in molecular evolution. Recently a lot of scholars have focused on the research of the field, and developed a number of methods.

Dendroscope [28] is a program for constructing rooted phylogenetic networks, which unites some methods such as the cluster network [29], the galled network [30] and the Cass [22]. Among all of the methods, the Cass can construct simpler networks than other methods, but it is extremely slow for large datasets. And the networks constructed by the Cass are highly dependent on the order of input data, i.e. the constructed phylogenetic networks are generally different for the same dataset when input orders are different. Then Wang *et al* improved the Cass algorithm, and designed two algorithms: the Lnetwork [31] and the BIMLR [32]. The Lnetwork and the BIMLR are faster than Cass and have less influence of input data order.

In the following, we refer to rooted phylogenetic networks as networks, unless otherwise stated.

Preliminaries

Given a set of taxa $\mathcal{X}$. A subset of $\mathcal{X}$ (except both $\emptyset$ and $\mathcal{X}$) is called a cluster. For two clusters C_1 and C_2 on $\mathcal{X}$, if they are disjoint or one is a subset of the other, i.e. $C_1 \cap C_2 = \emptyset$ or $C_1 \subseteq C_2$ or $C_2 \subseteq C_1$, we say that C_1 and C_2 are compatible, otherwise they are incompatible. Given a set of clusters $\mathcal{C}$ on $\mathcal{X}$. If any two clusters in $\mathcal{C}$ is compatible, $\mathcal{C}$ is called compatible, otherwise it is incompatible. The incompatibility graph $IG(\mathcal{C}) = (V, E)$ of $\mathcal{C}$ is defined as an undirected graph with node set $V = C$ and edge set E, where two clusters are connected by an edge if and only if they are incompatible.

Let S be a subset of $\mathcal{X}$. The restriction of $\mathcal{C}$ to S, denoted by $\mathcal{C}|_S$, is defined as the result of removing all of the taxa in $\mathcal{X} \backslash S$ (i.e. the taxa in $\mathcal{X}$ but not in S) from each cluster in $\mathcal{C}$. The incompatibility degree of $\mathcal{C}$, denoted by $d(\mathcal{C})$, is defined as the number of edges in $IG(\mathcal{C})$. Let $\mathcal{C}$ be a set of clusters on $\mathcal{X}$ and x a taxon in $\mathcal{X}$. The incompatibility degree of x, denoted by $d(x)$, is defined as the result of subtracting the incompatibility degree of $\mathcal{C}|_{\mathcal{X}\backslash\{x\}}$ from that of $\mathcal{C}$, i.e. $d(x) = d(\mathcal{C}) - d(\mathcal{C}|_{\mathcal{X}\backslash\{x\}})$. If the incompatibility degree of a taxon x is maximal among all of the taxa in $\mathcal{X}$, i.e. $d(x) = \max\{d(y)|y \in \mathcal{X}\}$, we call that x is the incompatibility taxon w.r.t. $\mathcal{C}$. The frequency of a taxon x w.r.t. $\mathcal{C}$, denoted by $f(x)$, is defined as the number of clusters which contain x, i.e. $f(x) = |\{C \in \mathcal{C}|x \in C\}|$.

A rooted phylogenetic network $N = (V, E)$ on $\mathcal{X}$ is a rooted directed acyclic graph (DAG for short), and its leaves are bijectively labelled as $\mathcal{X}$. The indegree of a node $v \in V$ is denoted as $\sigma(v)$. A node v is a reticulate node if $\sigma(v) \geq 2$; otherwise it is a tree node; particularly, a tree node is a root node if $\sigma(v) = 0$. An edge $e = (u, v)$ is a tree edge if v is a tree node, otherwise it is a reticulate edge. The reticulation number in a network $N = (V, E)$ is $\Sigma_{\sigma(v)>0}(\sigma(v) - 1) = |E| - |V| + 1$.

Given a cluster C and a rooted phylogenetic tree T. If there is an edge e in T such that the set of taxa reachable from e equals C, we say that T represents C. Given a network N, when connecting one incoming edge and disconnecting all other incoming edges for each reticulate node, if there exists a tree edge e such that the set of taxa reachable from e equals C, we say that N represents C in the softwired sense. Alternatively, if there is a tree edge e in N such that the set of taxa reachable from e equals C, we say that N represents C in the hardwired sense.

Let $N = (V, E)$ be a network representing the set of clusters $\mathcal{C}$. A cluster $C \in \mathcal{C}$ is often represented by more than one tree edge in N and a tree edge $e \in E$ often represents more than one cluster in $\mathcal{C}$. If there exists a mapping ϵ from $\mathcal{C}$ to the set of tree edges of N, where $\epsilon(C)$ is a tree edge representing C for $C \in \mathcal{C}$, such that for any two clusters $C_1, C_2 \in \mathcal{C}$, C_1 and C_2 lie in the same connected component of the incompatibility graph $IG(\mathcal{C})$ if and only if $\epsilon(C_1)$ and $\epsilon(C_2)$ are contained in the same biconnected component of N. Then we call that N is decomposable w.r.t. $\mathcal{C}$.

When constructing rooted phylogenetic networks from rooted phylogenetic trees, the methods first compute the clusters represented by the input trees, and then construct a rooted phylogenetic network representing all clusters.

The rooted phylogenetic networks can describe evolutionary history in the presence of reticulate events, such as horizontal gene transfers, hybridizations and recombinations. These reticulate events are rare in reality [33]. Accordingly, it is expected that the constructed network has the minimal number of reticulate nodes. Let N be a network constructed for the input cluster set $\mathcal{C}$. Assume that $\mathcal{C}^{'}$ is the set of clusters represented by N. In fact $\mathcal{C}^{'}$ contains more clusters than $\mathcal{C}$, i.e. $\mathcal{C} \subsetneq \mathcal{C}^{'}$. Here we define the redundant clusters $\mathcal{C}_0$ of N as the clusters which are in $\mathcal{C}^{'}$ but not in $\mathcal{C}$, i.e. $\mathcal{C}_0 = \mathcal{C}^{'} \backslash \mathcal{C}$. In phylogenetic analysis, the taxa in a cluster are putative monophyletic. Consequently, the ideal situation would be $\mathcal{C}_0 = \emptyset$, i.e. all clusters represented by the constructed network would be the clusters represented in the input trees, and no others. Therefore, by means of parsimony principle, the best constructed network is one that minimizes the number of redundant clusters, which is based on the prerequisite that it has minimal number of reticulate nodes.

The following section will introduce the main methods for constructing rooted phylogenetic networks from rooted phylogenetic trees.

Methods

So far, the main methods for constructing rooted phylogenetic networks from rooted phylogenetic trees are the cluster network, the galled network, the Cass, the Lnetwork and the BIMLR. The following will give a brief introduction to each method.

The cluster network is a method for constructing rooted phylogenetic networks, which is based on the Hasse diagram. Given a set of clusters $\mathcal{C}$. It first defines a partial order which is a binary relation $\preccurlyeq$ on $\mathcal{C}$: for $u, v \in \mathcal{C}$, $u \preccurlyeq v$ if and only if $u \subseteq v$. The $(\mathcal{C}, \preceq)$ is called a partially ordered set. Then it draws a Hasse diagram $H = (V, E)$ for $(\mathcal{C}, \preceq)$, which is a DAG with node set $V = \mathcal{C}$ and the edge set E, where there is an edge $e = (u, v)$ if and only if $v \precnapprox u$ and there exists no other node w in V such that $v \precnapprox w \precnapprox u$. Finally it labels the leaves of H by the taxa of $\mathcal{X}$ and assigns the root of H. The result DAG is the rooted phylogenetic network representing $\mathcal{C}$.

The galled network is a method based on the seed-growing algorithm. It first finds a set of taxa $S \subset \mathcal{X}$ by seed-growing algorithm such that the set $\mathcal{C}^{'} = \mathcal{C}|_{\mathcal{X} \backslash S}$ is compatible. Next it constructs a rooted tree T for $\mathcal{C}^{'}$. Finally it attaches the reticulate nodes to T under a certain amount of constraints, where the labels of nodes which are children of reticulate nodes are the taxa of S. The constructed network represents $\mathcal{C}$.

The Cass, the Lnetwork and the BIMLR are the methods based on the decomposition property of networks. They first find all non-trivial biconnected components $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_k$ of $IG(\mathcal{C})$; and then construct the subnetwork for $\mathcal{C}_i (1 \leq i \leq k)$; next integrate those subnetworks into a final network. The difference among them is the construction of the subnetworks.

We have known that a network N represents an incompatible set of clusters. After removing all reticulate nodes from N, N becomes a tree representing a compatible set of clusters. However, the construction of networks is the inverse process mentioned above. Given a set of clusters $\mathcal{C}$ on $\mathcal{X}$. We can construct a tree for $\mathcal{C}$ if it is compatible, otherwise, we first remove some taxa from $\mathcal{X}$, such that the result set $\mathcal{C}^{'}$ is compatible, then construct a tree for $\mathcal{C}^{'}$, finally append the removed taxa to the tree under certain conditions.

The Cass, the Lnetwork and the BIMLR construct subnetworks by the above description. Assume that the taxa set of $\mathcal{C}_i$ is $\mathcal{X}_i$ $(1 \leq i \leq k)$. When constructing the subnetworks for $\mathcal{C}_i$, they first remove a few taxa of $\mathcal{X}_i$ from each cluster in $\mathcal{C}_i$, such that the result set $\mathcal{C}_0$ is compatible, then construct a rooted tree T for $\mathcal{C}_0$, next attach some new reticulate nodes to T, finally add a new leaf below each reticulate node and label it as the removed taxon. From the process, we can see that the removed taxa in $\mathcal{X}_i$ are very pivotal.

The difference among the Cass, the Lnetwork and the BIMLR is the removed taxa. The Cass tries to remove a few taxa, i.e. by means of trial and error, it randomly removes some taxa, if it can construct a network representing the set of clusters, then it stops; otherwise it continues to remove other taxa. The Lnetwork removes the taxa computed by seed-growing algorithm. The BIMLR removes the incompatibility taxa with the maximal frequency. The Cass method aims at minimizing the number of reticulate nodes, while the Lnetwork and the BIMLR not only minimize the number of redundant of clusters but also let the reticulate nodes as few as possible.

When constructing networks, the Lnetwork and the BIMLR find all networks representing the cluster set, mainly to reduce the number of redundant clusters in the resulting network and lessen the influence of the input data order on the resulting network.

Results

In order to survey the performance of those methods, we do the experiments using both the practical and artificial data. The paper [31] has compared the Lnetwork, the Cass, the cluster network and the galled network; and the results show that the Lnetwork and the Cass can construct much simpler networks than the others. Here we just compare the Lnetwork, the Cass and the BIMLR using both the practical and artificial data (https://sites.google.com/site/cassalgorithm/data-sets).

All experiments were performed on a computer with an Intel Xeon E5504 2.0GHz CPU, 8 GB RAM and 147GB HDD. The operating system was Debian 4.1 32bit with Java 1.6 installed.

The experiments are used to compare two main aspects of these methods, on the one hand, the influence of input data order (Table 1 shows the results), on the other hand, the complexities of the constructed network, i.e. the reticulation number and the number of redundant clusters (Tables 2 and 3 show the results).

The papers [31, 32] have shown that the Lnetwork and the BIMLR are superior to the Cass in terms of the influence of input data order; and the two methods are faster than the Cass. Here we just compare the Lnetwork and the BIMLR on the influence of input data order, and the results are shown in Table 1. In the experiment, because each program needs to construct the network for every input order of dataset, the running time is factorial. Accordingly, here we just use the dataset with small scale. When comparing networks constructed by a method for the same dataset with different input orders, the difference of those networks is more small, the method is more stable, otherwise it is unstable. Here the difference among the constructed networks is measured by means of the tripartition distances [34].

Table 1 shows the number of constructed networks, the mean, the minimum (min) and the maximum (max) of tripartition distance of those networks constructed for each dataset. And the last row shows the average values. From the Table 1 we come to the following conclusions. First, for the same data with different input orders, the number of different networks constructed by the BIMLR is less than the number of different networks constructed by the Lnetwork except for the dataset with $|\mathcal{C}| = 21$ and $|\mathcal{X}| = 11$. Second, for almost datasets, the mean, min, max of the BIMLR are less than corresponding values of the Lnetwork except for the datasets with $|\mathcal{C}| = 22, |\mathcal{X}| = 13, |\mathcal{C}| = 21, |\mathcal{X}| = 11$ and $|\mathcal{C}| = 22, |\mathcal{X}| = 10$. The result shows that, when the input order of the data is changed, even the network constructed by the BIMLR is more than one, those networks are more similar to each other than the networks constructed by Lnetwork. Thus, the BIMLR are more stable than the Lnetwork.

Table 1. Results of LNetwork compared with BIMLR in terms of influence of input data order.

Data		LNetwork				BIMLR			
$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{X}\rvert$	***n***	**mean**	**min**	**max**	***n***	**mean**	**min**	**max**
35	22	1	0	0	0	1	0	0	0
25	15	1	0	0	0	1	0	0	0
22	13	2	1	1	1	2	1.5	1.5	1.5
27	15	2	1	1	1	2	1	1	1
25	13	3	1.2	0.5	1.5	1	0	0	0
22	11	1	0	0	0	1	0	0	0
17	10	3	1.3	1	1.5	3	1.3	1	1.5
13	8	2	1	1	1	1	0	0	0
23	11	2	1	1	1	2	1	1	1
18	10	3	2.5	1.5	3.5	3	1.5	0.5	2.5
21	11	1	0	0	0	2	0.5	0.5	0.5
12	10	1	0	0	0	1	0	0	0
21	10	2	1.5	1.5	1.5	2	0.5	0.5	0.5
13	7	2	1	1	1	1	0	0	0
22	10	1	0	0	0	2	0.5	0.5	0.5
21.3	11.7	1.8	1.2	1.1	1.4	1.6	0.6	0.4	0.7

Note: *n* represents the number of constructed networks and mean, min, max represent, respectively, the mean, the minimum, the maximum of tripartition distances of those networks.

doi:10.1371/journal.pone.0165834.t001

Table 2. Results of BIMLR, LNetwork and Cass for the artificial datasets.

Data		BIMLR			LNetwork			Cass		
$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{X}\rvert$	***t***	***r***	***c***	***t***	***r***	***c***	***t***	***r***	***c***
14	4	1 s	3	0	1 s	3	0	1 s	3	0
30	5	3 s	4	0	2 s	4	0	2 s	4	0
62	6	11 s	5	0	18 s	5	0	11 s	5	0
42	10	2 s	4	8	1 s	4	14	10 s	4	34
39	11	2 s	5	8	38 s	6	18	21 s	5	7
61	11	3 s	5	11	15 s	6	43	1 m 26 s	5	48
75	30	1 s	2	19	1 s	2	19	4 s	2	19
180	51	4 s	2	0	4 s	2	0	40 s	2	0
70	56	1 s	4	0	1 s	4	0	1 s	4	0
270	76	12 s	2	0	12 s	2	0	6 m 22 s	2	0
404	122	44 m	2	0	44 s	2	0	1 h 14 m	2	0
113.4	34.7	7 s	3.5	4.2	13 s	3.6	8.5	7 m 34 s	3.5	10

Note: *t*, *r* and *c* represent, respectively, the running time, the reticulation number and the redundant cluster number.

doi:10.1371/journal.pone.0165834.t002

Table 3. Results of BIMLR, LNetwork and Cass for the practical datasets.

Data		BIMLR			LNetwork			Cass		
$\|\mathcal{C}\|$	$\|\mathcal{X}\|$	***t***	***r***	***c***	***t***	***r***	***c***	***t***	***r***	***c***
86	37	3 s	8	23	13 s	8	11	3 s	8	27
38	20	2 s	6	25	6 s	6	15	4 s	6	25
43	22	1 s	5	11	1 s	5	3	1 s	4	12
72	27	3 s	7	29	1 s	7	19	15 s	7	43
52	22	9 s	8	15	1 m 12 s	8	15	17 s	7	33
79	27	2 m 40 s	10	52	39 s	8	44	7 m 21 s	8	89
38	16	5 s	8	25	14 m 12 s	9	36	15 s	7	50
41	16	1 s	5	7	3 s	5	4	2 s	5	29
12	8	1 s	2	0	1 s	2	0	0 s	2	2
45	20	4 s	7	47	31 s	7	28	4 h 14 m	7	66
22	11	1 s	3	4	1 s	3	1	0 s	3	5
17	10	1 s	3	7	1 s	3	4	1 s	3	8
46	16	9 s	8	22	21 s	8	15	23 s	7	34
22	11	3 s	5	21	10 s	4	13	2 s	4	23
22	10	2 s	5	19	10 s	4	12	2 s	4	21
42.3	18.2	14 s	6	20.5	1 m 11 s	5.8	14.7	18 m	5.5	31

Note: *t*, *r* and *c* are defined as in Table 2.

doi:10.1371/journal.pone.0165834.t003

We compare the BIMLR, the LNetwork and the Cass on several artificial datasets. Table 2 shows the results, which have the reticulation number *r*, the redundant cluster number *c*, the running time *t* in hours (h), minutes (m) and seconds (s), and their average values in last row. Table 2 shows that the BIMLR takes least time except for the datasets with $|\mathcal{C}| = 30, |\mathcal{X}| = 5$ and $|\mathcal{C}| = 404, |\mathcal{X}| = 122$. For every dataset, the reticulation number of the network constructed by the BIMLR is the same as that of the network constructed by Cass, which is less than that of the network constructed by the LNetwork. The networks constructed by LNetwork have fewer redundant clusters than the networks constructed by Cass for almost all datasets, while the networks constructed by BIMLR have fewest redundant clusters. Thus, the networks constructed by BIMLR are simplest in terms of the redundant clusters and the reticulation number contained in the constructed networks for the artificial datasets.

We compare the BIMLR, the LNetwork and the Cass on practical datasets. Table 3 shows the results, which have the reticulation number *r*, the redundant cluster number *c*, the running time *t* in hours (h), minutes (m) and seconds (s) and their average values in last row. Table 3 shows the BIMLR takes least time except for the dataset with $|\mathcal{C}| = 79, |\mathcal{X}| = 27$. For most datasets, the reticulation number of the networks constructed by the Cass is least, secondly LNetwork, at least BIMLR. The networks constructed by LNetwork have fewest redundant clusters, secondly the BIMLR, at least Cass. Table 3 shows that the average reticulation number of LNetwork and BIMLR is slightly more than that of Cass. Hence, the networks constructed by LNetwork are simplest for the practical dataset.

From the Tables 2 and 3, it follows that the LNetwork and the BIMLR take less time than the Cass; the networks constructed by LNetwork and BIMLR have fewer redundant clusters than those constructed by CASS; and the average reticulation number of LNetwork and BIMLR are slightly more than that of Cass.

Conclusion

We compared BIMLR, LNETWORK and CASS using one artificial and one practical dataset. The results show that the BIMLR is superior to the others for the artificial datasets, while the LNETWORK is superior to the others for the practical datsets. Accordingly in practice, the LNETWORK is the best option for the construction of networks.

Acknowledgments

I would like to thank the editor and two anonymous referees for remarks and suggestions which improved the exposition in this paper.

Author Contributions

Conceptualization: JW.

Data curation: JW.

Formal analysis: JW.

Funding acquisition: JW.

Investigation: JW.

Methodology: JW.

Project administration: JW.

Resources: JW.

Software: JW.

Supervision: JW.

Validation: JW.

Visualization: JW.

Writing – original draft: JW.

Writing – review & editing: JW.

References

1. Vellutini BC, Hejnol A. Expression of segment polarity genes in brachiopods supports a non-segmental ancestral role of engrailed for bilaterians. Scientific Reports. 2016; 6(6): 1994–1994. doi: 10.1038/srep32387 PMID: 27561213
2. Hejnol A, Pang K. Xenacoelomorpha's significance for understanding bilaterian evolution. Current Opinion in Genetics & Development. 2016; 39: 48–54. doi: 10.1016/j.gde.2016.05.019 PMID: 27322587
3. Huson DH, Rupp R, Scornavacca C. Phylogenetic Networks: Concepts, Algorithms and Applications. Cambridge University Press; 2011.
4. Zou Q, Guo J, Ju Y, Wu M, Zeng X, Hong Z. Improving tRNAscan-SE Annotation Results via Ensemble Classifiers. Molecular Informatics. 2015; 34(11-12): 2992–3000. doi: 10.1002/minf.201500031
5. Zou Q, Hu Q, Guo M, Wang G. HAlign: Fast multiple similar DNA/RNA sequence alignment based on the centre star strategy. Bioinformatics. 2015; 31(15): 2475–2481. doi: 10.1093/bioinformatics/btv177 PMID: 25812743
6. Linder CR, Rieseberg LH. Reconstructing patterns of reticulate evolution in plants. American Journal of Botany. 2004; 91(10): 1700–1708. doi: 10.3732/ajb.91.10.1700

7. Gusfield D, Hickerson D, Eddhu S. An efficiently computed lower bound on the number of recombinations in phylogenetic networks: Theory and empirical study. Discrete Applied Mathematics. 2007; 155 (6): 806–830. doi: 10.1016/j.dam.2005.05.044

8. Zou Q, Li XB, Jiang WR, Lin ZY, Li GL, Chen K. Survey of MapReduce frame operation in bioinformatics. Briefings in Bioinformatics. 2014; 15(4): 637–647. doi: 10.1093/bib/bbs088 PMID: 23396756

9. Liu Y, Zeng X, He Z, Zou Q. Inferring microRNA-disease associations by random walk on a heterogeneous network with multiple data sources. IEEE/ACM Transactions on Computational Biology & Bioinformatics, 2016:1–1 doi: 10.1109/TCBB.2016.2550432 PMID: 27076459

10. Bandelt HJ, Dress AW. A canonical decomposition theory for metrics on a finite set. Advances in mathematics. 1992; 92(1): 47–105. doi: 10.1016/0001-8708(92)90061-O

11. Bandelt HJ, Forster P, Röhl A. Median-joining networks for inferring intraspecific phylogenies. Molecular biology and evolution. 1999; 16(1): 37–48. doi: 10.1093/oxfordjournals.molbev.a026036 PMID: 10331250

12. Buneman P. The recovery of trees from measures of dissimilarity. Mathematics in the archaeological and historical sciences. Edinburgh University Press; 1971: 387–395.

13. Huson DH, Bryant D. Application of phylogenetic networks in evolutionary studies. Molecular biology and evolution. 2006; 23(2): 254–267. doi: 10.1093/molbev/msj030 PMID: 16221896

14. Bryant D, Moulton V. Neighbor-net: an agglomerative method for the construction of phylogenetic networks. Molecular biology and evolution. 2004; 21(2): 255–265. doi: 10.1093/molbev/msh018 PMID: 14660700

15. Huson DH, Dezulian T, Klopper T, Steel MA. Phylogenetic super-networks from partial trees. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB). 2004; 1(4): 151–158. doi: 10.1109/TCBB.2004.44 PMID: 17051697

16. Song YS, Hein J. Constructing minimal ancestral recombination graphs. Journal of Computational Biology. 2005; 12(2): 147–169. doi: 10.1089/cmb.2005.12.147 PMID: 15767774

17. Huson DH, Kloepper TH. Computing recombination networks from binary sequences. Bioinformatics. 2005; 21(suppl2): ii159–ii165. doi: 10.1093/bioinformatics/bti1126 PMID: 16204096

18. Gusfield D, Eddhu S, Langley C. Efficient reconstruction of phylogenetic networks with constrained recombination. Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE. IEEE; 2003: 363–374.

19. Gusfield D. Optimal, efficient reconstruction of root-unknown phylogenetic networks with constrained and structured recombination. Journal of Computer and System Sciences. 2005; 70(3): 381–398. doi: 10.1016/j.jcss.2004.12.009

20. Gusfield D, Bansal V. A fundamental decomposition theory for phylogenetic networks and incompatible characters.Research in Computational Molecular Biology. Springer; 2005: 217–232.

21. Semple C. Hybridization networks. Department of Mathematics and Statistics, University of Canterbury; 2006.

22. van Iersel L, Kelk S, Rupp R, Huson D. Phylogenetic networks do not need to be complex: using fewer reticulations to represent conflicting clusters. Bioinformatics. 2010; 26(12): i124–i131. doi: 10.1093/bioinformatics/btq202 PMID: 20529896

23. Collins J, Linz S, Semple C. Quantifying hybridization in realistic time. Journal of Computational Biology. 2011; 18(10): 1305–1318. doi: 10.1089/cmb.2009.0166 PMID: 21210735

24. Wu Y. Close lower and upper bounds for the minimum reticulate network of multiple phylogenetic trees. Bioinformatics. 2010; 26(12): i140–i148. doi: 10.1093/bioinformatics/btq198 PMID: 20529899

25. van Iersel L, Keijsper J, Kelk S, Stougie L, Hagen F, Boekhout T. Constructing level-2 phylogenetic networks from triplets. Computational Biology and Bioinformatics, IEEE/ACM Transactions on. 2009; 6(4): 667–681. doi: 10.1109/TCBB.2009.22

26. 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

27. Wang J. A new algorithm to construct phylogenetic networks from trees. Genetics and Molecular Research. 2014; 13(1): 1456–1464. doi: 10.4238/2014.March.6.4 PMID: 24634244

28. Huson DH, Scornavacca C. Dendroscope 3: an interactive tool for rooted phylogenetic trees and networks. Systematic Biology. 2012; 61(6): 1061–1067. doi: 10.1093/sysbio/sys062 PMID: 22780991

29. Huson DH, Rupp R. Summarizing multiple gene trees using cluster networks. Algorithms in Bioinformatics. Springer; 2008: 296–305. doi: 10.1007/978-3-540-87361-7_25

30. Huson DH, Rupp R, Berry V, Gambette P, Paul C. Computing galled networks from real data. Bioinformatics. 2009; 25(12): i85–i93. doi: 10.1093/bioinformatics/btp217 PMID: 19478021

31. Wang J, Guo M, Liu X, Liu Y, Wang C, Xing L, Che k. LNETWORK: an efficient and effective method for constructing phylogenetic networks. Bioinformatics. 2013; 29(18): 2269–2276. doi: 10.1093/bioinformatics/btt378 PMID: 23811095

32. Wang J, Guo M, Xing L, Che K, Liu X, Wang C. BIMLR: A Method for Constructing Rooted Phylogenetic Networks from Rooted Phylogenetic Trees. Gene. 2013; 527(1): 344–351. doi: 10.1016/j.gene.2013.06.036 PMID: 23816409

33. Linder CR, Moret BME, Nakhleh L, Warnow T. Network (Reticulate) Evolution: Biology, Models, and Algorithms. In The Ninth Pacific Symposium on Biocomputing (PSB. 2010). 2010.

34. Cardona G, Llabrés M, Rosselló F, Valiente G. Metrics for phylogenetic networks I: Generalizations of the Robinson-Foulds metric. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB). 2009; 6(1): 46–61. doi: 10.1109/TCBB.2008.70 PMID: 19179698