# A Distance-Based Method for Inferring Phylogenetic Networks in the Presence of Incomplete Lineage Sorting

Yun Yu(✉) and Luay Nakhleh

Department of Computer Science, Rice University, 6100 Main Street, Houston, TX 77005, USA
{yy9,nakhleh}@rice.edu

**Abstract.** Hybridization and incomplete lineage sorting (ILS) are two evolutionary processes that result in incongruence among gene trees and complicate the identification of the species evolutionary history. Although a wide array of methods have been developed for inference of species phylogeny in the presence of each of these two processes individually, methods that can account for both of them simultaneously have been introduced recently. However, these new methods are based on the optimization of certain criteria, such as parsimony and likelihood, and are thus computationally intensive. In this paper, we present a novel distance-based method for inferring phylogenetic networks in the presence of ILS that makes use of pairwise distances computed from multiple sampled loci across the genome. We show in simulation studies that the method infers accurate networks when the estimated pairwise distances have good accuracy. Furthermore, we devised a heuristic for post-processing the inferred network to remove potential false positive reticulation events. The method is computationally very efficient and is applicable to very large data sets.

## 1 Introduction

Understanding the evolutionary history of a set of species and the intricate relationships between the evolution of genes and genomes are two central questions in biology. It has long been acknowledged that the evolutionary history of a genomic region from a set of species is not necessarily congruent with that of the species [16], which is the classic gene tree/species tree problem. The incongruence among gene trees and species tree may be caused by various evolutionary processes. Incomplete lineage sorting (ILS), which is a result of random genetic drift in populations, is one common process, especially in evolutionary scenarios that involve rapid speciation and/or large population sizes. The occurrence of ILS and its extent have been reported in various data studies of very diverse sets of organisms; e.g., [29,24,14,32,4,33,8,30]. A large variety of methods have been developed to deal with it; see [26,5,15,23] for recent surveys of such methods.

A second evolutionary process that results in gene tree incongruence is reticulation, which includes horizontal gene transfer in asexual species and hybridization in sexual species. Hybridization is believed to play an important role in several groups of eukaryotic species [1,2,17,18,27]. Not only does hybridization result in gene tree incongruence, but it also results in non-treelike phylogenetic relationships among species, that are best represented by *phylogenetic networks*. The structure of a phylogenetic network

R. Harrison et al. (Eds.): ISBRA 2015, LNBI 9096, pp. 378–389, 2015.
DOI: 10.1007/978-3-319-19048-8_32

is a rooted, directed acyclic graph, which allows for nodes with more than one parents. Many methods have been devised to infer these phylogenetic networks by making use of gene tree incongruence; see [22,11,23] for recent surveys of such methods.

With increasingly available genomic data, patterns of cooccurrence of hybridization and incomplete lineage sorting are being observed, or suspected, in the data [7,6,28,3,20]. This has called for developing methods that can take both hybridization and incomplete lineage sorting into account. Methods that assume only ILS as the cause of incongruence would completely miss the possibility of hybridization, whereas methods that infer phylogenetic networks without accounting for ILS would end up grossly overestimating the amount of hybridization when ILS is also at play. To address this issue, several methods were proposed recently. However, given the the complexity of modeling such scenarios in general, most of these methods focused on special cases of the problem (typically with limited complexity); e.g., [31,9,19,13,12,38]. More recently, methods for inferring general networks based on parsimony and likelihood criteria were developed [35,34,37,36]. The applicability of these inference methods is currently limited to small data sets, given the hardness of the inference problems under these two criteria.

Distance-based methods have long been some of the fastest methods in phylogenetics, producing very good estimates on phylogenies with thousands of taxa in minutes. Even when the accuracy of inferences made by these methods is not very high, trees produced by distance-based methods are still used as initial trees for the most computationally intensive and detailed methods, such as maximum likelihood. Thus, distance-based method provide a very good tool in phylogenetics. In this paper, we introduce a novel distance-based method that infers a phylogenetic network from pairwise distance data in the presence of both hybridization and ILS. Our method builds on the GLASS method [21] that was recently introduced to infer species trees from pairwise distances obtained from multiple loci under the assumption that all incongruence is due to ILS. We studied the performance of our method on simulated data and found that it produces very good results, even when we perturbed the pairwise distances so as to simulate error in distance estimates. We also devised a heuristic for potentially eliminating false positive reticulations in order to minimize the overestimation of the number of reticulations.

It is important to note that accurate estimates of pairwise distances based on multiple loci is a requirement for a good performance of our method (just like they are a requirement for a good performance of GLASS). We view this as a major obstacle facing the application of this method to real data. Nevertheless, as we pointed out above, this method can still be used to quickly generate a good phylogenetic network to initialize the search employed by computationally intensive methods such as [37,36].

## 2 Methods

### 2.1 Phylogenetic Networks

In order to account for both hybridization and incomplete lineage sorting in the evolutionary history of a set of species (or, genomes), we use an evolutionary (rather than "data-display") *phylogenetic network* model [22]. For a node $v$ in a digraph, we denote by $d^-(v)$ and $d^+(v)$ the in- and out-degree of $v$. A (binary) phylogenetic $\mathscr{X}$-network $N$ is a rooted, directed, acyclic graph whose node-set $V(N)$ is partitioned into four sets:

- $\{r\}$, the root of $N$, with $d^-(r) = 0$ and $d^+(r) = 2$;
- The leaf-set $V_L = \{v \in V(N) : d^-(r) = 1, d^+(r) = 0\}$, which are bijectively labeled by $\mathscr{X}$;
- The internal tree nodes $V_T = \{v \in V : d^-(r) = 1, d^+(r) = 2\}$; and
- The reticulation nodes $V_N = \{v \in V : d^-(r) = 2, d^+(r) = 1\}$.

Every structure inferred by our algorithm (described below) is a phylogenetic network. However, it is important to point out that there are phylogenetic networks that cannot be inferred by our algorithm. This is not a limitation of the algorithm, but rather has to do with the reconstructibility of certain reticulation scenarios (e.g., a reticulation edge involving two nodes one of which falls on the path from the root to the other node). More generally, let us denote by $L(v)$ the set of taxa that label leaves that are descendants of node $v$. Given a phylogenetic network $N$, for each node $v$ in $\{r\} \cup V_T$, we define the set $dp(v) = \{L(v_1) - L(v_2), L(v_2) - L(v_1)\}$ where $v_1$ and $v_2$ are the two children of $v$. Then if a phylogenetic network contains two nodes $u, v \in \{r\} \cup V_T$ where $dp(u) = dp(v)$, one of these two nodes cannot be inferred by our method.

### 2.2 Inferring a Network from a Distance Matrix

We denote by $D_L$ a distance matrix over a set of taxa $L$ where $D_L(i, j)$ is the distance between taxa $i$ and $j$ in $L$. With respect to the nodes of a phylogenetic network, we define two functions $DMax(u, v, S)$ and $DMin(u, v, S)$, where $u$ and $v$ are two nodes and $S$ is a set of nodes, to be $DMax(u, v, S) = \max\{D_L(a, b) : a \in L(u) - L(v), b \in L(v) - L(u), \nexists w \in S \text{ s.t.} \{a, b\} \subseteq L(w)\}$ and $DMin(u, v, S) = \min\{D_L(a, b) : a \in L(u) - L(v), b \in L(v) - L(u), \nexists w \in S \text{ s.t.} \{a, b\} \subseteq L(w)\}$. See Figure 1 for an illustration.

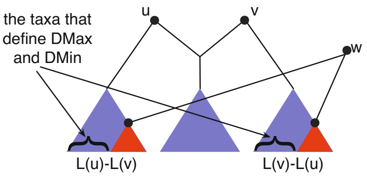


**Fig. 1.** An illustration of $DMax(u, v, S)$ and $DMin(u, v, S)$ computation on $S = \{u, v, w\}$

Assuming the pairwise distances are realizable by a phylogenetic network, the basic idea of our method is we start with a set of nodes $S$, each labeled by a taxon in $L$ and then we do the following until $S$ has only one node:

1. Let $X$ and $Y$ be two nodes in $S$ that have the minimum $DMin(X, Y, S)$.
2. If $DMin(X, Y, S) = DMax(X, Y, S)$, a speciation event is considered. We remove $X$ and $Y$ in $S$ and add node $XY$.
3. If $DMin(X, Y, S) \neq DMax(X, Y, S)$, a hybridization event is considered. We find the most parsimonious way to make a reticulation node(s), which can be one of the following:

- A reticulation node, say $u$, is added onto an edge whose tail is a descendant of $X$. We remove $Y$ from $S$ and add a new node whose children are $u$ and $Y$.
- A reticulation node, say $u$, is added onto an edge whose tail is a descendant of $Y$. We remove $X$ from $S$ and add a new node whose children are $u$ and $X$.
- Two reticulation nodes, say $u_x$ and $u_y$, are added onto an edge whose tail is a descendant of $X$ and an edge whose tail is a descendant of $Y$, respectively. We add a new node whose children are $u_x$ and $u_y$ to $S$.

More details can be found in Alg.1. See Fig. 2 for an example.

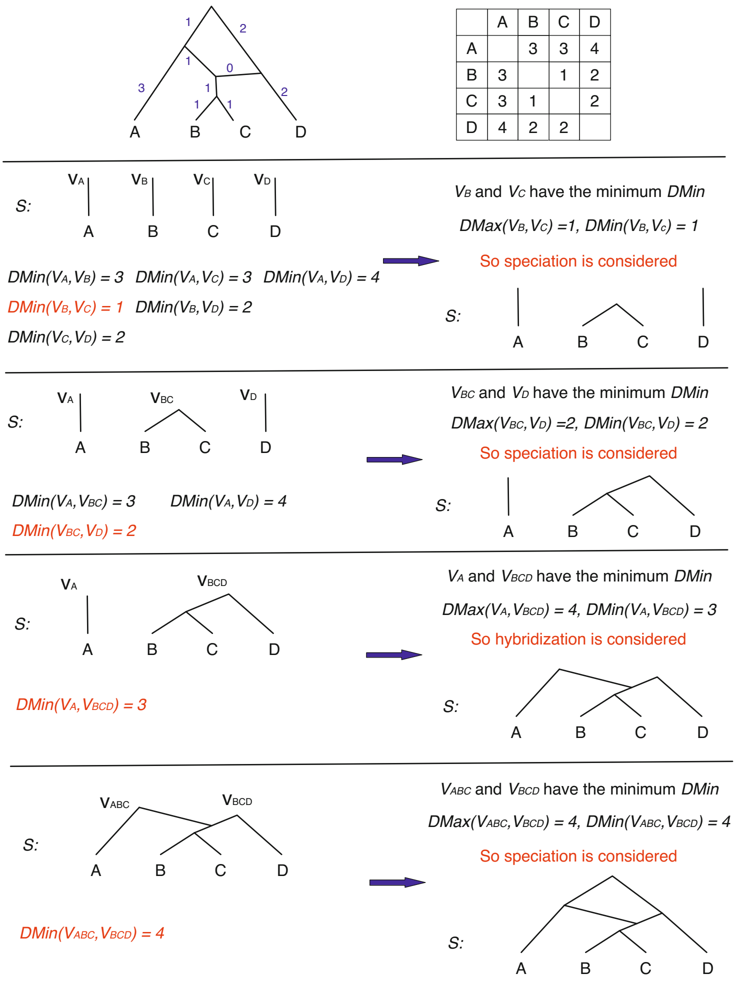
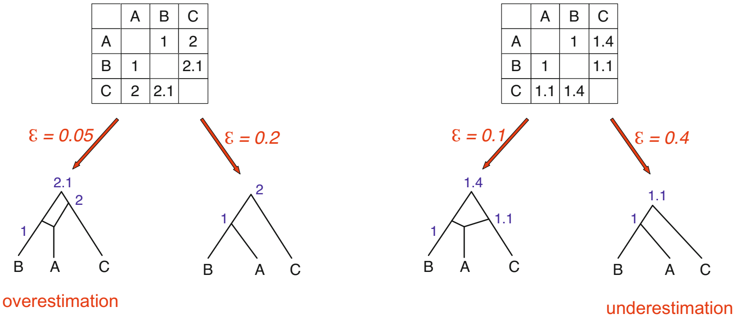

1
2
1
0
3
1
1
1
2
A
B
C
D
A B C D
A 3 3 4
B 3 1 2
C 3 1 2
D 4 2 2
S:
VA
VB
VC
VD
A
B
C
D
DMin(VA,VB) = 3 DMin(VA,VC) = 3 DMin(VA,VD) = 4
DMin(VB,VC) = 1 DMin(VB,VD) = 2
DMin(VC,VD) = 2
VB and VC have the minimum DMin
DMax(VB,VC) =1, DMin(VB,VC) = 1
So speciation is considered
S:
A
B
C
D
S:
VA
VBC
VD
A
B
C
D
DMin(VA,VBC) = 3 DMin(VA,VD) = 4
DMin(VBC,VD) = 2
VBC and VD have the minimum DMin
DMax(VBC,VD) =2, DMin(VBC,VD) = 2
So speciation is considered
S:
A
B
C
D
S:
VA
VBCD
A
B
C
D
DMin(VA,VBCD) = 3
VA and VBCD have the minimum DMin
DMax(VA,VBCD) = 4, DMin(VA,VBCD) = 3
So hybridization is considered
S:
A
B
C
D
S:
VABC
VBCD
A
B
C
D
DMin(VABC,VBCD) = 4
VABC and VBCD have the minimum DMin
DMax(VABC,VBCD) = 4, DMin(VABC,VBCD) = 4
So speciation is considered
S:
A
B
C
D

**Fig. 2.** An example of building a species network given true pairwise distances. The true species network and distance matrix are given on the top. For simplicity, the third parameter $S$ is omitted in $DMin$ and $DMax$ since the context is clear.

In practice, the pairwise distances are estimated from gene data, and it is important to account for the error inherent in these estimates. The GLASS method of [21] uses pairwise distances that are computed as the minimum interspecific coalescence times across all loci and then builds a species tree using simple clustering. The rationale behind this method is that when the number of loci goes to infinity, the minimum interspecies coalescence times across all loci should converge to the speciation times. Here, given multiple loci data, we computed pairwise distances exactly like what GLASS does which is using minimum interspecific coalescence times across all loci. Now suppose we have a node $(u, v)$ with time $t$ in a species tree. Then if the number of loci is large enough, we should see $D(a, b)$ very close to $t$ for all $a \in L(u), b \in L(v)$. To quantify if two numbers are "close", we used some $\varepsilon$ such that if $|x - y| \leq \varepsilon$ we say $x \approx y$. Then for two chosen nodes $X$ and $Y$ whether a speciation or a reticulation event should be considered depends on if $DMax(X, Y, S) - DMin(X, Y, S) \leq \varepsilon$ or not. It is clear that in this case our method would be very sensitive to the value of $\varepsilon$. If $\varepsilon$ is set to be too small, the method will overestimate the number of reticulations; if $\varepsilon$ is set to be too big, the method will underestimate the number of reticulations. When we vary $\varepsilon$ from a very small value gradually to a big one, we can expect the method to return species networks with fewer and fewer reticulations. So we need to set a criterion. Here we say that we want to infer a species network with the minimum number of unreasonably short edges with as few reticulations as possible. This is because when $\varepsilon$ is set to be too small, the overestimated reticulations will produce short edges in the inferred network. On the other hand, when $\varepsilon$ is set to be too big, the underestimation of reticulations will "squashed" the network to satisfy the distance matrix in which case short edges might also be produced. See Fig. 3 for simple illustration of these two cases. In our program, a value $\sigma$ that defines "short" branches needs to be specified as input. Then the program will try $\varepsilon$ equal to 1, 2, . . ., $k$ times of this value respectively and find the optimal network.



| | A | B | C |
|---|---|---|---|
| A | | 1 | 2 |
| B | 1 | | 2.1 |
| C | 2 | 2.1 | |

| | A | B | C |
|---|---|---|---|
| A | | 1 | 1.4 |
| B | 1 | | 1.1 |
| C | 1.1 | 1.4 | |

**Fig. 3.** Two examples of building species networks on different $\varepsilon$. Left: a small value of $\varepsilon$ caused an overestimation of reticulations. Right: a big value of $\varepsilon$ caused an underestimation of reticulations. Both of them result in short branches (of length 0.1) in the inferred network.

The details of our method are shown in Alg.1. It takes a distance matrix $D_L$, a value $\sigma$ that defines short branches, and a value $k$ that sets the values of $\varepsilon$ as we discussed above as input, and returns an inferred phylogenetic network. It reflects the basic idea of our method. In fact, we found that when we tried to find two nodes $X$ and $Y$ in $S$ that have the minimum $DMin(X,Y,S)$, there might be multiple pairs of nodes that share the same minimum value. To address this issue, we kept a stack in the program so that every time there were more than one optimal pair we added a flag in the stack. After a network was built from choosing one of the optimal pairs, the program read the flag on the top of the stack and rolled back to the point where that flag was added and tried another optimal pair, until all optimal pairs were tried. All equally optimal species networks would be returned.

```
Input: A distance matrix $D_L$, $\sigma$, $k$.
Output: A phylogenetic network $N$.
$numCloseNodes \leftarrow 0$;
$numReticulations \leftarrow 0$;
$N \leftarrow NIL$;
for $i = 1$ to $k$ do
    $\varepsilon \leftarrow i \times \sigma$;
    Let $S$ be a set of nodes each labeled by a taxon in $L$ and each node has time 0;
    while $|S| > 1$ do
        Let $X$ and $Y$ be two nodes in $S$ that has the minimum $DMin(X, Y, S)$;
        $t_{min} \leftarrow DMin(X, Y, S)$;
        if $DMax(X, Y, S) - t_{min} \leq \varepsilon$ then
            Remove $X$ and $Y$ from $S$;
            Add a new node $(X, Y)$ with time $t_{min}$ to $S$;
        end
        else
            $(w_x, v_x) \leftarrow \arg\max_{(w,v)}\{|L(v) - L(Y)| : DMin(v, Y, S) = t_{min}, DMax(v, Y, S) - t_{min} \leq \varepsilon, w$ is a descendant of $X\}$;
            if edge $(w_x, v_x)$ does not exist then
                $(w_y, v_y) \leftarrow \arg\max_{(w,v)}\{|L(v) - L(X)| : DMin(X, v, S) = t_{min}, DMax(X, v, S) - t_{min} \leq \varepsilon, w$ is a descendant of $Y\}$;
                if edge $(w_y, v_y)$ does not exist then
                    $(w_x, v_x), (w_y, v_y) \leftarrow \arg\max_{(w_1,v_1),(w_2,v_2)}\{|L(v_1) - L(v_2)| + |L(v_2) - L(v_1)| : DMin(v_1, v_2, S) = t_{min}, DMax(v_1, v_2, S) - t_{min} \leq \varepsilon, w_1$ is a descendant of $X$ and $w_2$ is a descendant of $Y\}$;
                    Add a new node whose children are $u_x$ and $u_y$ with time $t_{min}$ to $S$ where $u_x$ and $u_y$ are newly added nodes on $(w_x, v_x)$ and $(w_y, v_y)$ respectively;
                end
                else
                    Remove $X$ from $S$;
                    Add a new node whose children are $X$ and $u$ with time $t_{min}$ to $S$ where $u$ is a newly added node on $(w_y, v_y)$;
                end
            end
            else
                Remove $Y$ from $S$;
                Add a new node whose children are $u$ and $Y$ with time $t_{min}$ to $S$ where $u$ is a newly added node on $(w_x, v_x)$;
            end
        end
    end
    Let $N'$ be the network that rooted at the only node in $S$;
    Let $c$ be the number of branches of $N'$ whose branch length is less than $\sigma$;
    Let $r$ be the number of reticulations of $N'$;
    if $N = NIL$, or $c < numCloseNodes$, or $c = numCloseNodes$ and $r < numReticulations$
    then
        $N = N'$; $c = numCloseNodes$; $r = numReticulations$;
    end
end
return $N$;
```

**Algorithm 1.** inferNetworkFromDistanceMatrix

### 2.3 Removing Reticulations with Low Support

In our simulation study (see Results section), we found that our method tended to overestimate the number of reticulations, especially when the number loci is small. To address this issue, we employed a heuristics to remove reticulations with low bootstrap support. More specifically, assuming the original data contained $n$ loci, we randomly sampled $n$ loci with replacement and used them as the input of our method to infer a species network. This process was repeated 100 times. Then we removed reticulations of the species network inferred from the original dataset that were not well supported by the 100 species networks obtained from bootstrap. To do so, we first defined a function called **computeBootstrapSupport**, which takes a target species network and a set of bootstrap species networks and returns the target species network $N$ with bootstrap support for every edge. The support of an edge in the species network is calculated as the percentage of that edge present in the bootstrap networks. To see whether one edge in network $N_1$ exists in network $N_2$, we simply computed the hardwired cluster [11] induced from that edge and then check if there is any edge in $N_2$ inducing the same hardwired cluster. The detailed algorithm for removing reticulations of a species network with low support given a set of bootstrap networks and a bootstrap threshold is shown in Alg. 2, where $Support(u, v)$ means the support of edge $(u, v)$.

```
Input: A species network $N$, a set of bootstrap networks $BN$, $threshold$.
Output: a species network $N'$
$N' \leftarrow$ computeBootstrapSupport($N, BN$) ;
Let $numLowSupport$ be the number of edges in $N'$ that has low support;
foreach edge $(u, v)$ visited when post-traversing $N'$ do
    if $Support(u, v) < threshold$ then
        foreach child node $w$ of $v$ that is also a reticulation node do
            $N'' \leftarrow N'$;
            Remove reticulation edge $(v, w)$ of $N''$;
            $N'' \leftarrow$ computeBootstrapSupport($N'', BN$) ;
            Let $tnls$ be the number of edges in $N''$ that has low support;
            if $numLowSupport > tnls$ then
                return removeLowSuportEdges($N'', BN, threshold$);
            end
        end
    end
end
return $N'$;
```

**Algorithm 2.** removeLowSuportEdges

# 3 Results

We used synthetic datasets to test the performance of our method. We first generated 2 datasets, each consisting of 100 random species trees with 10 taxa of height 8 and 20 taxa of height 16 respectively using PhyloGen [25]. The height of tree is the total branch lengths from the root of the tree to any of its leaf. Then for each species tree, we randomly added 1, 2, 3, 4 and 5 reticulations respectively. To add a reticulation to a species network, we randomly chose two edges in the network and add an edge between their midpoints from the higher one to the lower one. Then the lower one became a new reticulation node and we randomly assigned an inheritance probability from $0.1$ to $0.9$. Within the branches of each species network we simulated 25, 50, 100, 200, 500, 1000, 2000 and 5000 gene trees respectively using program ms [10].

We run our method on these gene trees and compared the inferred species networks with true ones using hardwired cluster distance [11]. Note that in all simulations, we set parameters as $\sigma = 0.1$ and $k = 5$ (see Alg. 1). $\sigma$ was set to be $0.1$ because it is a good threshold of "short" branch when branch lengths are in coalescent units. We also tried different values and found that varying it slightly did not have much affect on results. For the setting of $k$, we found that in our simulations most optimal species networks were found at $\varepsilon = 2\sigma$ or $\varepsilon = 3\sigma$, and setting $k$ to be more than 5 would only change the results very slightly. The result is shown in Fig. 4. Note that when multiple equally optimal networks were returned, the average distance of those tie networks was calculated. We can see that overall our method made very accurate inferences. As expected, for both datasets, the error of the inferred networks increased slightly with the number of reticulations, because for the same number of taxa increasing the number of reticulations made the inference problem harder. Also, for both datasets, as the number of gene trees increased, the accuracy of the inferred networks increased. When comparing the results from the two datasets, we can see that the 20-taxon dataset actually produced slightly better result. This is because for the same number of reticulations the reticulations are expected to be more independent from each other on a network with more taxa, which makes the inference problem easier.

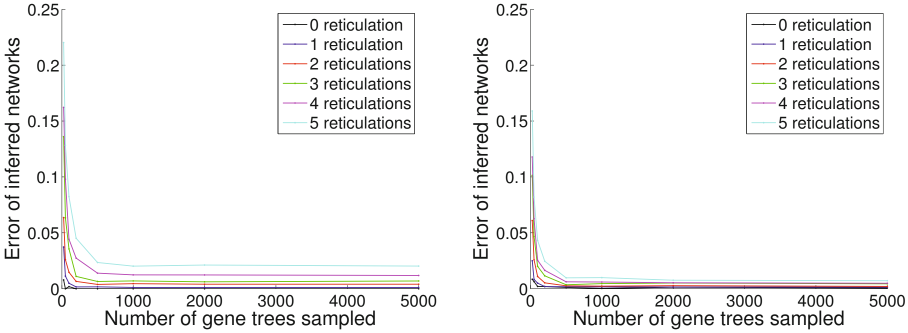


**Fig. 4.** Accuracy of the method using true gene trees. Results of the 10-taxon dataset and the 20-taxon dataset are shown in the left and right panels, respectively. The errors of the inferred networks were computed using hardwired cluster distance [11]. The results were averaged over 100 repetitions.

In order to test the robustness of our method to error in pairwise distance estimates, we synthetically perturbed the true distances. More specifically, the pairwise distances obtained above underwent 5 different perturbation experiments $i = 1, 2, 3, 4, 5$: In experiment $i$, each pairwise distance was multiplied by a (uniformly distributed) random number in the range $[1, 1 + i\epsilon]$ for $\epsilon = 0.1$. For example, in the results, the "$30\%$ error" data sets were obtained by multiplying each pairwise distance by a random number in $[1, 1.3]$ (each pairwise distance was multiplied by a potentially different number). The inference method was then applied to the perturbed data sets. The results of using these perturbed pairwise distances are shown in Fig. 5. We can see that overall our method still produced accurate results. As expected, on the same dataset, the accuracy of the inferred network decreased as the value of $i\epsilon$ increased. Further, the effect of the distance error on the network accuracy decreased with increasing the number of gene trees. It is important to note that the error has more impact on the "harder" datasets, that is, the ones with more reticulation nodes.

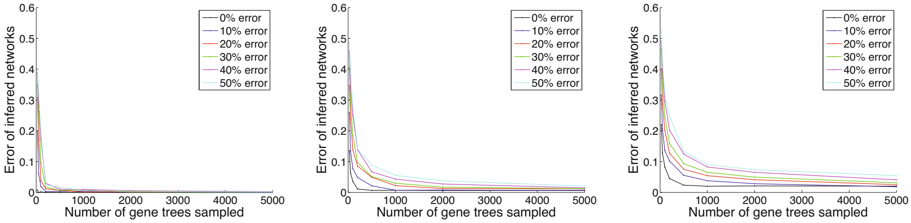


**Fig. 5.** Accuracy of the method using perturbed pairwise distances on 10-taxon dataset. Results of datasets containing true species networks with 0, 3 and 5 reticulations are shown from left to right columns, respectively. The errors of the inferred networks were computed using hardwired cluster distance [11]. The results were averaged over 100 repetitions.

We also examined the number of reticulations in the inferred species networks; see Fig. 6. As the results show, the estimates of the numbers of reticulations tend to the true values as the number of loci increases. However, when the number of loci is small, our method overestimates the number of reticulations, especially for datasets with high error values. To address this issue, we used a heuristics to remove reticulations that result in edges with low bootstrap support (see the Methods section).

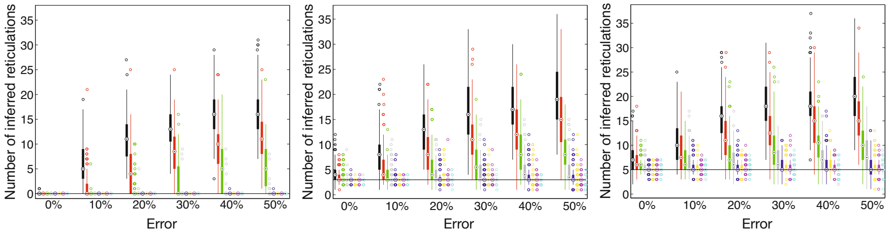


**Fig. 6.** The number of reticulations in inferred species networks of 10-taxon dataset. Results of datasets containing true species networks with 0, 3 and 5 reticulations are shown from left to right columns, respectively. In each subfigure, boxes from left to right (from black to cyan) in each group corresponds to datasets consisting of 25, 50, 100, 200, 500, 1000, 2000 and 5000 gene trees respectively. The solid horizontal black line in each subfigure indicates the true number of reticulations.

In Fig. 7, we show results based on the "hardest" dataset where the true species networks contain 10 taxa and 5 reticulations and the pairwise distances of taxa from gene trees were randomly perturbed by at most 50%. When multiple equally optimal species networks were returned, we chose a random one to which to apply the heuristic.

We varied the bootstrap threshold by using values 70, 80 and 90. As the results show, the heuristics successfully reduced the number of reticulations in the inferred species networks, especially for datasets with a small number of loci. For datasets with 25 gene trees, the mean number of reticulations was reduced from 15 to 7 when a bootstrap threshold of 70 was used. As expected, when a larger bootstrap threshold was used, the inferred species networks had fewer reticulations. On the other hand, the accuracy of the inferred species networks increased after reducing the number of reticulations.

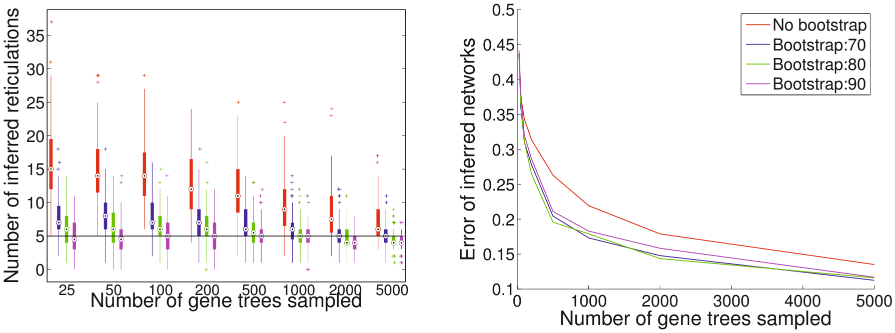


**Fig. 7.** Results of using heuristics to remove reticulations that result in edges with low bootstrap support based on 10-taxon and 5-reticulation datasets where distance matrices of gene trees were randomly perturbed by at most 50%. Left: the number of reticulations in the inferred species networks. The solid horizontal black line indicates the true number of reticulations. Right: the error of the inferred species networks.

In terms of the running time, for the largest and most complex dataset (20 taxa, 5 reticulations and 5000 gene trees), the program took, on average, around 3 minutes to complete the inference. For most of the datasets with 10 taxa, 5 reticulations and 5000 gene trees, the program finished in 10 seconds or less.

## 4 Conclusions

In this paper, we proposed a simple, yet effective distance-based method for inferring phylogenetic networks from pairwise distances in the presence of incomplete lineage sorting. Our method is a simple extension of the GLASS method [21]. It is important to note, though, that while GLASS has theoretical guarantees (the authors proved its statistical consistency), our method makes heuristic decisions and currently lack any theoretical guarantees. However, our simulation study demonstrate the method can obtain very good results, even when noise is added to the distance estimates. In practice, distance-based methods in general suffer from the lack of accurate methods for estimating pairwise distances. As the amount of molecular sequence data increases and more sophisticated methods are developed for more accurate estimates of pairwise distances, the application of distance-based methods would become more common, particularly for large data sets. Nevertheless, the speed of these methods make them appealing for rapid generation of a relatively accurate network to initialize the search of a more accurate, and computationally intensive method, such as maximum likelihood or Bayesian inference.

## References

1. Arnold, M.L.: Natural Hybridization and Evolution. Oxford University Press, Oxford (1997)
2. Barton, N.H.: The role of hybridization in evolution. Molecular Ecology 10(3), 551–568 (2001)
3. The Heliconius Genome Consortium: Butterfly genome reveals promiscuous exchange of mimicry adaptations among species. Nature 487(7405), 94–98 (2012)
4. Cranston, K.A., Hurwitz, B., Ware, D., Stein, L., Wing, R.A.: Species trees from highly incongruent gene trees in rice. Syst. Biol. 58, 489–500 (2009)
5. Degnan, J.H., Rosenberg, N.A.: Gene tree discordance, phylogenetic inference and the multispecies coalescent. Trends Ecol. Evol. 24(6), 332–340 (2009)
6. Eriksson, A., Manica, A.: Effect of ancient population structure on the degree of polymorphism shared between modern human populations and ancient hominins. Proceedings of the National Academy of Sciences 109(35), 13956–13960 (2012)
7. Green, R.E., Krause, J., Briggs, A.W., Maricic, T., Stenzel, U., Kircher, M., Patterson, N., Li, H., Zhai, W., Fritz, M.H.-Y., Hansen, N.F., Durand, E.Y., Malaspinas, A.-S., Jensen, J.D., Marques-Bonet, T., Alkan, C., Prfer, K., Meyer, M., Burbano, H.A., Good, J.M., Schultz, R., Aximu-Petri, A., Butthof, A., Hber, B., Hffner, B., Siegemund, M., Weihmann, A., Nusbaum, C., Lander, E.S., Russ, C., Novod, N., Affourtit, J., Egholm, M., Verna, C., Rudan, P., Brajkovic, D., Kucan, E., Guic, I., Doronichev, V.B., Golovanova, L.V., Lalueza-Fox, C., de la Rasilla, M., Fortea, J., Rosas, A., Schmitz, R.W., Johnson, P.L.F., Eichler, E.E., Falush, D., Birney, E., Mullikin, J.C., Slatkin, M., Nielsen, R., Kelso, J., Lachmann, M., Reich, D., Pbo, S.: A draft sequence of the Neandertal genome. Science 328(5979), 710–722 (2010)
8. Hobolth, A., Dutheil, J., Hawks, J., Schierup, M., Mailund, T.: Incomplete lineage sorting patterns among human, chimpanzee, and orangutan suggest recent orangutan speciation and widespread selection. Genome Research 21, 349–356 (2011)
9. Holland, B.R., Benthin, S., Lockhart, P.J., Moulton, V., Huber, K.T.: Using supernetworks to distinguish hybridization from lineage-sorting. BMC Evol. Biol. 8, 202 (2008)
10. Hudson, R.R.: Generating samples under a Wright-Fisher neutral model of genetic variation. Bioinformatics 18, 337–338 (2002)
11. Huson, D.H., Rupp, R., Scornavacca, C.: Phylogenetic Networks: Concepts, Algorithms and Applications. Cambridge University Press, New York (2010)
12. Joly, S., McLenachan, P.A., Lockhart, P.J.: A statistical approach for distinguishing hybridization and incomplete lineage sorting. Am. Nat. 174(2), E54–E70 (2009)
13. Kubatko, L.S.: Identifying hybridization events in the presence of coalescence via model selection. Syst. Biol. 58(5), 478–488 (2009)
14. Kuo, C.-H., Wares, J.P., Kissinger, J.C.: The Apicomplexan whole-genome phylogeny: An analysis of incongruence among gene trees. Mol. Biol. Evol. 25(12), 2689–2698 (2008)
15. Liu, L., Yu, L.L., Kubatko, L., Pearl, D.K., Edwards, S.V.: Coalescent methods for estimating phylogenetic trees. Mol. Phylogenet. Evol. 53, 320–328 (2009)
16. Maddison, W.P.: Gene trees in species trees. Syst. Biol. 46(3), 523–536 (1997)
17. Mallet, J.: Hybridization as an invasion of the genome. Trends Ecol. Evol. 20(5), 229–237 (2005)
18. Mallet, J.: Hybrid speciation. Nature 446, 279–283 (2007)
19. Meng, C., Kubatko, L.S.: Detecting hybrid speciation in the presence of incomplete lineage sorting using gene tree incongruence: A model. Theor. Popul. Biol. 75(1), 35–45 (2009)
20. Moody, M.L., Rieseberg, L.H.: Sorting through the chaff, nDNA gene trees for phylogenetic inference and hybrid identification of annual sunflowers (Helianthus sect Helianthus). Molecular Phylogenetics And Evolution 64, 145–155 (2012)

21. Mossel, E., Roch, S.: Incomplete lineage sorting: consistent phylogeny estimation from multiple loci. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 7(1), 166–171 (2010)
22. Nakhleh, L.: Evolutionary phylogenetic networks: models and issues. In: Heath, L., Ramakrishnan, N. (eds.) The Problem Solving Handbook for Computational Biology and Bioinformatics, pp. 125–158. Springer, New York (2010)
23. Nakhleh, L.: Computational approaches to species phylogeny inference and gene tree reconciliation. Trends in Ecology & Evolution 28(12), 719–728 (2013)
24. Pollard, D.A., Iyer, V.N., Moses, A.M., Eisen, M.B.: Widespread discordance of gene trees with species tree in Drosophila: evidence for incomplete lineage sorting. PLoS Genet. 2(10), e173 (2006)
25. Rambaut, A.: Phylogen v1.1 (2012), http://tree.bio.ed.ac.uk/software/phylogen/
26. Rannala, B., Yang, Z.: Phylogenetic inference using whole genomes. Annu. Rev. Genomics Hum. Genet. 9, 217–231 (2008)
27. Rieseberg, L.H.: Hybrid origins of plant species. Annu. Rev. Ecol. Syst. 28, 359–389 (1997)
28. Staubach, F., Lorenc, A., Messer, P.W., Tang, K., Petrov, D.A., Tautz, D.: Genome patterns of selection and introgression of haplotypes in natural populations of the house mouse (mus musculus). PLoS Genet. 8(8), e1002891 (2012)
29. Syring, J., Willyard, A., Cronn, R., Liston, A.: Evolutionary relationships among Pinus (Pinaceae) subsections inferred from multiple low-copy nuclear loci. Am. J. Bot. 92, 2086–2100 (2005)
30. Takuno, S., Kado, T., Sugino, R.P., Nakhleh, L., Innan, H.: Population genomics in bacteria: A case study of staphylococcus aureus. Molecular Biology and Evolution 29(2), 797–809 (2012)
31. Than, C., Ruths, D., Innan, H., Nakhleh, L.: Confounding factors in HGT detection: statistical error, coalescent effects, and multiple solutions. J. Comput. Biol. 14, 517–535 (2007)
32. Than, C., Sugino, R., Innan, H., Nakhleh, L.: Efficient inference of bacterial strain trees from genome-scale multi-locus data. Bioinformatics 24, i123–i131 (2008)
33. White, M.A., Ane, C., Dewey, C.N., Larget, B.R., Payseur, B.A.: Fine-scale phylogenetic discordance across the house mouse genome. PLoS Genetics 5, e1000729 (2009)
34. Yu, Y., Barnett, R.M., Nakhleh, L.: Parsimonious inference of hybridization in the presence of incomplete lineage sorting. Systematic Biology 62, 738–751 (2013)
35. Yu, Y., Degnan, J.H., Nakhleh, L.: The probability of a gene tree topology within a phylogenetic network with applications to hybridization detection. PLoS Genetics 8, e1002660 (2012)
36. Yu, Y., Dong, J., Liu, K., Nakhleh, L.: Maximum likelihood inference of reticulate evolutionary histories. Proceedings of the National Academy of Sciences 111, 16448–16453 (2014)
37. Yu, Y., Ristic, N., Nakhleh, L.: Fast algorithms and heuristics for phylogenomics under ils and hybridization. BMC Bioinformatics 14, S6 (2013)
38. Yu, Y., Than, C., Degnan, J.H., Nakhleh, L.: Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting. Systematic Biology 60, 138–149 (2011)