Network-thinking: novel inference tools and scalability challenges

Claudia Solis-Lemus

Contents

1 Introduction: the impact of gene flow

Typically, phylogenetic analyses estimate a species tree to represent the evolutionary relationships among a group of organisms. A species tree implicitly assumes the absence of reticulate evolution, and this assumption is violated when there is some form of gene flow, such as horizontal gene transfer, hybridization, hybrid speciation or introgression. In recent years, there has been an explosion of phylogenetic network methods that can represent reticulate evolution by estimating species networks, instead of species trees. However, there is still the belief that tree methods will be able to estimate the "major vertical signal" of the data, and network methods remain under-utilized in the field. It turns out that this belief can be erroneous in some real-life datasets, which motivates the development of more accurate and more scalable network methods for phylogenetic analyses.

Just as incomplete lineage sorting (ILS) creates a pattern in the sample of gene trees that is poorly represented with concatenation methods [32, 44], gene flow also causes gene tree discordance that is not captured by coalescent-based tree methods alone. In the first case, concatenation methods assume that all genes have the same phylogeny. Given that ILS can cause gene tree discordance, especially for species trees with short internal branch lengths, it has been proven that concatenation methods are not robust to the presence of ILS, and thus, can estimate the wrong phylogeny with high support [32, 44].

Coalescent-based tree methods such as ASTRAL [37, 58] and NJst [33] are a huge improvement over concatenation by explicitly modeling ILS via the multispecies coalescent model [29]. However, these methods are still limited under reticulate evolution as they do not account for gene flow. Indeed, it has been proven that when gene trees are simulated under some form of gene flow, tree methods can estimate the wrong species tree, with high support [48]. Thus, under certain patterns of gene flow, tree methods cannot recover the main "vertical signal" of the data.

The rationale for the lack of robustness of tree methods to gene flow lies in the existence of anomalous gene trees (AGT): gene trees that do not have the same topology as the species tree and have a higher probability under the coalescent model than gene trees that have the same topology as the species tree. These anomalous gene trees can be unrooted (AUGT) or rooted (AGT). Degnan and others [19, 18] shows that there are no AUGT for the case of 4 taxa for gene trees generated from a species tree with no gene flow under the coalescent tree model. This result leads to the accuracy of quartet-based methods, like ASTRAL, as quartets with higher frequencies in the sample could not be anomalous and then would serve to reconstruct the correct species tree. Under the ILS+gene flow scenario, however, this is not the case anymore. Solis-Lemus and others [48] showed that there are AUGT for the case of 4 taxa under the network coalescent model, and thus, species tree methods could reconstruct the wrong species tree by being misled by anomalous gene trees in the sample (see [35] for similar study under a model of continuous gene flow between sister species).

For example, in Figure 1 we see a 4-taxon network with one gene flow event shown with a blue arrow (more about networks description in Section 3). This gene flow event has a γ parameter associated with it that represents the proportion of genes that were transferred through this reticulation arrow. If $\gamma = 0$, then there is no gene flow and thus, it would be as if the blue arrow did not exist, and the figure would represent a tree, not a network. In fact, by ignoring this blue arrow, we get the "major tree" in black (more about displayed trees on a network in Section 3.3). It turns out that if we select very short lengths for the ancestral branches to (A,B) ($t_1 = t_2 = 0.01$ in coalescent units) and we simulate gene trees under the coalescent model on this network (ILS + gene flow), there will be some discordant gene trees (different from the major tree in black) with higher probability under the coalescent model than the gene trees that agree with the major tree. Intuitively, one might think that all gene trees should have the (A,B) clade because the gene flow is ancestral to the speciation event (see the two displayed trees in black in Figure 1 next to the network; both trees have the (A,B) clade). However, as we know from the coalescent model (**chapter X**), there is random coalescence in the populations ancestral to the (A,B) clade which is more variable as the internal branch shortens. If, in addition, there is gene flow in this branch, the majority of the simulated gene trees no longer agree with the species tree topology. In fact, as γ increases (table in Figure 1 right), the probability of observing the clade (A,B) in the unrooted gene trees is smaller than the probability of the other two possibilities: (A,C), and (A,D). Thus, the discordant trees (without (A,B) clade) appear in the sample with higher frequency than the gene tree in agreement with the species tree (with (A,B) clade). This results in a sample with a strong discordant signal that deceives tree methods into reconstructing a species tree without the (A,B) clade.

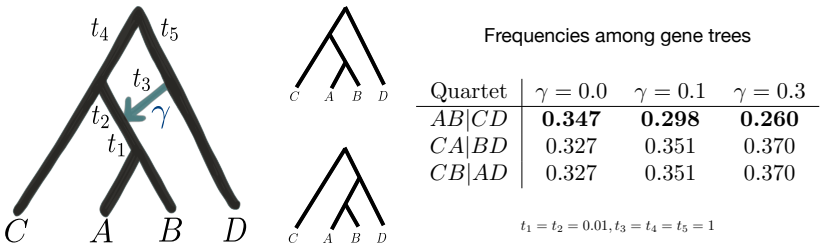

Figure 1: Anomalous gene trees under a network model. Left: Network model with one hybridization event (blue arrow). Center: By turning on/off the hybrid edge (blue arrow), we have two displayed trees, both containing clade (A,B). Right: The discordant gene trees with clade (A,C) or (A,D) have higher frequency than the gene tree that has the same clade as the species tree (A,B) for $\gamma > 0$ (gene flow) and short internal branches ($t_1 = t_2 = 0.01$ in coalescent units).

In conclusion, tree methods are not suited to handle gene flow. So, when there is

the possibility of reticulate evolution among the organisms under study, network methods should be used instead. However, this raises the question: when is this the case? That is, when is a tree model good enough to describe the sample of gene trees at hand, and when do we need to resort to network models instead?

In the next section, we will describe a goodness of fit test that will allow us to identify if a tree model is a good fit for the data at hand. Later, we will also describe what we mean by a species network (Section 3), and several difficulties in network thinking such as the concept of displayed trees (Section 3.3), and the difficulty in comparing different networks (Section 3.4). Finally, we will explain a scalable method to reconstruct phylogenetic networks (Section 4) as well as examples on how to use the software tool PhyloNetworks (Section 5).

2 Is a tree sufficient or do we need a network?

If we want to estimate the evolutionary relationships among a group of organisms, we can use a coalescent-based method which takes a sample of gene trees as input. A reasonable question is when to choose methods that estimate a species tree and when to choose methods that estimate a species network. Basically, we need a way to know if the data at hand follows a tree-like pattern of evolution (and thus, we can use a tree method), or if the data follows a non-tree-like pattern (and thus, we need a network method). There exist already multiple tests that allow us to identify whether there is hybridization within a given group of taxa. For example, the ABBA-BABA test for SNPs [24, 20] tries to identify if there is hybridization within a set of 4 taxa, HyDe [30, 12] tests if there is a potential hybridization among 3 taxa plus an outgroup using phylogenetic invariants, and MSCQuartets [38, 42] tests if the quartet frequencies of a 4-taxon subset follow the expected probabilities under the multispecies coalescent model. Despite the accuracy of these tests, the disadvantage is that they test one subset of taxa at a time.

Unlike, ABBA-BABA and HyDe, TICR (Tree Incongruence Checking in R) [50] is a goodness of fit test which combines all 4-taxon sets into a single test. TICR tests whether a specific species tree is a good enough fit to the data at hand by comparing the expected concordance factors under the coalescent tree model with the observed concordance factors from the sample of gene trees. The concordance factor (CF) of a given quartet (or split) is the proportion of genes whose true tree displays that quartet (or split) [9]. For example, if A is a hybrid intermediate between B and C, the CFs of $AB|CD$ and $AC|BD$ would be around 0.5 while the CF of $AD|BC$ would be near 0. On the contrary, if there is no hybridization among A, B, C, D, and the species tree has the split $AB|CD$, we would expect the two discordant splits: $AC|BD$ and $AD|BC$ to have equally minor CF, and the major split $AB|CD$ to have a higher CF than the other two (see Figure 2 top). These CFs depend on the branch lengths in the species tree, represented in coalescent units. We use the term 'CF' as opposed to 'probability' to emphasize that CFs measure genomic support, unlike probabilities (such as

posterior probabilities or bootstrap values) that most often measure statistical uncertainty [5].

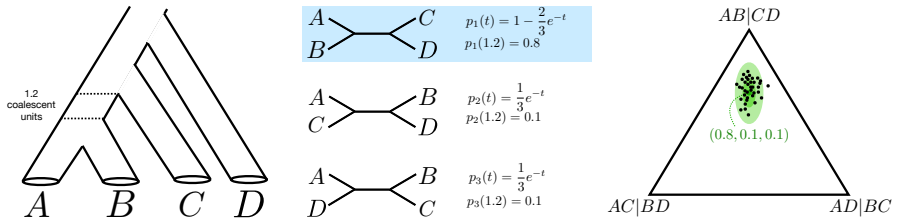

Figure 2: Left: 4-taxon species tree with internal branch length of 1.2 coalescent units. Center: Under the coalescent model, the expected CFs for the three quartets $(p_1(t), p_2(t), p_3(t))$ depend on the internal branch length. For $t = 1.2$, the expected CF for the major split $AB|CD$ is 0.8, and the expected CF for the minor splits $AC|BD, AD|BC$ are equal to 0.1. Right: Dirichlet distribution centered on the expected CFs $(0.8, 0.1, 0.1)$. Under the coalescent model, the two minor CF are expected to be equal. This equality is disrupted by gene flow.

The TICR goodness of fit test requires a given species tree with branch lengths in coalescent units, which can be estimated with the average CF of all quartets that map onto a given internal branch, converted to a branch length (in coalescent units) by inverting the equation in Figure 2: $t = -\log((3/2)(1 - CF_{major}))$.

With this metric tree, we calculate the expected CFs under the multispecies coalescent model for every 4-taxon set. For example, in Figure 2, the expected CFs: $(p_{AB|CD}(t), p_{AC|BD}(t), p_{AD|BC}(t)) = (1 - (2/3)\exp(-t), (1/3)\exp(-t), (1/3)\exp(-t))$ for any $t > 0$. Next, for each 4-taxon set A, B, C, D, the observed quartet CFs $(x_{AB|CD}, x_{AC|BD}, x_{AD|BC})$ are modeled by a Dirichlet distribution $D(\alpha_1, \alpha_2, \alpha_3)$, with parameters α_i that depend on the expected CFs under the multispecies coalescent model. To simplify notation, we will use the subscripts $1, 2, 3$ instead of $AB|CD, AC|BD, AD|BC$. For example, the multispecies coalescent model implies that the expected CF for the major quartet is $p_1(t) = \alpha_1/(\alpha_1 + \alpha_2 + \alpha_3) = 1 - (2/3)\exp(-t)$, if t is the internal branch length on the quartet tree. Similarly, the expected CF for the minor quartets is given by $p_2(t) = p_3(t) = (1 - p_1)/2 = (1/3)\exp(-t)$. The parameters α_i can thus be obtained from t.

All four-taxon sets are assumed to share the same concentration parameter $\alpha = \alpha_1 + \alpha_2 + \alpha_3$, and we can estimate this parameter by maximizing the pseudo-log-likelihood from the Dirichlet distribution assumed for the observed

quartet CFs, which has the form:

$$\begin{aligned}\log PL(\alpha) =& \sum_{j=1}^{M} \log\Gamma(\alpha) - \log\Gamma(\alpha p_1(t_j)) - 2\log\Gamma(\alpha p_2(t_j)) + (\alpha p_1(t_j) - 1)\log x_{1j} + \\ & (\alpha p_2(t_j) - 1)(\log x_{2j} + \log x_{3j}) \\ =& M\log\Gamma(\alpha) - \sum_{j=1}^{M}(\log\Gamma(\alpha p_1(t_j)) + 2\log\Gamma(\alpha p_2(t_j))) + \\ & \alpha M\widetilde{\log CF} - 3M\overline{\log CF},\end{aligned}$$

where t_j is the internal branch lengths of the tree reduced to the j^{th} 4-taxon set, $\widetilde{\log CF}$ is the average of the observed log CFs, weighted by their $p_i(t_j)$ values, (x_{1j}, x_{2j}, x_{3j}) are the observed CFs for the j^{th} 4-taxon set, and $\overline{\log CF}$ denotes the average log CF. Note that this is a pseudo-likelihood (not a true likelihood) because we are treating the M 4-taxon sets as independent (by summing over the log probabilities), when they are not as all the 4-taxon set are linked through the tree topology.

Using the estimated value of α, we can calculate a p-value for each 4-taxon set, based on how much the observed CF for the major quartet, x_1, departed from its expectation, p_1. Let d denote the distance between the expected and the observed CF: $d = |x_1 - p_1|$, the p-value for this 4-taxon set is then calculated as $p = P_\alpha(|X_1 - p_1| \geq d)$ from a Beta distribution with parameter $(\alpha p_1, \alpha(1-p_1))$.

After calculating a p-value for each of the 4-taxon sets, we can bin all these p-values into arbitrary categories to test if the bin frequencies depart from the expected proportions under the null hypothesis. We choose the categories as: $0-0.01, 0.01-0.05, 0.05-0.10, 0.1-1.0$. The overall test is then defined using a χ^2 test with 3 degrees of freedom to determine if the bin frequencies depart from the expected proportions $(0.01, 0.04, 0.05, 0.90)$. The null hypothesis is whether the given species tree used to compute the expected CF is a good enough fit to the observed CF. If this hypothesis is rejected by the overall p-value, then we would prefer to use a network method as there is evidence of reticulate evolution which does not follow the pattern of the coalescent model on a tree.

It is important to note that the TICR test has simplifying assumptions, but most of these assumptions have been explored in literature for their robustness in real applications. For example, while the pseudolikelihood function ignores the dependency of the quartets, this model has been explored to be robust in different studies [50, 34, 47, 57]. For more details about this and other assumptions, check out the TICR paper [50].

3 What is a species network?

Just like phylogenetic trees, networks can be rooted or unrooted. A rooted phylogenetic network on taxon set X is a connected directed acyclic graph with

vertices $V = \{r\} \cup V_L \cup V_H \cup V_T$, edges $E = E_H \cup E_T$ and a bijective leaf-labeling function $f : V_L \to X$ with the following characteristics. The root r has indegree 0 and outdegree 2. Any leaf $v \in V_L$ has indegree 1 and outdegree 0. Any tree node $v \in V_T$ has indegree 1 and outdegree 2. Any hybrid node $v \in V_H$ has indegree 2 and outdegree 1. A tree edge $e \in E_T$ is an edge whose child is a tree node (or a leaf node). A hybrid edge $e \in E_H$ is an edge whose child is a hybrid node. Unrooted phylogenetic networks are typically obtained by suppressing the root node and the direction of all edges. We also consider semi-directed unrooted networks (Figure 3 center), where the root node is suppressed and we ignore the direction of all tree edges, but we maintain the direction of hybrid edges, thus keeping information on which nodes are hybrids. The placement of the root is then constrained, because the direction of the two hybrid edges to a given hybrid node inform the direction of time at this node: the child edge must be a tree edge directed away from the hybrid node and leading to all the hybrid's descendants. Therefore the root cannot be placed on any descendant of any hybrid node, although it might be placed on some hybrid edges.

Throughout this chapter, we use the following notation

- n = the number of taxa,
- h = the number of hybridization events (in orange arrows in Figure 3) and
- k_i = the number of nodes in the undirected cycle created by the i^{th} hybrid node.

For example, Figure 3 shows a semi-directed network (center) with two possible rootings (left and right). In this network, there are 7 taxa ($n = 7$), and two hybridization events marked by the orange arrows ($h = 2$). One of the hybridization events (corresponding to γ_1) has three nodes in the cycle ($k_1 = 3$) and the second hybridization event (corresponding to γ_2) has four nodes in the cycle ($k_2 = 4$). The number of nodes inside each hybridization cycle will be important when we determine which hybridizations are identifiable, and which are not (Section 4.1.1).

The main parameter of interest is the topology $\mathcal{N}$ of the semi-directed network, which can later be rooted by a known outgroup just as a tree can. The other parameters of interest are $\mathbf{t}$, the vector of branch lengths in coalescent units, and a vector of inheritance probabilities γ, describing the proportion of genes inherited by a hybrid node from one of its hybrid parents (see Figure 3). Each hybrid edge has a $\gamma < 1$ associated with it, and we distinguish between major hybrid edge (with $\gamma > 0.5$) and minor hybrid edge (with $\gamma < 0.5$). Only identifiable branch lengths are considered in $\mathbf{t}$. For example, with only one sequenced individual per taxon, the lengths of external edges are not identifiable because branch lengths can only be estimated if there is the possibility of a coalescent event in the branch (not possible with only one taxon).

There are different classes of networks depending their complexity, that is, how far they are from a tree. In this chapter, we will assume that the true network is of level-1 [28] (see Figure 4), i.e. any given edge can be part of at most one

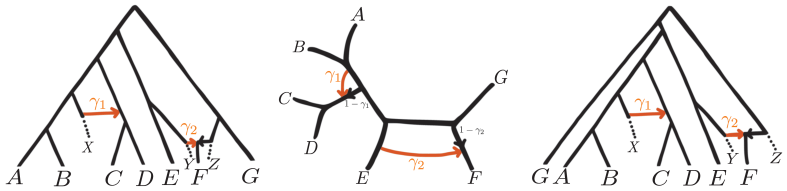

Figure 3: Example of rooted and semi-directed phylogenetic networks with $h = 2$ hybridization events and $n = 7$ sampled taxa. Inheritance probabilities γ represent the proportion of genes contributed by each parental population to a given hybrid node. Left: rooted network modeling several biological processes. Taxon F is a hybrid between two non-sampled taxa Y and Z with $\gamma_2 \approx 0.50$, and the lineage ancestral to taxa C and D has received genes introgressed from a non-sampled taxon X, for which $\gamma_1 \approx 0.10$. An alternative process at this event could be the horizontal transfer of only a handful of genes, corresponding to a very small fraction $\gamma_1 \approx 0.001$. Center: semi-directed network for the biological scenario just described. Although the root location is unknown, its position is constrained by the direction of hybrid edges (directed by arrows). For example, C, D or F cannot be outgroups. Right: rooted network obtained from the semi-directed network (center) by placing the root on the hybrid edge that leads to taxon F (labeled by $1 - \gamma_2$).

cycle. This means that there is no overlap between any two cycles. This is a huge limitation, and it is not entirely biologically reasonable. However, as we will see in the remainder of the chapter (in particular, Section 4.1.1), level-1 networks are already quite complex and it is not straight-forward to identify which reticulations can be detected and which cannot given the sample of gene trees. More research is needed in the area of identifiability of phylogenetic networks in order to relax the level-1 assumption and allow for more complex networks (refer to [28] for other types of evolutionary networks such as level-k or tree-child networks).

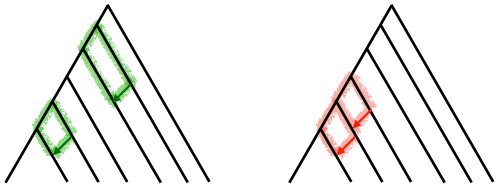

Figure 4: Left: Level-1 network in which the hybridization cycles do not intersect. Right: Level-2 network in which the hybridization cycles intersect.

We also note that the network model uses one-time events to summarize episodes of continuous gene flow. Furthermore, the network model does not say anything about what biological process – introgression, hybridization, etcetera – is at work

as all of these are modeled with the same network structure. However, in some cases, γ can provide some insight, for example, if $\gamma \approx 0.5$, we can suspect that a process of hybridization occurred.

Finally, visual artifacts can mislead the interpretation. For example, in Figure 5, all three networks represent the same reticulation event, but they would appear as different processes by the way they are drawn. That is, the network on the left has two perfectly horizontal edges coming together into the hybrid node (ancestor of blue clade) which seem to illustrate a hybridization. This is not the same interpretation for the network in the center, which only displays one green arrow with $\gamma = 0.2$ that flows into the black edge ancestral to the blue clade. This green arrow seems to represent gene flow or horizontal gene transfer (HGT) from the ancestral population of the pink clade to the ancestral population of the blue clade. Finally, the network on the right displays the edge with $\gamma = 0.2$ as the major edge (in black), so it seems as if there is gene flow (e.g. HGT) from the ancestral population of the yellow clade into the ancestral population of the blue clade. Mathematically, all these three networks are the same, suggesting a reticulate clade (blue) with 80% contribution from the ancestral population of yellow clade and 20% contribution from the ancestral population of pink clade.

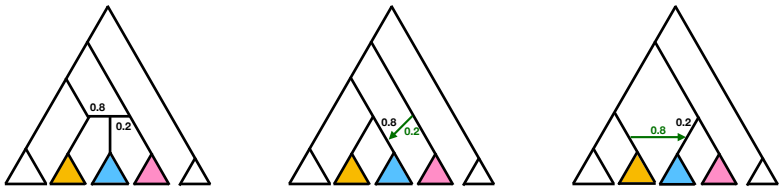

Figure 5: Visual artifacts can mislead the interpretation.

3.1 Explicit vs. implicit networks

A huge variety of phylogenetic networks have been proposed (see [28]), but we can categorize them into two main classes: explicit networks and implicit networks (or split networks). In explicit networks, each internal node is associated to a specific biological process (like speciation or hybridization), whereas in implicit networks, internal nodes need not correspond to any biological mechanism or ancestral population.

Thus, explicit networks can be easily interpreted as phylogenetic trees are. For example, in Figure 6 (left), there is a gene flow event represented by the green arrow from ancestral population of the bright yellow fish into the blue fish. As explained already in previous sections, this gene flow event has an extra parameter associated with it (inheritance probability γ), which represents the proportion of genes that were transferred through this arrow (17% in this case). There is also a time progression from root to tips, with the root and every internal node representing ancestral populations. On the other hand, in the

implicit network (Figure 6 right), the internal nodes do not represent ancestral species anymore, and the edges are not directed, so we lose the time progression from past to present. Furthermore, the repeated parallel edges do not represent any particular form of gene flow. They simply represent gene tree discordance, which can well be ILS or gene flow, or even estimation error.

Many methods have been proposed to reconstruct implicit networks [22, 25, 53], which are a great way to summarize the data and visualize patterns of discordance. However, they are not as clear when we want to study the evolutionary relationships of organisms. Methods to reconstruct explicit networks are booming [56, 48, 52, 59, 3], but they still cannot handle the sizes of data that split networks can. Explicit network methods need more data and more computational time, but in return, they produce a species network that can be interpreted in biological terms.

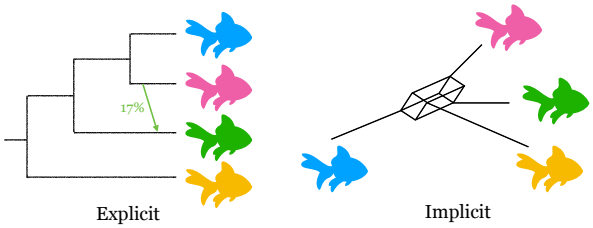

Figure 6: Explicit vs. implicit networks: in explicit networks, internal nodes represents specific biological processes like speciation or hybridization. In implicit network, internal nodes do not have any meaning.

3.2 Extended parenthetical format

To represent network objects, we use the extended Newick format [39, 17, 51] This format uses the concept of minor hybrid edges (edges with $\gamma < 0.5$) and major hybrid edges ($\gamma > 0.5$). By default, we detach the minor hybrid edge at each hybrid node to write the extended Newick description of a network as we would for a tree, with a repeated label, that of the hybrid node ('#H1' in Figure 7). This description can include edge information, formatted as :length:support:γ.

For example, the parenthetical format of the network in Figure 7 can include γ values:

```
(((A,(B)#H1:::0.8),(C,#H1:::0.2)),D);,
```

which are written after three colons, because there is no information about branch length nor support for the hybrid edges. Other internal edges (tree edges) have information about branch lengths, which follow just one colon. These tree edges do not have information about γ, because all tree edges have $\gamma = 1$.

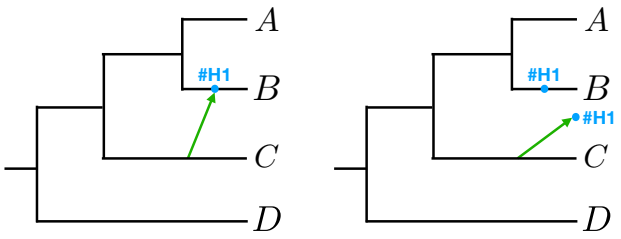

Figure 7: Extended Newick format for networks: (((A,(B)#H1),(C,#H1)),D); The network is written as a tree with two nodes having the same label #H1.

3.3 Displayed trees and subnetworks

As mentioned before, a $\gamma = 0$ in an explicit network means that the hybrid edge is not even present. Thus, by removing all the minor hybrid edges (the ones with $\gamma < 0.5$), we get a tree, denoted the "major tree" because it is the tree that the majority of the genes follow (Figure 8), and hence, it could be used to study the evolutionary relationships of the group. By removing certain hybrid edges, we get a collection of trees denoted the "displayed trees". These trees are obtained by choosing one parent hybrid edge at each hybrid node, and dropping the other parent hybrid edge.

For example, in Figure 8 there is one network (left), with its major and minor trees in the center. The major tree (turning off the minor hybrid edge $\gamma < 0.5$) keeps the (A,B) clade, while the minor tree (dropping the major hybrid edge $\gamma > 0.5$) has the clade (E,A) now, as A is of hybrid origin between E and B.

If the network had more than one hybridization event, we could also get subnetworks by removing specific major/minor hybrid edges, but leaving others. It is important to keep in mind the possibility of breaking down a network into displayed trees or subnetworks, as this will facilitate the comparison with other networks. That is, network estimation is difficult enough that recovering identical networks from different methods is unlikely. However, if we break down the networks into trees/subnetworks, we might be able to identify features that all the estimated networks have in common. As we will see in Section 4.1.1, some reticulations are much harder to detect than others.

3.4 Comparing networks

One important ingredient in network inference is the ability to compare networks. For the case of trees, the most common approach involves the computation of the Robinson-Foulds distance [43]. However, there is not an equivalent distance for the case of networks, as the notion of splits or clades is not straight-forward. Rooted networks can be compared by their hardwired cluster dissimilarity [28]: the number of hardwired clusters found in one network and not in the other. A hardwired cluster is associated with a given node in the rooted network, and

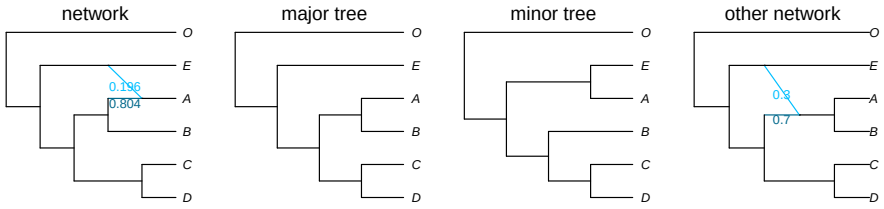

Figure 8: Network with its displayed trees, and a second network to illustrate the hardwired cluster distance of 4, which represents the number of hardwired clusters that are present in one network and not the other.

it is defined as the set of all taxa that inherited at least some genetic material from that node. That is, the hardwired cluster of a node is the set of all tips descendant from that node. The hardwired cluster dissimilarity relies on the root position, so unrooted or semi-directed networks need to be rooted with the same outgroup(s) first. Thus, this measure can be affected by rooting errors. Furthermore, the hardwired cluster dissimilarity is not a real distance for the space of all networks, but it is a distance for the class of level-1 networks. In fact, this dissimilarity is the Robinson–Foulds distance on trees. Recent work has produced distance measures on other classes of networks [40, 28, 15, 16].

To illustrate with an example, imagine that you want to compare two networks: one estimated with SNaQ (which will be described in Section 4) and one estimated with a different network method (Figure 8 left and right, respectively). Note that both networks need to be rooted in the same outgroup "O".

Visually, we can see that these two networks are different. The origin of gene flow is the same: E, but the target of gene flow is not: in the SNaQ network, it is the lineage ancestral to A, but, in the other network, it is the lineage ancestral to (A,B). If we calculate the hardwired distance between these two networks, we get 4. This means that there are 4 hardwired clusters that appear in one network and not the other. We can list the hardwired clusters of any network by looking at the descendant taxa for every given internal node. For example, the hardwired clusters for the network on the left are $\{CD, AB, AB, ABE, ABCD, ABCDE, ABCDEO\}$, and the ones for the network on the right are $\{CD, A, AB, AE, ABCD, ABCDE, ABCDEO\}$. We can see that the hardwired clusters that do not appear in both networks are 4: $\{AB, A, AE, ABE\}$. Finally, we can also note that if we compare the major trees (removing the minor hybrid edge), these trees are identical.

There is not a clear sense of whether 4 is small or big distance. Like the Robinson-Foulds distance, the hardwired cluster dissimilarity can be quite large for networks that are similar (i.e. having just one taxon in a different place). This is why the hardwired cluster dissimilarity is mostly used to determine whether two networks are equal (dissimilarity of 0). but see [14] for comparisons

on network metrics.

4 Fast reconstruction of species networks

Pseudolikelihood methods have been previously utilized to achieve computationally feasible inference when the full likelihood can be intractable. For example, Liu and others [34] used the pseudolikelihood approach to estimate phylogenetic trees under ILS in the software MP-Est. Here, we present a pseudolikelihood method to estimate phylogenetic networks denoted SNaQ (Species Networks applying Quartets) [48]. The rationale behind SNaQ is similar to MP-Est, except that SNaQ estimates phylogenetic networks, not trees, and SNaQ uses unrooted quartets in the computation of the pseudolikelihood, as opposed to rooted triplets. Using unrooted quartets is advantageous because it avoids potential rooting errors in the input gene trees.

4.1 Maximum pseudolikelihood estimation

Likelihood-based approaches to estimate phylogenetic networks rely on the computation of the likelihood of a network (with branch lengths and inheritance probabilities) given the data, which is a collection of gene trees estimated from DNA sequences. These methods can be very accurate as they can profit from all the statistical properties of maximum likelihood estimates. However, the computation of the likelihood can be very expensive, which, combined with the heuristic search in the space of networks (which is much bigger than the space of trees), results in an accurate method that is restricted to the case of small datasets (see [55, 56, 59] and Chapter 5).

To overcome the scalability problems of likelihood-based methods, Solis-Lemus and Ané [48] devised a pseudolikelihood method, which first summarizes the sample of gene trees into CFs (thus being scalable in the number of genes), and then breaks down the computation of the likelihood of the whole network into the computation of the likelihood for 4-taxon subsets. In this manner, only the likelihood of 4-taxon networks is computed. Despite the increase in the number of 4-taxon subsets as the number of taxa grows, this divide-and-conquer approach is still more scalable than computing the likelihood of the full network.

SNaQ (Species Networks applying Quartets) implements the statistical inference method in [48]. The procedure involves a numerical optimization of branch lengths and inheritance probabilities and a heuristic search in the space of phylogenetic networks. The full inference scheme from multi-locus sequences to phylogenetic network is shown in Figure 9.

As mentioned already, the pseudolikelihood of a network is based on the likelihoods of its 4-taxon subnetworks. That is, for a given network $\mathcal{N}$ with $n \geq 4$ taxa, we consider all 4-taxon subsets $\mathcal{S} = \{s = \{a, b, c, d\} : a, b, c, d \in X\}$ and combine the likelihood of each 4-taxon subnetwork to form the full network

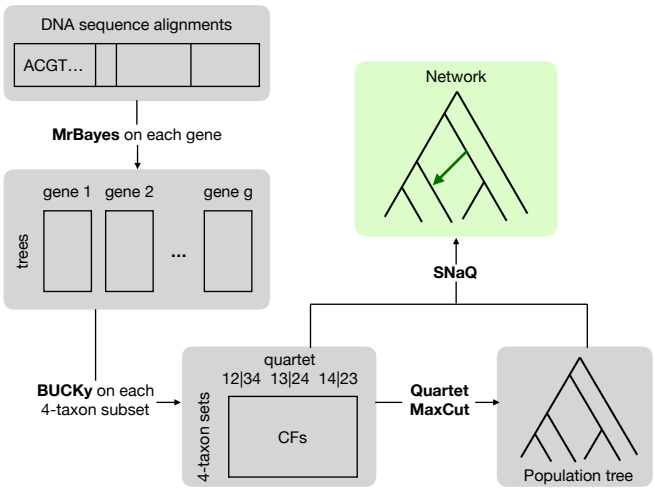

Figure 9: Flow chart of the procedure to estimate a network with SNaQ from multi-locus sequences.

pseudolikelihood:

$$L(\mathcal{N}) = \prod_{s \in \mathcal{S}} L(s) \tag{1}$$

where $L(s)$ is the likelihood of the subnetwork of a given 4-taxon subset s. These 4-taxon likelihoods are not independent, which is why we get a pseudolikelihood when we multiply them, instead of a true likelihood.

To calculate the likelihood of a 4-taxon network from gene trees $\mathcal{G} = (G_1, G_2, \cdots, G_g)$ at g loci, we note that for taxon set $s = \{a, b, c, d\}$, there are only three possible quartets, represented by the splits $q_1 = ab|cd$, $q_2 = ac|bd$ and $q_3 = ad|bc$. We then consider the number of gene trees $Y = (Y_{q_1}, Y_{q_2}, Y_{q_3})$ that display each of the three quartets. Assuming unlinked loci, Y follows a multinomial distribution with probabilities $(CF_{q_1}, CF_{q_2}, CF_{q_3})$, the theoretical CFs expected under the coalescent on the 4-taxon network. These theoretical CFs expected under the coalescent model are already known if the network is a species tree [4]. In that case, these CFs do not depend on the position of the root, and are given by $(1 - (2/3)\exp(-t), (1/3)\exp(-t), (1/3)\exp(-t))$ if the unrooted species tree is $q_1 = ab|cd$ with an internal edge of length t coalescent units. On a species network with reticulations, the probabilities of rooted gene trees were fully derived in [55] and more efficiently in [56], and for unrooted 4-taxon networks in [48]. We provide details on the computation of these probabilities in one example below.

By substituting the multinomial likelihood in Equation 1, we get

$$L(\mathcal{N}) \propto \prod_{s \in \mathcal{S}} (CF_{q_1})^{Y_{q_1}} (CF_{q_2})^{Y_{q_2}} (CF_{q_3})^{Y_{q_3}} \tag{2}$$

where $q_i = q_i(s)$ $(i = 1, 2, 3)$ are the 3 quartet resolutions on s. The data are summarized in the Y values, and the candidate network governs the CF values, which we explain below. The data, as mentioned already, could be either a collection of gene trees $\mathcal{G} = (G_1, G_2, \cdots, G_g)$, or the CFs estimated with BUCKy [5]. The advantage of estimating the CFs with BUCKy is that BUCKy tries to disentangle genomic discordance versus statistical discordance. Ideally, the CFs represent true genomic support: proportion of genes that support a given split. However, when we use estimated gene trees to calculate the CFs directly (by counting number of gene trees with a specific split), the resulting measure of support will have confounding effects of genomic support and statistical error, given that the gene trees can have some potential estimation error. When using BUCKy to estimate the CFs, BUCKy estimates the CFs as genomic support, accounting for statistical error in the reconstruction of the gene trees. Given that most coalescent-based network methods take a collection of estimated gene trees as perfectly known for the method's input, methods like SNaQ that can take estimated CFs as input instead are more robust to the estimation error in the gene trees.

For a given 4-taxon subnetwork, we want to calculate the expected CFs based on the multispecies coalescent model for networks [36, 55]. Take the subnetwork in Figure 10 (left) with $h = 1$ hybridization event. Each gene from taxon C has probability γ of having descended from the hybridization edge sister to D, and probability $1 - \gamma$ of having descended from the original tree branch, sister to (A,B). Therefore, the expected CFs are weighted averages of CFs obtained on 2 species trees with ILS. Because the quartet probabilities do not depend on the root placement in each species tree, they do not depend on the root placement in the original network either. Figure 10 (center) shows the corresponding semi-directed network, and all rooted networks displaying it share the same quartet CFs, obtained from the coalescent models on the 2 unrooted species trees shown in Figure 10 (right). These trees have the same topology but different branch lengths in this case. Therefore, we get that $CF_{ab|cd} = (1 - \gamma)(1 - (2/3)\exp(-t_1)) + \gamma(1 - (2/3)\exp(-t_1 - t_2))$ and the other 2 quartets occur with equal probabilities: $CF_{ac|bd} = CF_{ad|bc} = (1 - \gamma)(1/3)\exp(-t_1) + \gamma(1/3)\exp(-t_1 - t_2)$.

For 4-taxon subnetworks with $h = 1$ hybridization, there are 5 different semi-directed topologies up to tip re-labeling (Figure 11). Similar procedures to the one described for Figure 10 can be used to compute the expected CFs for each of the 5 cases (see [48]).

With more than 1 hybridization event ($h > 1$), there are an infinite number of semi-directed 4-taxon networks, but we can still calculate the quartet CFs if we assume that the cycles created by different reticulations do not share edges.

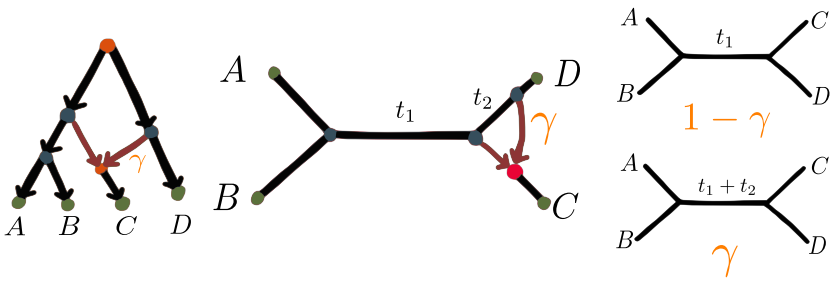

Figure 10: Rooted 4-taxon network (left) and its semi-directed version (center). Quartet CFs expected under the network do not depend on the root placement, and are weighted averages of quartet CFs expected under the unrooted trees (right).

Figure 11: Five different semi-directed 4-taxon networks with one hybridization event, up to tip re-labelling.

This is where the assumption of level-1 networks is used. By assuming that hybridization cycles do not intersect, we can reduce each network to an equivalent network with $h = 0$ or 1 with transformed branch lengths. For example, the network in Figure 10 leads to equal CFs of the 2 minor quartets $ac|bd$ and $ad|bc$, so it is equivalent to the unrooted species tree $ab|cd$ with internal branch length $t_3 = -\log((1-\gamma)\exp(-t_1)+\gamma\exp(-t_1-t_2))$ to ensure $(1/3)\exp(-t_3) = CF_{ac|bd}$ given above. This new quartet tree and the original 4-taxon network have the same expected quartet CFs. Thus, the assumption of a level-1 network guarantees non-overlapping reticulation cycles, such that we can find an equivalent 4-taxon network with $h = 0$ or 1 and the same expected quartet CFs.

The maximum pseudolikelihood (MPL) estimate is the network $\mathcal{N}$, branch lengths $\mathbf{t}$ and heritabilities γ that maximize the pseudolikelihood (Equation 2). This MPL optimization was fully implemented in SNaQ and is part of the open source package PhyloNetworks [46] in Julia [10]. The numerical optimization of branch lengths and γ parameters for a fixed topology is performed with a derivative-free methodology in the NLopt Julia package. The heuristic optimization of the network topology uses a strategy similar to that in [56]. Given a fixed maximum number of hybridizations (h_m), we search for the MPL network with at most h_m hybridizations. Since the pseudolikelihood can only improve when hybridizations are added, we expect the final network to have $h = h_m$ exactly.

For a given h_m, the search is initialized with a tree from a very fast quartet-based tree estimation method like ASTRAL [37] or Quartet MaxCut [45, 6]. The length of each branch is initialized using the average observed CF of the

quartets that span that branch exactly, $\overline{CF}$, transformed to coalescent units by $t = -\log(1-(3/2)\overline{CF})$, just as in the TICR test. The search then navigates the network space by altering the current network using one of 5 proposals, chosen at random: 1) move the origin of an existing hybrid edge, 2) move the target of an existing hybrid edge, 3) change the direction of an existing hybrid edge, 4) perform a nearest-neighbor interchange move (NNI) on a tree edge, and 5) add a hybridization if the current topology has $h < h_m$. Any new proposed network is checked to verify that it is a semi-directed level-1 network with $h \leq h_m$ and with at least one valid placement for the root (see Section 4.2).

Although the deletion of a current hybridization is not proposed (because the MPL network should have $h = h_m$), this deletion is still performed when suggested by the data, if the numerical optimization of parameters returns a $\hat{\gamma} = 0$. In this case, the corresponding hybrid edge is removed and the search attempts to add it back at random in the neighborhood of the original hybrid edge. If this attempt fails for all neighbors, the hybridization is deleted entirely and the search continues from a network with 1 fewer hybridization. Similarly, if the numerical optimization returns a branch of length $t = 0$, an NNI move is proposed immediately on that branch. The search continues until the pseudolikelihood converges or until the number of consecutive failed proposals reaches a limit.

In [26], Huber and others proved that the space of unrooted level-1 networks is connected by local subnetwork transfers, which generalize the NNI operations on trees and which are similar to our moves 1, 2 and 4. Although we do not have a formal proof that the MPL network can be reached from the starting tree using our proposals, the results in [26] suggest that it is the case.

After the pseudolikelihood estimation, we get the maximum pseudolikelihood estimated network (with estimated branch lengths in coalescent units and estimated inheritance probabilities) with $h = h_m$ hybridizations. A network can be estimated for various values of h_m, followed by a model selection procedure to select the appropriate number of hybridizations (see Section 4.3). Furthermore, we are interested in measuring statistical uncertainty as well. We have mentioned already one way to account for statistical uncertainty by using CFs estimated with BUCKy as input, instead of estimated gene trees. In addition, one can do a bootstrap analysis on the estimated network. This will be described in Section 4.4.

4.1.1 Identifiability: what we can and cannot learn from data

Identifiability is a basic requirement if one seeks to learn about parameters from data. Here our parameters are the network topology $\mathcal{N}$, branch lengths $\mathbf{t}$ and inheritance values γ. We already know that quartet CFs do not depend on the root placement, so the rooted network is not identifiable and we only consider semi-directed networks.

The pseudolikelihood model is identifiable if two different combinations of pa-

rameters $(\mathcal{N}, \mathbf{t}, \gamma)$ and $(\mathcal{N}', \mathbf{t}', \gamma')$ yield different sets of quartet CFs. It turns out that some reticulations and some parameters are impossible (or hard) to detect. By identifying the reticulations and parameters that can be recovered with the data at hand, SNaQ explores a reduced parameter space to avoid network and parameter combinations that are not identifiable.

On $n = 4$ taxa, we already showed that the network in Figure 10 is equivalent to a tree with some appropriate internal branch length. In fact, the same holds true for all 4-taxon networks with $k = 2$ or 3 nodes in their reticulation cycle: these reticulations cannot be detected. If $k = 4$ i.e. if the reticulation involves more distantly related taxa, then the presence of the hybridization can be detected based on the quartet CFs. However, networks with the same unrooted topology are unidentifiable from each other from only 4 taxa, like the 2 networks in Figure 12 if only D_1 is sampled ($n = 4$). They only differ in the placement of the hybrid node, which is therefore not identifiable with only $n = 4$, even if the presence of a reticulation is.

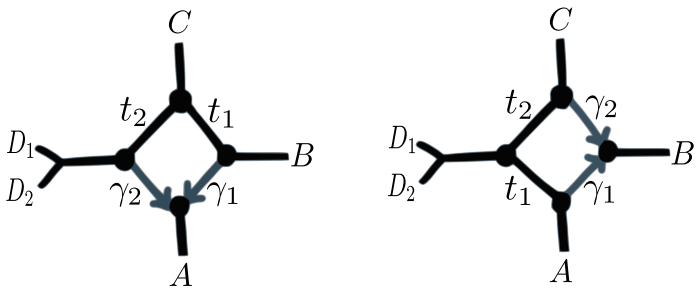

Figure 12: Networks with $k = 4$ nodes in the reticulation cycle and identical unrooted topologies. They differ in their hybrid position. If D2 is not sampled $(n = 4)$, the 2 networks are not distinguishable from each other. Furthermore, the set of numerical parameters is not identifiable, only $\gamma_i(1 - \exp(-t_i))$ for $i = 1, 2$ are identifiable.

In general, for networks with $n \geq 4$ taxa, the presence of the hybridization event of interest can be detected if the quartet CFs from $(\mathcal{N}, \mathbf{t}, \gamma)$ cannot all be equal to the quartet CFs from $(\mathcal{N}', \mathbf{t}', \gamma')$ simultaneously, where $\mathcal{N}'$ is the network topology obtained from $\mathcal{N}$ by removing the hybrid edge of interest. Assuming that all $\binom{n}{4}$ 4-taxon sets are used in the pseudolikelihood, the network $\mathcal{N}$ gives us $3\binom{n}{4}$ quartet CFs equations expected under the coalescent model. The assumption of level-1 networks allows us to consider one hybrid node at time in this identifiability study.

Intuitively, if both $\mathcal{N}$ (with h hybridization events) and $\mathcal{N}'$ (with $h-1$ hybridization events) produce the same set of CFs, then there is no possibility to detect the extra hybridization event in $\mathcal{N}$ with the CFs as input data. Thus, we want to identify the region in parameter space where $\mathcal{N}$ and $\mathcal{N}'$ produce different CFs, that is, the region in which we are certain that we can distinguish $\mathcal{N}$ and

$\mathcal{N}'$. To do this, we can match both systems of CF equations (one for $\mathcal{N}$ and one for $\mathcal{N}'$) using the algebraic geometry software Macaulay2 [23], and check the values of $(\mathbf{t}, \gamma)$ that produce the same CFs on both $\mathcal{N}$ and $\mathcal{N}'$. By avoiding these values, we can differentiate $\mathcal{N}$ and $\mathcal{N}'$ from the input CFs, and thus, we can detect the presence of the hybridization of interest.

Apart from the obvious case $\gamma = 0$ for the hybrid edge absent in $\mathcal{N}'$, $\mathcal{N}$ and $\mathcal{N}'$ are also not distinguishable when $t_b = 0$ or $t_b = \infty$ for some tree branches b, implying either a hard polytomy or a branch with no ILS. We can ignore these cases with the following reasonable assumption

A1: $t \in (0, \infty)$ for all tree branches and $\gamma \in (0, 1)$.

A1 is not a sufficient condition, however, to ensure that the presence of each hybridization in $\mathcal{N}$ can be detected. Increasing taxon sampling helps detect a hybridization only if the added taxa increase the size of the reticulation cycle. Namely, if the cycle only involves $k = 2$ nodes (see Figure 13), then $\mathcal{N}$ is not distinguishable from $\mathcal{N}'$, regardless of n. For $k = 3$, some hybridizations are detectable and some are not. If any two of the three subtrees defined by the hybridization cycle (Figure 13) have only one taxon, then the hybridization is not detectable [48]. It is, if instead at least two subtrees contain more than one taxon. In general, hybridizations with $k \geq 4$ can be detected if $n \geq 5$. Here and below, we use the terms detectable or identifiable in their generic sense [2, 1], which simply means that some conditions on $(\mathbf{t}, \gamma)$ are required, like **A1**, but that all these conditions are met except on a subset of measure zero.

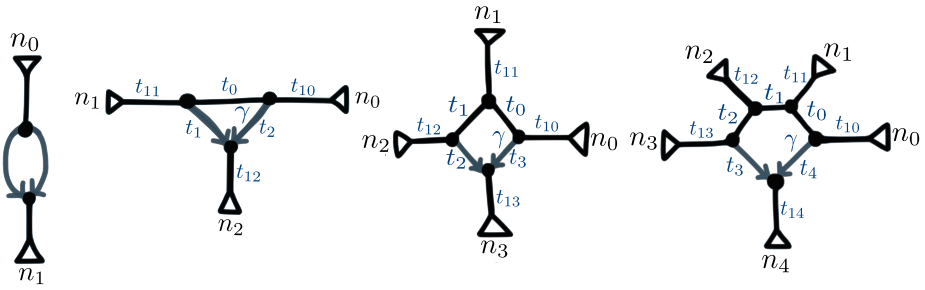

Figure 13: Networks with k nodes in a hybridization cycle: $k = 2, 3, 4$ and 5 from left to right. When $k = 2$, the presence of hybridization is not identifiable. When $k = 3$, the hybridization is detectable if $n_i, n_j \geq 2$ for any i, j, but the set of parameters is not identifiable. In this case, a **good triangle** corresponds to $n_1, n_2, n_3 \geq 2$, in which case setting $t_{12} = 0$ makes the other parameters identifiable. When $k = 4$, the hybridization is detectable if $n_i \geq 2$ for any i, but parameters are not all identifiable for the **bad diamond I** ($n_0 = n_2 = n_3 = 1$ but $n_1 \geq 2$) and for the **bad diamond II** ($n_0 = n_1 = n_2 = 1$ but $n_3 \geq 2$). When $k \geq 5$, the hybridization is detectable and all the parameters are identifiable.

Aside from the presence of the hybridization, we are also interested in knowing whether the direction of gene flow is identifiable. For example, Figure 12 shows

two networks that differ only in the placement of the hybrid node, but otherwise have the same unrooted topology. It turns out that these two networks yield different sets of quartet probabilities and therefore are distinguishable from each other, showing that the direction of the hybridization becomes identifiable when $n \geq 5$ [48]. However, in practice, it is quite common that the pseudolikelihood function from small sample size data behaves differently than the theoretical pseudolikelihood used in the identifiability proofs, in particular for the case of the direction of gene flow. Thus, it could be impossible to detect the correct direction of gene flow with a given sample size, and it is important to consider other networks with an alternative placement for the hybrid node in the hybridization cycle. This will be discussed further in Section 4.2.

Finally, we are also interested in whether numerical parameters like branch lengths and inheritance probabilities can be estimated from the input data of CFs. Like before, we determine under which conditions two different combinations of parameters $(\mathbf{t}, \gamma)$ and $(\mathbf{t}', \gamma')$ yield different sets of quartet probabilities for a fixed network $\mathcal{N}$. Just as before, the identifiability depends on the type of network (Figure 13). With only 4 taxa, there are more parameters than equations (3 quartet CFs), so $\mathbf{t}$ and γ are not separately identifiable, so to estimate these parameters, we need to have $n \geq 5$.

If $n \geq 5$, parameter identifiability is again easier if the reticulation involves more distantly related taxa. If $k \geq 5$, all the parameters are identifiable. If $k \leq 3$, parameters are not identifiable. If $k = 4$, parameters are identifiable if either $n_0 \geq 2$ (or n_2, symmetrically), or if both n_1 and $n_3 \geq 2$ (see Figure 13). We call this a good diamond. Parameters are not all identifiable in the remaining 2 cases, which we call bad diamonds I and II (see Figure 13). The bad diamond I already lacked identifiability under a different model in [41].

A naive search for the most likely network would get stuck alternating between non-distinguishable networks or parameter sets. Hence SNaQ reduces the searchable space to only consider networks whose reticulations involve enough nodes. Indeed, all reticulations with $k = 2$ and most with $k = 3$ are either not detectable at all, or their parameters are not all identifiable. Thus, SNaQ will skip proposed networks with $k = 2$, and most with $k = 3$, and it reparametrizes the cases when parameters are not identifiable.

4.2 Rooting semi-directed networks

SNaQ estimates semi-directed networks, where the root node is suppressed and we ignore the direction of all tree edges, but we maintain the direction of hybrid edges, thus keeping information on which nodes are hybrids. To interpret the estimated network, one might want to root it at a known outgroup. However, as mentioned before, unlike rooting trees, the placement of the root is constrained, because the direction of the two hybrid edges to a given hybrid node inform the direction of time at this node: the third edge must be a tree edge directed away from the hybrid node and leading to all the hybrid's descendants. Therefore

the root cannot be placed on any descendant of any hybrid node. For example, in Figure 14, the network on the left was estimated with SNaQ, but it has a randomly chosen root. If taxon "O" is meant to be the outgroup, then this network is clearly not rooted properly. In this case, the root cannot be placed in any descendant edge of the hybridization event. That is, the root cannot be placed on edges 9, 10, or 11.

4.2.1 What if the root conflicts with the direction of a reticulation?

As mentioned already, the direction of hybrid edges constrain the position of the root. The root cannot be downstream of hybrid edges. Any hybrid node has to be younger than, or of the same age as both of its parents. So time has to flow "downwards" of any hybrid node, and the root cannot be placed "below" a hybrid node.

For example, see Figure 14 (left) with edge numbers to illustrate where to root the network. Let's imagine that the A1 and A2 are outgroups, and this is the network estimated with SNaQ. According to this network, time must flow from the hybrid node toward A1 and A2. So any attempt to reroot the network with A1 as the outgroup, or with A2 as the outgroup, or with the A clade (on edge 11), will fail.

In this case, however, it is possible to root the network on either parent edge of the hybrid node (edges 5 and 12), resulting in the rooted versions in Figure 14 (two networks in the center). The network rooted on major edge 12 represents gene flow between E and an ancestral population to the clade (A1,A2), while the network rooted on minor edge 5 requires an unsampled or extinct taxa to be included, so that gene flow would be between the unsampled taxa and the ancestral population to the clade (A1,A2) (see Figure 14 right). The network rooted on the major edge 12 is more plausible if we think that the species tree is the major tree, meaning that any gene flow or introgression event replaced fewer than 50% of the genes in the recipient population.

In other cases, it may not be possible to re-root the network with a known outgroup: for example, if only A1 is the only outgroup, and if A2 was an ingroup taxon. In such a case, the outgroup knowledge tells us that our estimated network is wrong, as the placement of the reticulation contradicts the root position. Its placement might be correct, but its direction would be incorrect. In this situation, we can explore other candidate networks compatible with a known outgroup (see Section 4.2.2).

4.2.2 Candidate networks compatible with a known outgroup

When estimating semi-directed networks with SNaQ, there is always the possibility that the estimated network is impossible to root with a known outgroup. This is the case as SNaQ does not impose any rooting constraint on the network: the search for the lowest pseudolikelihood score considers all level-1 networks,

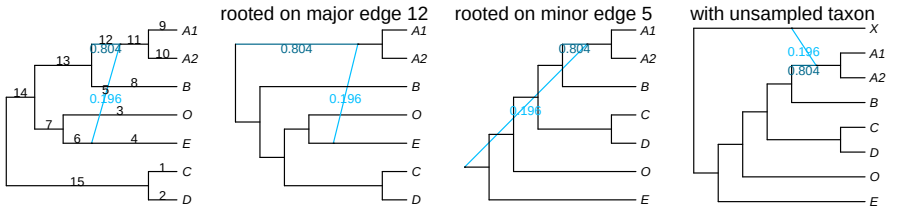

Figure 14: Left: Network estimated with SNaQ with a randomly chosen root. Black numbers denote edge labels. The root cannot be placed in edges 9, 10, or 11 as they are "below" the hybridization event. If the clade (A1,A2) is the outgroup, we need to root the network on the hybrid edges as following. Center left: Network rooted on the major hybrid edge (edge 12). Center right: Network rooted on minor hybrid edge (edge 5). This network requires an unsampled/extinct taxa. Right: Network rooted on the minor hybrid edge (edge 5) but with an added unsampled taxon X.

including those that are incompatible with a known outgroup. The monophyly of outgroups is not imposed either, as is the case for many other methods.

If the estimated network cannot be rooted with the known outgroup, we can recall the discussion in Section 4.1.1 where we noted that sometimes, it is impossible to detect the correct placement of the hybrid node in the hybridization cycle, depending on the specific network topology and the data at hand. Thus, we might want to compare the pseudolikelihood score of alternative placements of the hybrid node in the hybridization cycle on the estimate network (see Figure 12). If we find in these alternative placements of the hybrid node a modified network that has a score close to that of the best network, and that can be re-rooted with our known root position, then this modified network is a better candidate than the network with the best score. For example, in Figure 15 we show several networks: the best network estimated by SNaQ with pseudolikelihood score of 28.3 (the smaller, the better in this plot because the pseudolikelihood score denotes negative log-pseudolikelihood) with a randomly chosen root (top left), and the second best network with the direction of gene flow modified and score of 31.5 (not far from the best network), also with a random root (top right). Now imagine that our outgroup is taxon A. If we try to root the best network at "A" (edge 9), we will get an error. But we could root this network on the major parent edge to A (edge 10) in Figure 15 (top center).

For the second best network (Figure 15 top right), there are 2 ways to root it with A: on the external edge 8 to A (Figure 15 bottom center), or on its parent edge 10 (Figure 15 bottom left). These 2 options give quite different rooted versions of the network, one of which requires the existence of an unsampled taxon, sister to BOECD, that would have contributed to introgression into an ancestor of E (Figure 15 bottom right). The rooted version on edge 8 (bottom left) says that an ancestor of A contributed to the introgression into the ancestor of E.

Taxon A is the outgroup in both cases, but this last case is more parsimonious, in the sense that it does not require the existence of an unsampled taxon. If we compare this network (Figure 15 bottom left) with the best network rooted at A (Figure 15 top center), we can see that the only distinction is the direction of the minor hybrid edge: in one case going from A to E, and in the other case going from E to A. Both networks have similar pseudolikelihood score, so additional biological knowledge could be use to choose the more appropriate evolutionary interpretation.

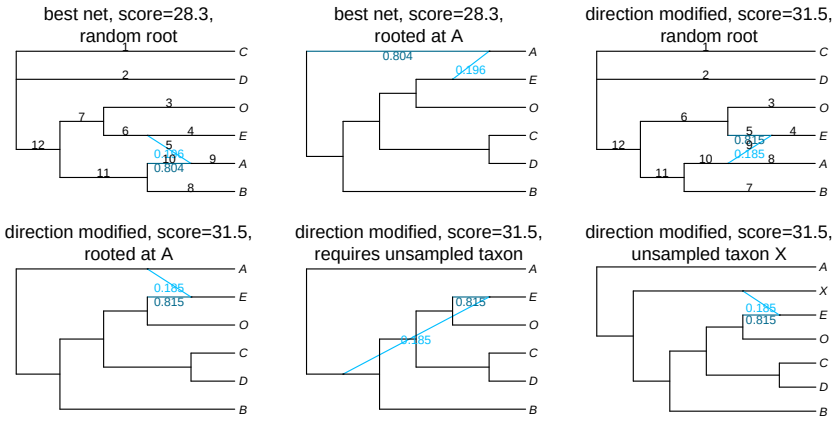

Figure 15: Top left: Best network estimated with SNaQ with random root placement. Black numbers represent edge labels. Top center: Same network, but rooted with the outgroup A. Top right: Second best network with the only difference being that the direction of the hybrid edge is from A to E instead of from E to A. The second best network allows us to root in A (bottom left). Two options to root the second best network at A. Bottom left: The second best network rooted at A on edge 8. Bottom center: Option 2 of the second best network rooted at A on edge 10. This option requires the existence of an unsampled taxon X. Bottom right: Second best network rooted at A, with the inclusion of the unsampled taxa X.

4.3 Goodness of fit tools

4.3.1 Choosing the number of hybridizations

When using SNaQ (and other network methods), users need to specify the maximum number of hybridizations allowed, because the pseudolikelihood will improve as h increases, in the same way that the likelihood [55, 56] and parsimony score [54] improve as hybridizations are added. Thus, model selection tools are necessary to estimate the number of hybridizations. Existing tools involve using cross-validation to determine the best parameter h [56]. For the pseudolikelihood framework, the cross-validation error could be measured from the difference

between the quartet CFs observed in the validation subset and the quartet CFs expected from the network estimated on the training set. Because K-fold cross-validation requires partitioning the loci into K subsets and re-estimating a network K times at each h value, this approach can be computationally heavy.

Information criteria have already been used to select h (e.g. [31]), but these criteria are inappropriate if the full likelihood is replaced by a pseudolikelihood. Theory is missing to compare the pseudolikelihood scores of different networks, because of the possible correlation between quartets from different 4-taxon sets. It can be shown, however, that quartets from two 4-taxon sets s_1 and s_2 are independent if s_1 and s_2 overlap by at most one taxon and if the true 4-taxon subnetworks share no internal edges. Future work could exploit this partial independence to construct hypothesis tests. Global tests like TICR have recently been extended to the case of networks [13], but there is still room for more research on model selection tools for networks.

One alternative is to use data-driven tools, for example, slope heuristics can indeed be used with contrast functions (like pseudolikelihood) for model selection in regression frameworks [11, 8]. In Figure 16 left, we show the log pseudolikelihood profile with h. A sharp improvement is expected until h reaches the best value and a slower, linear improvement thereafter. Based on this plot, we can estimate the best $h = 1$, as there is no longer an improvement in the pseudolikelihood for $h > 1$.

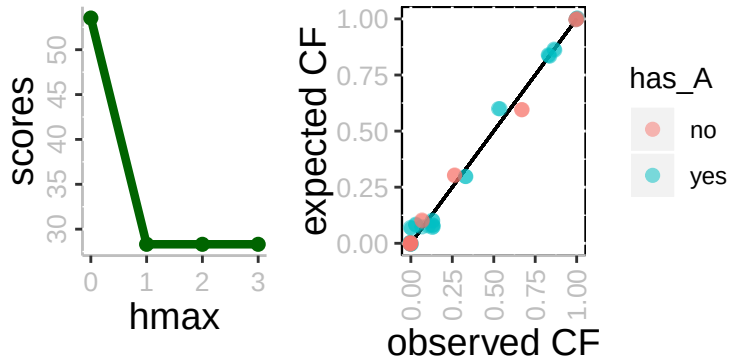

Figure 16: Left: Negative of log-pseudolikelihood used as score to choose the optimal number of hybridizations to be included in the network, here it is 1. Right: Comparison of quartet CFs observed in gene trees and quartet CFs expected from the network as a measure of goodness of fit. Color depends on whether the specific quartet (point) has taxon A (blue) or not (red).

4.3.2 Visualizing comparisons between observed CF vs expected CF

A good way to visualize the goodness of fit of a given estimated network to the data is to plot the observed CF from the input against the expected CF (Figure

16 right). There are always many points overlapping on the bottom-left corner: concordance factors of 0.0 for quartet resolutions not observed, and not expected. We could highlight quartets that include taxon A, say, if we suspect that it is an unrecognized hybrid. With this plot, there is not a discernible pattern, but if A was indeed an unrecognized hybrid, then we would expect the red dots to be farther from the $x = y$ line than the blue dots.

4.4 Bootstrap analysis

To measure uncertainty in the network, one may re-estimate the network on bootstrap data sets, and compare the bootstrap networks. In order to create bootstrap data sets, we can either use the credibility intervals of the CFs obtained with BUCKy to sample CFs for each quartet, or we can use bootstrap gene trees to sample input gene trees at random, one per gene.

As will be described in the next sections, to summarize the networks estimated from these bootstrap sets, we first calculate the support for edges being in the major tree: the tree obtained by suppressing the minor hybrid edge (with $\gamma < 0.5$) at each reticulation. We then summarize the support for the placement of each minor hybrid edge on that tree, considering 2 edges as equivalent if they are of the same type (hybrid or tree edges) and define the same clusters in the networks [56].

4.4.1 Summarizing support for tree edges

After sampling 100 bootstrap datasets (CFs or gene trees), we estimate 100 bootstrap networks, and we need to summarize what they have in common (highly supported features) and what they do not (areas of uncertainty). The first thing to do is to summarize the bootstrap support of tree edges, which can be easily done by extracting the major tree in each of the 100 bootstrap networks. This results in a sample of 100 bootstrap major trees, and usual techniques to summarize bootstrap support on tree edges work.

In Figure 18, we map the bootstrap support for internal tree edges (100 in every case) onto the estimated network (left). These bootstrap support values mean that all bootstrap major trees contain the same internal edges. So, the main vertical signal of the data is strong and highly supports the major tree in the estimated network.

4.4.2 Summarizing support for hybrid edges and hybrid nodes

Summarizing a set of networks is more complex than summarizing a set of trees, because edges in a network do not uniquely correspond to bipartitions of the taxon set. In the previous subsection, we just illustrated that for a tree edge, we calculate its bootstrap support as a major edge, that is, the proportion of bootstrap networks whose major tree displays the edge of interest.

To summarize bootstrap support for reticulation events, we consider each hybrid node separately. We remove the minor hybrid edges at all the other reticulation events, and only keep one reticulation event of interest. We then identify 3 clades: the hybrid clade, the major sister clade and the minor sister clade of the hybrid node (Figure 17). The descendants of a given hybrid node form the "recipient" or "hybrid" clade, which is obtained after removing all other reticulations. Because of the reticulation event, the hybrid clade has 2 sister clades, not 1: the major sister (through the major hybrid edge with $\gamma > 0.5$) and the minor sister (through the minor hybrid edge with $\gamma < 0.5$). We can calculate the frequency that each clade is a hybrid clade, or a major or minor sister for some other hybrid, in the bootstrap networks.

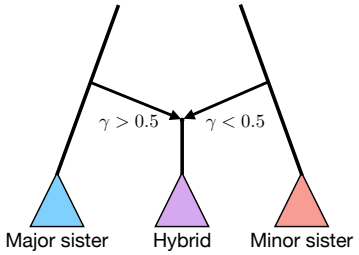

Figure 17: A hybrid clade is the set of all descendants of a hybrid node. The major (or minor) sister clade is the descendant set of the edge sister to the major (minor) hybrid edge. In unrooted networks, both sister clades are considered as splits (unrooted bipartitions) because the root can be placed anywhere outside the hybrid clade.

For example, in Figure 18 (left), we can plot the bootstrap values of the 2 hybrid edges in the best network: percentage of bootstrap networks with an edge from the same sister clade to the same hybrid clade. In this case, there is 32% bootstrap support for the minor hybrid edge (this hybrid edge appears in 32% of the bootstrap networks), and 33% for the major hybrid edge (this hybrid edge appears in 33% of the bootstrap networks). In addition, we could show the bootstrap support for the full reticulation relationship in the network, shown at the hybrid node in Figure 18 (left). In this case, the support for same hybrid with same sister clades is 32%. This means that in 32% of the bootstrap networks we have the same reticulation event (exact same hybrid clade, major hybrid clade and minor hybrid clade). Thus, this reticulation event (and the two hybrid edges) are not well-supported by the data. So, we next explore what are the other reticulation events that appear in the bootstrap networks.

4.4.2.1 Who are the hybrids in bootstrap networks? In Figure 18 (center left), we show the bootstrap support for alternative hybrid clades mapped on the parent edge of these nodes. For example, taxon A is estimated as a hybrid in only 33% of our bootstrap networks. This percentage seems to contradict

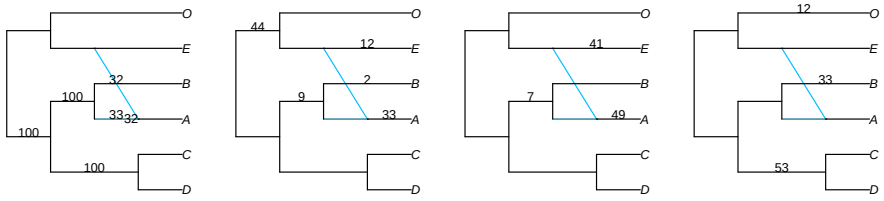

Figure 18: Left: Best network with bootstrap support on tree edges (all 100), on the two hybrid edges (32 and 33) and on the hybrid node (32). Black numbers represent bootstrap support. The bootstrap support value for the hybrid node represents the proportion of bootstrap networks that have the same two hybrid edges. Center left: Boostrap values on hybrid clades that appeared on the bootstrap networks. For example, the clade (O,E) was a hybrid clade in 44 out of the 100 bootstrap networks. Center right: Bootstrap values for the possible origin of gene flow (minor sister clade) that appeared in the bootstrap networks. For example, taxon E was the origin of gene flow in 41 out of 100 bootstrap networks. Recall that in 32 of those, the recipient of gene flow was taxon A, as shown by the bootstrap support of 32 on the minor hybrid edge (from E to A) in the network on the left. Right: Bootstrap values for the possible major sister clades in the bootstrap networks. For example, clade (C,D) was a major sister clade in 53 out of 100 bootstrap networks.

the 32% in the hybrid node in Figure 18 (left), however, these two quantities represent different proportions. In Figure 18 (center left), 33% represents the proportion of times that taxon A is a hybrid clade *regardless* of the sister clades, whereas in Figure 18 (left), 32% represents the proportion of times taxon A is a hybrid clade, *with E and B as sister clades*. Continuing with other hybrid clades in Figure 18 (center left), the lineage to (E,O) is estimated as being of hybrid origin in 44%, in 12% it is taxon E, in 9% it is clade (A,B), and in 2% it is taxon B. So, there is quite a bit of variability in the hybrid clades among the bootstrap networks, with clade (O,E) as the one more supported, but with only 44%.

4.4.2.2 Where is the origin of gene flow? Just as we explored the hybrid clades in the bootstrap networks (that is, the recipient clades of gene flow), we can ask which clades are the most supported as sister clades to the hybrid clade. That is, we are interested in identifying the clades that serve as the origin of gene flow in most of the bootstrap networks.

In Figure 18 (center right), we plot the support for the various placements of the gene flow origin (minor sister clade), mapped along the parent edge of these nodes. We filtered clades to only show those with bootstrap support $> 5\%$.

In the best network, the lineage to E is estimated as the origin of gene flow (Figure 18 (left)), but this is recovered in only 41% of our bootstrap networks. In another 49%, it is the lineage to A that is estimated as the origin of gene flow:

so gene flow is estimated in the opposite direction. Finally, there is 7% support for gene flow originating in (A,B).

Mapping the support for major sister clades might be interesting too (Figure 18 right). In the best network (Figure 18 (left)), B is estimated as the major sister clade, with 33% bootstrap support. The clade (C,D) has the higher bootstrap support as major sister clade with 53%.

These examples illustrate some of the complications of summarizing samples of networks. These tools can also be used to summarize a posterior distribution of networks generated by other programs (e.g., [52, 59]). PhyloNet can summarize a set of networks by listing the networks with highest support (and averaging branch lengths and inheritance probabilities for each unique topology). The tools described here provide summaries about local relationships on individual edges and nodes. This lets us identify hybridization events that are highly supported, regardless of the other hybridizations in the network, which is crucial as we showed that some hybridizations are harder to be detected than others.

5 Appendix: installation and use of the PhyloNetworks Julia package

Julia is a high-level and interactive programming language (like R or Matlab), but it is also high-performance (like C). The instructions to install Julia are in https://julialang.org/downloads/. The instructions to install the PhyloNetworks package are in http://crsl4.github.io/PhyloNetworks.jl/stable/.

SNaQ is a method implemented in the package to estimate a phylogenetic network from multiple molecular sequence alignments. There are two alternatives for the input data:

- A list of estimated gene trees for each locus, which can be obtained using MrBayes or RAxML,
- A table of concordance factors (CF), i.e. gene tree frequencies, for each 4-taxon subset. This table can be obtained from BUCKy, to account for gene tree uncertainty (see Figure 9).

In the package documentation, we present a pipeline to obtain the table of quartet CF needed as input for SNaQ (see also https://github.com/crsl4/PhyloNetworks.jl/wiki). The tutorial starts from the sequence alignments, runs MrBayes and then BUCKy (both parallelized), producing the table of estimated CFs and their credibility intervals. Additional details on this TICR pipeline describe how to insert data at various stages (e.g. after running MrBayes on each locus).

Besides the scalability of SNaQ compared to other network methods, SNaQ has the advantage of not using branch lengths in the estimated gene trees. By using only gene tree topologies, SNaQ avoids the dangerous assumptions that all genes

and/or lineages evolve at the same rate. For reconstructing species tree, methods that ignore branch lengths in gene trees tend to be more robust.

Also, SNaQ uses unrooted gene trees as input, which also prevents a potential layer of error if the outgroup is involved in ILS, or saturation or long branch attraction. For example, [21] shows that rooting errors explain incongruence in a yeast dataset.

5.1 Main functions in PhyloNetworks

First, we need to read the input data into Julia which can be a concordance factors table from BUCKy [5]:

```
buckyCF = readTableCF("bucky-tableCF.csv")
```

or a list of estimated gene trees from MrBayes [27] or RAxML [49]:

```
raxmlCF = readTrees2CF("raxml-trees.tre")
```

If the input data is a list of estimated gene trees, then the `readTrees2CF` function will create the table of CFs automatically.

Finally, we need to read a starting tree or network for the optimization. Usually, users estimate a species tree with a coalescent-based method like ASTRAL [37], which we can read into Julia with the following command:

```
tre = readTopology("astral.tre")
```

Next, we describe how to estimate a network in Julia. The function `snaq!` in the PhyloNetworks package estimates a semi-directed level-1 network from a collection of gene trees or a table of concordance factors. SNaQ also needs a starting topology for the optimization in the space of networks, and it needs to impose a constraint on the maximum number hybridizations allowed (`hmax`). Usually, one would start with `hmax=0`, and then use the resulting tree as starting point to estimate a network with one hybridization event (`hmax=1`), and so on.

We can run the `snaq!` function on a given starting tree (`tree`), input data (`CF`) and maximum number of hybridizations allowed (`hmax=0` in the first example). The function will return a network object (`net0`), which can be used in subsequent analyses like comparative methods [7], or simply plotting or re-rooting (see Section 4.2).

```
net0 = snaq!(tree, CF, hmax=0, filename="net0", seed=1234);
net1 = snaq!(net0, CF, hmax=1, filename="net1", seed=2345);
net2 = snaq!(net1, CF, hmax=2, filename="net2", seed=3456);
net3 = snaq!(net2, CF, hmax=3, filename="net3", seed=4567);
```

Here, we are estimating networks up to 3 hybridization events in a sequential manner: first we estimate a tree (`hmax=0`), then using this estimated tree as

starting topology, we estimate a network with one hybridization event (`hmax=1`), and so on.

For a full documentation and tutorial, see https://github.com/crsl4/PhyloNetworks.jl/.

References

[1] E. S. Allman and J. A. Rhodes. The identifiability of covarion models in phylogenetics. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1):76–88, 2009.

[2] Elizabeth S. Allman, Cécile Ané, and John A. Rhodes. Identifiability of a markovian model of molecular evolution with gamma-distributed rates. *Advances in Applied Probability*, 40(1):229–249, 2008.

[3] Elizabeth S Allman, Hector Baños, and John A Rhodes. Nanuq: a method for inferring species networks from gene trees under the coalescent model. *Algorithms for Molecular Biology*, 14(1):1–25, 2019.

[4] Elizabeth S. Allman, James H. Degnan, and John A. Rhodes. Identifying the rooted species tree from the distribution of unrooted gene trees under the coalescent. *Journal of Mathematical Biology*, 62(6):833–862, 2011.

[5] Cécile Ané, Bret Larget, David A Baum, Stacey D Smith, and Antonis Rokas. Bayesian estimation of concordance among gene trees. *Molecular biology and evolution*, 24(2):412–26, mar 2007.

[6] Eliran Avni, Reuven Cohen, and Sagi Snir. Weighted Quartets Phylogenetics. *Systematic Biology*, 64(2):233–242, 11 2014.

[7] Paul Bastide, Claudia Solís-Lemus, Ricardo Kriebel, Kenneth William Sparks, and Cécile Ané. Phylogenetic comparative methods on phylogenetic networks with reticulations. *Systematic Biology*, 67(5):800–820, 2018.

[8] Jean-Patrick Baudry, Cathy Maugis, and Bertrand Michel. Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2):455–470, 2012.

[9] David A Baum. Concordance trees, concordance factors, and the exploration of reticulate genealogy. *Taxon*, 56(May):417–426, 2007.

[10] Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman. Julia: A Fast Dynamic Language for Technical Computing. *arXiv:1209.5145*, pages 1–27, sep 2012.

[11] Lucien Birge and Pascal Massart. Minimal penalties for gaussian model selection. *Probability Theory and Related Fields*, 138:33–73, 2007.

[12] Paul D Blischak, Julia Chifman, Andrea D Wolfe, and Laura S Kubatko.

Hyde: A python package for genome-scale hybridization detection. *Systematic biology*, 67(5):821–829, 09 2018.

[13] Ruoyi Cai and Cécile Ané. Assessing the fit of the multi-species network coalescent to multi-locus data. *Bioinformatics*, 37(5):634–641, 2021.

[14] G. Cardona, M. Llabres, F. Rossello, and G. Valiente. Metrics for phylogenetic networks i: Generalizations of the robinson-foulds metric. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1):46–61, 2009.

[15] G. Cardona, F. Rossello, and G. Valiente. Comparison of tree-child phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(4):552–569, 2009.

[16] Gabriel Cardona, Mercé Llabrés, Francesc Rosselló, and Gabriel Valiente. A distance metric for a class of tree-sibling phylogenetic networks. *Bioinformatics*, 24(13):1481–1488, 05 2008.

[17] Gabriel Cardona, Francesc Rosselló, and Gabriel Valiente. Extended Newick: it is time for a standard representation of phylogenetic networks. *BMC bioinformatics*, 9:532, 2008.

[18] James H. Degnan. Anomalous unrooted gene trees. *Systematic Biology*, 62(4):574–590, 2013.

[19] James H. Degnan, Noah a. Rosenberg, and Tanja Stadler. A characterization of the set of species trees that produce anomalous ranked gene trees. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(6):1558–1568, 2012.

[20] Eric Y Durand, Nick Patterson, David Reich, and Montgomery Slatkin. Testing for ancient admixture between closely related populations. *Molecular biology and evolution*, 28(8):2239–2252, 08 2011.

[21] John Gatesy, Rob DeSalle, and Niklas Wahlberg. How Many Genes Should a Systematist Sample? Conflicting Insights from a Phylogenomic Matrix Characterized by Replicated Incongruence. *Systematic Biology*, 56(2):355–363, 04 2007.

[22] Olivier Gauthier and François-Joseph FJ François-joseph Lapointe. Hybrids and Phylogenetics Revisited: A Statistical Test of Hybridization Using Quartets. *Systematic Botany*, 32(1):8–15, 2007.

[23] DR Grayson and ME Stillman. Macaulay2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.

[24] Richard E Green, Johannes Krause, Adrian W Briggs, Tomislav Maricic, Udo Stenzel, Martin Kircher, Nick Patterson, Heng Li, Weiwei Zhai, Markus Hsi-Yang Fritz, Nancy F Hansen, Eric Y Durand, Anna-Sapfo Malaspinas, Jeffrey D Jensen, Tomas Marques-Bonet, Can Alkan, Kay Prüfer, Matthias

Meyer, Hernán A Burbano, Jeffrey M Good, Rigo Schultz, Ayinuer Aximu-Petri, Anne Butthof, Barbara Höber, Barbara Höffner, Madlen Siegemund, Antje Weihmann, Chad Nusbaum, Eric S Lander, Carsten Russ, Nathaniel Novod, Jason Affourtit, Michael Egholm, Christine Verna, Pavao Rudan, Dejana Brajkovic, Željko Kucan, Ivan Gušic, Vladimir B Doronichev, Liubov V Golovanova, Carles Lalueza-Fox, Marco de la Rasilla, Javier Fortea, Antonio Rosas, Ralf W Schmitz, Philip L F Johnson, Evan E Eichler, Daniel Falush, Ewan Birney, James C Mullikin, Montgomery Slatkin, Rasmus Nielsen, Janet Kelso, Michael Lachmann, David Reich, and Svante Pääbo. A draft sequence of the neandertal genome. *Science (New York, N.Y.)*, 328(5979):710–722, 05 2010.

[25] Stefan Grünewald, Andreas Spillner, Sarah Bastkowski, Anja Bögershausen, Vincent Moulton, Stefan Grünewald, and Anja Bögershausen. SuperQ: computing supernetworks from quartets. *IEEE/ACM transactions on computational biology and bioinformatics*, 10(1):151–60, 2013.

[26] Katharina T. Huber, Simone Linz, Vincent Moulton, and Taoyang Wu. Spaces of phylogenetic networks from generalized nearest-neighbor interchange operations. *Journal of Mathematical Biology. In press*, 2015.

[27] J P Huelsenbeck and F Ronquist. MrBayes: Bayesian inference of phylogeny. *Bioinformatics*, 17:754–755, 2001.

[28] Daniel Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic Networks*. Cambridge University Press, New York, NY, first edition, 2010.

[29] L. Lacey Knowles and Laura S Kubatko. *Estimating Species Trees: practical and theoretical aspects*. Wiley-Blackwell, 1st edition, 2010.

[30] Laura S. Kubatko and Julia Chifman. An invariants-based method for efficient identification of hybrid species from large-scale genomic data. *bioRxiv*, page 034348, 01 2015.

[31] Laura Salter Kubatko. Identifying Hybridization Events in the Presence of Coalescence via Model Selection. *Systematic Biology*, 58(5):478–488, oct 2009.

[32] Laura Salter Kubatko and James H. Degnan. Inconsistency of Phylogenetic Estimates from Concatenated Data under Coalescence. *Systematic Biology*, 56(1):17–24, 02 2007.

[33] Liang Liu and Lili Yu. Estimating Species Trees from Unrooted Gene Trees. *Systematic Biology*, 60(5):661–667, 03 2011.

[34] Liang Liu, Lili Yu, and Scott V Edwards. A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. *BMC Evolutionary Biology*, 10:302, jan 2010.

[35] Colby Long and Laura Kubatko. The Effect of Gene Flow on Coalescent-based Species-Tree Inference. *Systematic Biology*, 67(5):770–785, 03 2018.

[36] Chen Meng and Laura Salter Kubatko. Detecting hybrid speciation in the presence of incomplete lineage sorting using gene tree incongruence: a model. *Theoretical population biology*, 75(1):35–45, mar 2009.

[37] S Mirarab, R Reaz, Md S Bayzid, T Zimmermann, M S Swenson, and T Warnow. Astral: genome-scale coalescent-based species tree estimation. *Bioinformatics (Oxford, England)*, 30(17):i541–i548, 09 2014.

[38] Jonathan D Mitchell, Elizabeth S Allman, and John A Rhodes. Hypothesis testing near singularities and boundaries. *Electronic journal of statistics*, 13(1):2150, 2019.

[39] M. M. Morin and B. M E Moret. NetGen: Generating phylogenetic networks with diploid hybrids. *Bioinformatics*, 22(15):1921–1923, 2006.

[40] Luay Nakhleh. A metric on the space of reduced phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(2):218–222, 2009.

[41] JK Joseph K. Pickrell and JK Jonathan K. Pritchard. Inference of population splits and mixtures from genome-wide allele frequency data. *PLoS genetics*, 8(11):e1002967, jan 2012.

[42] John A Rhodes, Hector Baños, Jonathan D Mitchell, and Elizabeth S Allman. Mscquartets 1.0: quartet methods for species trees and networks under the multispecies coalescent model in r. *Bioinformatics*, 37(12):1766–1768, 2021.

[43] D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1):131–147, 1981.

[44] Sebastien Roch and Mike Steel. Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. *Theoretical Population Biology*, 100:56–62, 2015.

[45] Sagi Snir and Satish Rao. Quartet maxcut: a fast algorithm for amalgamating quartet trees. *Molecular phylogenetics and evolution*, 62(1):1—8, January 2012.

[46] C. Solís-Lemus, P. Bastide, and C. Ané. PhyloNetworks: a package for phylogenetic networks. *Molecular Biology and Evolution*, 34(12):3292–3298, 2017.

[47] Claudia Solís-Lemus and Cécile Ané. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLOS Genetics*, 12(3):e1005896, 2016.

[48] Claudia Solís-Lemus, Mengyao Yang, and Cécile Ané. Inconsistency of species-tree methods under gene flow. *Systematic Biology*, 2016.

[49] Alexandros Stamatakis. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 2014.

[50] Noah Stenz, Bret Larget, David A Baum, and Cecile Ane. Exploring tree-like and non-tree-like patterns using genome sequences: an example using the inbreeding plant species Arabidopsis thaliana (L.) Heynh. *Systematic Biology*, 64(5):809–823, 2015.

[51] Cuong Than, Derek Ruths, and Luay Nakhleh. PhyloNet: a software package for analyzing and reconstructing reticulate evolutionary relationships. *BMC bioinformatics*, 9:322, jan 2008.

[52] Dingqiao Wen, Yun Yu, and Luay Nakhleh. Bayesian inference of reticulate phylogenies under the multispecies network coalescent. *PLOS Genetics*, 12(5):e1006006–, 05 2016.

[53] Jialiang Yang, Stefan Grünewald, Yifei Xu, and Xiu-Feng Wan. Quartet-based methods to reconstruct phylogenetic networks. *BMC systems biology*, 8:21, 2014.

[54] Yun Yu, R. Matthew Barnett, and Luay Nakhleh. Parsimonious inference of hybridization in the presence of incomplete lineage sorting. *Systematic Biology*, 62(5):738–751, sep 2013.

[55] Yun Yu, James H. Degnan, and Luay Nakhleh. The probability of a gene tree topology within a phylogenetic network with applications to hybridization detection. *PLoS genetics*, 8(4):e1002660, jan 2012.

[56] Yun Yu, Jianrong Dong, Kevin J Liu, and Luay Nakhleh. Maximum Likelihood Inference of Reticulate Evolutionary Histories. *PNAS*, 111(46):16448–16453, 2014.

[57] Yun Yu and Luay Nakhleh. A maximum pseudo-likelihood approach for phylogenetic networks. *BMC genomics*, 16(10):1–10, 2015.

[58] Chao Zhang, Maryam Rabiee, Erfan Sayyari, and Siavash Mirarab. Astral-iii: polynomial time species tree reconstruction from partially resolved gene trees. *BMC Bioinformatics*, 19(6):153, 2018.

[59] Chi Zhang, Huw A Ogilvie, Alexei J Drummond, and Tanja Stadler. Bayesian Inference of Species Networks from Multilocus Sequence Data. *Molecular Biology and Evolution*, 35(2):504–517, 12 2017.