On the inference of complex phylogenetic networks by Markov Chain Monte-Carlo

Charles-Elie Rabier, Vincent Berry, Marnus Stoltz, Joao D. Santos, Wensheng Wang, Jean-Christophe Glaszmann, Fabio Pardi, Celine Scornavacca

On the inference of complex phylogenetic networks by Markov Chain Monte-Carlo

Charles-Elie Rabier [1,2,3,*], Vincent Berry [2], Marnus Stoltz[1], João D. Santos[4,5], Wensheng Wang[6], Jean-Christophe Glaszmann[4,5], Fabio Pardi[2] and Celine Scornavacca[1,*]

1 Institut des Sciences de l'Evolution (ISEM), Université de Montpellier, CNRS, EPHE, IRD, Montpellier, France
2 Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM), Université de Montpellier, CNRS, Montpellier, France
3 Institut Montpelliérain Alexander Grothendieck (IMAG), Université de Montpellier, CNRS, Montpellier, France
4 CIRAD, UMR AGAP, Montpellier, France
5 Amélioration Génétique et Adaptation des Plantes méditerranéennes et tropicales (AGAP), Université de Montpellier, CIRAD, INRAE, Institut Agro, Montpellier, France
6 Institute of Crop Sciences (ICS), Chinese Academy of Agricultural Sciences, Beijing, China

* charles-elie.rabier@umontpellier.fr (CER); * celine.scornavacca@umontpellier.fr (CS)

Abstract

For various species, high quality sequences and complete genomes are nowadays available for many individuals. This makes data analysis challenging, as methods need not only to be accurate, but also time efficient given the tremendous amount of data to process. In this article, we introduce an efficient method to infer the evolutionary history of individuals under the multispecies coalescent model in networks (MSNC). Phylogenetic networks are an extension of phylogenetic trees that can contain *reticulate* nodes, which allow to model complex biological events such as horizontal gene transfer, hybridization and introgression. We present a novel way to compute the likelihood of *biallelic* markers sampled along genomes whose evolution involved such events. This likelihood computation is at the heart of a Bayesian network inference method called SnappNet, as it extends the Snapp method inferring evolutionary trees under the multispecies coalescent model, to networks. SnappNet is available as a package of the well-known beast 2 software.

Recently, the `MCMC_BiMarkers` method, implemented in PhyloNet, also extended Snapp to networks. Both methods take biallelic markers as input, rely on the same model of evolution and sample networks in a Bayesian framework, though using different methods for computing priors. However, SnappNet relies on algorithms that are exponentially more time-efficient on non-trivial networks. Using simulations, we compare performances of SnappNet and `MCMC_BiMarkers`. We show that both methods enjoy similar abilities to recover simple networks, but SnappNet is more accurate than `MCMC_BiMarkers` on more complex network scenarios. Also, on complex networks, SnappNet is found to be extremely faster than `MCMC_BiMarkers` in terms of time required for the likelihood computation. We finally illustrate SnappNet performances on a rice data set. SnappNet infers a scenario that is consistent with previous results and provides additional understanding of rice evolution.

Author summary

Nowadays, to make the best use of the vast amount of genomic data at our disposal, there is a real need for methods able to model complex biological mechanisms such as hybridization and introgression. Understanding such mechanisms can help geneticists to elaborate strategies in crop improvement that may help reducing poverty and dealing with climate change. However, reconstructing such evolution scenarios is challenging. Indeed, the inference of phylogenetic networks, which explicitly model reticulation events such as hybridization and introgression, requires high computational resources. Then, on large data sets, biologists generally deduce reticulation events indirectly using species tree inference tools.

In this context, we present a new Bayesian method, called SnappNet, dedicated to phylogenetic network inference. Our method is competitive in terms of execution speed with respect to its competitors. This speed gain enables us to consider more complex evolution scenarios during Bayesian analyses. When applied to rice genomic data, SnappNet retrieved an evolution scenario that confirms the global triple foundation of the species and the origin of cBasmati as a hybrid derivative between Japonica cultivars and a local Indian form. It suggests that this hybridization is ancient and probably precedes the domestication of cAus.

Introduction

Complete genomes for numerous species in various life domains [1–5], and even for several individuals for some species [6, 7] are nowadays available thanks to next generation sequencing. This flow of data finds applications in various fields such as pathogenecity [8], crop improvement [9], evolutionary genetics [10] or population migration and history [11–13]. Generally, phylogenomic studies use as input thousands to millions genomic fragments sampled across different species. To process such a large amount of data, methods need not only to be accurate, but also time efficient. The availability of numerous genomes at both the intra and inter species levels has been a fertile ground for studies at the interface of population genetics and phylogenetics [14] that aim to estimate the evolutionary history of closely related species. In particular, the well-known coalescent model from population genetics [15] has been extended to the *multispecies coalescent* (MSC) model [16, 17] to handle studies involving populations or individuals from several species. Recent works show how to incorporate sequence evolution processes into the MSC [18, 19]. As a result, it is now possible to reconstruct evolutionary histories while accounting for both incomplete lineage sorting (ILS) and sequence evolution [20, 21].

For a given locus, ILS leads different individuals in a same population to have different alleles that can trace back to different ancestors. Then, if speciation occurs before the different alleles get sorted in the population, the locus tree topology can differ from the species history [22]. But incongruence between these trees can also result from biological phenomena that can cause a species to inherit lineages and/or genomic fragments from more than one parent species. Examples of such phenomena include hybrid speciation [23–26], introgression [27–29] and horizontal gene transfer [30, 31] (the latter is not addressed in this paper). As a consequence of these reticulate events, trees are not suited to represent species history, and should be replaced by phylogenetic networks. A rooted phylogenetic network is mainly a directed acyclic graph whose internal nodes can have several children, as in trees, but can also have several parents [32–34]. Various models of phylogenetic network have been proposed over time to explicitly represent reticulate evolution, such as hybridization networks [35] or

ancestral recombination graphs [36], along with dozens of inference methods [37,38].

Model-based methods have been proposed to handle simultaneously ILS and reticulate evolution, which is a desired feature to avoid bias in the inference [39–41]. These methods postulate a probabilistic model of evolution and then estimate its parameters –including the underlying network– from the data. The estimation of parameters such as branch lengths (hence speciation dates) and population sizes makes them more versatile than combinatorial methods [42]. On the down side, they usually involve high running times as they explore large parameter spaces. Two probabilistic models differentiate regarding the way a locus tree can be embedded within a network. In Kubatko's model [43,44], all lineages of a given locus tree coalesce within a single species tree *displayed* by the network. The model of Yu et al. [45] is more general as, at each reticulation node, a lineage of the locus tree is allowed to descend from a parental ancestor independently of which ancestors provide the other lineages. Works on the latter model extend in various ways the MSC model to consider network-like evolution, giving rise to the *multispecies network coalescent* (MSNC), intensively studied in recent years [38,41,46–55]. For this model, Yu et al. have shown how to compute the probability of a non-recombinant locus (*gene*) tree evolving inside a network, given the branch lengths and inheritance probabilities at each reticulation node of the network [46,48]. This opened the way to infer networks according to the well-known maximum likelihood and Bayesian statistical frameworks.

When the input data consists of multi-locus alignments, a first idea is to decompose the inference process in two steps: first, infer locus trees from their respective alignments, then look for networks that assign high probability to these trees. Following this principle, Yu et al. devised a maximum likelihood method [48], then a Bayesian sampling technique [51]. However, using locus trees as a proxy for molecular sequences loses some information contained in the alignments [16] and is subject to tree reconstruction errors. For this reasons, recent work considers jointly estimating the locus trees and the underlying network. This brings the extra advantage that better locus trees are likely to be obtained [56], but running time may become prohibitive already for inferences on few species. Wen et al. in the PhyloNet software [52] and Zhang et al. with the SpeciesNetwork method [53] both proposed Bayesian methods following this principle.

Though a number of trees for a same locus are considered during such inference processes, they are still considered one at a time, which may lead to a precision loss (and a time loss) compared to an inference process that would consider all possible trees for a given locus at once. When data consists of a set of *biallelic* markers (e.g., SNPs), the ground-breaking work of Bryant et al. [19] allows to compute likelihoods while integrating over all gene trees, under the MSC model (*i.e.*, when representing the history as a tree). This work was recently extended to the MSNC context by Zhu et al [54].

In this paper, we present a novel way to compute the probability of biallelic markers, given a network. This likelihood computation is at the heart of a Bayesian network inference method we called SnappNet, as it extends the Snapp method [19] to networks. SnappNet is available at `https://github.com/rabier/MySnappNet` and distributed as a package of the well-known Beast 2 software [57,58]. This package partly relies on code from Snapp [19] to handle sequence evolution and on code from SpeciesNetwork [53] to modify the network during the MCMC as well as to compute network priors.

Our approach differs from that of Zhang et al. [53] in that SnappNet takes a matrix of biallelic markers as input while SpeciesNetwork expects a set of alignments. Thus, the substitution models differ, as we consider only two states (alleles) while SpeciesNetwork deals with nucleotides. The computational approaches also differ as

our MCMC integrates over all locus trees for each sampled network, while SpeciesNetwork jointly samples networks and gene trees. Though summarizing the alignments by gene trees might be less flexible, this allows SpeciesNetwork to provide embeddings of the gene trees into the sampled networks, while in our approach this needs to be done in a complementary step after running SnappNet. However, managing the embeddings can also lead to computational issues as Zhang et al. report, since a topological change for the network usually requires a recomputation of the embeddings for all gene trees [53].

The SnappNet method we present here is much closer to the `MCMC_BiMarkers` method of Zhu et al. [54], which also extends the Snapp method [19] to network inference. Both methods take biallelic markers as input, rely on the same model of evolution and both sample networks in a Bayesian framework. However, they differ in two important respects: the way the Bayesian inference is conducted and, most importantly, in the algorithm to compute the likelihoods. The results we present here show that this often leads to tremendous differences in running time, but also to differences in convergence.

We note that reducing running times of model-based methods can also be done by approximating likelihoods, as done by *pseudo-likelihood* methods: the network likelihood is computed for subparts of its topology, these values being then assembled to approximate the likelihood of the full network. A decomposition of the network into rooted networks on three taxa (trinets) is proposed in the PhyloNet software [49, 59] and one into semi-directed networks on four taxa in the SNaQ method of the PhyloNetwork package [50]. Since pseudo-likelihood methods are approximate heuristics to compute a likelihood, they are usually much faster than full likelihood methods and can handle large genomic data sets. On the downside, these methods face, more often than the full-likelihood methods, serious identifiability problems since some networks simply cannot be recovered from topological substructures such as rooted triples, quartets or even embedded trees [49, 50, 60]. Here we focus on the *exact* computation of the *full* likelihood, for which identifiability issues are likely to be less serious [41, 61].

In the following, we first detail the mathematical model considered, then explain the SnappNet method, before illustrating its performances on simulated and real data.

Materials and methods

Input data

SnappNet considers as input data a matrix D containing an alignment of m biallelic markers sampled from a number of individuals. Each individual belongs to a given species. These species are in a 1-to-1 correspondence with the leaves of an unknown phylogenetic network, which is the main parameter that we wish to estimate. The markers can be SNPs or random sites sampled from chromosomes, including invariant sites. All markers are considered to be independent, so a certain distance must be preserved between genomic locations included in the matrix. We identify the two alleles with the colors red and green.

Each column D_i of the alignment corresponds to a different marker. The only information that is relevant to SnappNet's computations are the numbers of red and green alleles observed in D_i for the individuals of a given species. This implies that unphased data can be analyzed with SnappNet, as long as it is translated in the input format expected by the software.

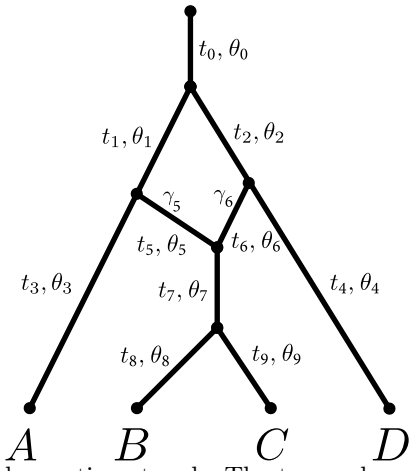


Fig 1. Example of a phylogenetic network. The top node represents the origin and its child node is called the root of the network. Time flows from the origin node to the leaves (here A, B, C, D) so branches are directed from the top to the leaves. Each branch x is associated to a length t_x, and to a population size θ_x. Additionally, branches x on top of a reticulation node have an inheritance probability γ_x representing their probability to have contributed to any individual at the top of the branch just below.

Mathematical model

In this paper, we refer to phylogenetic networks as directed acyclic graphs with branches oriented as the time flows, see Fig 1. At their extremities, networks have a single node with no incoming branch and a single outgoing branch —the *origin*— and a number of nodes with a single incoming branch and no outgoing branches —the *leaves*. All other nodes either have a single incoming branch and two outgoing branches —the *tree* nodes— or two incoming branches and a single outgoing branch —the *reticulation* nodes. Tree nodes and reticulation nodes represent speciations and hybridization events, respectively. For consistency with Zhang et al. [53], the immediate descendant of the origin – that is, the tree node representing the first speciation in the network – is called the *root*.

Each branch x in the network represents a population, and is associated to two parameters: a scaled population size θ_x and a branch length t_x. Any branch x on top of a reticulation node h is further associated with a probability $\gamma_x \in (0, 1)$, under the constraint that the probabilities of the two parent branches of h sum to 1. These probabilities are called *inheritance probabilities*. All these parameters have a role in determining how gene trees are generated by the model, and how markers evolve along these gene trees, as described in the next two subsections, respectively.

Gene tree model

Gene trees are obtained according to the MSNC model. The process starts at the leaves of the network, where a given number of lineages is sampled for each leaf, each lineage going backwards in time, until all lineages coalesce. Along the way, this process determines a gene tree whose branch lengths are each determined as the amount of time between two coalescences affecting a single lineage. Here and in what follows, "times" —and therefore branch lengths— are always measured in terms of expected number of mutations per site.

Within each branch x of the network, the model applies a standard coalescent process governed by θ_x. In detail, any two lineages within x coalesce at rate $2/\theta_x$,

meaning that the first coalescent time among k lineages follows an exponential distribution $\mathcal{E}(k(k-1)/\theta_x)$, since the coalescence of each combination of 2 lineages is equiprobable. Naturally, if the waiting time to coalescence exceeds the branch length t_x, the lineages are passed to the network branch(es) above x without coalescence. If there are two such branches y, z (i.e., the origin of x is a reticulation node), then each lineage that has arrived at the top of branch x chooses independently whether it goes to y or z with probabilities γ_y and $\gamma_z = 1 - \gamma_y$, respectively [45]. The process terminates when all lineages have coalesced and only one ancestral lineage remains.

Mutation model

As is customary for unlinked loci, we assume that the data is generated by a different gene tree for each biallelic marker. The evolution of a marker along the branches of this gene tree follows a two-states asymmetric continuous-time Markov model, scaled so as to ensure that 1 mutation is expected per time unit. This is the same model as Bryant et al. [19]. For completeness, we describe this mutation model below.

We represent the two alleles by red and green colors. Let u and v denote the instantaneous rates of mutating from red to green, and from green to red, respectively. Then, for a single lineage, $\mathbb{P}(\text{red at } t+\Delta t \mid \text{green at } t) = v\Delta t + o(\Delta t)$, and $\mathbb{P}(\text{green at } t+\Delta t \mid \text{red at } t) = u\Delta t + o(\Delta t)$, where $o(\Delta t)$ is negligible when Δt tends to zero. The stationary distribution for the allele at the root of the gene tree is green with probability $u/(u+v)$ and red with probability $v/(u+v)$. Under this model, the expected number of mutations per time unit is $2uv/(u+v)$. In order to measure time (branch lengths) in terms of expected mutations per site (i.e. genetic distance), we impose the constraint $2uv/(u+v) = 1$ as in [19]. When u and v are set to 1, the model is also known as the Haldane model [62] or the Cavender-Farris-Neyman model [63].

Bayesian framework

Posterior distribution

Let D_i be the data for the i-th marker. The posterior distribution of the phylogenetic network Ψ can be expressed as:

$$\begin{aligned}\mathbb{P}(\Psi|D_1,\ldots,D_m) &\propto \mathbb{P}\left(D_1,\ldots,D_m \mid \Psi\right)\cdot\mathbb{P}(\Psi)\\ &= \mathbb{P}(\Psi)\cdot\prod_{i=1}^{m}\mathbb{P}(D_i|\Psi) \qquad (1)\end{aligned}$$

where $\propto$ means "is proportional to", and where $\mathbb{P}\left(D_1,\ldots,D_m \mid \Psi\right)$ and $\mathbb{P}(\Psi)$ refer to the likelihood and the network prior, respectively.

Eq 1 —which relies on the independence of the data at different markers— allows us to compute a quantity proportional to the posterior by only using the prior of Ψ and the likelihoods of Ψ with respect to each marker, that is $\mathbb{P}(D_i|\Psi)$. While we could approximate $\mathbb{P}(D_i|\Psi)$ by sampling gene trees from the distribution determined by the species network, this is time-consuming and not necessary. Similarly to the work by Bryant et al. [19] for inferring phylogenetic *trees*, we show below that $\mathbb{P}(D_i|\Psi)$ can be computed for *networks* using dynamic programming.

SnappNet samples networks from their posterior distribution by using Markov chain Monte-Carlo (MCMC) based on Eq 1.

Priors

Before describing the network prior, let us recall the network components: the topology, the branch lengths, the inheritance probabilities and the populations sizes. In this context, we used the birth-hybridization process of Zhang et al. [53] to model the network topology and its branch lengths. This process depends on the speciation rate λ, on the hybridization rate ν and on the time of origin τ_0. Hyperpriors are imposed onto these parameters. An exponential distribution is used for the hyperparameters $d := \lambda - \nu$ and τ_0. The hyperparameter $r := \nu/\lambda$ is assigned a Beta distribution. We refer to [53] for more details. The inheritance probabilities are modeled according to a uniform distribution. Moreover, like SNAPP, SNAPPNET considers independent and identically distributed Gamma distributions as priors on population sizes θ_x associated to each network branch. This prior on each population size induces a prior on the corresponding coalescence rate (see [19] and SNAPP's code). Last, as in SNAPP, the user can specify fixed values for the u and v rates, or impose a prior for these rates and let them be sampled within the MCMC.

Partial likelihoods

In the next section we describe a few recursive formulae that we use to calculate the likelihood $\mathbb{P}(D_i|\Psi)$ using a dynamic programming algorithm. Here we introduce the notation that allows us to define the quantities involved in our computations. Unless otherwise stated, notations that follow are relative to the ith biallelic marker. To keep the notations light, the dependence on i is not explicit.

Given a branch x, we denote by $\overline{x}$ and $\underline{x}$ the top and bottom of that branch. We call $\overline{x}$ and $\underline{x}$ *population interfaces*. We say that two population interfaces are *incomparable* if neither is a descendant of the other (which also excludes them being equal). $N_{\overline{x}}$ and $N_{\underline{x}}$ are random variables denoting the number of gene tree lineages at the top and at the bottom of x, respectively. Similarly, $R_{\overline{x}}$ and $R_{\underline{x}}$ denote the number of red lineages at the top and bottom of x, respectively. See Fig 2 for illustration of these concepts and of the notation that we introduce in the following.

For simplicity, when x is a branch incident to a leaf, we identify $\underline{x}$ with that leaf. Two quantities that are known about each leaf are $r_{\underline{x}}$ and $n_{\underline{x}}$, which denote the number of red lineages sampled at $\underline{x}$ and the total number of lineages sampled at $\underline{x}$, respectively. Note that $N_{\underline{x}}$, in this case, is non-random: indeed, it must necessarily equal $n_{\underline{x}}$, which is determined by the number of individuals sampled from that species. On the other hand, the model we adopt determines a distribution for the $R_{\underline{x}}$. The probability of the observed values $r_{\underline{x}}$ for these random variables equals $\mathbb{P}(D_i|\Psi)$.

Now let $\mathbf{x}$ be an ordered collection (i.e. a vector) of population interfaces. We use $\mathbf{n_x}$ (or $\mathbf{r_x}$) to denote a vector of non-negative integers in a 1-to-1 correspondence with the elements of $\mathbf{x}$. Then $N_{\mathbf{x}} = \mathbf{n_x}$ is a shorthand for the equations expressing that the numbers of lineages in $\mathbf{n_x}$ are observed at their respective interfaces in $\mathbf{x}$. For example, if $\mathbf{x} = (\underline{x}, \overline{y})$ and $\mathbf{n_x} = (m, n)$, then $N_{\mathbf{x}} = \mathbf{n_x}$ is a shorthand for $N_{\underline{x}} = m, N_{\overline{y}} = n$. We use $R_{\mathbf{x}} = \mathbf{r_x}$ analogously to express the observation of the numbers of red lineages in $\mathbf{r_x}$ at $\mathbf{x}$.

In order to calculate the likelihood $\mathbb{P}(D_i|\Psi)$, we subdivide the problem into that of calculating quantities that are analogous to partial likelihoods. Given a vector of population interfaces $\mathbf{x}$, let $\mathbf{L}(\mathbf{x})$ denote a vector containing the leaves that descend from any element of $\mathbf{x}$, and let $\mathbf{r_{L(x)}}$ be the vector containing the numbers of red lineages $r_{\underline{x}}$ observed at each leaf $\underline{x}$ in $\mathbf{L}(\mathbf{x})$. Then we define:

$$\mathbf{F_x}\left(\mathbf{n_x}; \mathbf{r_x}\right) = \mathbb{P}\left(R_{\mathbf{L(x)}} = \mathbf{r_{L(x)}} \mid N_{\mathbf{x}} = \mathbf{n_x}, R_{\mathbf{x}} = \mathbf{r_x}\right) \cdot \mathbb{P}\left(N_{\mathbf{x}} = \mathbf{n_x}\right) \tag{2}$$

(see Fig 2). These quantities are generalizations of similar quantities defined by Bryant et al. [19]. We will call them partial likelihoods, although, as noted by these authors, strictly speaking this is an abuse of language.

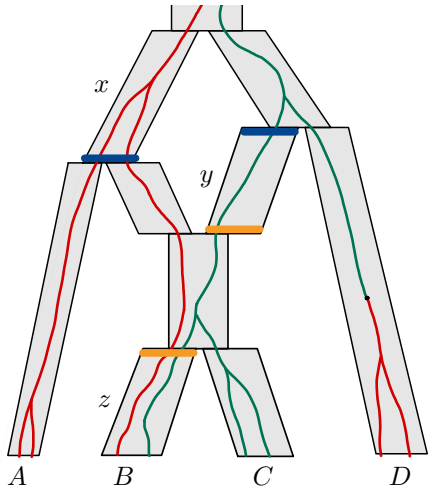


Fig 2. Illustration of the concepts and notation employed to describe likelihood computations. The species network topology is the same as that in Fig 1, but branches (populations) are now represented as grey parallelograms. A gene tree is drawn inside the species network (green and red lines). One mutation occurs in the branch above D. We focus on three branches: x, y and z. Colored horizontal bars represent the population interfaces $\underline{x}$, $\overline{y}$, $\underline{y}$ and $\overline{z}$. Note that $(\underline{x}, \overline{y})$ (blue) is a vector of incomparable population interfaces, while $(\underline{y}, \overline{z})$ (orange) is not, as $\overline{z}$ is a descendant of $\underline{y}$. Here, $n_A = n_B = n_C = n_D = 2$, $r_A = 2$, $r_B = 1$, $r_C = 0$, $r_D = 2$ are known, whereas the values of $N_{\underline{x}}, N_{\overline{y}}, N_{\underline{y}}, N_{\overline{z}}$ and $R_{\underline{x}}, R_{\overline{y}}, R_{\underline{y}}, R_{\overline{z}}$ are not observed, and depend on the gene tree generated by the MSNC process. For the gene tree shown, $N_{(\underline{x},\overline{y})} = (2, 1)$ and $R_{(\underline{x},\overline{y})} = (2, 0)$. Since z is incident to leaf B, we have $\underline{z} = B$ and $R_{\underline{z}} = r_B = 1$. Now note $\mathbf{L}((\underline{x}, \overline{y})) = (A, B, C)$. Then, $\mathbf{F}_{(\underline{x},\overline{y})}\left((n, n'); (r, r')\right) =$
$\mathbb{P}\left(R_A = r_A, R_B = r_B, R_C = r_C \mid N_{\underline{x}} = n, N_{\overline{y}} = n', R_{\underline{x}} = r, R_{\overline{y}} = r'\right) \mathbb{P}\left(N_{\underline{x}} = n, N_{\overline{y}} = n'\right)$.

Computing partial likelihoods: the rules

Here we show a set of rules that can be applied to compute partial likelihoods in a recursive way. Derivations and detailed proofs of the correctness of these rules can be found in Section 1 in S1 Text.

We use the following conventions. In all the rules that follow, vectors of population interfaces $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are allowed to be empty. The comma operator is used to concatenate vectors or append new elements at the end of vectors, for example, if $\mathbf{a} = (a_1, a_2, \ldots, a_k)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_h)$, then $\mathbf{a}, \mathbf{b} = (a_1, \ldots, a_k, b_1, \ldots, b_h)$ and $\mathbf{a}, c = (a_1, a_2, \ldots, a_k, c)$. Trivially, if $\mathbf{a}$ is empty, then $\mathbf{a}, \mathbf{b} = \mathbf{b}$ and $\mathbf{a}, c = (c)$. A vector $\mathbf{x}$ of incomparable population interfaces is one where all pairs of population interfaces are incomparable. Finally, for any branch x, let m_x denote the number of lineages sampled in the descendant leaves of x.

Rule 0: Let x be a branch incident to a leaf. Then,

$$\mathbf{F}_{(\underline{x})}\left((n); (r)\right) = \mathbb{1}\{n = n_{\underline{x}}\} \cdot \mathbb{1}\{r = r_{\underline{x}}\}$$

Rule 1: Let $\mathbf{x}, \underline{x}$ be a vector of incomparable population interfaces. Then,

$$\mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\overline{x}}; \mathbf{r}_{\mathbf{x}}, r_{\overline{x}}\right) = \sum_{n=n_{\overline{x}}}^{m_x} \sum_{r=0}^{n} \mathbf{F}_{\mathbf{x},\underline{x}}\left(\mathbf{n}_{\mathbf{x}}, n; \mathbf{r}_{\mathbf{x}}, r\right) \exp(\mathbb{Q}_x t_x)_{(n,r);(n_{\overline{x}},r_{\overline{x}})}$$

where t_x denotes the length of branch x, and $\mathbb{Q}_x$ is the rate matrix defined by Bryant et al. [19, p. 1922] that accounts for both coalescence and mutation (see also Section 1 in S1 Text).

Rule 2: Let $\mathbf{x}, \overline{x}$ and $\mathbf{y}, \overline{y}$ be two vectors of incomparable population interfaces, such that $\mathbf{L}(\mathbf{x}, \overline{x})$ and $\mathbf{L}(\mathbf{y}, \overline{y})$ have no leaf in common. Let x, y be the immediate descendants of branch z, as in Fig 3. Then,

$$\mathbf{F}_{\mathbf{x},\mathbf{y},\underline{z}}\left(\mathbf{n}_{\mathbf{x}}, \mathbf{n}_{\mathbf{y}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, \mathbf{r}_{\mathbf{y}}, r_{\underline{z}}\right) =$$

$$\sum_{n_{\overline{x}}} \sum_{r_{\overline{x}}} \mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\overline{x}}; \mathbf{r}_{\mathbf{x}}, r_{\overline{x}}\right) \mathbf{F}_{\mathbf{y},\overline{y}}\left(\mathbf{n}_{\mathbf{y}}, n_{\underline{z}} - n_{\overline{x}}; \mathbf{r}_{\mathbf{y}}, r_{\underline{z}} - r_{\overline{x}}\right) \binom{n_{\overline{x}}}{r_{\overline{x}}} \binom{n_{\underline{z}} - n_{\overline{x}}}{r_{\underline{z}} - r_{\overline{x}}} \binom{n_{\underline{z}}}{r_{\underline{z}}}^{-1}$$

The ranges of $n_{\overline{x}}$ and $r_{\overline{x}}$ in the summation terms are defined by
$\max(0, n_{\underline{z}} - m_y) \leq n_{\overline{x}} \leq \min(m_x, n_{\underline{z}})$ and
$\max(0, n_{\overline{x}} + r_{\underline{z}} - n_{\underline{z}}) \leq r_{\overline{x}} \leq \min(n_{\overline{x}}, r_{\underline{z}})$.

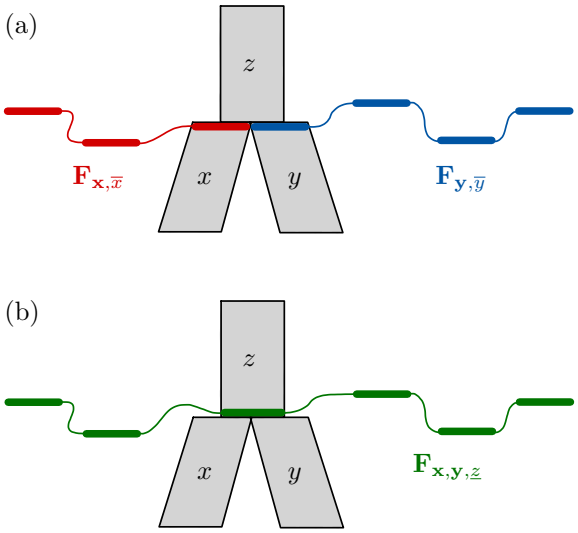


Fig 3. Illustration of Rule 2. Given (a) the partial likelihoods for the $\mathbf{x}, \overline{x}$ (red) vector of population interfaces and the partial likelihoods for the $\mathbf{y}, \overline{y}$ (blue) vector of population interfaces, Rule 2 allows us to compute the partial likelihoods for the (green) vector $\mathbf{x}, \mathbf{y}, \underline{z}$ (b).

Rule 3: Let $\mathbf{x}, \overline{x}$ be a vector of incomparable population interfaces, such that branch x's top node is a reticulation node. Let y, z be the branches immediately ancestral to x, as in Fig 4. Then,

$$\mathbf{F}_{\mathbf{x},\underline{y},\underline{z}}\left(\mathbf{n}_{\mathbf{x}}, n_{\underline{y}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, r_{\underline{y}}, r_{\underline{z}}\right) = \mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\underline{y}} + n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, r_{\underline{y}} + r_{\underline{z}}\right) \binom{n_{\underline{y}} + n_{\underline{z}}}{n_{\underline{y}}} \gamma_y^{n_{\underline{y}}} \cdot \gamma_z^{n_{\underline{z}}}$$

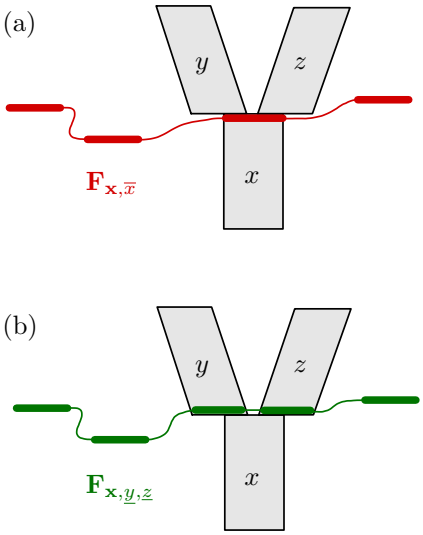


Fig 4. Illustration of Rule 3. Given (a) the partial likelihoods for the $\mathbf{x}, \overline{x}$ (red) vector of population interfaces, Rule 3 allows us to compute the partial likelihoods for the (green) vector $\mathbf{x}, \underline{y}, \underline{z}$ (b).

Rule 4: Let $\mathbf{z}, \overline{x}, \overline{y}$ be a vector of incomparable population interfaces, and let x, y be immediate descendants of branch z, as in Fig 5. Then,

$$\mathbf{F}_{\mathbf{z},\underline{z}}\left(\mathbf{n}_{\mathbf{z}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}\right) =$$
$$\sum_{n_{\overline{x}}} \sum_{r_{\overline{x}}} \mathbf{F}_{\mathbf{z},\overline{x},\overline{y}}\left(\mathbf{n}_{\mathbf{z}}, n_{\overline{x}}, n_{\underline{z}} - n_{\overline{x}}; \mathbf{r}_{\mathbf{z}}, r_{\overline{x}}, r_{\underline{z}} - r_{\overline{x}}\right) \binom{n_{\overline{x}}}{r_{\overline{x}}} \binom{n_{\underline{z}} - n_{\overline{x}}}{r_{\underline{z}} - r_{\overline{x}}} \binom{n_{\underline{z}}}{r_{\underline{z}}}^{-1}$$

The ranges of $n_{\overline{x}}$ and $r_{\overline{x}}$ in the sums are the same as those in Rule 2.

Note that, in the rules above, we assume that the vectors of population interfaces (VPIs from here on) on the right-hand side of each equation only contain incomparable population interfaces. This is necessary to ensure the validity of the rules (see Section 1 in S1 Text). It is easy to verify that, as a consequence of that assumption, also the VPIs on the left-hand side of each equation only contain incomparable population interfaces. Therefore, repeated application of the rules can only result in a partial likelihood $\mathbf{F}_{\mathbf{x}}\left(\mathbf{n}_{\mathbf{x}}; \mathbf{r}_{\mathbf{x}}\right)$ where $\mathbf{x}$ is a vector of incomparable population interfaces. All VPIs that we will encounter only contain incomparable population interfaces.

Repeated application of the rules above, performed by an algorithm described in the next subsection, leads eventually to the partial likelihoods for $\underline{\rho}$, the population interface immediately above the root of the network (i.e, ρ is the branch linking the origin to the root). From these partial likelihoods, the full likelihood $\mathbb{P}(D_i|\Psi)$ is computed as follows:

$$\mathbb{P}(D_i \mid \Psi) = \sum_{n=1}^{m_\rho} \sum_{r=0}^{n} \mathbf{F}_{(\underline{\rho})}(n; r) \cdot \mathbb{P}(R_{\underline{\rho}} = r \mid N_{\underline{\rho}} = n), \tag{3}$$

where the conditional probabilities $\mathbb{P}(R_{\underline{\rho}} = r \mid N_{\underline{\rho}} = n)$ are obtained as described by Bryant et al. [19]. Note that the length of branch ρ does not play any role in the computation of the likelihood, so it is not identifiable.

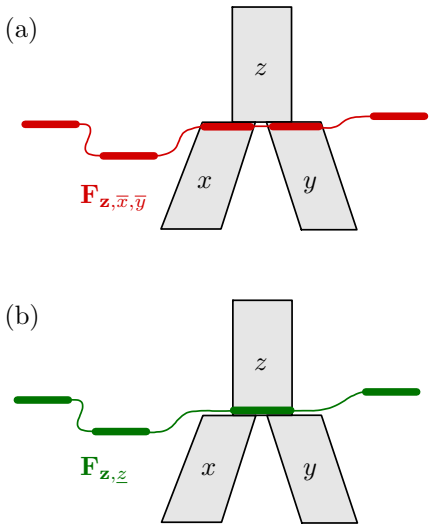


Fig 5. Illustration of Rule 4. Given (a) the partial likelihoods for the $\mathbf{z}, \overline{x}, \overline{y}$ (red) vector of population interfaces, Rule 4 allows us to compute the partial likelihoods for the (green) vector $\mathbf{z}, \underline{z}$ (b).

Likelihood computation

We now describe the algorithm that allows SnappNet to derive the full likelihood $\mathbb{P}(D_i|\Psi)$ using the rules introduced above. We refer to Section 2 in S1 Text for detailed pseudocode.

The central ingredient of this algorithm are the partial likelihoods for a VPI $\mathbf{x}$, which are stored in a matrix with potentially high dimension, denoted $\mathbf{F}_\mathbf{x}$. We say that a VPI $\mathbf{x}$ is *active* at some point during the execution of the algorithm, if: (1) $\mathbf{F}_\mathbf{x}$ has been computed by the algorithm, (2) $\mathbf{F}_\mathbf{x}$ has not yet been used to compute the partial likelihoods for another VPI. To reduce memory usage, we only store $\mathbf{F}_\mathbf{x}$ for active VPIs.

In a nutshell, the algorithm traverses each node in the network following a topological sort [64], that is, in an order ensuring that a node is only traversed after all its descendants have been traversed. Every node traversal involves deriving the partial likelihoods of a newly active VPI from those of at most two VPIs that, as a result, become inactive. Eventually, the root of the network is traversed, at which point the only active VPI is $(\underline{\rho})$ and the full likelihood of the network is computed from $\mathbf{F}_{(\underline{\rho})}$ using Eq 3.

In more detail, a node is ready to be traversed when all its child nodes have been traversed. At the beginning, only leaves can be traversed and their partial likelihoods $\mathbf{F}_{(\underline{x})}$ are obtained by application of Rule 0, followed by Rule 1 to obtain $\mathbf{F}_{(\overline{x})}$. Every subsequent traversal of a node d entails application of one rule among Rules 2, 3 or 4, depending on whether d is a tree node and on whether the branch(es) topped by d correspond to more than one VPI (see Figs 3-5). The selected rule computes $\mathbf{F}_\mathbf{x}$ for a newly active VPI $\mathbf{x}$. This is then followed by application of Rule 1 to replace every occurrence of any population interface $\underline{x}$ in $\mathbf{x}$ with $\overline{x}$.

It is helpful to note that at any moment, the set of active VPIs forms a frontier separating the nodes that have already been traversed, from those that have not yet been traversed (i.e., if branch $x = (d, e)$ with d not traversed and e traversed, then there must be an active VPI with $\underline{x}$ or $\overline{x}$ among its population interfaces). Any node that lies

immediately above this frontier can be the next one to be traversed. Thus, there is some latitude in the choice of the complete order in which nodes are traversed. Different orders will lead to different VPIs being activated by the algorithm, which in turn will lead to different running times. In fact, running times are largely determined by the sizes of the VPIs encountered. This point is explored further in the next section.

The correctness of our implementation of the algorithm above was confirmed by comparing the likelihoods we obtain to those computed with `MCMC_BiMarkers`, which also relies on biallelic marker data [54].

Time complexity of computing the likelihood

Our approach improves the running times by several orders of magnitude with respect to `MCMC_BiMarkers` [54]. This is clearly apparent for some experiments detailed in the Results section, but it can also be understood by comparing computational complexities.

Here, let n be the total number of individuals sampled, and let s denote the size of the species network Ψ (i.e. its number of branches or its number of nodes). Let us first examine the running time to process one node in Ψ. For any of Rules 0-4, let K be the number of population interfaces in the VPI for which partial likelihoods are being computed, that is, K is the number of elements of $\mathbf{x}, \overline{x}$ for Rule 1, that of $\mathbf{x}, \mathbf{y}, \underline{z}$ for Rule 2, and so on. These partial likelihoods are stored in a $2K$-dimensional matrix, with $O(n^{2K})$ elements. Each rule specifies how to compute an element of this matrix in at most $O(n^2)$ operations (in fact rules 0 and 3 only require $O(1)$ operations). Thus, any node in the network can be processed in $O(n^{2K+2})$ time.

Since the running time of any other step – i.e. computing Eq 3, and $\exp(\mathbb{Q}_x t_x)$ – is dominated by these terms, the total running time is $O(sn^{2\overline{K}+2})$, where $\overline{K}$ is the maximum number of population interfaces in a VPI activated by the given traversal.

Let us now compare this to the complexity of the likelihood computations described by Zhu et al. [54]. Processing a node d of the network in their algorithm involves at most $O(n^{4r_d+4})$ time, where r_d is the number of reticulation nodes which descend from d, and for which there exists a path from d that does not pass via a *lowest articulation node* (see definitions in Zhu et al. [54]). In Section 3 of S1 Text, we show that this entails a total running time of $O(sn^{4\ell+4})$, where ℓ is the *level* of the network [32, 65].

Thus, the improvement in running times with respect to the algorithm by Zhu et al. [54] relies on the fact that $2\overline{K}+2 \ll 4\ell+4$. One way of seeing this is to remark that, for any traversal of the network, $\overline{K} \leq \ell + 1$. We refer to Section 3 in S1 Text for a proof of this result. Assuming that $\overline{K}$ and ℓ are close, this would imply that the exponent of n in the worst-case time complexity is roughly halved with respect to Zhu et al. [54]. However, $\overline{K}$ is potentially much smaller than the level ℓ, as depicted in Fig 6.

We call the minimum value of $\overline{K}$ over all possible traversals of the network the *scanwidth* of the network [66]. The current implementation of SnappNet chooses an arbitrary traversal of the network, but research is ongoing to further lower running times by relying on more involved traversal algorithms producing VPIs with sizes closer to the scanwidth [66].

MCMC operators

SnappNet incorporates the MCMC operators of SpeciesNetwork [53] to move through the network space, and also benefits from operators specific to the mathematical model behind Snapp [19] (e.g. population sizes, mutation rates, etc.).

In order to explore the network space, we used the following topological operators

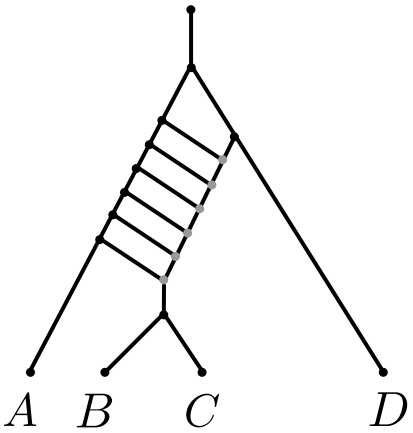


Fig 6. Example of a phylogenetic network where the level ℓ is equal to 6 (the reticulation nodes are depicted in grey), while $\overline{K} \in \{3, 4, 5, 6, 7\}$, depending on the traversal algorithm (not shown). A traversal ensuring that $\overline{K}$ remains close to the lower end of this interval (the scanwidth of the network) will be several orders of magnitude faster than algorithms whose complexity depends exponentially on ℓ. Increasing the number of reticulation nodes while keeping a "ladder" topology as above can make ℓ arbitrarily large, while the scanwidth remains constant. This topology may seem odd but it is intended as the backbone of a more complex and realistic network with subtrees hanging from the different internal branches of the ladder, in which case the complexity issue remains.

from SpeciesNetwork: (a) *addReticulation* and (b) *deleteReticulation* add and delete reticulation nodes respectively, (c) *flipReticulation* flips the direction of a reticulation branch and finally (d) *relocateBranch* and (e) *relocateBranchNarrow* relocate either the source or the destination of random branch. The operators on gene trees from SpeciesNetwork have been discarded since in SnappNet gene trees are integrated out. The following Snapp operators acting on continuous parameters are incorporated within SnappNet: (a) *changeUAndV* changes the values of the instantaneous rates u and v, (b) *changeGamma* and (c) *changeAllGamma* scale a single population size or all population sizes, respectively.

Last, SnappNet takes also advantage of a few SpeciesNetwork operators for continuous parameters: (a) *turnOverScale* and (b) *divrRateScale* allow to change respectively the hyperparameters r and d for the birth-hybridization process, (c) *inheritanceProbUniform* and (d) *inheritanceProbRndWalk* transform the inheritance probability γ at a random reticulation node by drawing either a uniformly distributed number or by applying a uniform sliding window to the logit of γ, (e) *networkMultiplier* and (f) *originMultiplier* scale respectively the heights of all internal nodes or of the origin node, (g) *nodeUniform* and (h) *nodeSlider* move the height of a random node uniformly or using a sliding window.

In summary, SnappNet relies on 16 MCMC operators, described in SnappNet's manual (`https://github.com/rabier/MySnappNet`). We refer to the original publications introducing these operators for more details [19, 53].

Simulation study

Simulated data

We implemented a simulator called SimSnappNet, an extension to networks of the SimSnapp software [19]. SimSnappNet handles the MSNC model whereas SimSnapp relies on the MSC model. SimSnappNet is available at https://github.com/rabier/SimSnappNet. In all simulations, we considered a given phylogenetic network, and a gene tree was simulated inside the network, according to the MSNC model. Next, a Markov process was generated along the branches of the gene tree, in order to simulate the evolution of a marker. Note that markers at different sites rely on different gene trees. In all cases, we set the u and v rates to 1. Moreover, we used the same $\theta = 0.005$ value, for all network branches. Our configuration differs slightly from the one of [54]. These authors considered $\theta = 0.006$ for external branches and $\theta = 0.005$ for internal branches. Indeed, since SnappNet considers the same prior distribution $\Gamma(\alpha, \beta)$ for all θ's, we found it more appropriate to generate data under SnappNet's assumptions.

Three numbers of markers were studied: 1,000, 10,000 or 100,000 biallelic sites were generated. Unless otherwise stated, constant sites were not discarded since SnappNet's mathematical formulas rely on random markers. When the analysis relied only on polymorphic sites, the gene tree and the associated marker were regenerated until it resulted in a polymorphic site. We considered 20 replicates for each simulation set up.

Phylogenetic networks studied

We studied the three phylogenetic networks shown in Fig 7. Networks A and B are rather simple networks that we wish our tool to be able to infer. They have been taken from [54] and this permits us to compare the performances of SnappNet and `MCMC_BiMarkers` on these networks, without having to rerun the latter. Networks A and B have one and two reticulations, respectively. Network C, like B, has two reticulations, but their relative positions are different: in C they are on top of one another, allowing us to investigate the influence of nested reticulations on the inference. In order to fully describe these networks, we give their extended Newick representation [67] in Section 4 in S1 Text.

We also studied networks C(3) and C(4), which are variants of network C (see Fig 8). Network C(k)—containing k reticulation nodes—is obtained by splitting species C into $k-1$ species, named C_1, C_2, ..., C_{k-1}, and by adding reticulations between them in the way depicted in Fig 8. The relative positions of reticulation nodes in these networks represents a significant computational challenge for network inference tools, and were therefore used to evaluate the efficiency of a single likelihood computation performed by SnappNet and `MCMC_BiMarkers`.

Bayesian analysis

In the experiments on networks A, B and C, we used a single tree as initial state of the MCMC. None of the starting trees were subtrees of the correct network topology. A few alternative starting trees were used to check the convergence of the MCMC, showing a limited effect of the starting tree on the posterior probabilities. All relevant Newick representations are reported in S1 Text.

As priors on population sizes, we considered $\theta \sim \Gamma(1, 200)$ for all branches. Since simulated data were generated by setting $\theta = 0.005$, the expected value of this prior distribution is exactly matching the true value ($\mathbb{E}(\theta) = 0.005$). For calibrating the

network prior, we chose the same distributions as suggested in [53]: $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1,1)$, $\tau_0 \sim \mathcal{E}(10)$. This network prior enables to explore a large network space, while imposing more weights on networks with 1 or 2 reticulations (see Fig A of S1 Text). Recall that network A is a 1-reticulation network, whereas networks B and C are 2-reticulation networks. However, in order to limit the computational burden for network C (and for estimating continuous parameters on network A), we modified slightly the prior by bounding the number of reticulations to 2. Last, on network B, the analysis was performed by bounding the number of reticulations to 3 in order to compare SnappNet's results with those obtained by `MCMC_BiMarkers` [54]. We refer to Figs B and C in S1 Text for illustrations of the "bounded" prior.

MCMC convergence

To track the behaviour of the Bayesian algorithm, we used the Effective Sample Size (ESS) criterion [68]. We assume that MCMC convergence was reached and that enough "independent" observations were sampled, when the ESS values for all model parameters are greater than 200 (see `https://beast.community/ess_tutorial`). This threshold is commonly adopted in the MCMC community. The first 10% samples were discarded as burn-in and the ESSs were computed on the remaining observations, using the Tracer software [69]. When we could not reach ESSs of 200, the ESS threshold is specified in the text. In the following, when speaking of a specific ESS value, we refer to the ESS computed for the posterior density function of the sampled networks (first value reported by Tracer). In order to estimate posterior distributions, we only sampled the MCMC every 1000 iterations. This was done to reduce autocorrelation across the sampled networks.

Note that here we do not attempt to measure an ESS of the network topologies sampled by the MCMC. While approaches to do this have been proposed for tree topologies [70], adapting such approaches to network topologies lies beyond the scope of this paper (see also the Discussion). Topological convergence was only assessed by inspecting the similarity between the results obtained for different MCMC replicates.

Accuracy of SnappNet

In order to evaluate SnappNet's ability to recover the true network topology, the posterior probability of the true topology was estimated by taking the proportion of sampled network topologies matching the true topology. Note that unlike previous works [54], we did not use a measure of topological dissimilarity, because most of the proposed measures can equal 0 even when the network topologies are different [38, 71]. In order to verify whether a sampled network and the true network have the same topologies, we used the isomorphism tester program available at `https://github.com/igel-kun/phylo_tools`. We report the average (estimated) posterior probability of the true network topology over the different replicates.

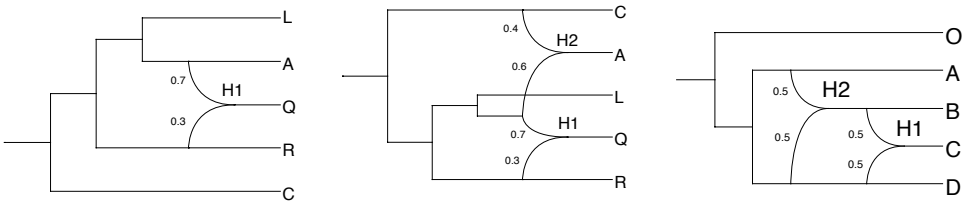
For some networks, we also investigated the ability of estimating continuous parameters, including network length (the sum of all branch lengths) and network height (the distance between the root and the leaves).

Real data study on rice

In order to assess the performance of our method on real data, we addressed the case of rice, both a prominent crop and a well-studied advanced plant model for which extensive data is available. We used genomic data extracted from [72] and [73]. We focused on 24

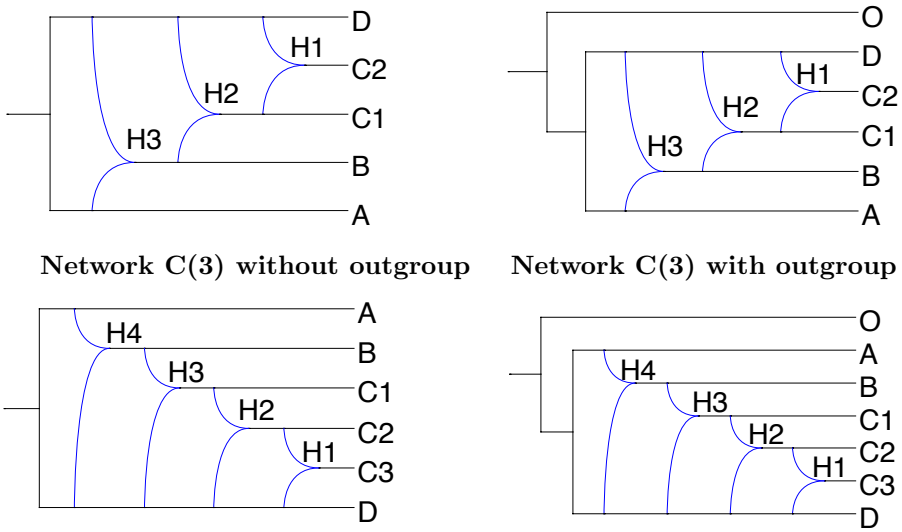
representative varieties (see Table C and Fig M in S1 Text) spanning the four main rice subpopulations of cultivars (Indica, Japonica, cAus and cBasmati) as well as the different types of wild rice *O. rufipogon* that are suspected to have been involved in the origin of cultivated rice. We built three random data sets, keeping a large panel of Asian countries. Data set 1 contains only one variety per subpopulation, whereas data sets 2 and 3 contain two varieties per subpopulation (cf. Tables D and E in S1 Text). For each of the 12 chromosomes, we sampled 1k SNPs having only homozygous alleles. Following recommendations of [19], the SNPs were chosen for each of the 12 chromosomes to be as separated as possible from one another to avoid linkage between loci, though [54] has shown this kind of analysis is quite robust to this bias. The concatenation of these SNPs lead to 12k whole-genome SNP data sets on the selected rice varieties.

SnappNet was run again discarding the first 10% of samples as burn-in and sampling the MCMC every 1000 iterations. The number of reticulations was bounded by two for data sets 1 and 3. On data set 2, in order to obtain results in a reasonable amount of time (cf. the Results section), only one reticulation was finally allowed.



Network A **Network B** **Network C**

Fig 7. The three phylogenetic networks used for simulating data. Networks A and B are taken from [54]. Branch lengths are measured in units of expected number of mutations per site (i.e. substitutions per site). Displayed values represent inheritance probabilities.



Network C(3) without outgroup **Network C(3) with outgroup**

Network C(4) without outgroup **Network C(4) with outgroup**

Fig 8. The networks from the C family, with either 3 or 4 reticulation nodes, and with or without outgroup O.

Hyperparameters \ Number of sites	Network A			Network B		
	1,000	10,000	100,000	1,000	10,000	100,000
True ($\alpha = 1$, $\beta = 200$, $\frac{\alpha}{\beta} = 0.005$)	0%	100%	100%	0%	81.25%	100%
Incorrect ($\alpha = 1$, $\beta = 1000$, $\frac{\alpha}{\beta} = 0.001$)	0%	94.73%	91.30%	0%	80%	95.65%
Incorrect ($\alpha = 1$, $\beta = 2000$, $\frac{\alpha}{\beta} = 5 \times 10^{-4}$)	0%	100%	80%	0%	85%	85.71%

Table 1. Average posterior probability of the correct topology (for networks A and B, see Fig 7) obtained by running SnappNet on simulated data. Results are given as a function of the number of sites and as a function of the hyperparameter values α and β for the prior on θ ($\theta \sim \Gamma(\alpha, \beta)$ and $\mathbb{E}(\theta) = \frac{\alpha}{\beta}$). Here, one lineage was simulated per species. Constant sites are included in the analysis, the rates u and v are considered as known, and 20 replicates are considered for each simulation set up (criterion ESS> 200 ; $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1, 1)$, $\tau_0 \sim \mathcal{E}(10)$ for the network prior).

Results

Simulations

First, we compare the performances of SnappNet and `MCMC_BiMarkers` on data simulated with networks A and B (cf. Fig 7), already studied in [54], and the more complex C network. Second, we compare the two tools in terms of CPU time and memory required to compute the likelihood of network C and its variants. This step is usually repeated million times in an MCMC analysis, and is therefore critical for its overall efficiency. Note that focusing on a single likelihood calculation allows us to exclude the effect of the prior on the overall efficiency of the MCMC, and to only test the computational efficiency of the new algorithm to compute the likelihood implemented in SnappNet.

Study of networks A and B

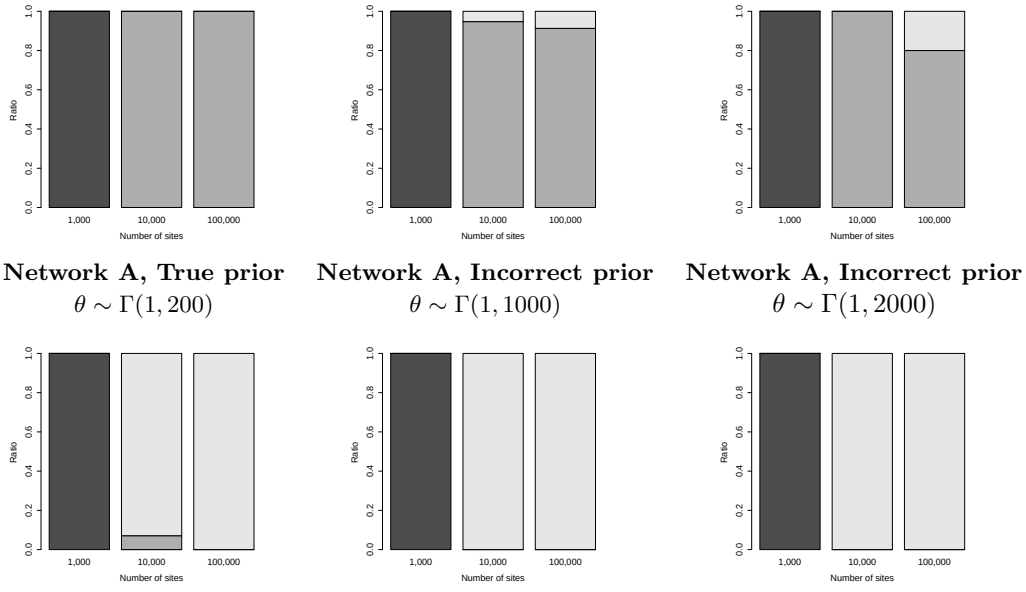
1) Ability to recover the network topology

Table 1 reports on the ability of SnappNet to recover the correct topology of networks A and B. As in [54], we simulated one individual for each species. Note that under this setting, population sizes θ corresponding to external branches are unidentifiable, as there is no coalescence event occurring along these branches. We studied different densities of markers and different priors on θ. Besides, we focused on either a) the true prior $\Gamma(1, 200)$ with $\mathbb{E}(\theta) = 0.005$, b) the incorrect prior $\Gamma(1, 1000)$ with $\mathbb{E}(\theta) = 0.001$, or c) the incorrect prior $\Gamma(1, 2000)$ with $\mathbb{E}(\theta) = 5 \times 10^{-4}$. Last, in order to compare our results with [54], we considered u and v, the mutation rates, as known parameters. Indeed, `MCMC_BiMarkers` relies on the operators of [52] that do not allow changes of these rates.

First consider simulations under the true prior. As shown in Table 1, in presence of a large number of markers, SnappNet recovered networks A and B with high posterior probability. In particular, when $m = 100,000$ sites were used, the posterior distributions were only concentrated on the true networks. For $m = 10,000$, the average posterior probability of network A is again 100%, whereas that of B is lower (81.25%). This is not surprising since network B is more complex than network A. Our results are consistent with those of [54], who found that `MCMC_BiMarkers` required 10,000 sites to infer precisely networks A and B. (Recall that we did not rerun `MCMC_BiMarkers` on data simulated from networks A and B.)

However, for a small number of sites ($m = 1,000$), we observed differences between SnappNet and `MCMC_BiMarkers`: SnappNet always inferred trees (see Fig 9), whereas `MCMC_BiMarkers` inferred networks. For instance, on Network A, `MCMC_BiMarkers` inferred a network in approximately 75% of cases, whereas SnappNet supported the tree ((((Q,A),L),R),C) with average posterior probability 78.71%. Interestingly, this tree can be obtained from network A by removing the hybridization branch with smallest inheritance probability. Details on the trees inferred by SnappNet for this setting are given in Table A of S1 Text.

Similarly, on network B that hosts 2 reticulations, for $m = 1,000$ `MCMC_BiMarkers` almost always inferred a 1-reticulation network [54], whereas SnappNet hesitated mainly between two trees, (((Q,R),L),(A,C)) and (((Q,L),R),(A,C)), with average posterior probabilities 35.28% and 28.54%, respectively. This different behavior among the two tools is most likely due to the fact that their prior models differ. With only 1,000 markers, `MCMC_BiMarkers` and SnappNet were both unable to recover network B.



Network A, True prior $\theta \sim \Gamma(1, 200)$ | **Network A, Incorrect prior** $\theta \sim \Gamma(1, 1000)$ | **Network A, Incorrect prior** $\theta \sim \Gamma(1, 2000)$

Network B, True prior $\theta \sim \Gamma(1, 200)$ | **Network B, Incorrect prior** $\theta \sim \Gamma(1, 1000)$ | **Network B, Incorrect prior** $\theta \sim \Gamma(1, 2000)$

Fig 9. The ratio of trees (black), 1-reticulation networks (dark grey), 2-reticulations networks (light gray), sampled by SnappNet, under the different simulations settings studied in Table 1. Recall that networks A and B contain 1 and 2 reticulations, respectively.

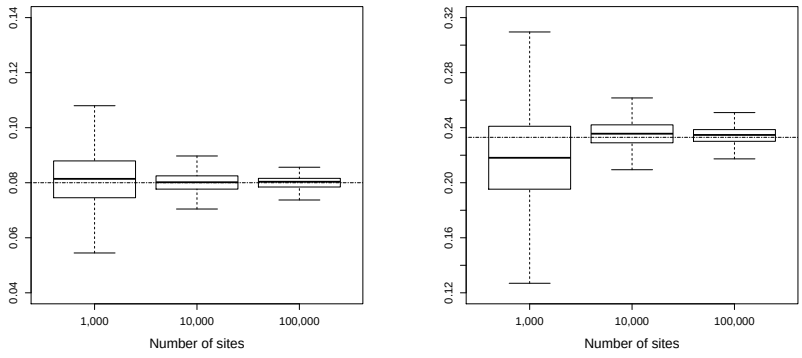
Now consider simulations based on incorrect priors. This mimics real cases where there is no or little information on the network underlying the data. Recall that these priors are incorrect since $\mathbb{E}(\theta)$ is either fixed to 0.001 or 5×10^{-4}, instead of being equal to the true value 0.005. In other words, these priors underestimate the number of ILS events in the data. When considering as few as 1,000 sites, SnappNet only inferred trees (cf. Table A in S1 Text), whereas `MCMC_BiMarkers` mostly inferred networks [54]. For $m = 10,000$ and $m = 100,000$ sites, SnappNet inferred network A with high posterior probability. In the rare cases where the true network was not sampled, SnappNet inferred a network with two reticulations (see Fig 9). The bias induced by incorrect priors (underestimating ILS) led the method to fit the data by adding supplementary edges to the network. On network B, SnappNet's posterior distribution

remained concentrated on the correct topology, and interestingly, for $m = 10,000$ and $m = 100,000$ sites, SnappNet sampled exclusively 2-reticulation networks (see Fig 9). To sum up, SnappNet's ability to recover the correct network topology did not really deteriorate with incorrect priors.

2) Ability to estimate continuous parameters for network A

Recall that in our modelling, the continuous parameters are branch lengths, inheritance probabilities γ, population sizes θ and instantaneous rates (u and v). As in [53], we also studied the network length and the network height, that is the sum of the branch lengths and the distance between the root node and the leaves, respectively. In order to evaluate SnappNet's ability to estimate continuous parameters, we will focus here exclusively on network A (following [54]). Analogous results for networks B and C can be found in Figs D-G in S1 Text.

For network A, we considered the case of two lineages in each species. Indeed, under this setting, θ values are now identifiable for external branches: the expected coalescent time is here $\theta/2$, that is to say 2.5×10^{-3}, which is a smaller value than all external branch lengths. In other words, a few coalescent events should happen along external branches. For these analyses, we considered exclusively the true prior on θ and we bounded the number of reticulations to 2 (as in [54]) in order to limit the computational burden. In the following, we consider the cases where a) input markers can be invariant or polymorphic, and b) only polymorphic sites are considered.



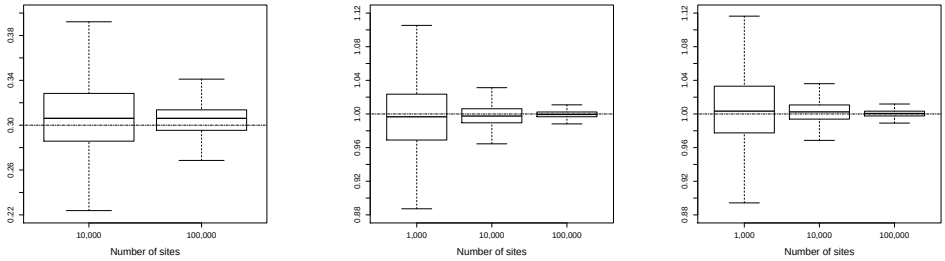
Network height **Network length**

Fig 10. Estimated height and length for network A (see Fig 7), as a function of the number of sites. Heights and lengths are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. Two lineages per species were simulated. Constant sites are included in the analysis, and 20 replicates are considered for each simulation set up (criterion ESS$>$ 200 ; $\theta \sim \Gamma(1, 200)$, $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1, 1)$, $\tau_0 \sim \mathcal{E}(10)$ for the priors, number of reticulations bounded by 2 when exploring the network space).

2a) Constant sites included in the analysis

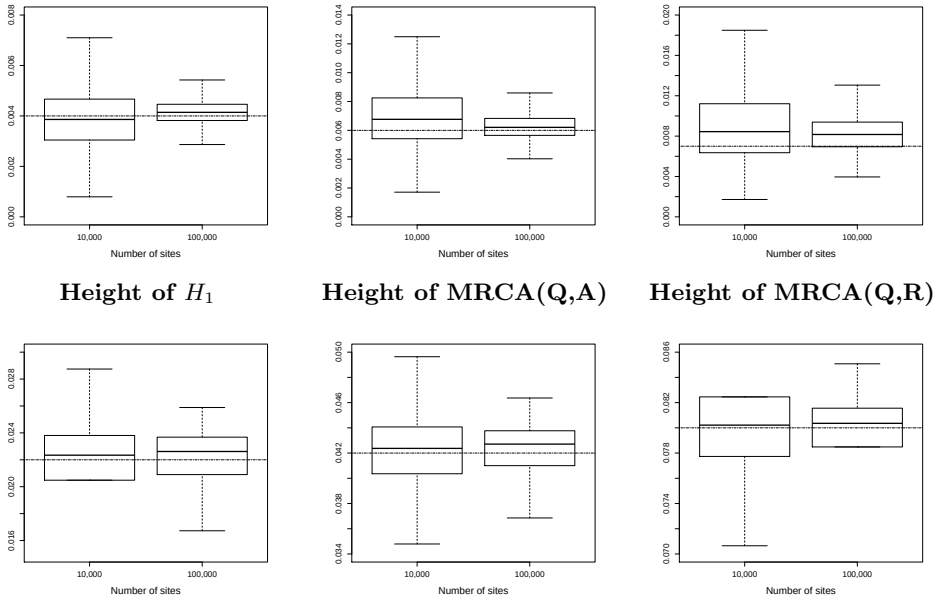
Before describing results on continuous parameters, let us first mention results regarding the topology. Although the number of lineages was increased in comparison with the previous experiment, SnappNet still sampled exclusively trees for $m = 1,000$, and always recovered the correct topology for $m = 10,000$ and $m = 100,000$. Note that for $m = 1,000$, we observed that generated data sets contained 78% invariant sites on average given the parameters of the simulation, so that such simulated data sets only

contained on average 220 variable sites.



Inheritance probability γ **Instantaneous rate u** **Instantaneous rate v**

Fig 11. Estimated inheritance probability and instantaneous rates for network A (see Fig 7), as a function of the number of sites. True values are given by the dashed horizontal lines. Same framework as in Fig 10.



Height of H_1 **Height of MRCA(Q,A)** **Height of MRCA(Q,R)**

Height of MRCA(L,A) **Height of MRCA(L,R)** **Height of MRCA(L,C)**

Fig 12. Estimated node heights of network A (see Fig 7), as a function of the number of sites. Heights are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. Same framework as in Fig 10. The initials MRCA stand for "Most Recent Common Ancestor".

In order to limit the computational burden, the analysis for $m = 100,000$ relied only on 17 replicates with ESS$>$ 200. Fig 10 reports on the estimated network height and the estimated network length. As expected, the accuracy increased with the number of sites. Fig 11 shows the same behaviour, regarding the inheritance probability γ, the rates u and v. Fig 12 is complementary to Fig 10, since it reports on the estimated node heights. All node heights were estimated quite accurately, which is not surprising in view of the results on the network length. Fig 13 is dedicated to population sizes. For

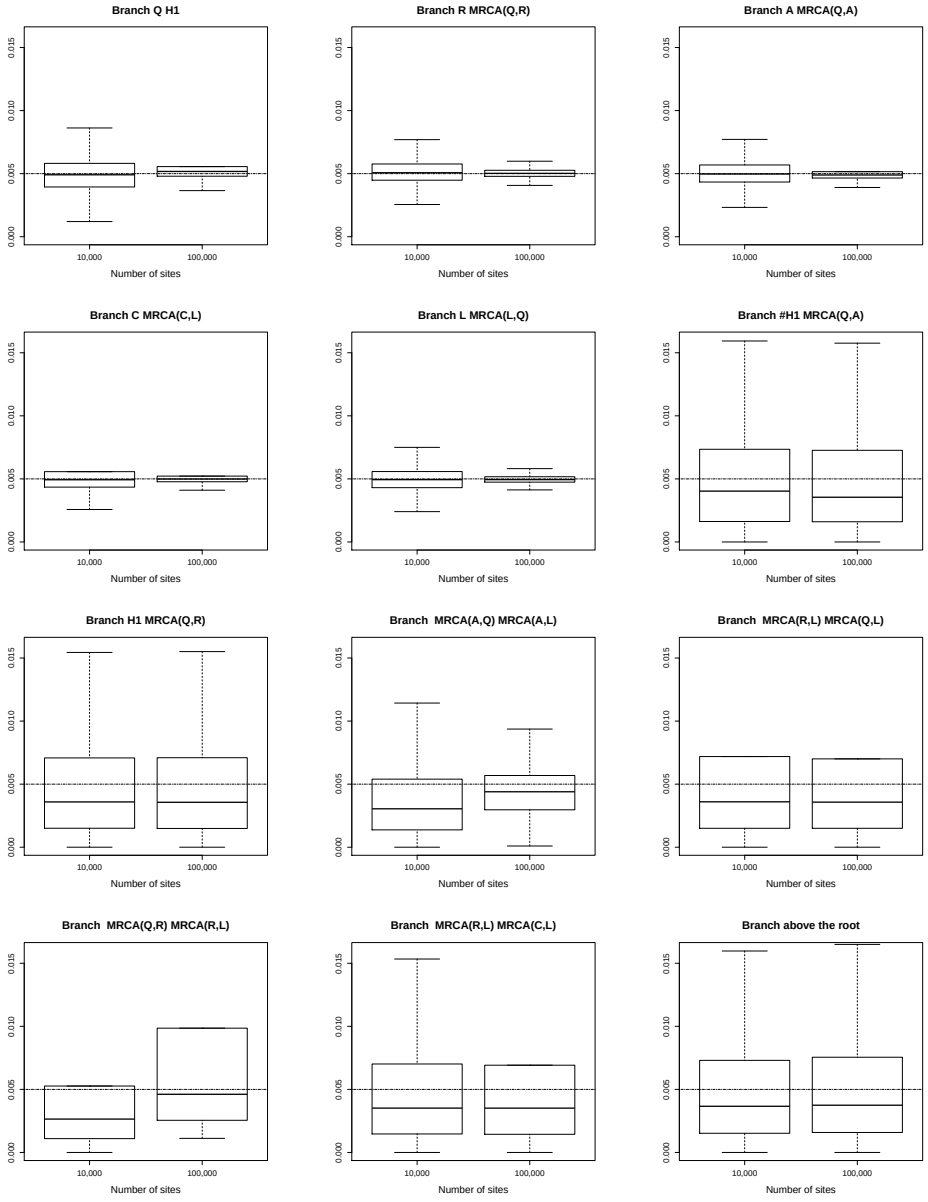


Fig 13. Estimated population sizes θ for each branch of network A (see Fig 7), as a function of the number of sites. True values are given by the dashed horizontal lines. Same framework as in Fig 10. The initials MRCA stand for "Most Recent Common Ancestor".

external branches, SnappNet's was able to estimate θ values very precisely. Performances slightly deteriorated on internal branches (see the box plots, from number 6 to number 12) whose θ values were underestimated (see the medians) and showed a higher posterior variance. This phenomenon was also observed for `MCMC_BiMarkers` [54, Fig 7 obtained under a different setting].

2b) Only polymorphic sites included in the analysis

In order to control for the fact that this analysis relies only on polymorphic sites, the

likelihood of the data for a network Ψ becomes a conditional likelihood equal to $\mathbb{P}(X_1, \ldots, X_m \mid \Psi) / \mathbb{P}(\text{“the m sites are polymorphic”}|\Psi)$, due to Bayes' rule.

Before focusing on continuous parameters, let us describe results regarding the topology. As mentioned in [54], polymorphic sites are considered as most informative to recover the topology. For $m = 1,000$, SnappNet now recovers the correct topology of network A with high frequency in 94.45% of samples). SnappNet always sampled the true network for $m = 10,000$ and $m = 100,000$. In order to reduce the computational burden for $m = 100,000$, our analysis relied on the 12 replicates that achieved ESS> 100.

Next, the same analysis was performed without applying the correction factor $\mathbb{P}(\text{“the m sites are polymorphic”}|\Psi)$, which is done by toggling an option within the software. For $m = 1,000$, the average posterior probability of network A dropped to 23.81%, while for $m = 10,000$ and $m = 100,000$, it remained relatively high (i.e., 95.24% and 95.65%, respectively). Using the correct likelihood computation is important here.

We also highlight that for $m = 100,000$, the sampler efficiency (i.e. the ratio ESS/nb iterations without burn-in) was much larger when the additional term was omitted (1.75×10^{-4} vs. 2.55×10^{-5}). It enabled us to consider 20 replicates with ESS> 200 in this new experiment.

Let us move on to the estimation of continuous parameters. Figs H-K in S1 Text illustrate results obtained from the experiment incorporating the correction factor. As previously, the network height, the network length, the rates u and v, the inheritance probability γ and the node heights were estimated very precisely. As expected, the accuracy increased with the number of sites. Estimated θ values were very satisfactory for external branches, whereas a slight bias was still introduced on internal branches. Last, for the analysis without the correction factor, we observed a huge bias regarding network height and network length (cf Fig L in S1 Text). Surprisingly, the rates u and v were still very accurately estimated.

		Number of sites		
Number of lineages for B and for C		**1,000**	**10,000**	**100,000**
1	PP	0% (20 replicates)	7.87% (20 replicates)	54.9% (20 replicates)
	SE	3.18×10^{-4}	3.47×10^{-4}	4.84×10^{-3}
4	PP	0% (20 replicates)	50.00% (18 replicates)	49.6% (8 replicates)
	SE	7.63×10^{-3}	3.89×10^{-4}	2.65×10^{-4}

Table 2. Average posterior probability (PP) of the topology of network C obtained by running SnappNet on data simulated from network C. Results are given as a function of the number of sites and as a function of the number of lineages sampled in hybrid species B and C (either both 1 or both 4). Only one lineage was sampled in every other species. Constant sites are included in the analysis and the rates u and v are considered as known. Posterior probabilities are computed on the basis of replicates for which the criterion ESS> 100 is fulfilled. The sampler efficiency (SE) is also indicated (true hyperparameter values for the prior on θ, i.e. $\theta \sim \Gamma(1, 200)$; as a network prior $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1, 1)$, $\tau_0 \sim \mathcal{E}(10)$; number of reticulations bounded by 2 when exploring the network space).

Study of network C and its variants

We focus here on network C (Fig 7) and its variants (Fig 8).

1) Ability to recover the network topology

Tables 2 and 3 report the ability of SnappNet and `MCMC_BiMarkers`, respectively,

		Number of sites		
Number of lineages for B and for C		**1,000**	**10,000**	**100,000**
1	PP	0% (20 replicates)	4.84% (20 replicates)	0% (20 replicates)
	SE	9.70×10^{-5}	3.10×10^{-5}	3.60×10^{-5}
	$\overline{\text{ESS}}$	126.08	40.38	46.80
4	PP	0% (20 replicates)	0% (12 replicates)	0% (9 replicates)
	SE	2.38×10^{-4}	8.53×10^{-5}	1.03×10^{-5}
	$\overline{\text{ESS}}$	309.00	110.90	159.96

Table 3. Average posterior probability (PP) of the topology of network C obtained by running `MCMC_BiMarkers` on data simulated from network C. Results are given as a function of the number of sites and as a function of the number of lineages sampled in hybrid species B and C (either both 1 or both 4). Only one lineage was sampled in every other species, constant sites are included in the analysis, and the rates u and v are considered as known. 1.5×10^6 iterations are considered. $\overline{\text{ESS}}$ is the average ESS over the different replicates, and SE stands for the sampler efficiency.

to recover the correct topology of network C. We considered one lineage in species O, A and D, and let the number of lineages in species B and C vary. We studied either a) 1 lineage, or b) 4 lineages, in these hybrid species. In order to limit the computational burden for SnappNet, the ESS criterion was decreased to 100 and the number of reticulations was also bounded by 2.

In order to closely mimic what was done in [54] for networks A and B, we let `MCMC_BiMarkers` run for 1,500,000 iterations instead of adopting an ESS criterion. Data were simulated with `simBiMarker` [54]. Indeed, like SimSnapp, SimSnappNet generates only count data (the number of alleles per site and per species). In contrast, `simBiMarker` generates actual sequences, a prerequisite for running `MCMC_BiMarkers`. The commands used under the 4 lineages scenario are given in Section 5 of S1 Text. Note that, to calibrate the network prior of `MCMC_BiMarkers`, the maximum number of reticulations was set to 2, and the prior Poisson distribution on the number of reticulation nodes was centered on 2.

As expected, SnappNet's ability to recover the correct network topology increased with the number of sites and with the number of lineages in the hybrid species (see Table 2). For instance, in the presence of one lineage in hybrid species B and C, the posterior probability of network C increased from 7.87% for $m = 10,000$ to 54.90% for $m = 100,000$. In the same way, when 4 lineages were considered instead of a single lineage, we observed an increase from 7.87% to 50.00% for $m = 10,000$. Note that the average posterior probability of 49.60% reported for $m = 100,000$ and 4 lineages, is based only on 8 replicates.

Surprisingly, in most cases studied, `MCMC_BiMarkers` was unable to recover the true topology of network C. The different behaviors of `MCMC_BiMarkers` and SnappNet may be due to the different network priors. Indeed, while the frequency of trees, 1-reticulation networks and 2-reticulations networks sampled by the two methods were globally similar (cf. Fig 14), we remarked that `MCMC_BiMarkers` seems to be unable, for these data sets, to sample networks with two reticulations on top of each other. Alternatively, we may be in the presence of failed or partial convergence of the MCMC process. Note the small ESS values for `MCMC_BiMarkers`, especially when only one lineage was sampled in hybrid species B and C. However, we attempted increasing the number of iterations from 1.5×10^6 to 12×10^6 and `MCMC_BiMarkers` was still unable to recover network C, despite larger ESS values (see Table B in S1 Text). We note here that SnappNet was ran for a maximum 804,000 iterations for 10,000 sites, and a

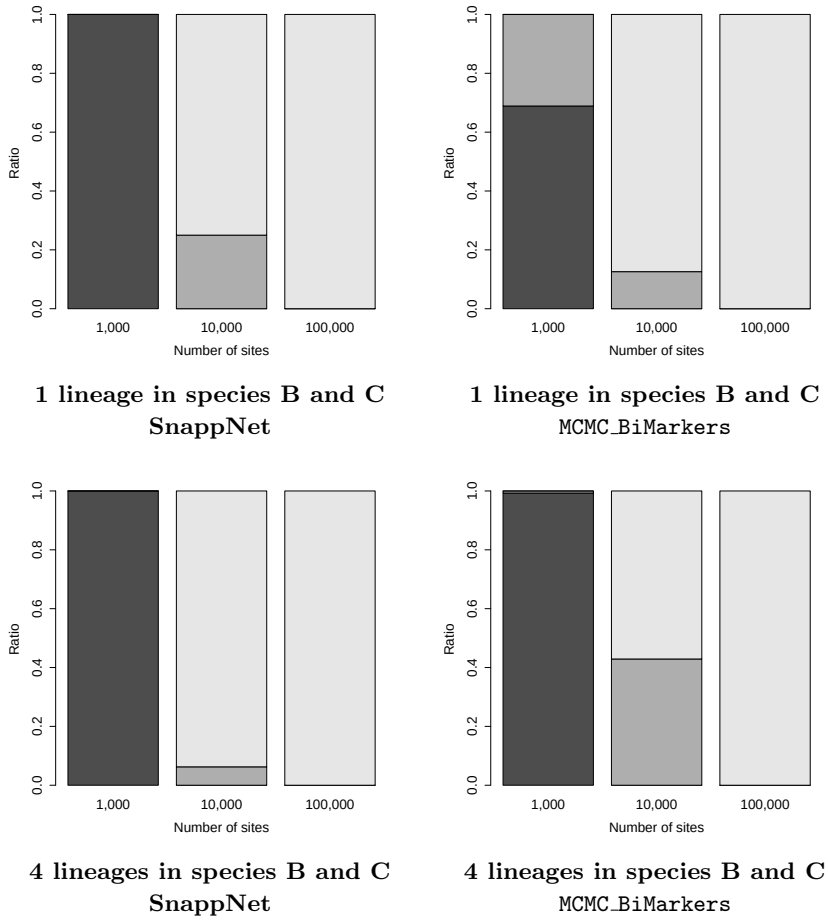


Fig 14. Frequency of trees (black), 1-reticulation networks (dark grey), 2-reticulations networks (light gray) sampled by SnappNet and `MCMC_BiMarkers`, when data were simulated from Network C (see Tables 2 and 3). Recall that network C contains 2 reticulations.

maximum of 555,000 iterations for 100,000 sites.

2) CPU time and required memory

To compare the CPU time and memory required by SnappNet and `MCMC_BiMarkers` on a single likelihood calculation, we focused on network C (see Fig 7), with and without outgroup (i.e. the species O), and networks C(3) and C(4), again with and without outgroup (see Fig 8). The simulations protocol used here is similar to that used in the previous sections, where here we fixed 10 lineages in species C and one lineage in the other species, $m = 1,000$ sites and 20 replicates per each network. The likelihood calculations were run on the true network.

The experiments were executed on a full quad socket machine with a total of 512GB of RAM (4 * 2.3 GHz AMD Opteron 6376 with 16 Cores, each with a RDIMM 32Go Quad Rank LV 1333MHz processor). The jobs that did not finish within two weeks, or required more than 128 GB, were discarded.

The results are reported in Table 4. SnappNet managed to run for all the scenarios

	CPU time		Memory	
	SnappNet (in **minutes**)	`MCMC_BiMarkers` (in **hours**)	SnappNet (max in GB)	`MCMC_BiMarkers` (max in GB)
Network C without outgroup	2.62 ± 0.04	14.58 ± 0.50	1.67 ± 0.03	8.76 ± 0.02
Network C	5.63 ± 0.16	33.46 ± 1.31	2.00 ± 0.09	8.79 ± 0.02
Network C(3) without outgroup	14.21 ± 0.56	?	2.19 ± 0.01	< 64
Network C(3)	24.69 ± 0.64	?	2.21 ± 0.06	< 64
Network C(4) without outgroup	45.47 ± 1.44	?	2.63 ± 0.60	> 128
Network C(4)	$70.98 \pm 3,16$	?	3.17 ± 0.81	> 128

Table 4. Computational efficiency of calculating a single likelihood value in SnappNet and `MCMC_BiMarkers` for networks C, C(3) and C(4). 10 lineages are sampled in species C and 1 lineage in other species. Average and standard deviation are reported.

within the two weeks limit: on average within 2.62 minutes and using 1.67 GB on network C without outgroup, within 5.63 minutes and using 2 GB on network C with outgroup, within 14.21 minutes and using 2.19 GB on network C(3) without outgroup, within 24.69 minutes and using 2.21 GB on network C(3) with outgroup, within 45.47 minutes and using 2.63 GB on network C(4) without outgroup, and finally, within 70.98 minutes and using 3.17 GB on network C(4) with outgroup.

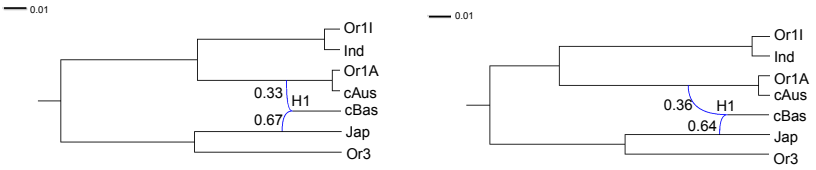
We were able to run `MCMC_BiMarkers` for all replicates of the network C, and we can thus compare its performance with that of SnappNet. From Table 4, we see that SnappNet is remarkably faster that `MCMC_BiMarkers`, needing on average only 0.29% of the time and 21% of the memory required by `MCMC_BiMarkers`. `MCMC_BiMarkers` needed more than 2 weeks for all scenarios on the C(3) network (requiring less than 64 GB), thus no run time is available for these scenarios. The same holds for the C(4) network scenarios, but for a different reason: all runs needed more than 128 GB each, and were discarded.

In Section 8 of S1 Text we provide the results of additional experiments on simulated data. In Section 8.1, we assess whether SnappNet's MCMC sampler can adequately sample from network space. In Section 8.2 we assess how population size priors and network priors influence SnappNet 's inferences.

Real data analysis

Real data derived from recent studies on rice were used to illustrate the application of SnappNet.

Diversity among Asian rice cultivars is structured around two major types which display worldwide distributions, namely Japonica and Indica, and two types localised around the Himalayas, namely *circum* Aus (cAus) and *circum* Basmati (cBasmati) [73, 74]. Japonica and Indica each have several subgroups with geographical contrast (see [73] as the most detailed description). Domestication scenarios that have been put forwards since the availability of whole genome sequences propose one to three domestications corresponding either to an early pivotal process in Japonica [72], or to multiple parallel dynamics in Japonica, Indica and cAus [12, 27], depending on whether

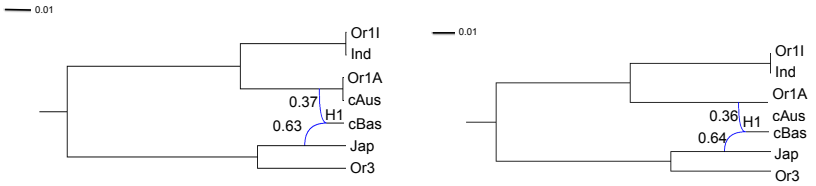


First sampling **Second sampling**

Fig 15. The two networks obtained for data set 1 with only one variety per subpopulation. Each network corresponds to the posterior mean of the distribution sampled by SnappNet. Inheritance probabilities are reported above reticulation edges and branch lengths are given in units of expected number of mutations per site (see the scale at the top left).

they consider the contribution of domestication alleles by the Japonica origin as predominant or as one among others. cBasmati has been posited as a specific lineage within Japonica [72] or as a secondary derivative from admixture between Japonica and a local wild rice close to cAus [75], or between Japonica and cAus with the contribution of one or several additional cryptic sources [76].

The most advanced studies of wild rice [72] recognize three populations designated Or-I to Or-III (Or for *Oryza rufipogon*), of which Or-I and Or-III are closely related to cultivars and Or-II is not. Using a data set constructed in [73], we compared wild rices to cultivars on the basis of ca. 2.5 million SNPs (cf. Fig M in S1 Text) and we selected representatives of Japonica, Indica, cAus and cBasmati as well as wild rices Or-III, closer to Japonica and cBasmati, and Or-I, closer either to Indica (Or-Ii) or to cAus (Or-Ia). For clarity in our subsequent use, we call the wild forms Or3, Or1I and Or1A, respectively. We made data sets of different sample sizes, including either one or two varieties per subpopulation. The studied subpopulations are the 4 groups of cultivars (Japonica, Indica, cAus, cBasmati), and different types of wild rice (Or3, Or1A, Or1I) , consistent with the classification by [72]. The 3 data sets we constructed are described in the Materials and methods.



First sampling **Second sampling**

Fig 16. The two networks obtained for data set 2 with two varieties per subpopulation. Each network corresponds to the posterior mean of the distribution sampled by SnappNet. Inheritance probabilities are reported above reticulation edges and branch lengths are given in units of expected number of mutations per site (see the scale at the top left).

In Fig 15, we report results for data set 1, which includes only one variety per subpopulation (cf. Table D in S1 Text). We studied two different samplings of 12k SNPs along the whole genome alignment. For each sampling, we ran two independent Markov chains with different starting points, for 10 million iterations. To assess the convergence of SnappNet on data set 1, (a) the ESS of the posterior distribution was checked for each chain, (b) the trace plots of the different parameters and their

associated ESS were examined and (c) the two posterior distributions corresponding to the two independent chains were compared (see Fig N and Table F in S1 Text). In view of these results, SnappNet reached stationarity. The ESS of the posterior distribution took the values 844 (resp. 971), 1159 (resp. 535) for the two different chains of the first (resp. second) sampling. All the networks sampled by the MCMC had the same topology with one reticulation only. For both genome samplings the lineages associate Or1I with Ind, Or1A with cAus and Or3 with Jap, respectively, while the reticulation conjugates Jap with (Or1A/cAus), the common precursor of Or1-A/cAus, with a dosage ratio close to 2:1, to yield cBas.

Next, we tackled a larger data set, data set 2, containing two varieties per subpopulation (see Table E in S1 Text). Two different chains corresponding to two different samplings of 12k SNPs along the whole genome alignment were run. The number of reticulations was bounded by one in order to reach convergence in a reasonable amount of time: after three months and half of computations, the ESS took the values 227 and 201 for the first and the second chain, respectively. Fig 16 illustrates the two networks obtained for the two different samplings. Each network corresponds to the posterior mean of the sampled distribution. Note that in both cases, the posterior distribution was concentrated on a single topology. The two genome samplings yield networks very similar to one another and remarkably close to that revealed with data set 1. The reticulation that was allowed again conjugates the Jap lineage with the common precursor of subpopulations Or1A and cAus. In contrast, after 6 months of calculations, SnappNet had still not reached the stationary regime for the two different samplings, when a maximum of 2 reticulations was imposed.

We also investigated another data set, data set 3, including two varieties per subpopulation and 12k SNPs for a different taxon sampling (see Table E in S1 Text). In this case, large ESS values were observed when SnappNet was allowed to infer networks with 2 reticulations: the ESS was estimated at 373 after having let SnappNet run for 7 months. The maximum a posteriori (MAP) network is represented in Fig 17. For this data set, the resulting topology again features a single reticulation, although two were allowed. It also conjugates Jap with the precursor (Or1A/cAus) of Or1A and cAus to produce cBas. Yet the composition is more unbalanced towards Jap (0.85) and (Or1A/cAus) appears involved very close to the Or1A vs cAus initial divergence. Given this proximity, it was useful to describe the three networks retained by SnappNet during the MCMC process (Fig 18). The first one (67%) features a conjugation between Jap and (Or1A/cAus), while the second one (23%) conjugates Jap with cAus and the third one (10%) conjugates Jap with Or1A in the origin of cBas.

Altogether the various networks inferred by SnappNet reveal stable features:

- the correspondence between wild subpopulations and cultivated subpopulations which point at three pillars for rice, namely Japonica, Indica and cAus
- the early divergence of Japonica, that predates the one between Indica and cAus
- the earlier divergence between wild and cultivated forms within the Japonica pillar
- the mobilisation of early Japonica cultivars to combine with the cAus pillar to produce the fourth varietal type cBas
- the indication that this hybridization may have occured before the domestication of cAus.

The latter item yet displays uneven strength levels between the various data sets. The first four items confirm the latest interpretations of massive analyses among rice specialists. Wild rice displays broad diversity and some of the wild subpopulations have

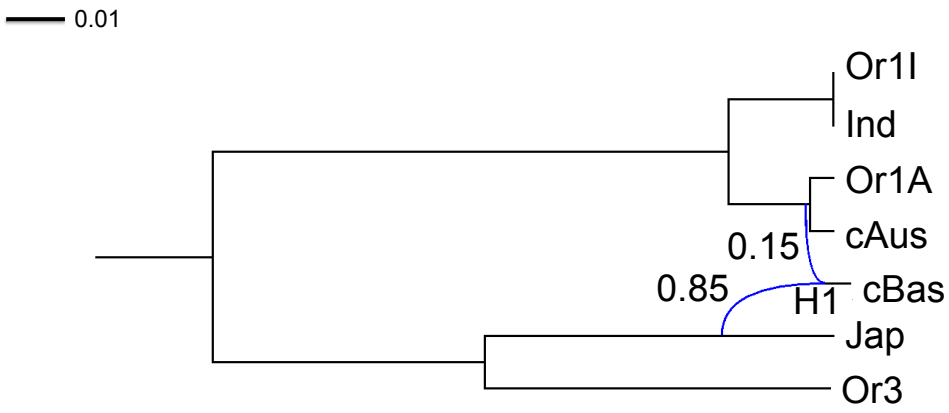
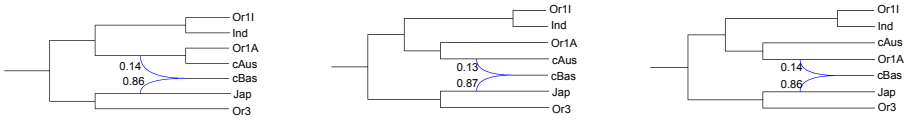


Fig 17. The MAP phylogenetic network obtained for data set 3 with two varieties per subpopulation. Inheritance probabilities are reported above reticulation edges and branch lengths are given in units of expected number of mutations per site (see the scale at the top left).



First topology 67.04%

Second topology 22.99%

Third topology 9.97%

Fig 18. The three topologies sampled by SnappNet when data set 3 was considered. Reported inheritance probabilities for each topology are averages on sampled observations.

been specifically involved in the emergence of cultivated forms. While the most ancient domestication occurred in China to produce Japonica cultivars, two other important foundations, namely Indica and cAus, contributed to the diversity of current rice cultivars. Early hybridization between Japonica cultivars and an ancestor, presumably wild, of current cAus cultivars and related wild forms resulted in the evolution of cBasmati cultivars.

Discussion

In this paper, we introduced a new Bayesian method, SnappNet, dedicated to phylogenetic network inference. SnappNet has similar goals as `MCMC_BiMarkers`, a method recently proposed by Zhu et al. [54], but differs from this method in two main aspects. The first difference is due to the way the two methods handle the complexity of the sampled networks. Unlike binary trees that have a fixed number of branches given the number of considered species, network topologies can be of arbitrary complexity. Their complexity directly depends on the number of reticulations they contain. In MCMC processes, the complexity of sampled networks is regulated by the prior. `MCMC_BiMarkers` uses descriptive priors: more precisely, it assumes a Poisson

distribution for the number of reticulation nodes and an exponential distribution for the *diameter* of reticulation nodes [51, 52, 54]. In contrast, SnappNet's prior is based on that of Zhang et al., which explicitly relies on speciation and hybridization rates and is extendable to account for extinction and incomplete sampling [53].

Our simulation study may provide some insight on the influence of these different priors. On two networks of moderate complexity (networks A and B), SnappNet and `MCMC_BiMarkers` presented globally similar results. Indeed, when we considered numbers of sites that are largely achieved in current phylogenomic studies (i.e. 10,000 or 100,000 sites), both methods were able to recover the true networks under this realistic framework. However, in presence of only a few sites (1,000 sites) which is unusual nowadays but still can be the case for poorly sequenced organisms, `MCMC_BiMarkers` recovered the correct topology with higher posterior probability than SnappNet. On the other hand, when focusing on a more complex network (network C) containing reticulation nodes on top of one another, the converse appeared to be true. With sufficiently large datasets, SnappNet recovered the correct scenario in approximately 50% of samples whereas `MCMC_BiMarkers` inferred this history in less than 5% of cases. Although these differences may be due to the different network priors used by the two methods, more work is needed to elucidate the reasons behind them. To conclude the discussion on priors, we also observed that, on simulated data, SnappNet's accuracy did not really deteriorate with incorrect priors on population sizes, although assuming a prior distribution skewed towards small population sizes has a tendency to favor hybridization over ILS as an explanation for non tree-like signals. Similar robustness properties were observed by [54] for `MCMC_BiMarkers`.

The second major difference between `MCMC_BiMarkers` and SnappNet lies in the way they compute the likelihood of a network. This step is at the core of the Bayesian analysis. According to the authors of `MCMC_BiMarkers`, this remains a major computational bottleneck and limits the applicability of their methods [59]. To understand the origin of this bottleneck, recall that the MCMC process of a Bayesian sampling explores a huge network space and that, at each exploration step, computing the likelihood is by far the most time consuming operation. Moreover, we need sometimes millions of runs before the chain converges. Thus, likelihood computation is a key factor on which to operate to be able to process large data sets.

The likelihood computation of `MCMC_BiMarkers` consists in a bottom-up traversal, from the leaves to the root. Each time a reticulation node r is visited, the partial likelihoods must be decomposed following all the possible ways the lineages reaching r can be assigned to the two parent populations of r. These partial likelihoods will be merged back only when the traversal reaches a lowest articulation node [54], or in other words the root of the blob to which r belongs (a *blob* is a maximal biconnected subgraph [65], see also S1 Text). For every other reticulation r' reached before the root of the blob, the decomposition above is applied again. As a result, the time required to process a blob grows exponentially with the number of reticulations it contains. More precisely, the time complexity of the likelihood computation in `MCMC_BiMarkers` is in $O(sn^{4\ell+4})$, where ℓ is the *level* of the network and s is the size of the species network.

Similarly to `MCMC_BiMarkers`, we compute the likelihood in a bottom-up traversal and when reaching a reticulation node r, we also take into account the various ways lineages could have split. But the originality of SnappNet is to compute *joint conditional probabilities* for branches above a same reticulation node r (see the Materials and methods). The set of branches jointly considered increases when crossing other reticulation nodes in a same blob, but it can also decrease when crossing tree-nodes in the blob (i.e. nodes having one ancestor and several children). Of course, the time to compute each partial likelihood increases in proportion with the number of

branches considered together. More precisely, SnappNet runs in $O(sn^{2\overline{K}+2})$, where $\overline{K}$ is the maximum number of branches simultaneously considered in a partial likelihood. The interest in depending on $\overline{K}$ instead of ℓ (the number of reticulations in a blob), is that for some blobs, we can resort to a bottom-up traversal of the blob that limits $\overline{K}$ to a small constant and process the blob in polynomial time in n, while `MCMC_BiMarkers` still requires an exponential time in ℓ.

Our results from simulated data confirm the above theoretical discussion. For a single likelihood evaluation, SnappNet was found to be orders of magnitude faster than `MCMC_BiMarkers` on networks containing reticulation nodes on top of one another. Besides, SnappNet required substantially less memory than `MCMC_BiMarkers`. These gains enable us to consider complex evolution scenarios in our Bayesian analyses.

In practice, SnappNet is a very useful tool for analyzing complex genomic data, as evidenced by our study about rice. Indeed, the most recent extensive genetic studies on this crop confirm and document the extent of genetic exchanges in various directions. Yet the same species consistently displays the reality of a simple classification scheme with only a few predominant types. Thus rice appears as a chance and a challenge for testing methods aiming to tackle phylogenetic resolution within a hybrid swarm. The application of SnappNet proves very efficient in resolving the three main phylogenetic pillars of current diversity in Asian rice [12, 77] and revealing a hybrid origin for the iconic varietal group cBasmati [75, 76]. The various data sets treated here suggest a contribution of Japonica cultivars at a high level, between 0.6 and 0.85. This rather broad range is not surprising given that this hybrid origin probably reflects numerous recent individual stories for very specific varieties rather than an old common story for a homogeneous lineage. On the other side, the second component of cBasmati derived from local sources in the North of the Indian subcontinent seems to date from before the evolution of cAus varieties. Here again, it is likely that many diverse events occurred resulting in a very rich diversity. Full resolution of the origin of cBasmati may require further investigation given the vast diversity it encompasses [78, 79]. SnappNet provides here a consistent and convincing set of results. Its integration in Beast may provide easier applicability than previous methods, potentially making it a method of choice to expand analysis of complex pictures generated by crop evolution and adaptation. Further applicability advantages may come from the fact that SnappNet can be used to compute the likelihoods of a set of networks of interest, and then to penalize more complex models with the AIC [80] and BIC [81] criteria.

In the future, in order to handle more sites in practice, the MSNC model should be extended to allow recombination events between loci. Recall that we have limited our rice study to 12,000 markers sampled along the genome because our model assumes independence between sampled sites, as does also Snapp's model, from which we inherit. As mentioned in the review of [38], in order to model recombination properly, the study of gene networks within species networks is an area for future research. A possibility would be to exploit previous work on Ancestral Recombination Graphs (see for instance [82]).

Another important research topic for MCMC inference of phylogenetic networks is the question of how to properly assess the autocorrelation between the topologies of the sampled networks, or, in other words, how to estimate the effective sample size (ESS) of the sampled topologies. Indeed, a large ESS for continuous parameters in a phylogenetic model does not necessarily imply a large ESS for the sampled topologies. Methods to estimate the ESS of a sample of tree topologies have been recently proposed [70]. They rely on measures of the distance between pairs of trees in the sample—which enable to assess autocorrelation—or on translating tree topologies into numbers (e.g., the distance from a focal tree), which are then treated as continuous parameters—for which an ESS

can then be computed using standard approaches. These methods to estimate topological ESS can be in principle adapted to networks. However some research will be needed for this, as standard tree metrics (e.g. the Robinson-Foulds distance [83] or the path-lengths difference [84]) do not have unique, easy to compute, natural extensions for networks (see [38] for a discussion on this). In the present work, different MCMC replicates led to consistent results, but we have not attempted to evaluate autocorrelation for the sampled topologies and/or their ESS. This is a limitation of all Bayesian approaches for network inference proposed so far [51, 53, 54].

Related to the issue above, it would be useful to conduct an in-depth investigation of the efficiency of the MCMC operators for the exploration of network topology space. In this work, we rely on the operators by Zhang et al. [53], who identified this as a major bottleneck of their approach (but they also had operators to change the gene tree embeddings, a feature that we do not need here). Although some important progress has been made in the last 20 years [85], in 2004 Felsenstein aptly wrote (speaking about trees): "At the moment the choice of a good proposal distribution involves the burning of incense, casting of chicken bones, magical incantations and invoking the opinions of more prestigious colleagues" [14]. Since network space is significantly more complex than tree space, it is easy to predict that this topic will keep researchers busy for a long time. A good starting point to address convergence issues in SnappNet would be to integrate it to the new Beast 2 package coupled MCMC [86], which tackles local optima issues thanks to heated chains.

Also note that in this work we limited our experiments to relatively simple networks, with few reticulations and few species (leaves). While the number of reticulations represents a strong limitation of all existing Bayesian approaches, the number of species is a much weaker limiting factor. Networks over more species can already be inferred by SnappNet and related approaches, but MCMC inference for such networks will then necessitate much more complex downstream analyses than the ones used here. For example, the posterior probability of any single network topology will be very small, and thus it will be much more interesting to look at the probability of individual splits, or to develop a network analog of consensus trees. These are not simple tasks, because all the underlying algorithmic problems (checking the presence of a split/clade in a network, or that of a subtree etc.) are computationally hard to solve on large networks [87].

Last, it would be interesting to study the identifiability of the model underlying SnappNet. For example, it is easy to see that if only one lineage is sampled from a given species at each locus, then the population size θ of that species is non-identifiable (because no coalescence can ever occur in it, and thus the likelihood does not depend on θ). Similarly, if only one lineage is sampled below a reticulation node, then the height of that node is non-identifiable [41, 61]. Intuitively, the more lineages can co-exist in a part of the species network, the more information there will be for the reconstruction of that part of the network. These aspects should be further investigated in future works.

Many methodological questions on Bayesian inference of phylogenetic networks remain open. The present work focused on the efficient calculation of likelihood for a single network, which is the key component of any Bayesian approach. At the end of their paper, the authors of `MCMC_BiMarkers` [54] concluded by mentioning that "An important direction for future research is improving the computational requirements of the method to scale up to data sets with many taxa". Our present work is a first answer to this demand.

Supporting information

S1 Text: Supplementary material for the manuscript

Fig A in S1 Text
Density probabilities for 5-tips networks, simulated with a prior corresponding to a birth hybridization process with parameters $d = 10$, $r = 1/2$ and $\tau_0 = 0.1$, using the SpeciesNetwork package [53]. The figure is obtained for 10,000 replicates. The means are given by the dashed vertical lines.

Fig B in S1 Text
Density probabilities for 5-tips networks with at most two reticulations, simulated with a prior corresponding to a birth hybridization process with parameters $d = 10$, $r = 1/2$ and $\tau_0 = 0.1$, using the SpeciesNetwork package [53]. Figures are drawn for the 4,377 cases in 10,000 where the network had at most two reticulations. The means are given by the dashed vertical lines.

Fig C in S1 Text
Density probabilities regarding the 5-tips network with a maximum of 3 reticulations, simulated under the birth hybridization process ($d = 10$, $r = 1/2$, $\tau_0 = 0.1$, 5,837 replicates), using the SpeciesNetwork package [53]. The means are given by the dashed vertical lines.

Fig D in S1 Text
Estimated node heights of network B. 10,000 sites are considered and 2 lineages per species. Constant sites are included in the analysis, and the estimated heights are based on the 12 replicates (over 14 replicates) for which network B was recovered by SnappNet (criterion ESS$>$ 200 ; $\theta \sim \Gamma(1, 200)$, $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1, 1)$, $\tau_0 \sim \mathcal{E}(10)$ for the priors, number of reticulations bounded by 3 when exploring the network space). Heights are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. The initials MRCA stand for "Most Recent Common Ancestor".

Fig E in S1 Text
Estimated population sizes θ for each branch of network B. Same framework as Figure D in S1 Text. True values are given by the dashed horizontal lines. The initials MRCA stand for "Most Recent Common Ancestor".

Fig F in S1 Text
Same framework as Figure E in S1 Text.

Fig G in S1 Text
Estimated node heights of network C as a function of the number of sites. Same experiment as in Table 2 of the main manuscript: 1 lineage in species O, A and D, and 4 lineages in species B and C. The estimated heights are based on the replicates for which network C was recovered by SnappNet. True values are given by the dashed horizontal lines. The initials MRCA stand for "Most Recent Common Ancestor".

Fig H in S1 Text
Estimated height and length for network A, as a function of the number of sites. Heights and lengths are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. Two lineages per species were simulated. Only polymorphic sites are included in the analysis, and 20 replicates are considered for each simulation set up (criterion ESS$>$ 200 for m=1,000 and m=10,000 ,

and criterion ESS> 100 for m=100,000; $\theta \sim \Gamma(1, 200)$, $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1, 1)$, $\tau_0 \sim \mathcal{E}(10)$ for the priors, number of reticulations bounded by 2 when exploring the network space). Same framework as in Figure 10 of the main paper, except that only polymorphic sites are taken into account.

Fig I in S1 Text
Estimated inheritance probability and instantaneous rates for network A, as a function of the number of sites. True values are given by the dashed horizontal lines. Same framework as in Figure 11 of the main paper, except that only polymorphic sites are taken into account.

Fig J in S1 Text
Estimated node heights of network A, as a function of the number of sites. Heights are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. Same framework as in Figure 12 of the main paper, except that only polymorphic sites are taken into account. The initials MRCA stand for "Most Recent Common Ancestor".

Fig K in S1 Text
Estimated population sizes θ for each branch of network A, as a function of the number of sites. True values are given by the dashed horizontal lines. Same framework as in Figure 13 of the main paper, except that only polymorphic sites are taken into account. The initials MRCA stand for "Most Recent Common Ancestor".

Fig L in S1 Text
Experiments on Network A and based only on polymorphic sites. Same framework as in Figures H and I in S1 Text, except that the correction factor is not used in the calculations (criterion ESS> 200 in all cases).

Fig M in S1 Text
Summary of rice molecular diversity used for selecting our sample of rice cultivated varieties and wild types. (A) unweighted neighbour joining (UWNJ) tree reflecting dissimilarities among 899 accessions based on 2.48 million SNPs as described in [73]; the accessions are colored according to their classification into wild population types or cultivar groups. (B, C) UWNJ tree using the same data for the 24 accessions we selected for assessing SnappNet performance, and showing their accessions number (B) and their country of origin (C); the colors are as in A.

Fig N in S1 Text
Trace plots obtained according to the Tracer software when data set 1 was analyzed with SnappNet. (a) and (b) refer to the first sampling of 12 kSNPs along the whole genome, whereas (c) and (d) focus on the second sampling. Two chains were considered for each sampling.

Fig O in S1 Text
Birth-hybridisation model with speciation rate 20 and hybridisation rate 1 (mean number of reticulations close to zero) and a normal prior with mean 0.1 and standard deviation of 0.01 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

Fig P in S1 Text
Birth-hybridisation model with speciation rate 20 and hybridisation rate 2 (mean number of reticulations close to one) and normal prior with mean 0.1 and standard

deviation of 0.01 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

Figure Q in S1 Text
Birth-hybridisation model with speciation rate 20 and hybridisation rate 3 (mean number of reticulations close to two) and normal prior with mean 0.1 and standard deviation of 0.01 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

Fig R in S1 Text
Birth-hybridisation model with speciation rate 20 and hybridisation rate 1 (mean number of reticulations close to zero) and an exponential prior with mean 0.1 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

Fig S in S1 Text
Birth-hybridisation model with speciation rate 20 and hybridisation rate 2 (mean number of reticulations close to one) and an exponential prior with mean 0.1 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

Fig T in S1 Text
Birth-hybridisation model with speciation rate 20 and hybridisation rate 3 (mean number of reticulations close to two) and an exponential prior with mean 0.1 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

Fig U in S1 Text
Summary distributions of all chains with correct population size priors (chain numbers 1,2,9,10,17,18) given data simulated from network A. We summarize the MCMC chains by combining them, that is: Chains 1 and 2 are indicated by the blue line (mean reticulations close to zero); Chains 9 and 10 are indicated by the orange line (mean reticulations close to one); Chains 17 and 18 are indicated by the green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.

Fig V in S1 Text
Summary distributions of all chains with incorrect population size priors Gamma(1,20) (chain numbers 3,4,11,12,19,20) given data simulated from network A. We summarize the MCMC chains by combining them, that is: Chains 3 and 4 are indicated by the blue line (mean reticulations close to zero); Chains 11 and 12 are indicated by the orange line (mean reticulations close to one); Chains 19 and 20 are indicated by the green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network

length used to simulate data are indicated by red lines.

Fig W in S1 Text
Summary distributions of all chains with correct population size priors (chain numbers 1,2,9,10,17,18 given data simulated under network B. We summarize the MCMC chains by combining them, that is: Chains 1 and 2 are indicated by the blue line (mean reticulations close to zero); Chains 9 and 10 are indicated by the orange line (mean reticulations close to one); Chains 17 and 18 are indicated by the green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.

Fig X in S1 Text
Summary distributions of all chains with incorrect population size priors (chain numbers 3,4,7,8,11,12) given data simulated from network B. We summarize the MCMC chains by combining them, that is: Chains 3 and 4 are indicated by blue line (mean reticulations close to zero); Chains 7 and 8 are indicated by orange line (mean reticulations close to one); Chains 11 and 12 are indicated by green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.

Fig Y in S1 Text
In this we figure we plot summary distributions of all chains with incorrect population size priors Gamma(1,20) (chain numbers 5,6,13,14,21,22) given data simulated from Network B. We summarize the MCMC chains by combining them, that is: Chains 5 and 6 are indicated by blue line (mean reticulations close to zero); Chains 13 and 14 are indicated by orange line (mean reticulations close to one); Chains 21 and 22 are indicated by green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.

Fig Z in S1 Text
In this we figure we plot summary distributions of all chains with incorrect population size priors (chain numbers 7,8,15,16,23,24) given data simulated from network B. We summarize the MCMC chains by combining them, that is: Chain 7 and 8 are indicated by blue line (mean reticulations close to zero); Chain 15 and 16 is indicated by orange line (mean reticulations close to one); Chain 23 and 24 are indicated by green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.

Table A in S1 Text
Table linked to Table 1 of the main manuscript. Trees inferred by SnappNet when m=1,000 sites were considered.

Table B in S1 Text
Average posterior probability (PP) of the topology of network C obtained by running `MCMC_BiMarkers` on data simulated from network C. Same as Table 3 of the main manuscript except that 12×10^6 iterations are considered, and only one lineage is sampled in hybrid species B and C. $\overline{\text{ESS}}$ is the average ESS over the different replicates, and SE stands for the sampler efficiency.

Table C in S1 Text

Description of the 24 rice varieties considered in our study. These varieties are either representative cultivars spanning the four main rice subpopulations (Indica, Japonica, *circum* Aus and *circum* Basmati), or wild types (Or1I, Or1A, Or3).

Table D in S1 Text
Data set 1, that includes only one variety per subpopulation. These varieties were chosen from Table C in S1 Text.

Table E in S1 Text
Data sets 2 and 3, that include two varieties per subpopulation. These varieties were chosen from Table C in S1 Text.

Table F in S1 Text
Informations obtained according to the Tracer software, when data set 1 was analyzed with SnappNet. Two different samplings of 12 kSNPs were considered, and also two chains for each sampling.

Table G in S1 Text
BH(birth rate, hybridisation rate) refers to the birth-hybridisation process of Zhang et al. with the specified birth and hybridisation rates. For data simulated with network A, only chains 1,2,3,4,9,10,11,12,17,18,19,20 were run. We indicate the mean number of reticulation for the Birth-Hybridization model given an exponential prior with mean 0.1 on network origin. Note that we only used the exponential prior in the experiment in Section 8.2 of S1 Text.

Table H in S1 Text
MCMC summary statistics for network A (correct population size priors).

Table I in S1 Text
MCMC summary statistics for network A (incorrect priors).

Table J in S1 Text
MCMC summary statistics for Network B (correct population size priors).

Table K in S1 Text
MCMC summary statistics for Network B (incorrect population size priors Gamma(1,20)).

Table L in S1 Text
MCMC summary statistics for Network B (incorrect population size priors Gamma(1,1000)).

Table M in S1 Text
MCMC summary statistics for Network B (incorrect population size priors Gamma(1,2000)).

Table N in S1 Text
MCMC acceptance rates for Network B (correct population size priors).

Table O in S1 Text
MCMC acceptance rates for Network B (incorrect population size priors $\Gamma(1, 1000)$).

Table P in S1 Text
MCMC acceptance rates for Network B (incorrect population size priors $\Gamma(1, 2000)$).

Acknowledgments

VB thanks University Montpellier for a 6-month sabbatical period.

References

1. Denoeud F, Carretero-Paulet L, Dereeper A, Droc G, Guyot R, Pietrella M, et al. The coffee genome provides insight into the convergent evolution of caffeine biosynthesis. Science. 2014;345(6201):1181–1184.

2. Badouin H, Gouzy J, Grassa CJ, Murat F, Staton SE, Cottret L, et al. The sunflower genome provides insights into oil metabolism, flowering and Asterid evolution. Nature. 2017;546(7656):148.

3. Garsmeur O, Droc G, Antonise R, Grimwood J, Potier B, Aitken K, et al. A mosaic monoploid reference sequence for the highly complex genome of sugarcane. Nature Communications. 2018;9(1):2638.

4. Cornillot E, Hadj-Kaddour K, Dassouli A, Noel B, Ranwez V, Vacherie B, et al. Sequencing of the smallest Apicomplexan genome from the human pathogen Babesia microti. Nucleic Acids Research. 2012;40(18):9102–9114.

5. Marra NJ, Stanhope MJ, Jue NK, Wang M, Sun Q, Bitar PP, et al. White shark genome reveals ancient elasmobranch adaptations associated with wound healing and the maintenance of genome stability. Proceedings of the National Academy of Sciences. 2019;116(10):4446–4455.

6. Consortium IH, et al. The international HapMap project. Nature. 2003;426(6968):789.

7. 3 RGP. The 3,000 rice genomes project. GigaScience. 2014;3(1):2047–217X.

8. Bamshad MJ, Ng SB, Bigham AW, Tabor HK, Emond MJ, Nickerson DA, et al. Exome sequencing as a tool for Mendelian disease gene discovery. Nature Reviews Genetics. 2011;12(11):745.

9. Mansueto L, Fuentes RR, Chebotarov D, Borja FN, Detras J, Abriol-Santos JM, et al. SNP-Seek II: A resource for allele mining and analysis of big genomic data in Oryza sativa. Current Plant Biology. 2016;7:16–25.

10. Hernandez RD, Kelley JL, Elyashiv E, Melton SC, Auton A, McVean G, et al. Classic selective sweeps were rare in recent human evolution. Science. 2011;331(6019):920–924.

11. Gravel S, Henn BM, Gutenkunst RN, Indap AR, Marth GT, Clark AG, et al. Demographic history and rare allele sharing among human populations. Proceedings of the National Academy of Sciences. 2011;108(29):11983–11988.

12. Civáň P, Craig H, Cox CJ, Brown TA. Three geographically separate domestications of Asian rice. Nature Plants. 2015;1(11):15164.

13. Rouard M, Droc G, Martin G, Sardos J, Hueber Y, Guignon V, et al. Three new genome assemblies support a rapid radiation in Musa acuminata (wild banana). Genome Biology and Evolution. 2018;10(12):3129–3140.

14. Felenstein J. Inferring phylogenies. vol. 2. Sinauer associates Sunderland, MA; 2004.

15. Kingman JF. On the genealogy of large populations. Journal of Applied Probability. 1982;19(A):27–43.

16. Rannala B, Yang Z. Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. Genetics. 2003;164(4):1645–1656.

17. Knowles LL, Kubatko LS. Estimating species trees: practical and theoretical aspects. John Wiley and Sons; 2011.

18. RoyChoudhury A, Felsenstein J, Thompson EA. A two-stage pruning algorithm for likelihood computation for a population tree. Genetics. 2008;180(2):1095–1105.

19. Bryant D, Bouckaert R, Felsenstein J, Rosenberg NA, RoyChoudhury A. Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. Molecular Biology and Evolution. 2012;29(8):1917–1932.

20. Ebersberger I, Galgoczy P, Taudien S, Taenzer S, Platzer M, Von Haeseler A. Mapping human genetic ancestry. Molecular Biology and Evolution. 2007;24(10):2266–2276.

21. Degnan JH, Rosenberg NA. Gene tree discordance, phylogenetic inference and the multispecies coalescent. Trends in Ecology & Evolution. 2009;24(6):332–340.

22. Maddison WP. Gene Trees in Species Trees. Systematic Biology. 1997 09;46(3):523–536.

23. Mallet J. Hybrid speciation. Nature. 2007;446(7133):279.

24. Morales L, Dujon B. Evolutionary role of interspecies hybridization and genetic exchanges in yeasts. Microbiology and Molecular Biology Reviews. 2012;76(4):721–739.

25. Cui R, Schumer M, Kruesi K, Walter R, Andolfatto P, Rosenthal GG. Phylogenomics reveals extensive reticulate evolution in Xiphophorus fishes. Evolution. 2013;67(8):2166–2179.

26. Glemin S, Scornavacca C, Dainat J, Burgarella C, Viader V, Ardisson M, et al. Pervasive hybridizations in the history of wheat relatives. Science Advances. 2019;5(5):eaav9188.

27. Civáň P, Brown TA. Role of genetic introgression during the evolution of cultivated rice (Oryza sativa L.). BMC Evolutionary Biology. 2018;18(1):57.

28. Minamikawa MF, Nonaka K, Kaminuma E, Kajiya-Kanegae H, Onogi A, Goto S, et al. Genome-wide association study and genomic prediction in citrus: potential of genomics-assisted breeding for fruit quality traits. Scientific Reports. 2017;7(1):4721.

29. Duranton M, Allal F, Fraïsse C, Bierne N, Bonhomme F, Gagnaire PA. The origin and remolding of genomic islands of differentiation in the European sea bass. Nature Communications. 2018;9(1):2518.

30. Koonin EV, Makarova KS, Aravind L. Horizontal gene transfer in prokaryotes: quantification and classification. Annual Reviews in Microbiology. 2001;55(1):709–742.

31. Szöllősi GJ, Davín AA, Tannier E, Daubin V, Boussau B. Genome-scale phylogenetic analysis finds extensive gene transfer among fungi. Phil Trans R Soc B. 2015;370(1678):20140335.

32. Huson DH, Rupp R, Scornavacca C. Phylogenetic networks: concepts, algorithms and applications. Cambridge University Press; 2010.

33. Nakhleh L. Evolutionary phylogenetic networks: models and issues. In: Problem solving handbook in computational biology and bioinformatics. Springer; 2010. p. 125–158.

34. Morrison DA. Introduction to Phylogenetic Networks. RJR Productions; 2011.

35. Baroni M, Semple C, Steel M. A framework for representing reticulate evolution. Annals of Combinatorics. 2005;8(4):391–408.

36. Hudson RR. Properties of a neutral allele model with intragenic recombination. Theoretical Population Biology. 1983;23(2):183–201.

37. Huson DH, Scornavacca C. A survey of combinatorial methods for phylogenetic networks. Genome Biology and Evolution. 2011;3:23–35.

38. Degnan JH. Modeling hybridization under the network multispecies coalescent. Systematic Biology. 2018;67(5):786–799.

39. Fontaine MC, Pease JB, Steele A, Waterhouse RM, Neafsey DE, Sharakhov IV, et al. Extensive introgression in a malaria vector species complex revealed by phylogenomics. Science. 2015;347(6217):1258524.

40. Marcussen T, Sandve SR, Heier L, Spannagl M, Pfeifer M, Jakobsen KS, et al. Ancient hybridizations among the ancestral genomes of bread wheat. Science. 2014;345(6194):1250092.

41. Zhu S, Degnan JH. Displayed trees do not determine distinguishability under the network multispecies coalescent. Systematic Biology. 2017;66(2):283–298.

42. Huson DH, Scornavacca C. A Survey of Combinatorial Methods for Phylogenetic Networks. Genome Biology and Evolution. 2010 11;3:23–35. Available from: https://doi.org/10.1093/gbe/evq077.

43. Kubatko LS. Identifying hybridization events in the presence of coalescence via model selection. Systematic Biology. 2009;58(5):478–488.

44. Meng C, Kubatko LS. Detecting hybrid speciation in the presence of incomplete lineage sorting using gene tree incongruence: a model. Theoretical Population Biology. 2009;75(1):35–45.

45. Yu Y, Than C, Degnan JH, Nakhleh L. Coalescent histories on phylogenetic networks and detection of hybridization despite incomplete lineage sorting. Systematic Biology. 2011;60(2):138–149.

46. Yu Y, Degnan JH, Nakhleh L. The probability of a gene tree topology within a phylogenetic network with applications to hybridization detection. PLoS Genetics. 2012;8(4):e1002660.

47. Yu Y, Ristic N, Nakhleh L; BioMed Central. Fast algorithms and heuristics for phylogenomics under ILS and hybridization. BMC bioinformatics. 2013;14(15):S6.

48. Yu Y, Dong J, Liu KJ, Nakhleh L. Maximum likelihood inference of reticulate evolutionary histories. Proceedings of the National Academy of Sciences. 2014;111(46):16448–16453.

49. Yu Y, Nakhleh L. A maximum pseudo-likelihood approach for phylogenetic networks. BMC Genomics. 2015;16(10):S10.

50. Solís-Lemus C, Ané C. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. PLoS Genetics. 2016;12(3):e1005896.

51. Wen D, Yu Y, Nakhleh L. Bayesian inference of reticulate phylogenies under the multispecies network coalescent. PLoS Genetics. 2016;12(5):e1006006.

52. Wen D, Nakhleh L. Coestimating reticulate phylogenies and gene trees from multilocus sequence data. Systematic Biology. 2018;67(3):439–457.

53. Zhang C, Ogilvie HA, Drummond AJ, Stadler T. Bayesian inference of species networks from multilocus sequence data. Molecular Biology and Evolution. 2017;35(2):504–517.

54. Zhu J, Wen D, Yu Y, Meudt HM, Nakhleh L. Bayesian inference of phylogenetic networks from bi-allelic genetic markers. PLoS Computational Biology. 2018;14(1):e1005932.

55. Elworth RL, Ogilvie HA, Zhu J, Nakhleh L. Advances in computational methods for phylogenetic networks in the presence of hybridization. In: Bioinformatics and Phylogenetics. Springer; 2019. p. 317–360.

56. Bayzid MS, Warnow T. Naive binning improves phylogenomic analyses. Bioinformatics. 2013;29(18):2277–2284.

57. Bouckaert R, Heled J, Kühnert D, Vaughan T, Wu C, Xie D, et al. BEAST 2: a software platform for Bayesian evolutionary analysis. PLoS Computational Biology. 2014;10(4):e1003537.

58. Bouckaert R, Vaughan TG, Barido-Sottani J, Duchêne S, Fourment M, Gavryushkina A, et al. BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis. PLoS Computational Biology. 2019;15(4):e1006650.

59. Zhu J, Nakhleh L. Inference of species phylogenies from bi-allelic markers using pseudo-likelihood. Bioinformatics. 2018;34(13):i376–i385.

60. Pardi F, Scornavacca C. Reconstructible phylogenetic networks: do not distinguish the indistinguishable. PLoS Computational Biology. 2015;11(4):e1004135.

61. Cao Z, Liu X, Ogilvie HA, Yan Z, Nakhleh L. Practical aspects of phylogenetic network analysis using PhyloNet. bioRxiv. 2019;p. 746362.

62. Haldane J. The combination of linkage values and the calculation of distances between the loci of linked factors. Journal of Genetics. 1919;8(29):299–309.

63. Cavender JA. Taxonomy with confidence. Mathematical Biosciences. 1978;40(3-4):271–280.

64. Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms, Third Edition. 3rd ed. The MIT Press; 2009.

65. Gambette P, Berry V, Paul C. The structure of level-k phylogenetic networks. In: Annual Symposium on Combinatorial Pattern Matching. Springer; 2009. p. 289–300.

66. Berry V, Scornavacca C, Weller M. Scanning Phylogenetic Networks is NP-hard. International Conference on Current Trends in Theory and Practice of Informatics. Springer; 2020. p. 519–530.

67. Cardona G, Rosselló F, Valiente G. Extended Newick: it is time for a standard representation of phylogenetic networks. BMC Bioinformatics. 2008;9(1):532.

68. Liu JS. Monte Carlo strategies in scientific computing. Springer Science & Business Media; 2008.

69. Rambaut A, Drummond AJ, Xie D, Baele G, Suchard MA. Posterior summarization in Bayesian phylogenetics using Tracer 1.7. Systematic Biology. 2018;67(5):901–904.

70. Lanfear R, Hua X, Warren DL. Estimating the effective sample size of tree topologies from Bayesian phylogenetic analyses. Genome Biology and Evolution. 2016;8(8):2319–2332.

71. Nakhleh L. A metric on the space of reduced phylogenetic networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2009;7(2):218–222.

72. Huang X, Kurata N, Wang ZX, Wang A, Zhao Q, Zhao Y, et al. A map of rice genome variation reveals the origin of cultivated rice. Nature. 2012;490(7421):497.

73. Wang W, Mauleon R, Hu Z, Chebotarov D, Tai S, Wu Z, et al. Genomic variation in 3,010 diverse accessions of Asian cultivated rice. Nature. 2018;557(7703):43–49.

74. Glaszmann JC. Isozymes and classification of Asian rice varieties. Theoretical and Applied genetics. 1987;74(1):21–30.

75. Civáň P, Ali S, Batista-Navarro R, Drosou K, Ihejieto C, Chakraborty D, et al. Origin of the aromatic group of cultivated rice (Oryza sativa L.) traced to the Indian subcontinent. Genome Biology and Evolution. 2019;11(3):832–843.

76. Santos JD, Chebotarov D, McNally KL, Bartholomé J, Droc G, Billot C, et al. Fine scale genomic signals of admixture and alien introgression among Asian rice landraces. Genome Biology and Evolution. 2019;11(5):1358–1373.

77. Civáň P, Brown TA. Misconceptions regarding the role of introgression in the origin of Oryza sativa subsp. indica. Frontiers in Plant Science. 2018;9:1750.

78. Myint KM, Courtois B, Risterucci AM, Frouin J, Soe K, Thet KM, et al. Specific patterns of genetic diversity among aromatic rice varieties in Myanmar. Rice. 2012;5(1):1–13.

79. Choi JY, Lye ZN, Groen SC, Dai X, Rughani P, Zaaijer S, et al. Nanopore sequencing-based genome assembly and evolutionary genomics of circum-basmati rice. Genome Biology. 2020;21(1):21.

80. Akaike H. Information theory and an extension of the maximum likelihood principle. In: Selected papers of hirotugu akaike. Springer; 1998. p. 199–213.

81. Schwarz G, et al. Estimating the dimension of a model. The Annals of Statistics. 1978;6(2):461–464.

82. Gusfield D. ReCombinatorics: the algorithmics of ancestral recombination graphs and explicit phylogenetic networks. MIT press; 2014.

83. Robinson DF, Foulds LR. Comparison of phylogenetic trees. Mathematical Biosciences. 1981;53(1-2):131–147.

84. Steel MA, Penny D. Distributions of tree comparison metrics—some new results. Systematic Biology. 1993;42(2):126–141.

85. Lakner C, Van Der Mark P, Huelsenbeck JP, Larget B, Ronquist F. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. Systematic Biology. 2008;57(1):86–103.

86. Mueller NF, Bouckaert R. Adaptive Metropolis-coupled MCMC for BEAST 2. PeerJ. 2020;8:e9473.

87. Kanj IA, Nakhleh L, Than C, Xia G. Seeing the trees and their branches in the network is hard. Theoretical Computer Science. 2008;401(1-3):153–164.

S1 Text. Supplementary material for the manuscript **"On the inference of complex phylogenetic networks by Markov Chain Monte-Carlo"**

August 6, 2021

Contents

1 A closer look at the rules

Here, we first provide proofs of correctness for the rules to compute the partial likelihoods introduced in the main text (Sec. 1.1). Then we explain the rationale behind the ranges used for the summation terms in Rules 2 and 4 (Sec. 1.2).

1.1 Correctness of the rules for partial likelihoods.

Recall the definition of the partial likelihoods, which will be used in each of the proofs below:

$$\mathbf{F}_{\mathbf{x}}\left(\mathbf{n}_{\mathbf{x}};\mathbf{r}_{\mathbf{x}}\right)=\mathbb{P}\left(R_{\mathbf{L}(\mathbf{x})}=\mathbf{r}_{\mathbf{L}(\mathbf{x})}\mid N_{\mathbf{x}}=\mathbf{n}_{\mathbf{x}},R_{\mathbf{x}}=\mathbf{r}_{\mathbf{x}}\right)\times\mathbb{P}\left(N_{\mathbf{x}}=\mathbf{n}_{\mathbf{x}}\right),\tag{1}$$

where $\mathbf{L}(\mathbf{x})$ is a vector of population interfaces (VPI) containing exactly once each leaf that descends from any element of $\mathbf{x}$.

We will also use the following equation (proven by Bryant et al. [2, 3] and based on [4]):

$$\mathbb{P}\left(R_{\underline{x}}=r_{\underline{x}}\mid N_{\underline{x}}=n_{\underline{x}},N_{\overline{x}}=n_{\overline{x}},R_{\overline{x}}=r_{\overline{x}}\right)=\frac{\exp(\mathbb{Q}_x t_x)_{(n_{\underline{x}},r_{\underline{x}});(n_{\overline{x}},r_{\overline{x}})}}{\mathbb{P}(N_{\overline{x}}=n_{\overline{x}}\mid N_{\underline{x}}=n_{\underline{x}})}\tag{2}$$

where $\mathbb{Q}_x=(q_{(n,r);(n',r')})$ denotes the matrix with the following entries:

$$\begin{aligned}
q_{(n,r);(n,r-1)}&=(n-r+1)v && 0<r\leq n,\\
q_{(n,r);(n,r+1)}&=(r+1)u && 0\leq r<n,\\
q_{(n,r);(n-1,r)}&=\frac{(n-1-r)n}{\theta_x} && 0\leq r<n,\\
q_{(n,r);(n-1,r-1)}&=\frac{(r-1)n}{\theta_x} && 0\leq r\leq n,\\
q_{(n,r);(n,r)}&=-\frac{n(n-1)}{\theta_x}-(n-r)v-ru && 0\leq r\leq n,\\
q_{(n,r);(n',r')}&=0 && \text{for all other entries.}
\end{aligned}$$

Finally, we note that many statements of conditional independence that we require in our proofs depend on the fact that the involved VPIs are incomparable.

Rule 0. *Let x be a branch incident to a leaf. Then,*

$$\mathbf{F}_{(\underline{x})}\left((n);(r)\right)=\mathbb{1}\{n=n_{\underline{x}}\}\times\mathbb{1}\{r=r_{\underline{x}}\}$$

Proof. Recall that the number of lineages sampled from species $\underline{x}$ is known and equal to $n_{\underline{x}}$. Then, applying definition (1) above with $\mathbf{x}=(\underline{x})$, we have:

$$\begin{aligned}
\mathbf{F}_{(\underline{x})}\left((n);(r)\right)&=\mathbb{P}\left(R_{\underline{x}}=r_{\underline{x}}\mid N_{\underline{x}}=n,R_{\underline{x}}=r\right)\times\mathbb{P}\left(N_{\underline{x}}=n\right)\\
&=\mathbb{1}\{r_{\underline{x}}=r\}\times\mathbb{1}\{n_{\underline{x}}=n\}.
\end{aligned}$$

□

Rule 1. *Let* $\mathbf{x}, \underline{x}$ *be a vector of incomparable population interfaces. Then,*

$$\mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\overline{x}}; \mathbf{r}_{\mathbf{x}}, r_{\overline{x}}\right) = \sum_{n=n_{\overline{x}}}^{m_x} \sum_{r=0}^{n} \mathbf{F}_{\mathbf{x},\underline{x}}\left(\mathbf{n}_{\mathbf{x}}, n; \mathbf{r}_{\mathbf{x}}, r\right) \exp(\mathbb{Q}_x t_x)_{(n,r);(n_{\overline{x}},r_{\overline{x}})}$$

Proof. First, note that, because $R_{\mathbf{L}(\mathbf{x},\overline{x})}$ is independent of $N_{\overline{x}}, R_{\overline{x}}$, when given $N_{\underline{x}}, R_{\underline{x}}$, and because $\mathbf{L}(\mathbf{x},\overline{x}) = \mathbf{L}(\mathbf{x},\underline{x})$:

$$\begin{aligned}
&\mathbb{P}\left(R_{\mathbf{L}(\mathbf{x},\overline{x})} = r_{\mathbf{L}(\mathbf{x},\overline{x})} \mid N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}, N_{\underline{x}} = n, R_{\underline{x}} = r, N_{\overline{x}} = n_{\overline{x}}, R_{\overline{x}} = r_{\overline{x}}\right) \\
&= \mathbb{P}\left(R_{\mathbf{L}(\mathbf{x},\underline{x})} = r_{\mathbf{L}(\mathbf{x},\underline{x})} \mid N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}, N_{\underline{x}} = n, R_{\underline{x}} = r\right)
\end{aligned}$$

Writing down the definition of $\mathbf{F}_{\mathbf{x},\overline{x}}$, then summing over all possible values of $N_{\underline{x}}$ and $R_{\underline{x}}$, and then using the identity above, we obtain:

$$\begin{aligned}
&\mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\overline{x}}; \mathbf{r}_{\mathbf{x}}, r_{\overline{x}}\right) \\
&= \sum_{n=n_{\overline{x}}}^{m_x} \sum_{r=0}^{n} \mathbb{P}\left(R_{\mathbf{L}(\mathbf{x},\underline{x})} = r_{\mathbf{L}(\mathbf{x},\underline{x})} \mid N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}, N_{\underline{x}} = n, R_{\underline{x}} = r\right) \\
&\times \mathbb{P}\left(N_{\underline{x}} = n, R_{\underline{x}} = r \mid N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}, N_{\overline{x}} = n_{\overline{x}}, R_{\overline{x}} = r_{\overline{x}}\right) \\
&\times \mathbb{P}\left(N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, N_{\overline{x}} = n_{\overline{x}}\right)
\end{aligned}$$

Moreover,

$$\begin{aligned}
&\mathbb{P}\left(N_{\underline{x}} = n, R_{\underline{x}} = r \mid N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}, N_{\overline{x}} = n_{\overline{x}}, R_{\overline{x}} = r_{\overline{x}}\right) \\
&= \mathbb{P}\left(R_{\underline{x}} = r \mid N_{\underline{x}} = n, N_{\overline{x}} = n_{\overline{x}}, R_{\overline{x}} = r_{\overline{x}}\right) \times \mathbb{P}\left(N_{\underline{x}} = n \mid N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, N_{\overline{x}} = n_{\overline{x}}\right),
\end{aligned}$$

where we have used that $R_{\underline{x}}$ is independent of $N_{\mathbf{x}}$ and $R_{\mathbf{x}}$, when given $N_{\underline{x}}, N_{\overline{x}}, R_{\overline{x}}$.

We then have:

$$\begin{aligned}
&\mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\overline{x}}; \mathbf{r}_{\mathbf{x}}, r_{\overline{x}}\right) \\
&= \sum_{n=n_{\overline{x}}}^{m_x} \sum_{r=0}^{n} \mathbb{P}\left(R_{\mathbf{L}(\mathbf{x},\underline{x})} = r_{\mathbf{L}(\mathbf{x},\underline{x})} \mid N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}, N_{\underline{x}} = n, R_{\underline{x}} = r\right) \\
&\times \mathbb{P}\left(R_{\underline{x}} = r \mid N_{\underline{x}} = n, N_{\overline{x}} = n_{\overline{x}}, R_{\overline{x}} = r_{\overline{x}}\right) \times \mathbb{P}(N_{\underline{x}} = n, N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, N_{\overline{x}} = n_{\overline{x}})
\end{aligned}$$

Using the fact that $N_{\overline{x}}$ is independent of $N_{\mathbf{x}}$, when given $N_{\underline{x}}$, the last term in the product can be rewritten as follows:

$$\mathbb{P}(N_{\underline{x}} = n, N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, N_{\overline{x}} = n_{\overline{x}}) = \mathbb{P}(N_{\overline{x}} = n_{\overline{x}} \mid N_{\underline{x}} = n) \times \mathbb{P}(N_{\underline{x}} = n, N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}})$$

Using Equation (2), we finally obtain:

$$\begin{aligned}
&\mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\overline{x}}; \mathbf{r}_{\mathbf{x}}, r_{\overline{x}}\right) \\
&= \sum_{n=n_{\overline{x}}}^{m_x} \sum_{r=0}^{n} \mathbb{P}\left(R_{\mathbf{L}(\mathbf{x},\underline{x})} = r_{\mathbf{L}(\mathbf{x},\underline{x})} \mid N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}, N_{\underline{x}} = n, R_{\underline{x}} = r\right) \\
&\times \mathbb{P}(N_{\underline{x}} = n, N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}) \times \exp(\mathbb{Q}_x t_x)_{(n,r);(n_{\overline{x}},r_{\overline{x}})} \\
&= \sum_{n=n_{\overline{x}}}^{m_x} \sum_{r=0}^{n} \mathbf{F}_{\mathbf{x},\underline{x}}(\mathbf{n}_{\mathbf{x}}, n; \mathbf{r}_{\mathbf{x}}, r) \times \exp(\mathbb{Q}_x t_x)_{(n,r);(n_{\overline{x}},r_{\overline{x}})}
\end{aligned}$$

□

In the following proofs, to make the mathematics more readable, we denote each event $A = a$ inside a probability simply as a, whenever the left-hand side of $A = a$ is unambiguously determined by the right-hand side. For example:

$$\begin{aligned}
\mathbf{n}_{\mathbf{x}} \quad &\text{means} \quad N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, \\
\mathbf{r}_{\mathbf{x}} \quad &\text{means} \quad R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}, \\
n_{\overline{x}} \quad &\text{means} \quad N_{\overline{x}} = n_{\overline{x}}, \\
r_{\overline{x}} \quad &\text{means} \quad R_{\overline{x}} = r_{\overline{x}}, \\
n_{\underline{x}} \quad &\text{means} \quad N_{\underline{x}} = n_{\underline{x}}, \\
r_{\underline{x}} \quad &\text{means} \quad R_{\underline{x}} = r_{\underline{x}}.
\end{aligned}$$

We will still write the full version in those cases where the left-hand side cannot be inferred from the right-hand side.

Rule 2. *Let $\mathbf{x}, \overline{x}$ and $\mathbf{y}, \overline{y}$ be two vectors of incomparable population interfaces, such that $\mathbf{L}(\mathbf{x}, \overline{x})$ and $\mathbf{L}(\mathbf{y}, \overline{y})$ have no leaf in common. Let x, y be the immediate descendants of branch z. Then,*

$$\begin{aligned}
&\mathbf{F}_{\mathbf{x},\mathbf{y},\underline{z}}\left(\mathbf{n}_{\mathbf{x}}, \mathbf{n}_{\mathbf{y}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, \mathbf{r}_{\mathbf{y}}, r_{\underline{z}}\right) \\
&= \sum_{n_{\overline{x}}} \sum_{r_{\overline{x}}} \mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\overline{x}}; \mathbf{r}_{\mathbf{x}}, r_{\overline{x}}\right) \mathbf{F}_{\mathbf{y},\overline{y}}\left(\mathbf{n}_{\mathbf{y}}, n_{\underline{z}} - n_{\overline{x}}; \mathbf{r}_{\mathbf{y}}, r_{\underline{z}} - r_{\overline{x}}\right) \binom{n_{\overline{x}}}{r_{\overline{x}}} \binom{n_{\underline{z}} - n_{\overline{x}}}{r_{\underline{z}} - r_{\overline{x}}} \binom{n_{\underline{z}}}{r_{\underline{z}}}^{-1}
\end{aligned}$$

The ranges of $n_{\overline{x}}$ and $r_{\overline{x}}$ in the summation terms are defined by
$\max(0, n_{\underline{z}} - m_y) \leq n_{\overline{x}} \leq \min(m_x, n_{\underline{z}})$ *and*
$\max(0, n_{\overline{x}} + r_{\underline{z}} - n_{\underline{z}}) \leq r_{\overline{x}} \leq \min(n_{\overline{x}}, r_{\underline{z}})$.

Proof. By definition,

$$\mathbf{F}_{\mathbf{x},\mathbf{y},\underline{z}}\left(\mathbf{n}_{\mathbf{x}}, \mathbf{n}_{\mathbf{y}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, \mathbf{r}_{\mathbf{y}}, r_{\underline{z}}\right) = \mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{x},\mathbf{y},\underline{z})} \mid \mathbf{n}_{\mathbf{x}}, \mathbf{n}_{\mathbf{y}}, n_{\underline{z}}, \mathbf{r}_{\mathbf{x}}, \mathbf{r}_{\mathbf{y}}, r_{\underline{z}}\right) \times \mathbb{P}\left(\mathbf{n}_{\mathbf{x}}, \mathbf{n}_{\mathbf{y}}, n_{\underline{z}}\right)$$

We then sum over all possible realizations of $N_{\overline{x}}$ and $R_{\overline{x}}$, and obtain:

$$\begin{aligned}
&\mathbf{F}_{\mathbf{x},\mathbf{y},\underline{z}}\left(\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}};\mathbf{r}_{\mathbf{x}},\mathbf{r}_{\mathbf{y}},r_{\underline{z}}\right)=\\
&\sum_{n_{\overline{x}}}\sum_{r_{\overline{x}}}\mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{x},\mathbf{y},\underline{z})}\mid\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}},\mathbf{r}_{\mathbf{x}},\mathbf{r}_{\mathbf{y}},r_{\underline{z}},n_{\overline{x}},r_{\overline{x}}\right)\\
&\times\mathbb{P}\left(n_{\overline{x}},r_{\overline{x}}\mid\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}},\mathbf{r}_{\mathbf{x}},\mathbf{r}_{\mathbf{y}},r_{\underline{z}}\right)\times\mathbb{P}\left(\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}}\right),
\end{aligned}$$

where the ranges in the summation terms are the same as those in the statement.

Now recall that $\mathbf{L}(\mathbf{x},\overline{x})$ and $\mathbf{L}(\mathbf{y},\overline{y})$ are disjoint vectors and note that their concatenation is equivalent to $\mathbf{L}(\mathbf{x},\mathbf{y},\underline{z})$. This means that $\mathbf{r}_{\mathbf{L}(\mathbf{x},\mathbf{y},\underline{z})}$ can also be written as $\mathbf{r}_{\mathbf{L}(\mathbf{x},\overline{x})},\mathbf{r}_{\mathbf{L}(\mathbf{y},\overline{y})}$. Moreover, $N_{\underline{z}}=n_{\underline{z}}$ and $N_{\overline{x}}=n_{\overline{x}}$ implies $N_{\overline{y}}=n_{\underline{z}}-n_{\overline{x}}$, and similarly $R_{\underline{z}}=r_{\underline{z}}$ and $R_{\overline{x}}=r_{\overline{x}}$ implies $R_{\overline{y}}=r_{\underline{z}}-r_{\overline{x}}$. We can then write:

$$\begin{aligned}
&\mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{x},\mathbf{y},\underline{z})}\mid\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}},\mathbf{r}_{\mathbf{x}},\mathbf{r}_{\mathbf{y}},r_{\underline{z}},n_{\overline{x}},r_{\overline{x}}\right)\\
&=\mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{x},\overline{x})},\mathbf{r}_{\mathbf{L}(\mathbf{y},\overline{y})}\mid\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}},\mathbf{r}_{\mathbf{x}},\mathbf{r}_{\mathbf{y}},r_{\underline{z}},n_{\overline{x}},r_{\overline{x}},N_{\overline{y}}=n_{\underline{z}}-n_{\overline{x}},R_{\overline{y}}=r_{\underline{z}}-r_{\overline{x}}\right)\\
&=\mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{x},\overline{x})}\mid\mathbf{n}_{\mathbf{x}},\mathbf{r}_{\mathbf{x}},n_{\overline{x}},r_{\overline{x}}\right)\times\mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{y},\overline{y})}\mid\mathbf{n}_{\mathbf{y}},\mathbf{r}_{\mathbf{y}},N_{\overline{y}}=n_{\underline{z}}-n_{\overline{x}},R_{\overline{y}}=r_{\underline{z}}-r_{\overline{x}}\right).
\end{aligned}$$

In the last equality above, we used the fact that $R_{\mathbf{L}(\mathbf{x},\overline{x})}$ and $R_{\mathbf{L}(\mathbf{y},\overline{y})}$ are independent random variables, given $N_{\mathbf{x},\overline{x}},R_{\mathbf{x},\overline{x}}$ and $N_{\mathbf{y},\overline{y}},R_{\mathbf{y},\overline{y}}$, respectively.

Moreover,

$$\begin{aligned}
&\mathbb{P}\left(n_{\overline{x}},r_{\overline{x}}\mid\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}},\mathbf{r}_{\mathbf{x}},\mathbf{r}_{\mathbf{y}},r_{\underline{z}}\right)\\
=\;&\mathbb{P}\left(r_{\overline{x}}\mid n_{\overline{x}},\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}},\mathbf{r}_{\mathbf{x}},\mathbf{r}_{\mathbf{y}},r_{\underline{z}}\right)\times\mathbb{P}\left(n_{\overline{x}}\mid\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}},\mathbf{r}_{\mathbf{x}},\mathbf{r}_{\mathbf{y}},r_{\underline{z}}\right)\\
=\;&\mathbb{P}\left(r_{\overline{x}}\mid n_{\overline{x}},n_{\underline{z}},r_{\underline{z}}\right)\times\mathbb{P}\left(n_{\overline{x}}\mid\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}}\right),
\end{aligned}$$

where in the last equality we have used the fact that $R_{\overline{x}}$ is independent of $N_{\mathbf{x}},N_{\mathbf{y}},R_{\mathbf{x}},R_{\mathbf{y}}$, when given $N_{\overline{x}},N_{\underline{z}}$, $R_{\underline{z}}$, and the fact that $N_{\overline{x}}$ is independent of $R_{\mathbf{x}},R_{\mathbf{y}},R_{\underline{z}}$, when given $N_{\mathbf{x}},N_{\mathbf{y}},N_{\underline{z}}$.

Putting all this together, we get:

$$\begin{aligned}
&\mathbf{F}_{\mathbf{x},\mathbf{y},\underline{z}}\left(\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}};\mathbf{r}_{\mathbf{x}},\mathbf{r}_{\mathbf{y}},r_{\underline{z}}\right)=\\
&\sum_{n_{\overline{x}}}\sum_{r_{\overline{x}}}\mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{x},\overline{x})}\mid\mathbf{n}_{\mathbf{x}},\mathbf{r}_{\mathbf{x}},n_{\overline{x}},r_{\overline{x}}\right)\times\mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{y},\overline{y})}\mid\mathbf{n}_{\mathbf{y}},\mathbf{r}_{\mathbf{y}},N_{\overline{y}}=n_{\underline{z}}-n_{\overline{x}},R_{\overline{y}}=r_{\underline{z}}-r_{\overline{x}}\right)\\
&\times\mathbb{P}\left(r_{\overline{x}}\mid n_{\overline{x}},n_{\underline{z}},r_{\underline{z}}\right)\times\mathbb{P}\left(n_{\overline{x}},\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}}\right).
\end{aligned}$$

Now note that

$$\begin{aligned}
\mathbb{P}\left(n_{\overline{x}},\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\underline{z}}\right)&=\mathbb{P}\left(\mathbf{n}_{\mathbf{x}},\mathbf{n}_{\mathbf{y}},n_{\overline{x}},N_{\overline{y}}=n_{\underline{z}}-n_{\overline{x}}\right)\\
&=\mathbb{P}\left(\mathbf{n}_{\mathbf{x}},n_{\overline{x}}\right)\times\mathbb{P}\left(\mathbf{n}_{\mathbf{y}},N_{\overline{y}}=n_{\underline{z}}-n_{\overline{x}}\right),
\end{aligned}$$

where the last equality is due to the independence between the lineages from $\mathbf{L}(\mathbf{x},\overline{x})$ and those from $\mathbf{L}(\mathbf{y},\overline{y})$.

Finally, $R_{\overline{x}}$, given $N_{\overline{x}} = n_{\overline{x}}, N_{\underline{z}} = n_{\underline{z}}, R_{\underline{z}} = r_{\underline{z}}$ follows a hypergeometric distribution:

$$\mathbb{P}\left(r_{\overline{x}} \mid n_{\overline{x}}, n_{\underline{z}}, r_{\underline{z}}\right) = \binom{n_{\overline{x}}}{r_{\overline{x}}}\binom{n_{\underline{z}} - n_{\overline{x}}}{r_{\underline{z}} - r_{\overline{x}}}\binom{n_{\underline{z}}}{r_{\underline{z}}}^{-1}, \tag{3}$$

which allows us to conclude:

$$\begin{aligned}
&\mathbf{F}_{\mathbf{x},\mathbf{y},\underline{z}}\left(\mathbf{n}_{\mathbf{x}}, \mathbf{n}_{\mathbf{y}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, \mathbf{r}_{\mathbf{y}}, r_{\underline{z}}\right) = \sum_{n_{\overline{x}}}\sum_{r_{\overline{x}}} \mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{x},\overline{x})} \mid \mathbf{n}_{\mathbf{x}}, \mathbf{r}_{\mathbf{x}}, n_{\overline{x}}, r_{\overline{x}}\right) \times \mathbb{P}\left(\mathbf{n}_{\mathbf{x}}, n_{\overline{x}}\right) \\
&\times \mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{y},\overline{y})} \mid \mathbf{n}_{\mathbf{y}}, \mathbf{r}_{\mathbf{y}}, N_{\overline{y}} = n_{\underline{z}} - n_{\overline{x}}, R_{\overline{y}} = r_{\underline{z}} - r_{\overline{x}}\right) \times \mathbb{P}\left(\mathbf{n}_{\mathbf{y}}, N_{\overline{y}} = n_{\underline{z}} - n_{\overline{x}}\right) \\
&\times \binom{n_{\overline{x}}}{r_{\overline{x}}}\binom{n_{\underline{z}} - n_{\overline{x}}}{r_{\underline{z}} - r_{\overline{x}}}\binom{n_{\underline{z}}}{r_{\underline{z}}}^{-1} \\
&= \sum_{n_{\overline{x}}}\sum_{r_{\overline{x}}} \mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\overline{x}}; \mathbf{r}_{\mathbf{x}}, r_{\overline{x}}\right) \mathbf{F}_{\mathbf{y},\overline{y}}\left(\mathbf{n}_{\mathbf{y}}, n_{\underline{z}} - n_{\overline{x}}; \mathbf{r}_{\mathbf{y}}, r_{\underline{z}} - r_{\overline{x}}\right) \binom{n_{\overline{x}}}{r_{\overline{x}}}\binom{n_{\underline{z}} - n_{\overline{x}}}{r_{\underline{z}} - r_{\overline{x}}}\binom{n_{\underline{z}}}{r_{\underline{z}}}^{-1}.
\end{aligned}$$

□

Rule 3. *Let $\mathbf{x}, \overline{x}$ be a vector of incomparable population interfaces, such that branch x's top node is a reticulation node. Let y, z be the branches immediately ancestral to x. Then,*

$$\mathbf{F}_{\mathbf{x},\underline{y},\underline{z}}\left(\mathbf{n}_{\mathbf{x}}, n_{\underline{y}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, r_{\underline{y}}, r_{\underline{z}}\right) = \mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\underline{y}} + n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, r_{\underline{y}} + r_{\underline{z}}\right)\binom{n_{\underline{y}} + n_{\underline{z}}}{n_{\underline{y}}}\gamma_y^{n_{\underline{y}}}\cdot\gamma_z^{n_{\underline{z}}}$$

Proof. First note that

$$\mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{x},\underline{y},\underline{z})} \mid \mathbf{n}_{\mathbf{x}}, n_{\underline{y}}, n_{\underline{z}}, \mathbf{r}_{\mathbf{x}}, r_{\underline{y}}, r_{\underline{z}}\right) = \mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{x},\overline{x})} \mid \mathbf{n}_{\mathbf{x}}, N_{\overline{x}} = n_{\underline{y}} + n_{\underline{z}}, \mathbf{r}_{\mathbf{x}}, R_{\overline{x}} = r_{\underline{y}} + r_{\underline{z}}\right).$$

Then, using the definitions of $\mathbf{F}_{\mathbf{x},\underline{y},\underline{z}}$ and $\mathbf{F}_{\mathbf{x},\overline{x}}$:

$$\frac{\mathbf{F}_{\mathbf{x},\underline{y},\underline{z}}\left(\mathbf{n}_{\mathbf{x}}, n_{\underline{y}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, r_{\underline{y}}, r_{\underline{z}}\right)}{\mathbf{F}_{\mathbf{x},\overline{x}}\left(\mathbf{n}_{\mathbf{x}}, n_{\underline{y}} + n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, r_{\underline{y}} + r_{\underline{z}}\right)} = \frac{\mathbb{P}\left(\mathbf{n}_{\mathbf{x}}, n_{\underline{y}}, n_{\underline{z}}\right)}{\mathbb{P}\left(\mathbf{n}_{\mathbf{x}}, N_{\overline{x}} = n_{\underline{y}} + n_{\underline{z}}\right)}$$

But

$$\frac{\mathbb{P}\left(\mathbf{n}_{\mathbf{x}}, n_{\underline{y}}, n_{\underline{z}}\right)}{\mathbb{P}\left(\mathbf{n}_{\mathbf{x}}, N_{\overline{x}} = n_{\underline{y}} + n_{\underline{z}}\right)} = \mathbb{P}\left(n_{\underline{y}}, n_{\underline{z}} \mid \mathbf{n}_{\mathbf{x}}, N_{\overline{x}} = n_{\underline{y}} + n_{\underline{z}}\right) = \binom{n_{\underline{y}} + n_{\underline{z}}}{n_{\underline{y}}}\gamma_y^{n_{\underline{y}}}\cdot\gamma_z^{n_{\underline{z}}},$$

where the first equality applies the definition of conditional probability, and the second equality uses the fact that $N_{\underline{y}}$ and $N_{\underline{z}}$ are binomially distributed, when given $N_{\overline{x}}$. The Rule trivially follows. □

Rule 4. *Let $\mathbf{z}, \overline{x}, \overline{y}$ be a vector of incomparable population interfaces, and let x, y be immediate descendants of branch z. Then,*

$$\begin{aligned}&\mathbf{F}_{\mathbf{z},\underline{z}}\left(\mathbf{n}_{\mathbf{z}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}\right)\\&= \sum_{n_{\overline{x}}} \sum_{r_{\overline{x}}} \mathbf{F}_{\mathbf{z},\overline{x},\overline{y}}\left(\mathbf{n}_{\mathbf{z}}, n_{\overline{x}}, n_{\underline{z}} - n_{\overline{x}}; \mathbf{r}_{\mathbf{z}}, r_{\overline{x}}, r_{\underline{z}} - r_{\overline{x}}\right) \binom{n_{\overline{x}}}{r_{\overline{x}}} \binom{n_{\underline{z}} - n_{\overline{x}}}{r_{\underline{z}} - r_{\overline{x}}} \binom{n_{\underline{z}}}{r_{\underline{z}}}^{-1}\end{aligned}$$

The ranges of $n_{\overline{x}}$ and $r_{\overline{x}}$ in the sums are the same as those in Rule 2.

Proof. Use the definition of $\mathbf{F}_{\mathbf{z},\underline{z}}$ and then sum over all possible realizations of $N_{\overline{x}}$ and $R_{\overline{x}}$:

$$\begin{aligned}&\mathbf{F}_{\mathbf{z},\underline{z}}\left(\mathbf{n}_{\mathbf{z}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}\right) = \mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{z},\underline{z})} \mid \mathbf{n}_{\mathbf{z}}, n_{\underline{z}}, \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}\right) \times \mathbb{P}\left(\mathbf{n}_{\mathbf{z}}, n_{\underline{z}}\right) =\\&\sum_{n_{\overline{x}}} \sum_{r_{\overline{x}}} \mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{z},\underline{z})} \mid \mathbf{n}_{\mathbf{z}}, n_{\underline{z}}, \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}, n_{\overline{x}}, r_{\overline{x}}\right) \times \mathbb{P}(n_{\overline{x}}, r_{\overline{x}} \mid \mathbf{n}_{\mathbf{z}}, n_{\underline{z}}, \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}) \times \mathbb{P}(\mathbf{n}_{\mathbf{z}}, n_{\underline{z}})\end{aligned}$$

Now note that $\mathbf{L}(\mathbf{z}, \underline{z}) = \mathbf{L}(\mathbf{z}, \overline{x}, \overline{y})$, and that

$$\begin{aligned}&N_{\underline{z}} = n_{\underline{z}}, R_{\underline{z}} = r_{\underline{z}}, N_{\overline{x}} = n_{\overline{x}}, R_{\overline{x}} = r_{\overline{x}} \quad \text{if and only if}\\&N_{\overline{x}} = n_{\overline{x}}, R_{\overline{x}} = r_{\overline{x}}, N_{\overline{y}} = n_{\underline{z}} - n_{\overline{x}}, R_{\overline{y}} = n_{\underline{z}} - n_{\overline{x}},\end{aligned}$$

meaning that

$$\begin{aligned}&\mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{z},\underline{z})} \mid \mathbf{n}_{\mathbf{z}}, n_{\underline{z}}, \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}, n_{\overline{x}}, r_{\overline{x}}\right)\\&= \mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{z},\overline{x},\overline{y})} \mid \mathbf{n}_{\mathbf{z}}, \mathbf{r}_{\mathbf{z}}, n_{\overline{x}}, r_{\overline{x}}, N_{\overline{y}} = n_{\underline{z}} - n_{\overline{x}}, R_{\overline{y}} = n_{\underline{z}} - n_{\overline{x}}\right).\end{aligned}$$

Moreover,

$$\begin{aligned}&\mathbb{P}(n_{\overline{x}}, r_{\overline{x}} \mid \mathbf{n}_{\mathbf{z}}, n_{\underline{z}}, \mathbf{r}_{\mathbf{z}}, r_{\underline{z}})\\&= \mathbb{P}(r_{\overline{x}} \mid n_{\overline{x}}, \mathbf{n}_{\mathbf{z}}, n_{\underline{z}}, \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}) \times \mathbb{P}(n_{\overline{x}} \mid \mathbf{n}_{\mathbf{z}}, n_{\underline{z}}, \mathbf{r}_{\mathbf{z}}, r_{\underline{z}})\\&= \mathbb{P}(r_{\overline{x}} \mid n_{\overline{x}}, n_{\underline{z}}, r_{\underline{z}}) \qquad \times \mathbb{P}(n_{\overline{x}} \mid \mathbf{n}_{\mathbf{z}}, n_{\underline{z}}),\end{aligned}$$

where in the last equality we have used that $r_{\overline{x}}$ is independent of $\mathbf{n}_{\mathbf{z}}, \mathbf{r}_{\mathbf{z}}$, when given $n_{\overline{x}}, n_{\underline{z}}, r_{\underline{z}}$, and the fact that $n_{\overline{x}}$ is independent of $\mathbf{r}_{\mathbf{z}}, r_{\underline{z}}$, when given $n_{\underline{z}}$.

Now use again Equation (3) to express $\mathbb{P}(r_{\overline{x}} \mid n_{\overline{x}}, n_{\underline{z}}, r_{\underline{z}})$ and conclude:

$$\begin{aligned}&\mathbf{F}_{\mathbf{z},\underline{z}}\left(\mathbf{n}_{\mathbf{z}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}\right)\\&= \sum_{n_{\overline{x}}} \sum_{r_{\overline{x}}} \mathbb{P}\left(\mathbf{r}_{\mathbf{L}(\mathbf{z},\overline{x},\overline{y})} \mid \mathbf{n}_{\mathbf{z}}, \mathbf{r}_{\mathbf{z}}, n_{\overline{x}}, r_{\overline{x}}, N_{\overline{y}} = n_{\underline{z}} - n_{\overline{x}}, R_{\overline{y}} = n_{\underline{z}} - n_{\overline{x}}\right)\\&\times \mathbb{P}(n_{\overline{x}} \mid \mathbf{n}_{\mathbf{z}}, n_{\underline{z}}) \times \mathbb{P}(\mathbf{n}_{\mathbf{z}}, n_{\underline{z}}) \times \binom{n_{\overline{x}}}{r_{\overline{x}}} \binom{n_{\underline{z}} - n_{\overline{x}}}{r_{\underline{z}} - r_{\overline{x}}} \binom{n_{\underline{z}}}{r_{\underline{z}}}^{-1}\\&= \sum_{n_{\overline{x}}} \sum_{r_{\overline{x}}} \mathbf{F}_{\mathbf{z},\overline{x},\overline{y}}\left(\mathbf{n}_{\mathbf{z}}, n_{\overline{x}}, n_{\underline{z}} - n_{\overline{x}}; \mathbf{r}_{\mathbf{z}}, r_{\overline{x}}, r_{\underline{z}} - r_{\overline{x}}\right) \binom{n_{\overline{x}}}{r_{\overline{x}}} \binom{n_{\underline{z}} - n_{\overline{x}}}{r_{\underline{z}} - r_{\overline{x}}} \binom{n_{\underline{z}}}{r_{\underline{z}}}^{-1}\end{aligned}$$

□

1.2 About ranges

We start this section with a general discussion about the values that the random variables $N_{\underline{x}}, N_{\overline{x}}, R_{\underline{x}}, R_{\overline{x}}$ can take for any population interface in the network. As usual, we will use lower-case letters for their realizations, i.e. $n_{\underline{x}}, n_{\overline{x}}, r_{\underline{x}}, r_{\overline{x}}$. Our remarks will allow us to derive the ranges used in our rules as simple consequences of a few equations.

1.2.1 Observable number of lineages across the network

The number of lineages $n_{\underline{x}}, n_{\overline{x}}, r_{\underline{x}}, r_{\overline{x}}$ observed at any population interface in the network must satisfy a few simple and obvious constraints, which we list below:

- For any branch x, the number of lineages at the top of the branch is at least 1, unless there were no lineages at the bottom of the branch, and at most equal to the number of lineages at the bottom. That is,

$$\mathbb{1}\{n_{\underline{x}} > 0\} \leq n_{\overline{x}} \leq n_{\underline{x}} \tag{4}$$

- At any population interface, the number of red and green lineages cannot exceed the total number of lineages. That is, for any branch x:

$$0 \leq r_{\underline{x}} \leq n_{\underline{x}} \tag{5}$$
$$0 \leq r_{\overline{x}} \leq n_{\overline{x}} \tag{6}$$

- For any internal node u, the numbers of red and green lineages entering u are the same as the numbers of red and green lineages exiting u. That is, if u is a tree node with ingoing branch z and outgoing branches x, y:

$$n_{\underline{z}} = n_{\overline{x}} + n_{\overline{y}} \tag{7}$$
$$r_{\underline{z}} = r_{\overline{x}} + r_{\overline{y}} \tag{8}$$

  (Note that these two equations also imply that the numbers of green lineages entering and exiting u are the same.)

  If u is a reticulation with ingoing branches x, y and outgoing branch z:

$$n_{\overline{z}} = n_{\underline{x}} + n_{\underline{y}} \tag{9}$$
$$r_{\overline{z}} = r_{\underline{x}} + r_{\underline{y}} \tag{10}$$

- A simple consequence of Equations (4), (7) and (9) is that the number of lineages in any branch x cannot exceed the total number of lineages at the leaves that descend from x, that is:

$$n_{\underline{x}}, n_{\overline{x}} \leq m_x \tag{11}$$

  (This can easily be proven by induction on the height of x.)

Constraints (4)-(10) above are not only necessary, but also sufficient to describe all possible values of $n_{\underline{x}}, n_{\overline{x}}, r_{\underline{x}}, r_{\overline{x}}$ across the network. In theory they could be used to infer the precise ranges for these variables, starting from the leaves and moving up the network.

In practice, however, this is unnecessary. SnappNet only ensures that for any population interface $\underline{x}$ or $\overline{x}$, the following two equations are satisfied:

$$0 \leq r_{\underline{x}} \leq n_{\underline{x}} \leq m_x \tag{12}$$
$$0 \leq r_{\overline{x}} \leq n_{\overline{x}} \leq m_x \tag{13}$$

These equations also specify the ranges for which $\mathbf{F_x}(\mathbf{n_x}; \mathbf{r_x})$ is defined and stored in memory.

Note that equations (12) and (13) permit a few more values for the n arguments than are actually possible. For example $n_{\underline{x}}$ is allowed to be 0, even when this is not possible (e.g. when x lies on all paths from a leaf with sampled individuals to the root). Whenever this occurs, the probability term within $\mathbf{F_x}(\mathbf{n_x}; \mathbf{r_x})$ equals 0. As a result, the partial likelihood itself is 0 and does not contribute to the calculation of any partial likelihood higher up in the network.

1.2.2 Ranges of the sums in Rules 2 and 4

It is now easy to justify the ranges in the sums in Rules 2 and 4. Recall that both these rules describe the behavior of the algorithm when traversing a tree node with ingoing branch z and outgoing branches x, y. Also recall that these rules sum over the possible values for $n_{\overline{x}}$ and $r_{\overline{x}}$. Note that, because conservation constraints (7) and (8) must hold here, these values also determine the values of $n_{\overline{y}} = n_{\underline{z}} - n_{\overline{x}}$ and $r_{\overline{y}} = r_{\underline{z}} - r_{\overline{x}}$.

Let's first consider the range for $n_{\overline{x}}$. By applying constraint (13) to $n_{\overline{x}}$ and then $n_{\overline{y}}$, we must ensure:

$$0 \leq n_{\overline{x}} \leq m_x$$
$$0 \leq n_{\underline{z}} - n_{\overline{x}} \leq m_y$$

The second equation is equivalent to $n_{\underline{z}} - m_y \leq n_{\overline{x}} \leq n_{\underline{z}}$ and therefore we get:

$$\max(0, n_{\underline{z}} - m_y) \leq n_{\overline{x}} \leq \min(m_x, n_{\underline{z}})$$

As for $r_{\overline{x}}$, by applying constraint (13) to $r_{\overline{x}}$ and then $r_{\overline{y}}$, we must ensure:

$$0 \leq r_{\overline{x}} \leq n_{\overline{x}}$$
$$0 \leq r_{\underline{z}} - r_{\overline{x}} \leq n_{\underline{z}} - n_{\overline{x}}$$

The second equation is equivalent to $n_{\overline{x}} + r_{\underline{z}} - n_{\underline{z}} \leq r_{\overline{x}} \leq r_{\underline{z}}$ and therefore we get:

$$\max(0, n_{\overline{x}} + r_{\underline{z}} - n_{\underline{z}}) \leq r_{\overline{x}} \leq \min(n_{\overline{x}}, r_{\underline{z}}).$$

2 Likelihood computation in detail

SnappNet uses Algorithm 1 to compute the full likelihood of a network Ψ with respect to D_i, the data from marker i. The algorithm starts by initializing the data structures that will subsequently be used and then processes all nodes of the network Ψ using the rules presented in the main text. Rules 2, 3 and 4 are applied respectively in Algorithm 3, 4 and 5, together with suitable modifications of data structures.

The data structures are the following: ReadyNodesQ, a queue storing the nodes that are ready to be processed; Processed, which stores whether a node has already been processed or not; and CurrF, a dictionary that associates any branch x to the $\mathbf{F_x}$ having $\overline{x}$ in $\mathbf{x}$. In this pseudocode, $\mathbf{F_x}$ represents a data structure holding all the relevant values of $\mathbf{F_x}(\mathbf{n_x}, \mathbf{r_x})$, as well as the vector of population interfaces $\mathbf{x}$. We also note that, to reduce memory usage, we only store the $\mathbf{F_x}$ associated to branches that separate an unprocessed node to a processed node, as these are the only ones that will be used in future computations. Note that unlike in the main text, nodes are denoted u, u' and u_p in S1 Text.

Algorithm 1: Compute the likelihood for one marker

Input: Network Ψ, and the data D_i for one marker
Output: The likelihood $\mathbb{P}(D_i|\Psi)$

```
// Defining global data structures shared by all algorithms
Let ReadyNodesQ be an empty queue
Let CurrF and Processed be empty dictionaries
Initialize_Data_Structures(D_i)
while ReadyNodesQ ≠ ∅ do
    u ← Dequeue(ReadyNodesQ)
    if u has two outgoing branches e_1 and e_2 then // u is a tree node
        if CurrF[e_1] ≠ CurrF[e_2] then          // comparing pointers
            Apply_Rule_2(u)
        else Apply_Rule_4(u)
    else Apply_Rule_3(u)                  // u is a reticulation node
end
Let ρ be the root branch in Ψ
Compute ℙ(D_i|Ψ) from F_(ρ) using Equation (3)
return ℙ(D_i|Ψ)
```

Algorithm 2: `Initialize_Data_Structures(`D_i`)`

foreach *leaf* $\underline{x}$ *in* Ψ **do**
  Compute $n_{\underline{x}}$ and $r_{\underline{x}}$ from D_i
  Compute $\mathbf{F}_{(\underline{x})}$ using Rule 0
  Compute $\mathbf{F}_{(\overline{x})}$ using Rule 1
  CurrF$[x] \leftarrow \mathbf{F}_{(\overline{x})}$
  Processed$[\underline{x}] \leftarrow$ true
end
foreach *internal node* u *in* Ψ **do**
  Processed$[u] \leftarrow$ false
  if *all children of* u *are leaves* **then** `Enqueue` (ReadyNodesQ,u)
end

Algorithm 3: `Apply_Rule_2(`u`)` `// u is a tree node of Ψ`

Let x, y be u's outgoing branches and let z be u's incoming branch
$\mathbf{F}_{\mathbf{x},\overline{x}} \leftarrow$ CurrF$[x]$
$\mathbf{F}_{\mathbf{y},\overline{y}} \leftarrow$ CurrF$[y]$
Apply Rule 2 to obtain $\mathbf{F}_{\mathbf{x},\mathbf{y},\underline{z}}$ from $\mathbf{F}_{\mathbf{x},\overline{x}}$ and $\mathbf{F}_{\mathbf{y},\overline{y}}$
if u *is the root node of* Ψ **then return**
Apply Rule 1 to obtain $\mathbf{F}_{\mathbf{x},\mathbf{y},\overline{z}}$ from $\mathbf{F}_{\mathbf{x},\mathbf{y},\underline{z}}$
foreach *branch* w *with an interface in* $\mathbf{x}, \mathbf{y}, \overline{z}$ **do**
  CurrF$[w] \leftarrow \mathbf{F}_{\mathbf{x},\mathbf{y},\overline{z}}$ `// copying pointers only`
Processed$[u] \leftarrow$ true
`CheckParentIsReady(`z`)`

Algorithm 4: `Apply_Rule_3(`u`)` `// u is a reticulation node of Ψ`

Let x be u's outgoing branch and let y, z be u's incoming branches
$\mathbf{F}_{\mathbf{x},\overline{x}} \leftarrow \textsc{CurrF}[x]$
Apply Rule 3 to obtain $\mathbf{F}_{\mathbf{x},\underline{y},\underline{z}}$ from $\mathbf{F}_{\mathbf{x},\overline{x}}$
Apply Rule 1 twice to obtain $\mathbf{F}_{\mathbf{x},\overline{y},\overline{z}}$ from $\mathbf{F}_{\mathbf{x},\underline{y},\underline{z}}$

foreach *branch* w *with an interface in* $\mathbf{x}, \overline{y}, \overline{z}$ **do**
  $\textsc{CurrF}[w] \leftarrow \mathbf{F}_{\mathbf{x},\overline{y},\overline{z}}$ `// copying pointers only`

$\textsc{Processed}[u] \leftarrow$ `true`
`CheckParentIsReady(`y`)`
`CheckParentIsReady(`z`)`

Algorithm 5: `Apply_Rule_4(`u`)` `// u is a tree node of Ψ`

Let x, y be u's outgoing branches and let z be u's incoming branch
`// recall that here` $\textsc{CurrF}[x] = \textsc{CurrF}[y]$
$\mathbf{F}_{\mathbf{z},\overline{x},\overline{y}} \leftarrow \textsc{CurrF}[x]$
Apply Rule 4 to obtain $\mathbf{F}_{\mathbf{z},\underline{z}}$ from $\mathbf{F}_{\mathbf{z},\overline{x},\overline{y}}$
if u *is the root node of* Ψ **then return**
Apply Rule 1 to obtain $\mathbf{F}_{\mathbf{z},\overline{z}}$ from $\mathbf{F}_{\mathbf{z},\underline{z}}$

foreach *branch* w *with an interface in* $\mathbf{z}, \overline{z}$ **do**
  $\textsc{CurrF}[w] \leftarrow \mathbf{F}_{\mathbf{z},\overline{z}}$

$\textsc{Processed}[u] \leftarrow$ `true`
`CheckParentIsReady(`z`)`

Algorithm 6: `CheckParentIsReady(`x_p`)` `// x_p is a branch of Ψ`

Result: Updated data structures, where the origin of x_p is added to ReadyNodesQ if all its descendants have already been processed

Let u_p and u be the nodes respectively at the origin and end of x_p
if u_p *has two parents* **then** `// u_p is a reticulation node`
  `Enqueue(`ReadyNodesQ`,`u_p`)`
else `// u_p is a tree node`
  Let u' be the child of u_p different from u
  if $\textsc{Processed}[u']$ **then**
    `Enqueue` (ReadyNodesQ, u_p)
  end
end

3 Other computational complexity results

In this section, we shall use the weak definition of connectivity in a directed graph: we say that two nodes in Ψ are *connected* is there is an undirected path between them in Ψ. The same holds for the notion of *biconnected*, see below.

3.1 Time complexity of the algorithm by Zhu et al. [1]

Although the time complexity stated by Zhu and coauthors is $O(sn^{4r+4})$, where r is the number of reticulation nodes in the network, they also note that *all labelled partial likelihoods (LPLs) at a lowest articulation node can be merged into a single LPL, thus avoiding carrying forth all that information* [1]. This means that, as we stated in the main text, the time complexity to process a node with their algorithm is actually $O(n^{4r_u+4})$, where r_u is the number of reticulation nodes which descend from u, and for which there exists a directed path from u that does not pass via a lowest articulation node. Note that r_u is potentially much smaller than r. We refer to the original paper by Zhu and coauthors for the definition of LPL and the full description of their algorithm [1].

Here we prove that, since the time complexity to process a node is $O(n^{4r_u+4})$, then the whole algorithm runs in $O(sn^{4\ell+4})$ time, where ℓ is the *level* of the network [5, 6].

Let us first recall some definitions from the theory of phylogenetic networks that are fundamental to analyse the complexity of the algorithm by Zhu et al. [1]. A subgraph G of Ψ is *biconnected* if the removal of any one node in G leaves the remainder of G connected. A *biconnected component* of Ψ is a maximal biconnected subgraph of Ψ. The nodes of Ψ that belong to two or more biconnected components are called *articulation nodes*. (Equivalently, articulation nodes are the nodes in Ψ whose removal cause the network to become disconnected.) An articulation node is said to be a *lowest articulation node* if all of its children are not articulation nodes. The *level* of a phylogenetic network is the maximum number of reticulation nodes in one of its biconnected components.

It is easy to see that a phylogenetic network has two kinds of biconnected components: those that only consist of two adjacent nodes — which we call *trivial* biconnected components — and more complex ones — which we call nontrivial biconnected components or *blobs*. Every articulation node of Ψ is found at the root of a biconnected component. The lowest articulation nodes of a network coincide with the roots of the network's blobs.

Recall that r_u is defined as the number of reticulation nodes which descend from u, and for which there exists a directed path from u that does not pass via a lowest articulation node. Now note that every directed path that ends in a reticulation node v and does not pass via a lowest articulation node can only be from a node u in the same blob as v. Then, r_u is at most equal to the

number of reticulation nodes in the same biconnected component as u. In turn, the number of reticulation nodes in the same biconnected component as u is at most equal to ℓ, the level of Ψ. We can then conclude that $r_u \leq \ell$ and that each node is processed in at most $O(n^{4\ell+4})$ time, giving a total running time of $O(sn^{4\ell+4})$.

3.2 SnappNet's $\overline{K}$ and the level of the network

Here we prove that for any traversal of the network Ψ, we have $\overline{K} \leq \ell+1$, where ℓ is the level of Ψ (Proposition 1 below).

We let $B(\mathbf{x})$ denote the set of branches x for which there exists a population interface $\underline{x}$ or $\overline{x}$ in the VPI $\mathbf{x}$. Moreover we let $G^{\Psi}_{\mathbf{x}}$ denote the subgraph of Ψ induced by all the descendant nodes of the branches in $B(\mathbf{x})$.

The intuition behind the proof is that, for any VPI activated by the traversal algorithm, the branches in $B(\mathbf{x})$ must all belong to the same biconnected component of Ψ. Moreover, $|B(\mathbf{x})|$ cannot exceed 1 + the number of reticulations within that biconnected component, which implies $\overline{K} \leq \ell + 1$.

Lemma 1. *Let* $\mathbf{x}$ *be a VPI activated by any traversal algorithm using Rules 0-4. Then,* $G^{\Psi}_{\mathbf{x}}$ *is connected.*

Proof. If $\mathbf{x} = (\underline{x})$ is activated by Rule 0, then $G^{\Psi}_{\mathbf{x}}$ consists of a single leaf and is trivially connected. Thus, we just need to prove that every subsequent application of Rules 1-4 can only activate a VPI $\mathbf{x}$ with connected $G^{\Psi}_{\mathbf{x}}$, assuming that this property is satisfied by the VPI or VPIs that the rule uses as input.

For Rule 1, this is trivially true as $G^{\Psi}_{\mathbf{x},\overline{x}} = G^{\Psi}_{\mathbf{x},\underline{x}}$. For Rule 2, let's assume that $G^{\Psi}_{\mathbf{x},\overline{x}}$ is connected and that $G^{\Psi}_{\mathbf{y},\overline{y}}$ is connected. This implies that $G^{\Psi}_{\mathbf{x},\mathbf{y},\underline{z}}$ is connected, as x and y appear in $G^{\Psi}_{\mathbf{x},\mathbf{y},\underline{z}}$ and ensure that all nodes in $G^{\Psi}_{\mathbf{x},\overline{x}}$ are connected to all nodes in $G^{\Psi}_{\mathbf{y},\overline{y}}$. For Rule 3 and 4, the thesis is again trivial, because $G^{\Psi}_{\mathbf{x}}$ for the newly active VPI only differs from the one for the input VPI by inclusion of a single new vertex, which is easily seen to be connected to the rest of $G^{\Psi}_{\mathbf{x}}$. □

Corollary 1. *Let* $\mathbf{x}$ *be a VPI activated by any traversal algorithm using Rules 0-4. Then, all the branches in* $B(\mathbf{x})$ *belong to the same biconnected component of* Ψ.

Proof. If $|B(\mathbf{x})| = 1$, this is trivial. If $B(\mathbf{x})$ contains at least two branches x and y, it is now easy to see that x and y belong to a cycle obtained by attaching the following two disjoint paths: (1) the path within $G^{\Psi}_{\mathbf{x}}$ from the bottom of x to the bottom of y — which exists because of Lemma 1 — and (2) the path from the bottom of x to the bottom of y, going via x and y and only using branches that are ancestral to x and y. The existence of this cycle implies the thesis. □

Lemma 2. *Let* $\mathbf{x}$ *be a VPI activated by any traversal algorithm using Rules 0-4, and let* $R(\mathbf{x})$ *be the set of reticulation nodes that descend from any branch in* $B(\mathbf{x})$ *and belong to the same biconnected component as the one of* $B(\mathbf{x})$*. Then,* $|B(\mathbf{x})| \leq |R(\mathbf{x})| + 1$.

Proof. To make notation light, let $b(\mathbf{x}) = |B(\mathbf{x})|$ and $r(\mathbf{x}) = |R(\mathbf{x})|$. As in the proof of Lemma 1, we start by noting that if $\mathbf{x} = (\underline{x})$ is activated by Rule 0, then the thesis trivially holds, as $b((\underline{x})) = 1$ and $r((\underline{x})) = 0$.

We then consider the other rules, and show that if the thesis holds for the VPIs that have already been activated, then it must hold for the newly activated VPI. For Rule 1, $b(\mathbf{x}, \overline{x}) = b(\mathbf{x}, \underline{x})$ and $r(\mathbf{x}, \overline{x}) = r(\mathbf{x}, \underline{x})$, so $b(\mathbf{x}, \underline{x}) \leq r(\mathbf{x}, \underline{x}) + 1$ trivially implies $b(\mathbf{x}, \overline{x}) \leq r(\mathbf{x}, \overline{x}) + 1$.

For Rule 2, we assume $b(\mathbf{x}, \overline{x}) \leq r(\mathbf{x}, \overline{x}) + 1$ and $b(\mathbf{y}, \overline{y}) \leq r(\mathbf{y}, \overline{y}) + 1$. Now note that $b(\mathbf{x}, \mathbf{y}, \underline{z}) = b(\mathbf{x}, \overline{x}) + b(\mathbf{y}, \overline{y}) - 1$, and $r(\mathbf{x}, \mathbf{y}, \underline{z}) = r(\mathbf{x}, \overline{x}) + r(\mathbf{y}, \overline{y})$ which imply:

$$\begin{aligned} b(\mathbf{x}, \mathbf{y}, \underline{z}) &= b(\mathbf{x}, \overline{x}) + b(\mathbf{y}, \overline{y}) - 1 \\ &\leq (r(\mathbf{x}, \overline{x}) + 1) + (r(\mathbf{y}, \overline{y}) + 1) - 1 \\ &= r(\mathbf{x}, \overline{x}) + r(\mathbf{y}, \overline{y}) + 1 \\ &= r(\mathbf{x}, \mathbf{y}, \underline{z}) + 1, \end{aligned}$$

thus proving the thesis for VPI $\mathbf{x}, \mathbf{y}, \underline{z}$.

For Rule 3, we assume $b(\mathbf{x}, \overline{x}) \leq r(\mathbf{x}, \overline{x}) + 1$. Now note that

$$\begin{aligned} b(\mathbf{x}, \underline{y}, \underline{z}) &= b(\mathbf{x}, \overline{x}) + 1, \\ r(\mathbf{x}, \underline{y}, \underline{z}) &= r(\mathbf{x}, \overline{x}) + 1, \end{aligned}$$

which implies $b(\mathbf{x}, \underline{y}, \underline{z}) \leq r(\mathbf{x}, \underline{y}, \underline{z}) + 1$.

Finally, for Rule 4, we assume $b(\mathbf{z}, \overline{x}, \overline{y}) \leq r(\mathbf{z}, \overline{x}, \overline{y}) + 1$. Now distinguish between two cases. Either (i) $\mathbf{z}$ is nonempty, in which case $B(\mathbf{z}, \overline{x}, \overline{y})$ and $B(\mathbf{z}, \underline{z})$ are in the same biconnected component and

$$\begin{aligned} b(\mathbf{z}, \underline{z}) &= b(\mathbf{z}, \overline{x}, \overline{y}) - 1, \\ r(\mathbf{z}, \underline{z}) &= r(\mathbf{z}, \overline{x}, \overline{y}). \end{aligned}$$

In this case we therefore have $b(\mathbf{z}, \underline{z}) \leq r(\mathbf{z}, \underline{z})$, which implies the thesis.

Alternatively, (ii) $\mathbf{z}$ is empty, in which case

$$\begin{aligned} b(\mathbf{z}, \underline{z}) &= 1, \\ r(\mathbf{z}, \underline{z}) &= 0. \end{aligned}$$

Thus $b(\mathbf{z}, \underline{z}) \leq r(\mathbf{z}, \underline{z}) + 1$ is again satisfied. □

We now have all we need to prove the main result of this section:

Proposition 1. *For any traversal algorithm using Rules 0-4 to process a network of level ℓ, $\overline{K} \leq \ell + 1$.*

Proof. Note that

$$\overline{K} = \max\{|B(\mathbf{x})| \text{ such that } \mathbf{x} \text{ is activated by the given traversal algorithm}\}.$$

Thus, using Lemma 2, and the definition of the level ℓ:

$$\begin{aligned}\overline{K} &\leq \max\{|R(\mathbf{x})| + 1 \text{ such that } \mathbf{x} \text{ is activated by the given traversal algorithm}\} \\ &\leq \ell + 1.\end{aligned}$$

□

4 Newick representations

Network A:
((C:0.08,((R:0.007,(Q:0.004)#H1:0.003):0.035,((A:0.006,#H1:0.002):0.016,
L:0.022):0.02):0.038):0);

Network B:
((((R:0.014,(Q:0.004)#H1:0.01):0.028,(((A:0.003)#H2:0.003,#H1:0.002)
:0.016,L:0.022):0.02):0.038,(C:0.005,#H2:0.002):0.075):0);

Network C:
((O:0.08,((A:0.012,((B:0.002,(C:0.001)#H1:0.001):0.002)#H2:0.008):0.038,
((D:0.003,#H1:0.002):0.017,#H2:0.016):0.03):0.03):0);

Starting tree for networks A and B:
(((C:0.05,R:0.05):0.05,((A:0.05,L:0.05):0.025,Q:0.075):0.025):0);

Alternative starting trees for networks A and B (only used to check the influence of the starting tree):
(((A:0.05,Q:0.05):0.05,((C:0.05,L:0.05):0.025,R:0.075):0.025):0);
((((C:0.05,A:0.05):0.05,((R:0.05,Q:0.05):0.025,L:0.075):0.025):0);

Starting tree for network C:
(((O:0.05,A:0.05):0.05,((C:0.05,D:0.05):0.025,B:0.075):0.025):0);

5 MCMCBiMarkers commands

For m=100,000 , data were generated in the following way:

```
SimBiMarkersinNetwork -pi0 0.5 -sd 17000 -num 100000
-tm <A:A_0;B:B_0,B_1,B_2,B_3;
C:C_0,C_1,C_2,C_3;D:D_0;O:O_0>
-truenet "[0.005](O:0.08:0.005,((A:0.012:0.005,((B:0.002:0.005,
(C:0.001:0.005)I1#H1:0.001:0.005:0.5)I2:0.002:0.005)I3#H2:0.008
:0.005:0.5)I4:0.038:0.005,((D:0.003:0.005,
I1#H1:0.002:0.005:0.5)I5:0.017:0.005,I3#H2:0.016:0.005:0.5)
I6:0.03:0.005)I7:0.03:0.005);"
;
```

Next, the following commands, were successively used to run `MCMCBiMarkers`. The first step consists in a pre-burnin phase relying on 3 chains of different temperatures.

```
MCMC_BiMarkers -cl 1500000 -sf 1000 -bl 200000 -prebl 10000
-premc3 (2.0,4.0) -premr 1 -pi0 0.5 -varytheta
-pp 2.0 -ee 2.0 -mr 2
```

```
-pl 1
-esptheta -sd 12345678
-taxa (A_0,B_0,B_1,B_2,B_3,C_0,C_1,C_2,C_3,D_0,O_0)
-tm <A:A_0;B:B_0,B_1,B_2,B_3;C:C_0,C_1,C_2,C_3;D:D_0;O:O_0>
;
```

The second step consists in MCMC sampling during 1.5×10^6 iterations.

```
MCMC_BiMarkers -cl 1500000 -sf 1000 -bl 200000
-pi0 0.5 -varytheta
-pp 2.0 -ee 2.0 -mr 2
-pl 1
-esptheta -sd 12345678
-taxa (A_0,B_0,B_1,B_2,B_3,C_0,C_1,C_2,C_3,D_0,O_0)
-tm <A:A_0;B:B_0,B_1,B_2,B_3;C:C_0,C_1,C_2,C_3;D:D_0;O:O_0>
-snet"..."
;
```

Note that the "-snet" option refers to the starting network obtained from the pre-burnin phase. Besides, the options "-mr" and "-pp" allow to specify the network prior: the maximum number of reticulations was set to 2, and the prior Poisson distribution on the number of reticulation nodes was centered on 2.

6 Supplementary results for the simulation study

Table A. Table linked to Table 1 of the main manuscript. Trees inferred by SnappNet when m=1,000 sites were considered.

Hyperparameters	Network A	Network B
True ($\alpha = 1$, $\beta = 200$, $\frac{\alpha}{\beta} = 0.005$)	78.71% tree ((((Q,A),L),R),C)	35.28% tree (((Q,R),L),(A,C)) 28.54% tree (((Q,L),R),(A,C))
True ($\alpha = 1$, $\beta = 1000$, $\frac{\alpha}{\beta} = 0.001$)	82.82% tree ((((Q,A),L),R),C)	45.27% tree (((Q,R),L),(A,C)) 40.35% tree (((Q,L),R),(A,C))
True ($\alpha = 1$, $\beta = 2000$, $\frac{\alpha}{\beta} = 5 \times 10^{-4}$)	82.92% tree ((((Q,A),L),R),C)	48.40% tree (((Q,R),L),(A,C)) 38.16% tree (((Q,L),R),(A,C))

Table B. Average posterior probability (PP) of the topology of network C obtained by running `MCMCBiMarkers` on data simulated from network C. Same as Table 3 of the main manuscript except that 12×10^6 iterations are considered, and only one lineage is sampled in hybrid species B and C. $\overline{\text{ESS}}$ is the average ESS over the different replicates, and SE stands for the sampler efficiency.

	Number of sites		
	1,000	**10,000**	**100,000**
PP	5.5×10^{-6} (20 replicates)	5.10% (19 replicates)	0% (16 replicates)
SE	2.32×10^{-5}	8.11×10^{-6}	1.96×10^{-5}
$\overline{\textbf{ESS}}$	250.88	87.63	211.57

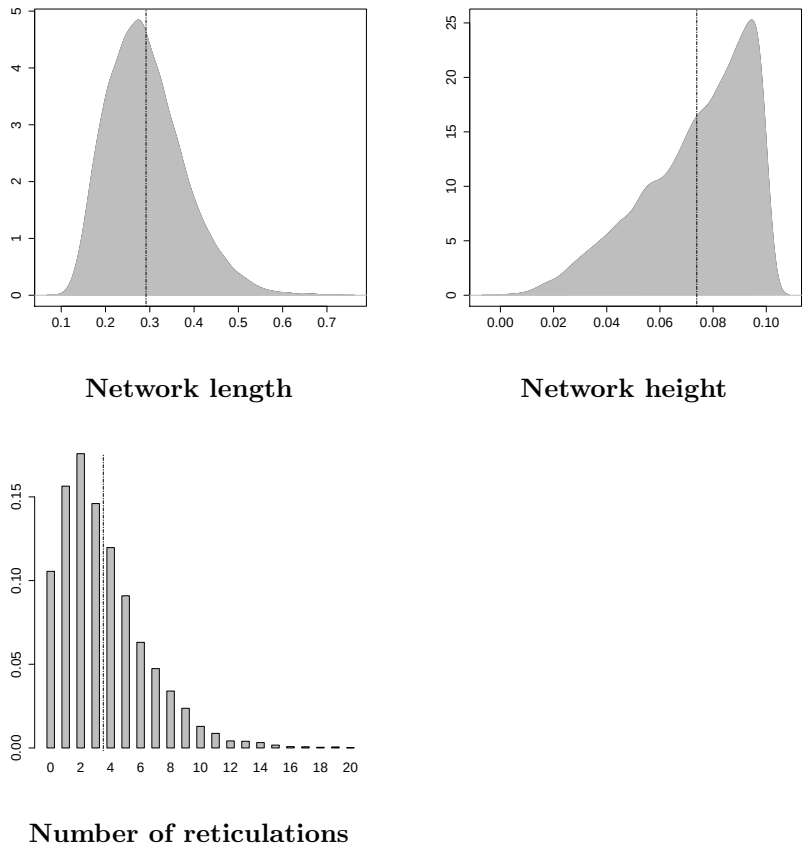


Figure A. Density probabilities for 5-tips networks, simulated with a prior corresponding to a birth hybridization process with parameters $d = 10$, $r = 1/2$ and $\tau_0 = 0.1$, using the SpeciesNetwork package [7]. The figure is obtained for 10,000 replicates. The means are given by the dashed vertical lines.

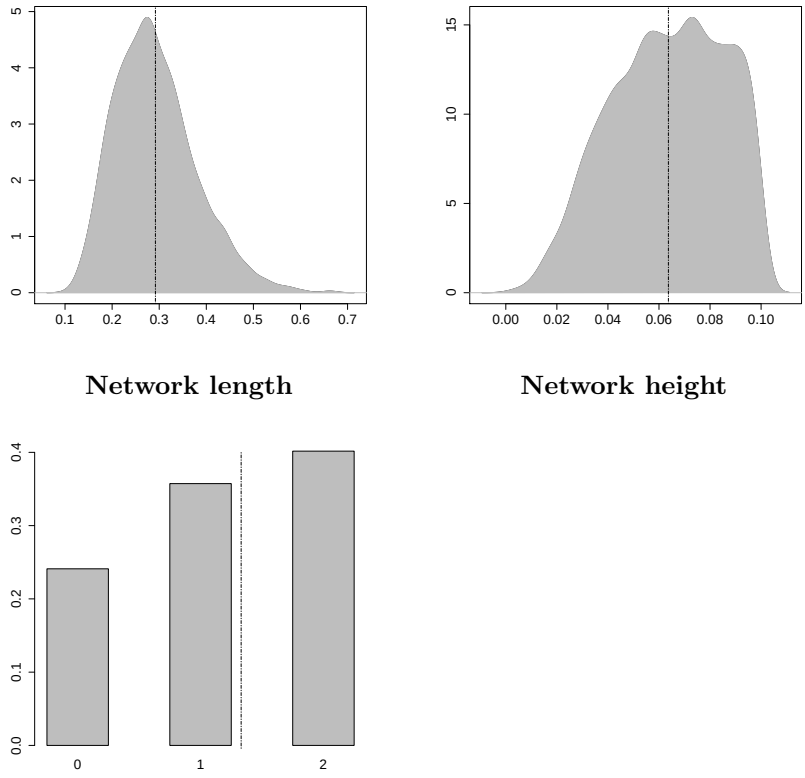


Figure B. Density probabilities for 5-tips networks with at most two reticulations, simulated with a prior corresponding to a birth hybridization process with parameters $d = 10$, $r = 1/2$ and $\tau_0 = 0.1$, using the SpeciesNetwork package [7]. Figures are drawn for the 4,377 cases in 10,000 where the network had at most two reticulations. The means are given by the dashed vertical lines.

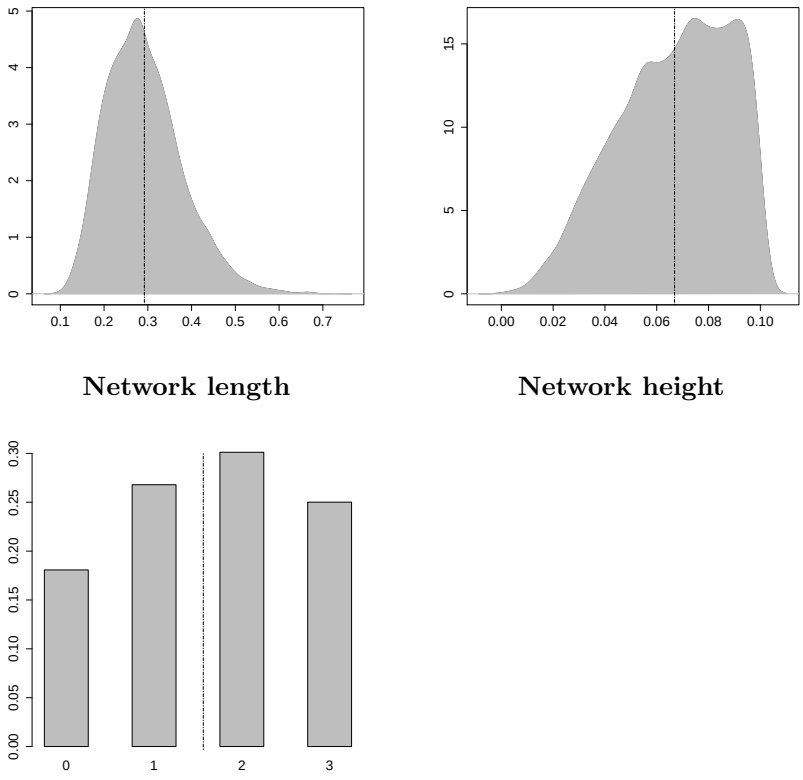


Figure C. Density probabilities regarding the 5-tips network with a maximum of 3 reticulations, simulated under the birth hybridization process ($d = 10$, $r = 1/2$, $\tau_0 = 0.1$, 5,837 replicates), using the SpeciesNetwork package [7]. The means are given by the dashed vertical lines.

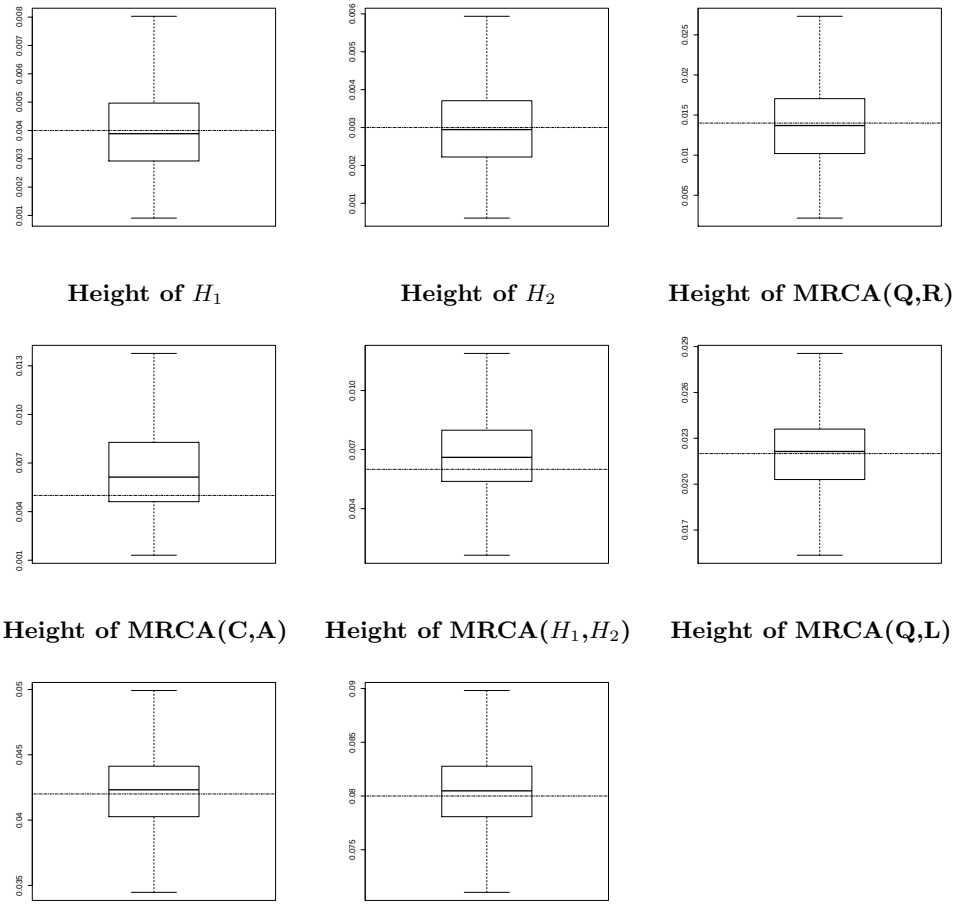


Figure D. Estimated node heights of network B. 10,000 sites are considered and 2 lineages per species. Constant sites are included in the analysis, and the estimated heights are based on the 12 replicates (over 14 replicates) for which network B was recovered by SnappNet (criterion ESS>200 ; $\theta \sim \Gamma(1, 200)$, $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1, 1)$, $\tau_0 \sim \mathcal{E}(10)$ for the priors, number of reticulations bounded by 3 when exploring the network space). Heights are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. The initials MRCA stand for "Most Recent Common Ancestor".

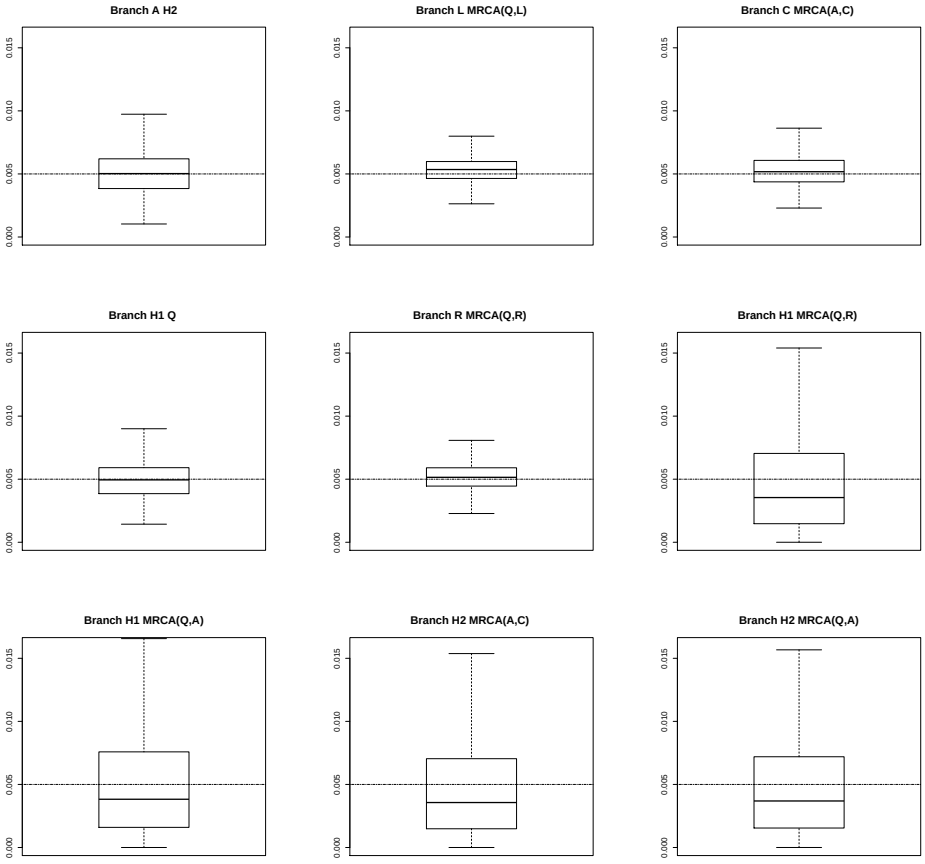


Figure E. Estimated population sizes θ for each branch of network B. 10,000 sites are considered and 2 lineages per species. Same framework as Figure D. True values are given by the dashed horizontal lines. The initials MRCA stand for "Most Recent Common Ancestor".

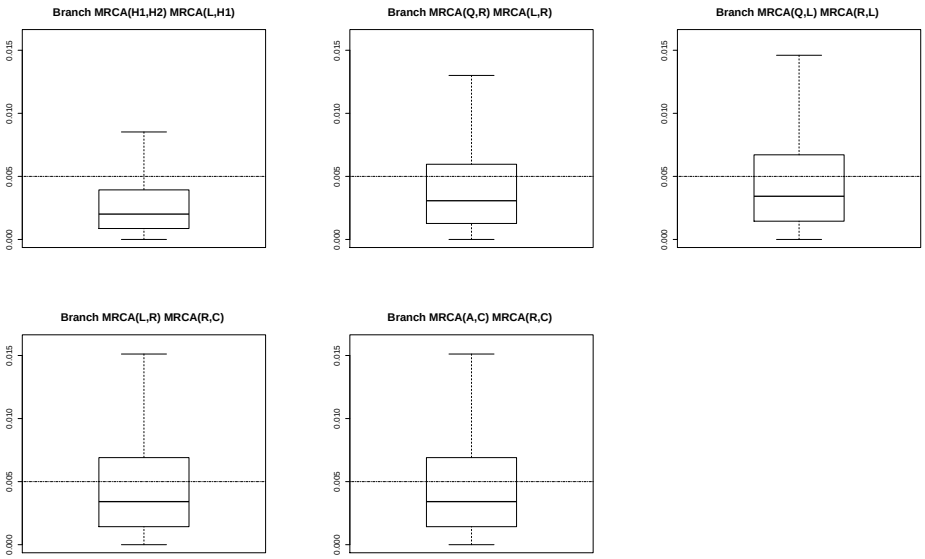
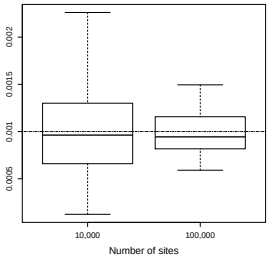
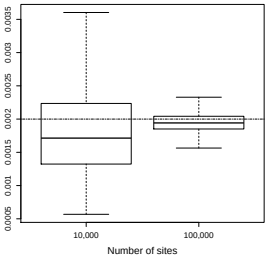


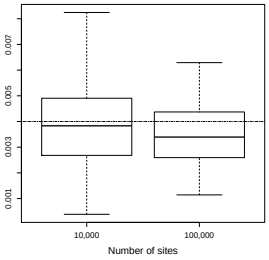
Figure F. Same framework as Figure E.



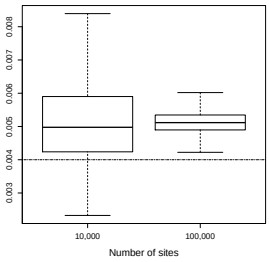
Height of H_1



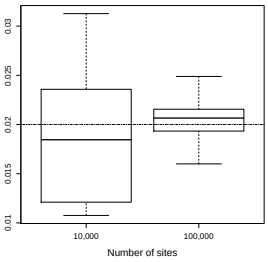
Height of MRCA(B,C)



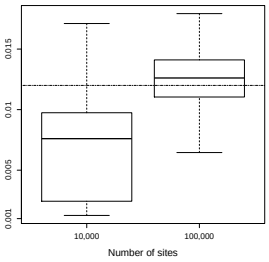
Height of MRCA(D,C)



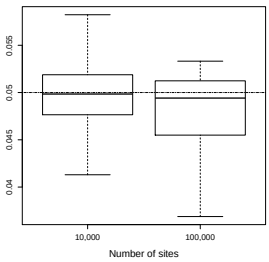
Height of H_2



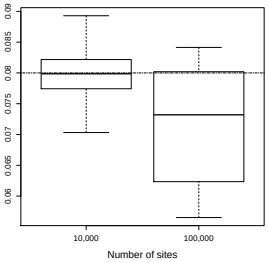
Height of MRCA(B,D)



Height of MRCA(A,B)

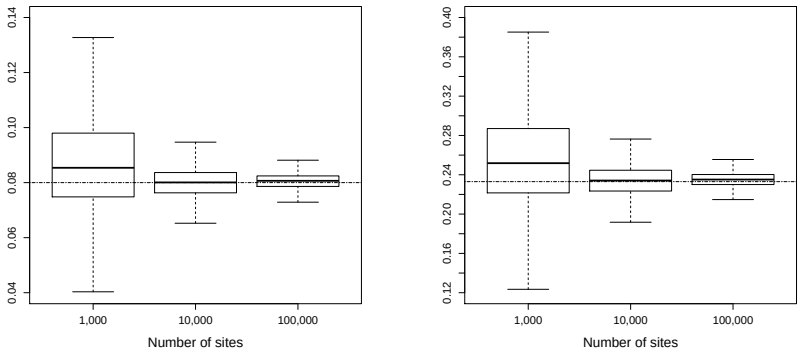


Height of MRCA(A,D)



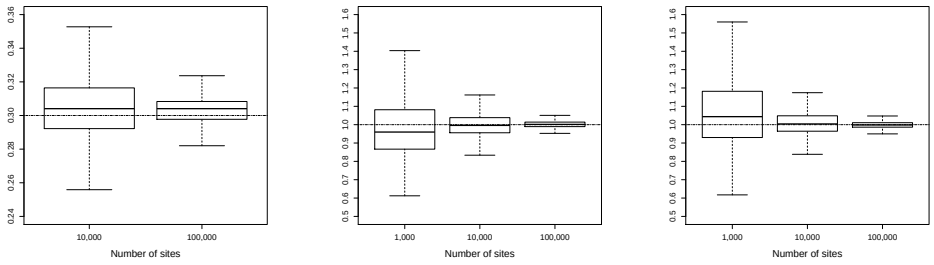
Height of MRCA(O,D)

Figure G. Estimated node heights of network C as a function of the number of sites. Same experiment as in Table 2 of the main manuscript: 1 lineage in species O, A and D, and 4 lineages in species B and C. The estimated heights are based on the replicates for which network C was recovered by SnappNet. True values are given by the dashed horizontal lines. The initials MRCA stand for "Most Recent Common Ancestor".



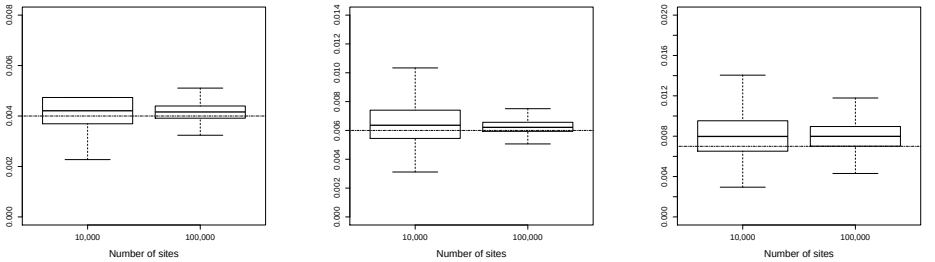
Network height **Network length**

Figure H. Estimated height and length for network A, as a function of the number of sites. Heights and lengths are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. Two lineages per species were simulated. Only polymorphic sites are included in the analysis, and 20 replicates are considered for each simulation set up (criterion ESS$>$ 200 for m=1,000 and m=10,000 , and criterion ESS$>$ 100 for m=100,000; $\theta \sim \Gamma(1, 200)$, $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1, 1)$, $\tau_0 \sim \mathcal{E}(10)$ for the priors, number of reticulations bounded by 2 when exploring the network space). Same framework as in Figure 10 of the main paper, except that only polymorphic sites are taken into account.

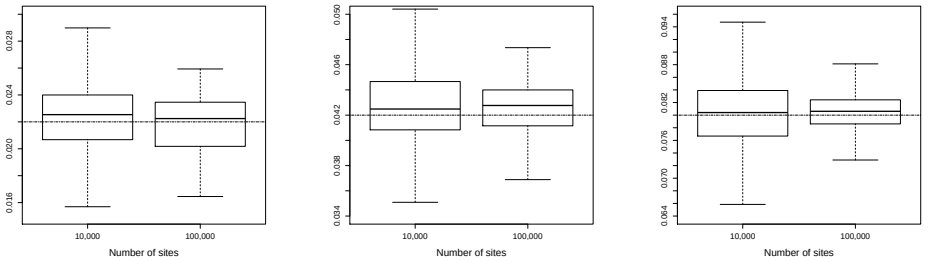


Inheritance probability γ **Instantaneous rate u** **Instantaneous rate v**

Figure I. Estimated inheritance probability and instantaneous rates for network A, as a function of the number of sites. True values are given by the dashed horizontal lines. Same framework as in Figure 11 of the main paper, except that only polymorphic sites are taken into account.



Height of H_1 **Height of MRCA(Q,A)** **Height of MRCA(Q,R)**



Height of MRCA(L,A) **Height of MRCA(L,R)** **Height of MRCA(L,C)**

Figure J. Estimated node heights of network A, as a function of the number of sites. Heights are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. Same framework as in Figure 12 of the main paper, except that only polymorphic sites are taken into account. The initials MRCA stand for "Most Recent Common Ancestor".

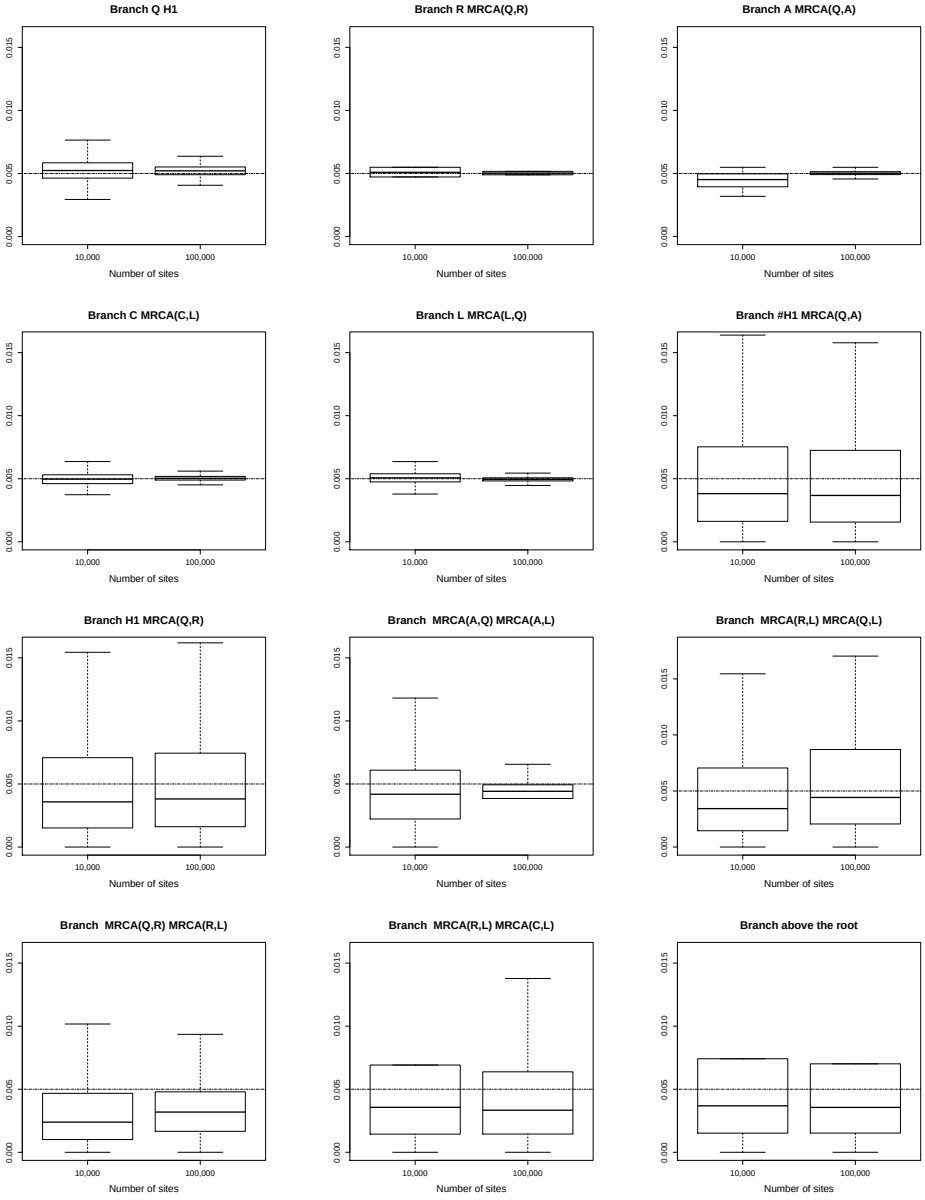
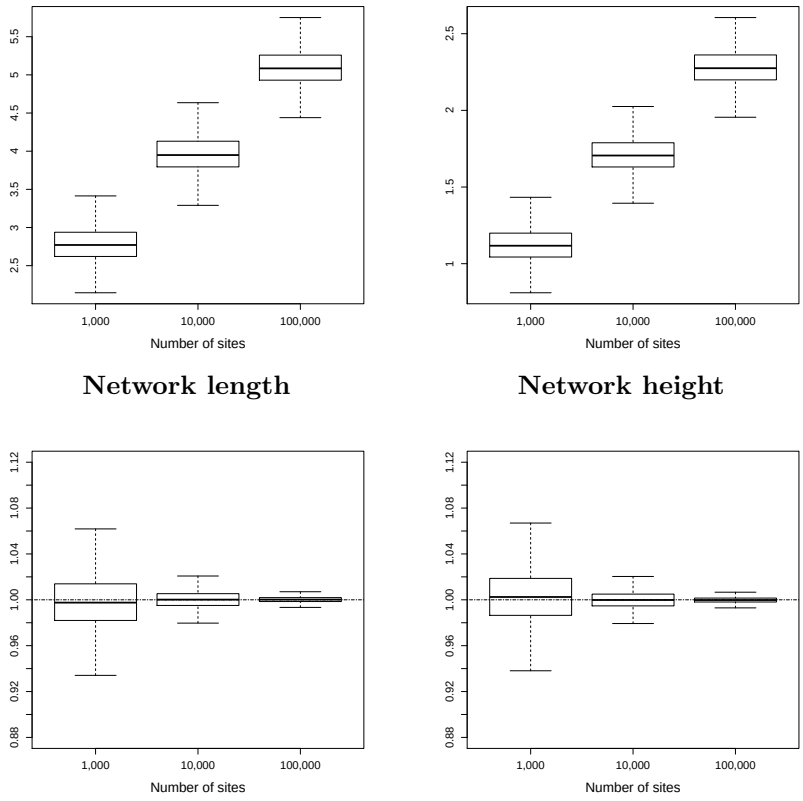


Figure K. Estimated population sizes θ for each branch of network A, as a function of the number of sites. True values are given by the dashed horizontal lines. Same framework as in Figure 13 of the main paper, except that only polymorphic sites are taken into account. The initials MRCA stand for "Most Recent Common Ancestor".



Network length **Network height**

Instantaneous rate u **Instantaneous rate** v

Figure L. Experiments on Network A and based only on polymorphic sites. Same framework as in Figures H and I above, except that the correction factor is not used in the calculations (criterion ESS$>$ 200 in all cases).

7 Supplementary informations on rice real data

Table C. Description of the 24 rice varieties considered in our study. These varieties are either representative cultivars spanning the four main rice subpopulations (Indica, Japonica, *circum* Aus and *circum* Basmati), or wild types (Or1I, Or1A, Or3).

Subpopulation	Variety ID	Country	Variety name
circum Aus	IRIS-313-11058	Bangladesh	AUS 329
	IRIS 313-11737	India	CHUNDI
	IRIS-313-10852	India	ARC 7336
	IRIS-313-11027	Pakistan	JHONA 101
circum Basmati	IRIS-313-11062	Bangladesh	BEGUNBICHI 33
	IRIS-313-8326	India	JC1
	IRIS-313-11258	India	ARC 13502
	IRIS-313-12094	Bangladesh	ARC KASHA
Indica	IRIS-313-11819	Myanmar	PADINTHUMA
	IRIS-313-11089	Cambodia	SRAU THMOR
	IS-313-11646	India	NCS771 A
	IRIS-313-11741	SriLanka	HERATH BANDA
Japonica	B204	China	LONGHUAMAOHU
	IRIS-313-10577	Philippines	IFUGAO RICE
	IRIS-313-11691	Bhutan	SHANGYIPA
	IRIS-313-7883	Indonesia	GANIGI
	B269	China	YUEFU
Or1I	W1117	India	W1117
	W1559	Thailand	W1559
Or1A	W0574	Malaya	W0574
	W1747	India	W1747
Or3	W3042	China	W3042
	W3073	China	W3073
	W3048	China	W3048

Table D. Data set 1, that includes only one variety per subpopulation. These varieties were chosen from Table C.

Subpopulation	Variety ID	Country	Variety name
circum Aus	IRIS-313-10852	India	ARC 7336
circum Basmati	IRIS-313-12094	Bangladesh	ARC KASHA
Indica	IRIS-313-11741	SriLanka	HERATH BANDA
Japonica	B269	China	YUEFU
Or1I	W1559	Thailand	W1559
Or1A	W0574	Malaya	W0574
Or3	W3073	China	W3073

Table E. Data sets 2 and 3, that include two varieties per subpopulation. These varieties were chosen from Table C.

Subpopulation	Data set	Variety ID	Country	Variety name
circum Aus	2	IRIS-313-11058	Bangladesh	AUS 329
		IRIS-313-10852	India	ARC 7336
	3	IRIS 313-11737	India	CHUNDI
		IRIS-313-11027	Pakistan	JHONA 101
circum Basmati	2	IRIS-313-11062	Bangladesh	BEGUNBICHI 33
		IRIS-313-11258	India	ARC 13502
	3	IRIS-313-8326	India	JC1
		IRIS-313-12094	Bangladesh	ARC KASHA
Indica	2	IRIS-313-11819	Myanmar	PADINTHUMA
		IS-313-11646	India	NCS771 A
	3	IRIS-313-11741	SriLanka	HERATH BANDA
		IRIS-313-11089	Cambodia	SRAU THMOR
Japonica	2	B204	China	LONGHUAMAOHU
		IRIS-313-11691	Bhutan	SHANGYIPA
	3	IRIS-313-10577	Philippines	IFUGAO RICE
		IRIS-313-7883	Indonesia	GANIGI
Or1I	2,3	W1117	India	W1117
		W1559	Thailand	W1559
Or1A	2,3	W0574	Malaya	W0574
		W1747	India	W1747
Or3	2	W3042	China	W3042
		W3073	China	W3073
	3	W3048	China	W3048
		W3073	China	W3073

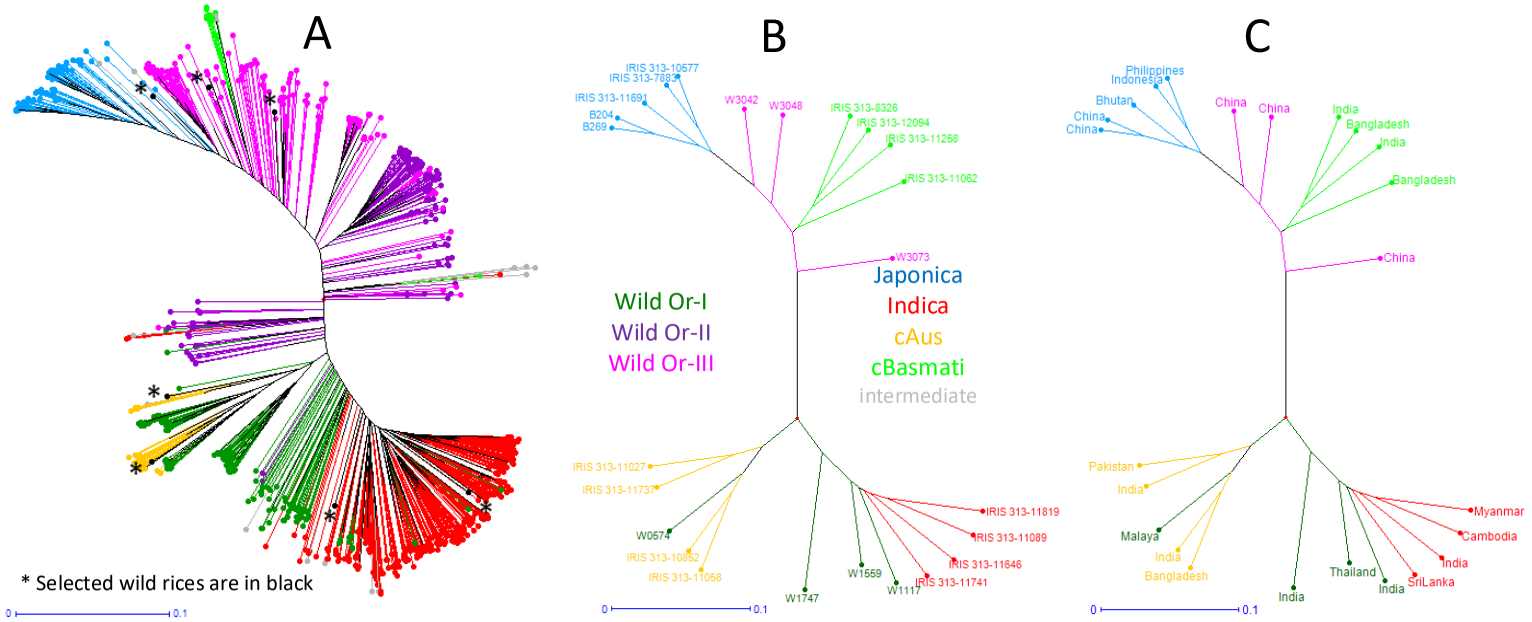
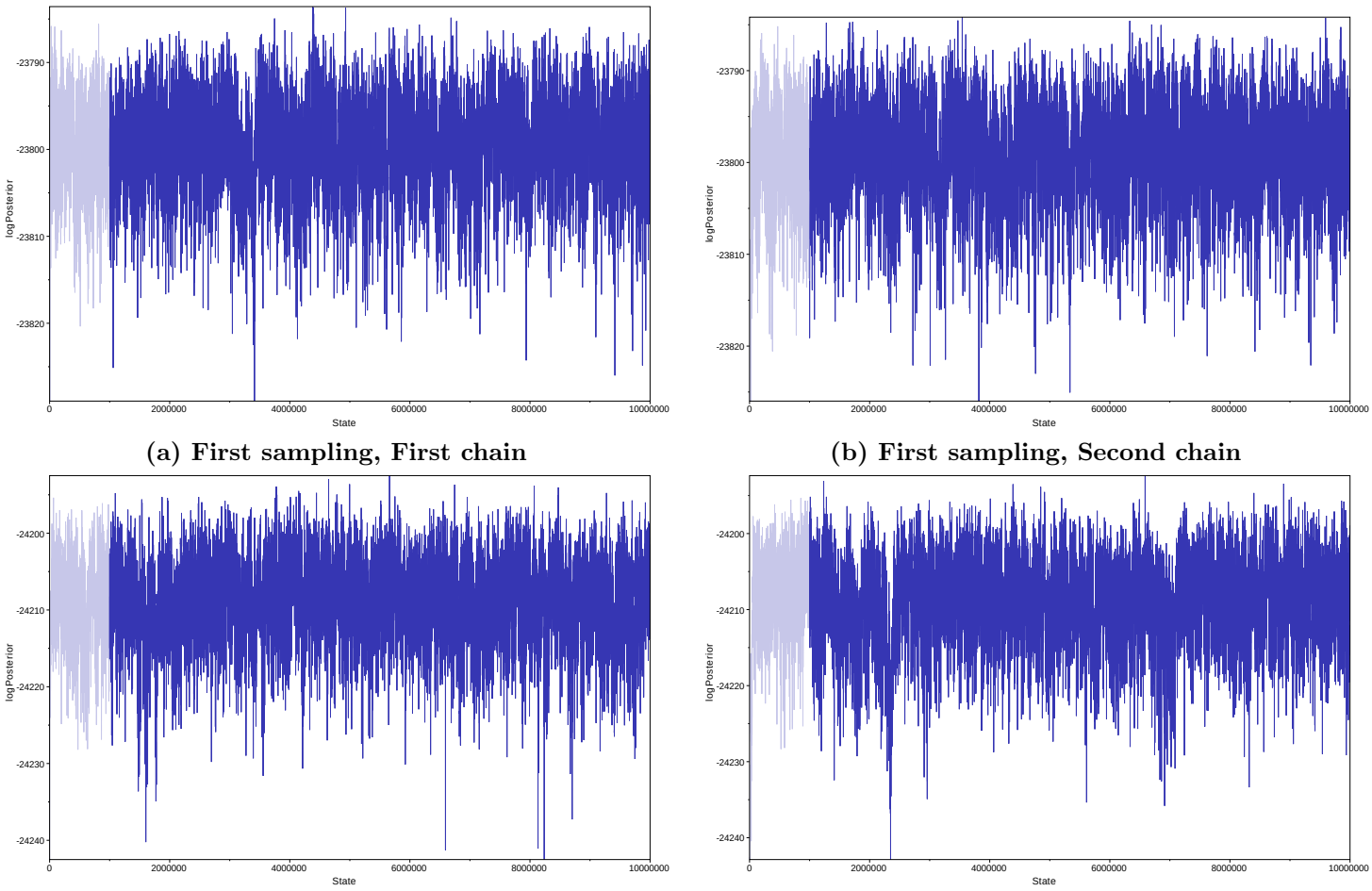


Figure M. Summary of rice molecular diversity used for selecting our sample of rice cultivated varieties and wild types. A: unweighted neighbour joining (UWNJ) tree reflecting dissimilarities among 899 accessions based on 2.48 million SNPs as described in [8]; the accessions are colored according to their classification into wild population types or cultivar groups. B and C: UWNJ tree using the same data for the 24 accessions we selected for assessing SnappNet performance, and showing their accessions number (B) and their country of origin (C); the colors are as in A.



(a) First sampling, First chain

(b) First sampling, Second chain

(c) Second sampling, First chain

(d) Second sampling, Second chain

Figure N. Trace plots obtained according to the Tracer software when data set 1 was analyzed with SnappNet. (a) and (b) refer to the first sampling of 12 kSNPs along the whole genome, whereas (c) and (d) focus on the second sampling. Two chains were considered for each sampling.

Table F. Informations obtained according to the Tracer software, when data set 1 was analyzed with SnappNet. Two different samplings of 12 kSNPs were considered, and also two chains for each sampling.

		First Sampling		Second Sampling	
		Chain 1	**Chain 2**	**Chain 1**	**Chain 2**
LogPosterior	mean	-23799.1709	-23798.9118	-24208.6649	-24208.8018
	stdev	5.7064	5.5892	5.867	5.9986
	median	-23798.6288	-23798.4253	-24208.1221	-24208.2887
	auto-correlation time	10667.9058	7764.256	9263.5725	16818.357
	effective sample size	843.7	1159.3	971.7	535.2
LogLikelihood	mean	-23610.9374	-23610.7083	-24021.9798	-24021.9297
	stdev	4.1712	4.0226	3.9336	4.0567
	median	-23610.6061	-23610.3848	-24021.6408	-24021.5582
	auto-correlation time	34252.7407	28373.4116	31661.3188	68782.4427
	effective sample size	262.8	317.2	284.3	130.9
LogPrior	mean	-188.2335	-188.2035	-186.6851	-186.8721
	stdev	5.7941	5.4802	5.3762	5.4702
	median	-187.8239	-187.7722	-186.2485	-186.3268
	auto-correlation time	17357.6687	11954.8279	9509.4182	15383.8939
	effective sample size	518.6	752.9	946.5	585.1
$\boldsymbol{u}$	mean	0.5567	0.5567	0.5583	0.5583
	stdev	9.4491E-4	9.5177E-4	9.7888E-4	9.5855E-4
	median	0.5567	0.5567	0.5583	0.5582
	auto-correlation time	1898.9027	2043.8061	1913.3736	1932.747
	effective sample size	4740.1	4404	4704.3	4657.1
$\boldsymbol{v}$	mean	4.9094	4.9073	4.7922	4.7909
	stdev	0.0734	0.0739	0.0721	0.0706
	median	4.9072	4.9062	4.7903	4.7925
	auto-correlation time	1896.7939	2044.0868	1944.4961	1936.7774
	effective sample size	4745.4	4403.4	4629	4647.4
$\boldsymbol{d}$	mean	10.7192	10.9194	9.8551	10.0755
	stdev	5.2274	5.3009	4.7636	4.8326
	median	10.0307	10.2443	9.2427	9.4077
	auto-correlation time	3692.7762	2098.2587	5591.047	4875.9969
	effective sample size	2437.5	4289.7	1609.9	1846
$\boldsymbol{r}$	mean	0.2387	0.2349	0.2217	0.2159
	stdev	0.1707	0.1667	0.1633	0.1558
	median	0.1996	0.1952	0.1799	0.179
	auto-correlation time	6276.9007	2088.925	1786.1765	1610.1668
	effective sample size	1434	4308.9	5039.3	5590.1

8 Additional experiments on SnappNet's MCMC sampler

In the following, we describe a few experiments that were conducted to better understand the behavior of the MCMC sampler employed by SnappNet— in particular its efficiency at sampling from network space, and how this efficiency is affected by the priors on phylogenetic network and population sizes. The prior on phylogenetic networks is specified in terms of the birth-hybridization model by Zhang et al. [7].

8.1 Experiment with no data

8.1.1 Protocol

In the first experiment we assess whether the MCMC sampler employed by SnappNet can adequately sample from network space. We specify a posterior distribution over 5-taxon phylogenetic networks with high variance across multiple number of reticulations. We ran the MCMC sampler so that it sampled from a posterior distribution specified in terms of a birth-hybridization model prior, origin height prior and a null likelihood function (always returns zero regardless of the input data). We then compared the sampled networks with 5-taxon networks simulated directly from the birth-hybridization model. Theoretically we expect the distributions of sampled and simulated networks to match.

We studied three different cases of the birth-hybridization model prior, for each case we either specified a normal prior with mean 0.1 and standard deviation of 0.01 on the origin height or an exponential prior with mean 0.1 on the origin height (that is a total of six different scenarios): In the first case we used a birth-hybridization model with speciation rate 20 and hybridization rate 1 (mean number of reticulations close to zero). In the second case we used a birth-hybridization model with speciation rate 20 and hybridization rate 2 (mean number of reticulations close to one). In the third case we used a birth-hybridization model with speciation rate 20 and hybridization rate 3 (mean number of reticulations close to two). We only kept simulated networks with 5 leaves.

In each case we simulated 1000 networks directly from the birth-hybridization model and sampled 2,000,000 networks using the SnappNet sampler (burning half the chain and logging every 1000th sample thereafter). Note that it is possible to fix the birth and hybridization rates in the prior used by SnappNet by fixing corresponding values for parameters d and r. We used Tracer to assess convergence of the MCMC chain by visually inspecting the trace and computing the ESS (effective sample size). Thereafter we compared the simulated networks with sampled networks in terms of the number of reticulations, time until first reticulation, network height and network length.

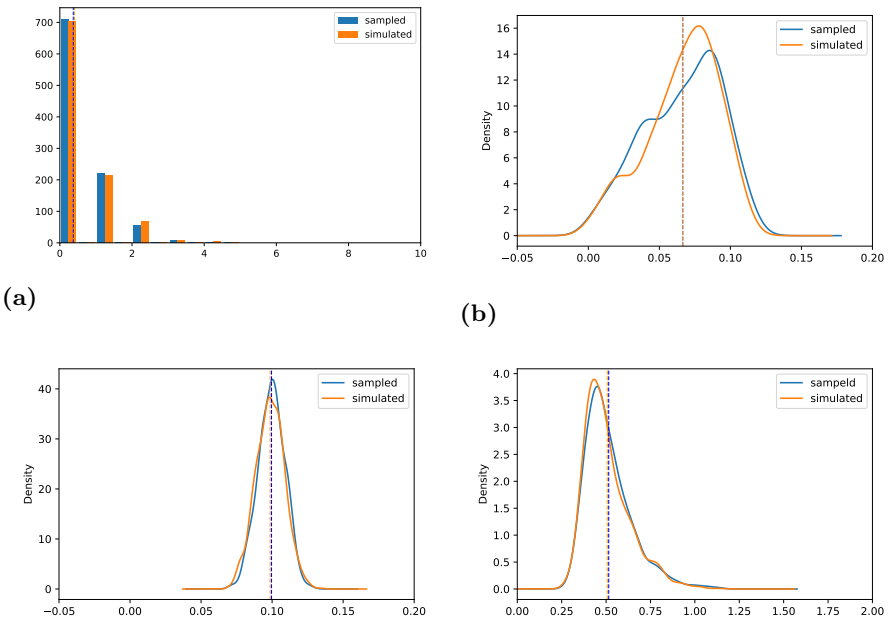
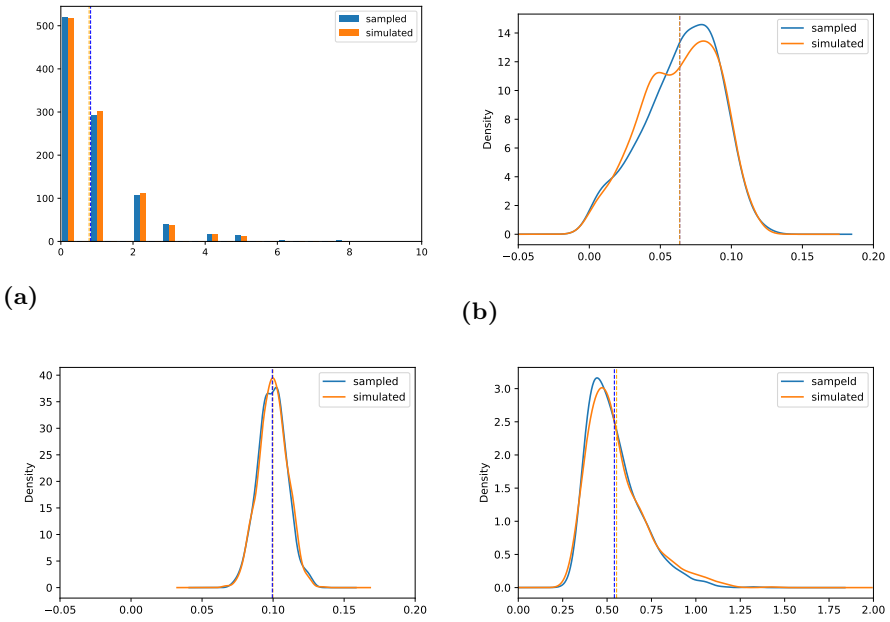


Figure O. Birth-hybridisation model with speciation rate 20 and hybridisation rate 1 (mean number of reticulations close to zero) and a normal prior with mean 0.1 and standard deviation of 0.01 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

8.1.2 Results

In the first experiment the sampler converged to the specified prior in all three cases (for both origin height priors) based on the computed summary statistics (see Figs O-Q and Figs R-T). The convergence of the sampler in all cases is a good indication that the implemented moves worked well enough. The ESS for the sampled networks given the normal prior on the network origin were: 1001 for the first case (mean number of reticulation close to zero); 844 for the second case (mean number of reticulations close to one); 1001 for the third case (mean number of reticulations close to two). The ESS for the sampled networks given the exponential prior on the network origin were: 872 for the first case; 955 for the second case; 838 for the third case.
Note that the normal and exponential priors on the origin height permit to describe different knowledge on the expected number of reticulations, see Figs Q(a) and T(a).



(a) **(b)**

(c) **(d)**

Figure P. Birth-hybridisation model with speciation rate 20 and hybridisation rate 2 (mean number of reticulations close to one) and normal prior with mean 0.1 and standard deviation of 0.01 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

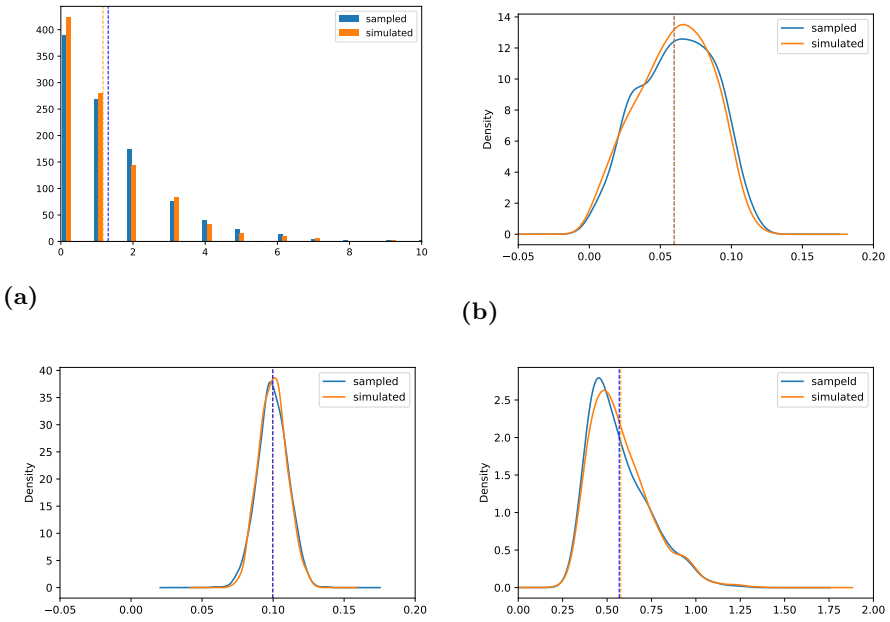


Figure Q. Birth-hybridisation model with speciation rate 20 and hybridisation rate 3 (mean number of reticulations close to two) and normal prior with mean 0.1 and standard deviation of 0.01 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

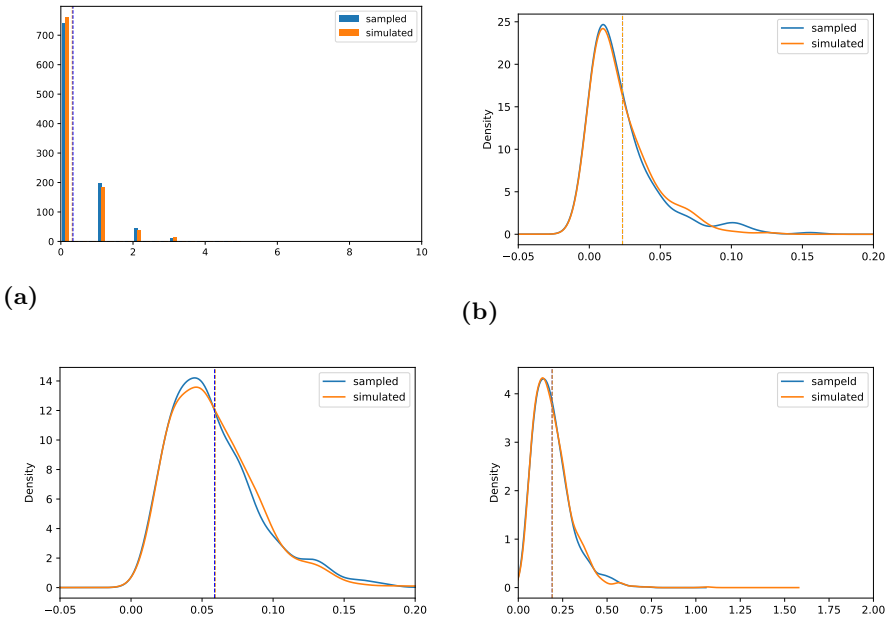


Figure R. Birth-hybridisation model with speciation rate 20 and hybridisation rate 1 (mean number of reticulations close to zero) and an exponential prior with mean 0.1 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

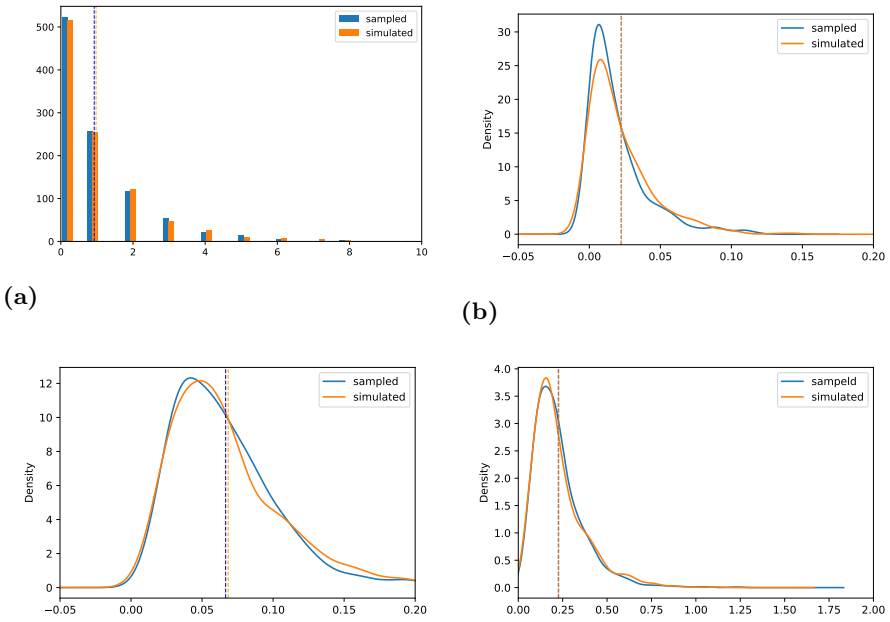
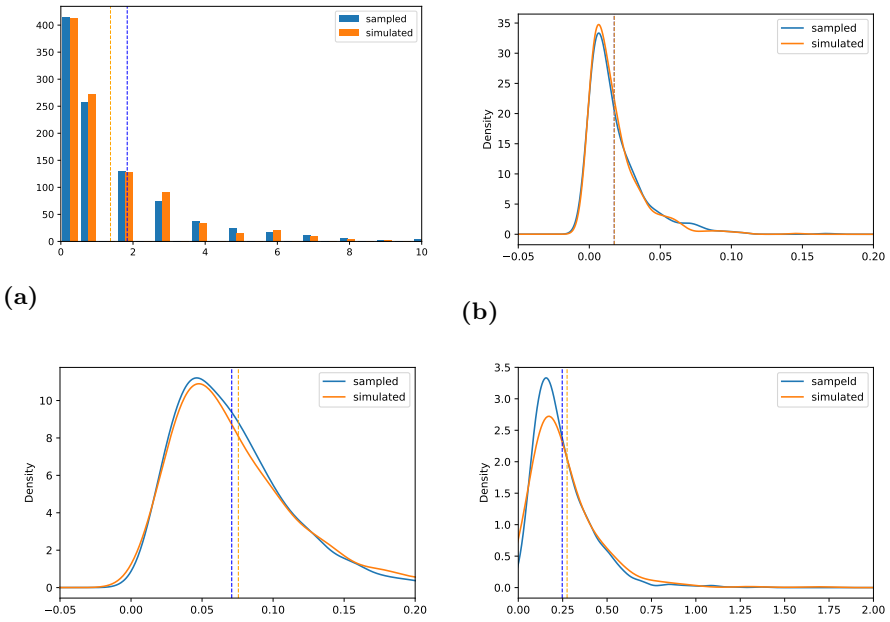


Figure S. Birth-hybridisation model with speciation rate 20 and hybridisation rate 2 (mean number of reticulations close to one) and an exponential prior with mean 0.1 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.



(a) **(b)**

(c) **(d)**

Figure T. Birth-hybridisation model with speciation rate 20 and hybridisation rate 3 (mean number of reticulations close to two) and an exponential prior with mean 0.1 on the origin height. We plot the simulated networks (orange) against the sampled networks (blue) summarising the networks under: (a) Number of reticulations (b) Time until first reticulation (c) Height of the network (d) Length of the network.

#Chain	Network prior	Mean reticulations	Pop size prior
1,2	BH(20,1)	0.371	$\Gamma(1, 200)$
3,4	BH(20,1)	0.371	$\Gamma(1, 20)$
5,6	BH(20,1)	0.371	$\Gamma(1, 1000)$
7,8	BH(20,1)	0.371	$\Gamma(1, 2000)$
9,10	BH(20,2)	0.861	$\Gamma(1, 200)$
11,12	BH(20,2)	0.861	$\Gamma(1, 20)$
13,14	BH(20,2)	0.861	$\Gamma(1, 1000)$
15,16	BH(20,2)	0.861	$\Gamma(1, 2000)$
17,18	BH(20,3)	2.265	$\Gamma(1, 200)$
19,20	BH(20,3)	2.265	$\Gamma(1, 20)$
21,22	BH(20,3)	2.265	$\Gamma(1, 1000)$
23,24	BH(20,3)	2.265	$\Gamma(1, 2000)$

Table G. BH(birth rate, hybridisation rate) refers to the birth-hybridisation process of Zhang et al. with the specified birth and hybridisation rates. For data simulated with network A, only chains 1,2,3,4,9,10,11,12,17,18,19,20 were run. We indicate the mean number of reticulation for the Birth-Hybridization model given an exponential prior with mean 0.1 on network origin. Note that we only used the exponential prior in the experiment in Section 8.2.

8.2 Experiments on 10,000 simulated sites

8.2.1 Protocol

In the second experiment we assess how population size priors and network priors influence SnappNet's inferences, in particular the rate of convergence and sampling efficiency of the MCMC sampler. Recall that the network prior specifies a hybridization rate, whereas the prior on population sizes affects the probability of coalescence, and therefore that of ILS. Thus, these two priors have an important role in determining the relative probability of hybridization and ILS as causes of incongruent (non-tree-like) signals in the data.

We simulated 10,000 SNPs for network A and network B under the multispecies network coalescent using SimSnappNet. For each of these two simulated SNP datasets, we ran 12 (for network A) or 24 (for network B) MCMC chains, for 500,000 iterations each. See Table G for details on the priors specified for each chain. In this experiment we only use the exponential prior with mean 0.1 on the network origin.

Briefly, as in the experiment of Sec. 8.1, we specified a network prior using the birth-hybridisation model of Zhang et al. [7]. Again, we fixed the birth rate to 20 for all MCMC chains and chose a hybridisation rate so that the mean number of reticulations is close to zero, one or two. Furthermore we specified either a 'correct' or 'incorrect' prior on population size ('correct' implies the mean of the prior distribution corresponds to the population size parameter used to simulate the SNP dataset). The 'correct' population size prior on each

branch was specified as $\Gamma(1, 200)$. The 'incorrect' population size prior on each branch was specified as $\Gamma(1, 20)$. For network B we considered two additional incorrect population size priors, namely $\Gamma(1, 1000)$ and $\Gamma(1, 2000)$. Note that the rest of the priors of the model used the default SnappNet settings. In order to assess convergence we ran two MCMC chains for each prior setting (as specified in Table G). We randomly drew initial networks and population sizes for each MCMC chain from the prior distribution. Also note that, here we do not impose any upper bound on the number of reticulations in the sampled networks.

8.2.2 Results for network A

We summarize results for data simulated under network A in Fig U (MCMC chains with correct population size priors) and Fig V (MCMC chains with incorrect population size priors). We also give detailed summary statistics in Table H (MCMC chains with correct population size priors) and Table I (MCMC chains with incorrect population size priors). We note that all chains with correct population size priors converged to the correct topology, network height and network length (see Figs U(c) and U(d)). We assume convergence for network topology since there was only one unique topology for each posterior distribution of the chains with correct population size priors. In each case the unique topology matched up with the topology of network A. Furthermore in Fig U(b) all chains have similar prior distributions. This could be due to the topology of network A that is very unlikely under all the specified birth-hybridization model priors (similar to sampling from a flat prior). We also note a much lower ESS under the model prior with reticulation mean close to zero (see ESS in Table H).

Chains with incorrect population size priors also converged to the correct topology. Similar to correct priors, there was only one unique topology for all chains. However the chains did not converge to the correct network height or network length. This is not unexpected since the length of a branch and its associated population size are correlated (see Bryant et al. [2] for more detail). Furthermore the ESS for chains with incorrect population size priors is significantly lower than chains with correct population size priors (see ESS in Table H and Table I). There is also a difference in ESS between chains with different topology priors. In this case ESS is highest when the mean number of reticulations on the network topology prior is close to one and lowest when mean number of reticulations for the network topology prior is close to two. This seems to suggest that specifying a prior with correct mean number of reticulations can improve sampling efficiency.

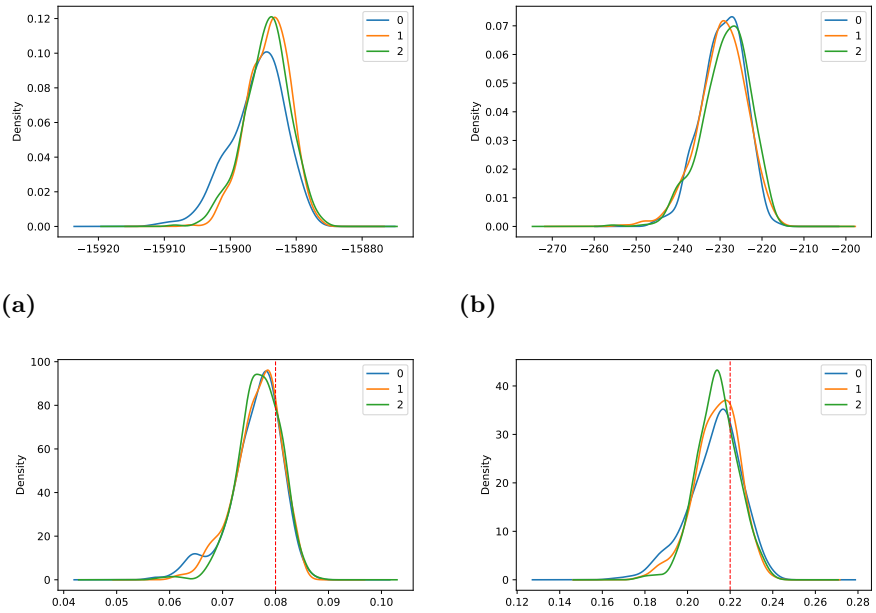
Table H. MCMC summary statistics for network A (correct population size priors)

Posterior	0	1	2
mean	-16124.9199	-16123.2496	-16123.8596
stdev	4.7795	4.9024	5.0388
median	-16124.601	-16122.7308	-16123.2257

95% HPD Interval	[-16134.1, -16116.4]	[-16133.4, -16114.9]	[-16133.0, -16114.7]
Auto-correlation time	4537.3247	1336.3834	1454.4489
Effective sample size	198.5752	674.2077	619.4786
Network height	0	1	2
mean	0.076	0.0769	0.0767
stdev	5.12E-03	4.09E-03	4.34E-03
median	0.0768	0.0772	0.0774
95% HPD Interval	[0.0647, 0.0854]	[0.0686, 0.0844]	[0.0682, 0.0847]
Auto-correlation time	10376.3331	3552.1422	3800.8964
Effective sample size	86.8322	253.6498	237.0493
Network length	0	1	2
mean	0.213	0.2144	0.2137
stdev	0.0124	9.93E-03	0.0107
variance	1.54E-04	9.87E-05	1.14E-04
95% HPD Interval	[0.1882, 0.2369]	[0.1967, 0.2357]	[0.1934, 0.2344]
Auto-correlation time	7533.4457	3113.3204	3818.75
Effective sample size	119.6	289.4016	235.9411

Table I. MCMC summary statistics for network A (incorrect priors)

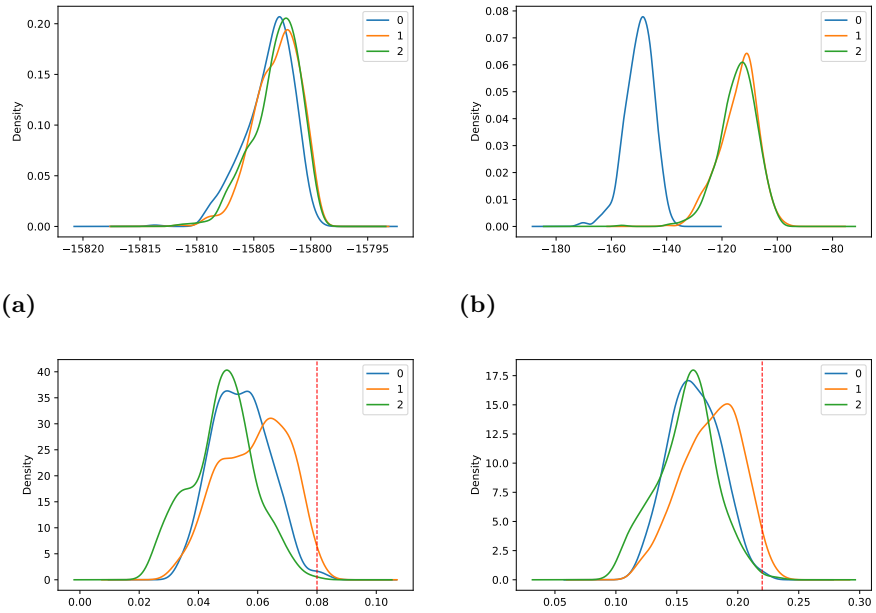
Posterior	0	1	2
mean	-15953.476	-15917.2941	-15917.7676
stdev	5.1764	7.0277	6.7969
median	-15952.9488	-15916.1712	-15917.1603
95% HPD Interval	[-15962.4, -15943.6]	[-15932.0, -15905.4]	[-15930.3, -15905.5]
Auto-correlation time	2395.3008	2164.8158	3002.7331
Effective sample size	167.4111	185.2352	133.545
Network height	0	1	2
mean	0.0548	0.0588	0.047
stdev	9.30E-03	0.0121	0.0112
median	0.0548	0.06	0.0476
95% HPD Interval	[0.0394, 0.073]	[0.0383, 0.0791]	[0.025, 0.0662]
Auto-correlation time	16248.1423	1.00E+05	34725.0867
Effective sample size	24.6797	4.0042	11.5478
Network length	0	1	2
mean	0.1655	0.1783	0.156
stdev	0.0206	0.0252	0.0255
median	0.1649	0.1805	0.158
95% HPD Interval	[0.1264, 0.2024]	[0.1331, 0.2263]	[0.1064, 0.201]
Auto-correlation time	11802.0222	74214.3724	27882.5047
Effective sample size	33.9772	5.4033	14.3818



(a) **(b)**

(c) **(d)**

Figure U. Summary distributions of all chains with correct population size priors (chain numbers 1,2,9,10,17,18) given data simulated from network A. We summarize the MCMC chains by combining them, that is: Chains 1 and 2 are indicated by the blue line (mean reticulations close to zero); Chains 9 and 10 are indicated by the orange line (mean reticulations close to one); Chains 17 and 18 are indicated by the green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.



(a) **(b)**

(c) **(d)**

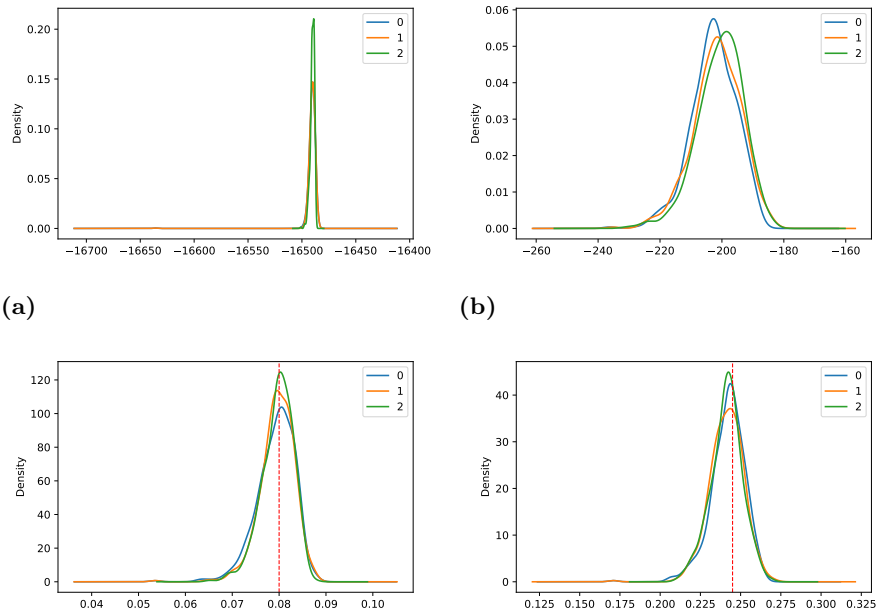
Figure V. Summary distributions of all chains with incorrect population size priors Gamma(1,20) (chain numbers 3,4,11,12,19,20) given data simulated from network A. We summarize the MCMC chains by combining them, that is: Chains 3 and 4 are indicated by the blue line (mean reticulations close to zero); Chains 11 and 12 are indicated by the orange line (mean reticulations close to one); Chains 19 and 20 are indicated by the green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.

8.2.3 Results for network B

We summarize results for data simulated under network B in Fig W (MCMC chains with correct population size priors), Fig X (MCMC chains with incorrect population size priors $\Gamma(1, 20)$), Fig Y (MCMC chains with incorrect population size priors $\Gamma(1, 1000)$) and Fig Z (MCMC chains with incorrect population size priors $\Gamma(1, 2000)$). We also give detailed summary statistics in Table J (MCMC chains with correct population size priors), Table K (MCMC chains with incorrect population size priors $\Gamma(1, 20)$), Table L (MCMC chains with incorrect population size priors $\Gamma(1, 1000)$) and Table M (MCMC chains with incorrect population size priors $\Gamma(1, 2000)$). We note that all chains with correct population size priors converged to the correct topology (posterior distribution contained only one network topology), network height and network length (see Figs W(c) and W(d)). Chains with incorrect population size priors also converged to the correct topology in most cases except for two cases: {BH(20,1),$\Gamma(1, 1000)$} and {BH(20,1),$\Gamma(1, 2000)$}. Therefore we were able to recover the correct topology 83.33% of the time. This is consistent with results in the simulation study of the main text. There is also a difference in ESS between chains with different topology priors. However in this case it is not clear how the prior affects the sampling efficiency (see ESS of Posterior distribution in Table J, Table K, Table L and Table M).

Table J. MCMC summary statistics for Network B (correct population size priors)

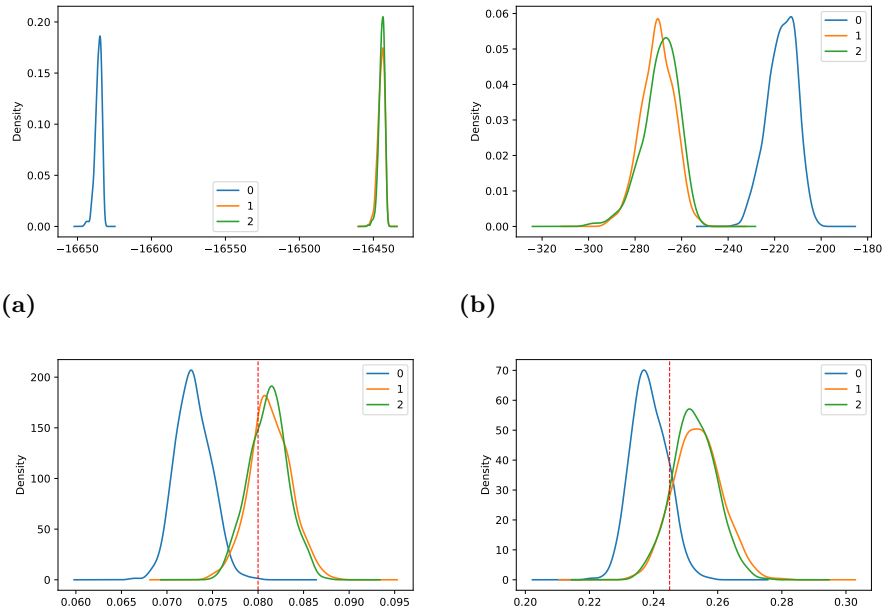
Posterior	0	1	2
mean	-16693.3667	-16692.247	-16690.8344
stdev	10.9267	11.2221	7.5266
median	-16692.4881	-16691.493	-16690.2388
95% HPD Interval	[-16706., -16678.2155]	[-16705.0995, -1667389	[-16705., -16676.7232]
Auto-correlation time	1000	1032	1138.8635
Effective sample size	501	489	439.9122
Network height	0	1	2
mean	0.0793	0.0796	0.0797
stdev	4.08E-03	3.71E-03	3.37E-03
median	0.0799	0.0797	0.08
95% HPD Interval	[0.0714, 0.086]	[0.072, 0.0858]	[0.0729, 0.0852]
Auto-correlation time	4047.2001	2842.2081	2047.1454
Effective sample size	123.7893	176.2714	244.731
Network length	0	1	2
mean	0.2421	0.2412	0.2412
stdev	0.0106	0.0106	9.82E-03
median	0.2431	0.2417	0.2418
95% HPD Interval	[0.2223, 0.2621]	[0.2229, 0.2625]	[0.2193, 0.2588]
Auto-correlation time	3262.683	2788.3608	1948.3473
Effective sample size	153.5546	179.6755	257.141



(a) **(b)**

(c) **(d)**

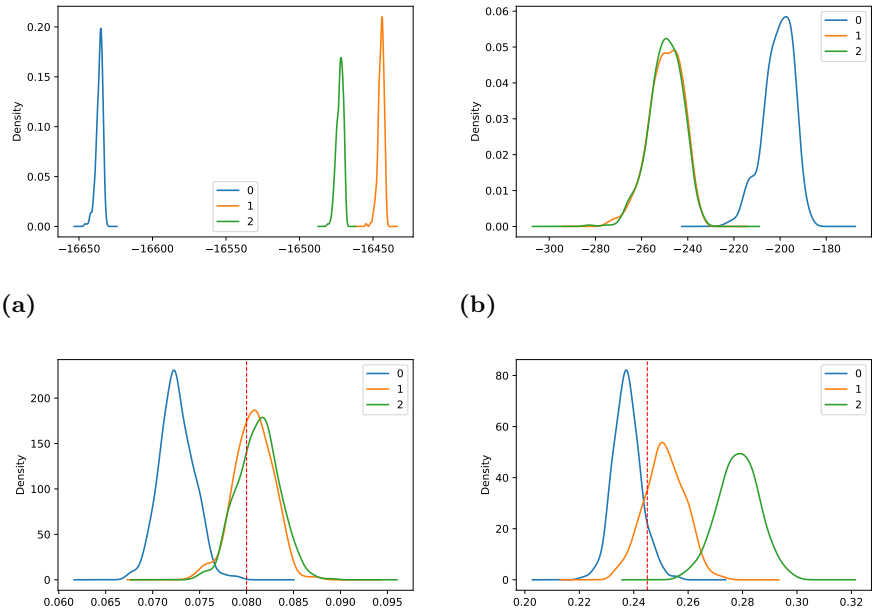
Figure W. Summary distributions of all chains with correct population size priors (chain numbers 1,2,9,10,17,18 given data simulated under network B. We summarize the MCMC chains by combining them, that is: Chains 1 and 2 are indicated by the blue line (mean reticulations close to zero); Chains 9 and 10 are indicated by the orange line (mean reticulations close to one); Chains 17 and 18 are indicated by the green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.



(a) **(b)**

(c) **(d)**

Figure X. Summary distributions of all chains with incorrect population size priors (chain numbers 3,4,7,8,11,12) given data simulated from network B. We summarize the MCMC chains by combining them, that is: Chains 3 and 4 are indicated by blue line (mean reticulations close to zero); Chains 7 and 8 are indicated by orange line (mean reticulations close to one); Chains 11 and 12 are indicated by green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.



(a) **(b)**

(c) **(d)**

Figure Y. In this figure we plot summary distributions of all chains with incorrect population size priors Gamma(1,20) (chain numbers 5,6,13,14,21,22) given data simulated from Network B. We summarize the MCMC chains by combining them, that is: Chains 5 and 6 are indicated by blue line (mean reticulations close to zero); Chains 13 and 14 are indicated by orange line (mean reticulations close to one); Chains 21 and 22 are indicated by green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.

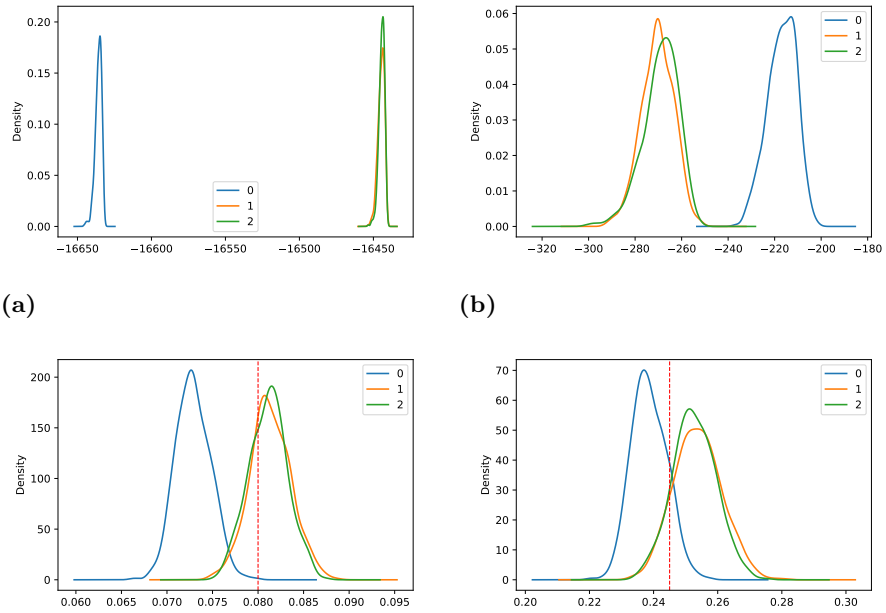


Figure Z. In this figure we plot summary distributions of all chains with incorrect population size priors (chain numbers 7,8,15,16,23,24) given data simulated from network B. We summarize the MCMC chains by combining them, that is: Chain 7 and 8 are indicated by blue line (mean reticulations close to zero); Chain 15 and 16 is indicated by orange line (mean reticulations close to one); Chain 23 and 24 are indicated by green line (mean reticulations close to two); We plot the following distributions (a) Likelihood (b) Prior (c) Network height (d) Network length. Note that network height and network length used to simulate data are indicated by red lines.

Table K. MCMC summary statistics for Network B (incorrect population size priors Gamma(1,20))

Posterior	0	1	2
mean	-16632.0842	-16630.1919	-16629.0074
stdev	8.0517	8.0343	8.3013
median	-16631.0958	-16629.7991	-16627.8702
95% HPD Interval	[-16647.6, -16617.2]	[-16646.3, -16615.2]	[-16644.3, -16613.8]
Auto-correlation time	3035.0395	2302.8325	1819.3308
Effective sample size	132.1235	174.1334	220.4107
Network height	0	1	2
mean	0.0639	0.0568	0.0578
stdev	0.011	0.0119	0.0117
median	0.0645	0.0558	0.0571
95% HPD Interval	[0.044, 0.0838]	[0.039, 0.083]	[0.0392, 0.0789]
Auto-correlation time	25480.1793	59294.2576	31555.2738
Effective sample size	15.7377	6.7629	12.7079
Network length	0	1	2
mean	0.1958	0.1826	0.1831
stdev	0.024	0.0253	0.0255
median	0.196	0.1836	0.1827
95% HPD Interval	[0.1543, 0.2412]	[0.1398, 0.2311]	[0.1378, 0.2273]
Auto-correlation time	19208.3139	36324.2679	23849.9552
Effective sample size	20.8764	11.0395	16.8134

Table L. MCMC summary statistics for Network B (incorrect population size priors Gamma(1,1000))

Posterior	0	1	2
mean	-16836.9572	-16693.995	-16722.2755
stdev	6.9406	7.7519	7.5566
median	-16836.2496	-16693.5673	-16722.1075
95% HPD Interval	[-16850.7, -16824.8]	[-16709.2, -16679.9]	[-16738.3, -16709.7]
auto-correlation time (ACT)	1665.166	1018.3201	1230.9875
effective sample size (ESS)	240.8168	393.7858	325.7547
Network height	0	1	2
mean	0.0726	0.0807	0.0811
stdev	1.88E-03	2.13E-03	2.20E-03
median	0.0724	0.0808	0.0811
95% HPD Interval	[0.0695, 0.0764]	[0.0768, 0.0847]	[0.0774, 0.0854]
auto-correlation time (ACT)	1000	1199.9843	1160.6579
effective sample size (ESS)	401	334.171	345.4937
Network length	0	1	2
mean	0.2375	0.2511	0.2781
stdev	5.38E-03	7.50E-03	7.67E-03
median	0.2371	0.2509	0.2784
95% HPD Interval	[0.228, 0.2492]	[0.2374, 0.2664]	[0.2633, 0.2928]

auto-correlation time (ACT)	1000	1221.2116	1413.5422
effective sample size (ESS)	401	328.3624	283.6845

Table M. MCMC summary statistics for Network B (incorrect population size priors Gamma(1,2000))

Posterior	0	1	2
mean	-16852.5586	-16715.134	-16713.4537
stdev	6.4594	7.6484	7.6353
95% HPD Interval	[-16864.9, -16841.0]	[-16731.4, -16701.6]	[-16727.7, -16698.8]
auto-correlation time (ACT)	1216.2537	1762.3287	1120.84
effective sample size (ESS)	329.7009	227.5398	357.7674
Network height	0	1	2
mean	0.0729	0.0813	0.0811
stdev	1.92E-03	2.25E-03	2.10E-03
95% HPD Interval	[0.0697, 0.0769]	[0.0767, 0.0855]	[0.0771, 0.0852]
auto-correlation time (ACT)	1455.1062	1073.7888	1425.0583
effective sample size (ESS)	275.5813	373.444	281.392
Network length	0	1	2
mean	0.2386	0.2539	0.2524
stdev	5.61E-03	7.39E-03	6.74E-03
variance	3.14E-05	5.46E-05	4.55E-05
95% HPD Interval	[0.2284, 0.2493]	[0.239, 0.2668]	[0.239, 0.2653]
auto-correlation time (ACT)	1519.2302	1462.2703	1020.1866
effective sample size (ESS)	263.9495	274.2311	393.0654

8.2.4 Operator acceptance rates

To better understand the behavior of the MCMC sampler, we inspect the acceptance rates for the 5 operators acting on the network topology (*AddReticulation, DeleteReticulation, FlipReticulation RelocateBranch, RelocateBranchNarrow*), the 4 operators updating branch lengths (*NodeSlider, NodeUniform, NetworkMultiplier, OriginMultiplier*) and the 2 operators updating population sizes (*ChangeGamma, ChangeAllGamma*).

We summarize the acceptance rates for network B in Table N, Table O and Table P. Each table focuses on a different population size prior, while averaging across the topology priors.

We observe that MCMC moves that update topology have a much lower acceptance rate than MCMC moves that update branch lengths and population sizes. *FlipReticulation* moves, which flip the direction of a reticulation branch, are the least likely to be accepted. There is no clear difference in the acceptance rates between different population size priors. More work is needed to determine what the proposal weights should be in order to optimally sample from the posterior distribution.

Table N. MCMC acceptance rates for Network B (correct population size priors).

Id	Pr_accept	Pr_proposed	Pr_proposed	Pr_accept
Topology moves				
AddReticulation	1.43E-04		2.32E-02	3.31E-06
DeleteReticulation	4.55E-05		2.32E-02	1.06E-06
FlipReticulation	8.05E-06		2.35E-02	1.89E-07
RelocateBranch	3.07E-02		2.34E-02	7.20E-04
RelocateBranchNarrow	1.81E-03		2.33E-02	4.21E-05
Branch length				
NodeSlider	5.29E-01		2.32E-02	1.23E-02
NodeUniform	2.73E-01		2.32E-02	6.33E-03
NetworkMultiplier	2.92E-01		1.15E-02	3.36E-03
OriginMultiplier	7.50E-01		1.18E-02	8.86E-03
Population size				
ChangeGamma	3.16E-01		3.49E-01	1.10E-01
ChangeAllGamma	2.80E-01		3.48E-01	9.75E-02

Table O. MCMC acceptance rates for Network B (incorrect population size priors $\Gamma(1, 1000)$).

Id	Pr_accept	Pr_proposed	Pr_proposed	Pr_accept
Topology moves				
AddReticulation	1.52E-04		2.33E-02	3.55E-06
DeleteReticulation	6.72E-05		2.32E-02	1.56E-06
FlipReticulation	1.44E-05		2.33E-02	3.33E-07

RelocateBranch	2.59E-02	2.33E-02	6.02E-04
RelocateBranchNarrow	6.01E-04	2.34E-02	1.41E-05
Branch length			
NodeSlider	5.20E-01	2.33E-02	1.21E-02
NodeUniform	2.68E-01	2.32E-02	6.22E-03
NetworkMultiplier	2.58E-01	1.15E-02	2.97E-03
OriginMultiplier	7.42E-01	1.17E-02	8.67E-03
Population size			
ChangeGamma	3.36E-01	3.49E-01	1.17E-01
ChangeAllGamma	3.15E-01	3.49E-01	1.10E-01

Table P. MCMC acceptance rates for Network B (incorrect population size priors $\Gamma(1, 2000)$).

Id	Pr_accept	Pr_proposed	Pr_proposed	Pr_accept
Topology moves				
AddReticulation	1.14E-04		2.32E-02	2.64E-06
DeleteReticulation	3.25E-05		2.32E-02	7.54E-07
FlipReticulation	2.43E-05		2.33E-02	5.68E-07
RelocateBranch	2.71E-02		2.33E-02	6.31E-04
RelocateBranchNarrow	5.49E-04		2.33E-02	1.28E-05
Branch length				
NodeSlider	5.11E-01		2.33E-02	1.19E-02
NodeUniform	2.53E-01		2.32E-02	5.86E-03
NetworkMultiplier	2.65E-01		1.15E-02	3.04E-03
OriginMultiplier	7.46E-01		1.18E-02	8.77E-03
Population size				
ChangeGamma	3.69E-01		3.49E-01	1.29E-01
ChangeAllGamma	3.80E-01		3.49E-01	1.32E-01

References

[1] Zhu J, Wen D, Yu Y, Meudt HM, Nakhleh L. Bayesian inference of phylogenetic networks from bi-allelic genetic markers. PLoS computational biology. 2018;14(1):e1005932.

[2] Bryant D, Bouckaert R, Felsenstein J, Rosenberg NA, RoyChoudhury A. Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis. Molecular biology and evolution. 2012;29(8):1917–1932.

[3] Bryant D, RoyChoudhury A, Bouckaert R, Felsenstein J, Rosenberg N. Exact coalescent likelihoods for unlinked markers in finite-sites mutation models. arXiv preprint arXiv:11093525. 2011;.

[4] Griffiths RC, Tavare S. Computational methods for the coalescent. IMA Volumes in Mathematics and its Applications. 1997;87:165–182.

[5] Huson DH, Rupp R, Scornavacca C. Phylogenetic networks: concepts, algorithms and applications. Cambridge University Press; 2010.

[6] Gambette P, Berry V, Paul C. The structure of level-k phylogenetic networks. In: Annual Symposium on Combinatorial Pattern Matching. Springer; 2009. p. 289–300.

[7] Zhang C, Ogilvie HA, Drummond AJ, Stadler T. Bayesian inference of species networks from multilocus sequence data. Molecular biology and evolution. 2017;35(2):504–517.

[8] Wang W, Mauleon R, Hu Z, Chebotarov D, Tai S, Wu Z, et al. Genomic variation in 3,010 diverse accessions of Asian cultivated rice. Nature. 2018;557(7703):43–49.