



HAL
open science

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

► **To cite this version:**

Charles-Elie Rabier, Vincent Berry, Marnus Stoltz, Joao D. Santos, Wensheng Wang, et al.. On the inference of complex phylogenetic networks by Markov Chain Monte-Carlo. PLoS Computational Biology, In press, pp.1-38. 10.1101/2020.10.07.329425 . hal-03287030v1

HAL Id: hal-03287030

<https://hal.science/hal-03287030v1>

Submitted on 15 Jul 2021 (v1), last revised 10 Sep 2021 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

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

Charles-Elie Rabier^{1,2,3,*}, Vincent Berry², Marnus Stoltz¹, João D. Santos^{4,5}, Wensheng Wang⁶, Jean-Christophe Glaszmann^{4,5}, Fabio Pardi² 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 Montpellierain Alexander Grothendieck (IMAG), Université de Montpellier, CNRS, Montpellier, France

4 CIRAD, UMR AGAP, F-34398 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, celine.scornavacca@umontpellier.fr

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 [1] 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 [2], implemented in PhyloNet [3], 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 [4–8], and even for several individuals for some species [9, 10] are nowadays available thanks to next generation sequencing. This flow of data finds applications in various fields such as pathogenicity [11], crop improvement [12], evolutionary genetics [13] or population migration and history [14–16]. 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 [17] that aim to estimate the evolutionary history of closely related species. In particular, the well-known coalescent model from population genetics [18] has been extended to the *multispecies coalescent* (MSC) model [19, 20] to handle studies involving populations or individuals from several species. Recent works show how to incorporate sequence evolution processes into the MSC [1, 21]. As a result, it is now possible to reconstruct evolutionary histories while accounting for both incomplete lineage sorting (ILS) and sequence evolution [22, 23].

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 [24]. 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 [25–28], introgression [29–31] and horizontal gene transfer [32, 33] (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 [34–36]. Various models of phylogenetic network have been proposed over time to explicitly represent reticulate evolution, such as hybridization networks [37] or

ancestral recombination graphs [38], along with dozens of inference methods [39, 40].

Model-based methods have been proposed to handle simultaneously ILS and reticulate evolution, which is a desired feature to avoid bias in the inference [41–43]. 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 [44]. 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 [45, 46], all lineages of a given locus tree coalesce within a single species tree *displayed* by the network. The model of Yu et al. [47] 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 [2, 40, 43, 48–56]. 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 [48, 50]. 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 [50], then a Bayesian sampling technique [53]. However, using locus trees as a proxy for molecular sequences loses some information contained in the alignments [19] 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 [57], but running time may become prohibitive already for inferences on few species. Wen et al. in the PHYLONET software [54] and Zhang et al. with the SPECIESNETWORK method [55] 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. [1] 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 [2].

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 [1] to networks. SNAPPNET is available at <https://github.com/rabier/MySnappNet> and distributed as a package of the well-known BEAST 2 software [58, 59]. This package partly relies on code from SNAPP [1] to handle sequence evolution and on code from SPECIESNETWORK [55] to modify the network during the MCMC as well as to compute network priors.

Our approach differs from that of Zhang et al. [55] 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 [55].

The SNAPPNET method we present here is much closer to the MCMC_BiMarkers method of Zhu et al. [2], which also extends the SNAPP method [1] 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 [51, 60] and one into semi-directed networks on four taxa in the SNAQ method of the PHYLONETWORK package [52]. 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 [51, 52, 61]. Here we focus on the *exact* computation of the *full* likelihood, for which identifiability issues are likely to be less serious [43, 62].

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.

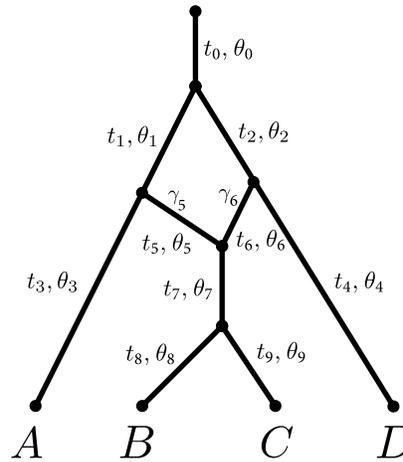


Fig 1. Example of a phylogenetic network. The top node represents the origin and its child node is called the root of the network. Time flows from the top to the leaves (here A, B, C, D) so branches are directed from the top to the leaves. Each branch x is associated to a length t_x , and to a population 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 Figure 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. [55], 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 [47]. 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. [1]. 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 [1]. When u and v are set to 1, the model is also known as the Haldane model [63] or the Cavender-Farris-Neyman model [64].

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:

$$\begin{aligned} \mathbb{P}(\Psi \mid D_1, \dots, D_m) &\propto \mathbb{P}(D_1, \dots, D_m \mid \Psi) \cdot \mathbb{P}(\Psi) \\ &= \mathbb{P}(\Psi) \cdot \prod_{i=1}^m \mathbb{P}(D_i \mid \Psi) \end{aligned} \quad (1)$$

where \propto means “is proportional to”, and where $\mathbb{P}(D_1, \dots, D_m \mid \Psi)$ and $\mathbb{P}(\Psi)$ refer to the likelihood and the network prior, respectively.

Equation (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 $\mathbb{P}(D_i \mid \Psi)$. While we could approximate $\mathbb{P}(D_i \mid \Psi)$ 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. [1] for inferring phylogenetic *trees*, we show below that $\mathbb{P}(D_i \mid \Psi)$ can be computed for *networks* using dynamic programming.

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

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. [55] 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 [55] 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 [1] 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 i th biallelic marker. To keep the notations light, the dependence on i is not explicit.

Given a branch x , we denote by \bar{x} and \underline{x} the top and bottom of that branch. We call \bar{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). $N_{\bar{x}}$ 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, $R_{\bar{x}}$ 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 $\mathbf{n}_{\mathbf{x}}$ (or $\mathbf{r}_{\mathbf{x}}$) to denote a vector of non-negative integers in a 1-to-1 correspondence with the elements of \mathbf{x} . Then $N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}$ is a shorthand for the equations expressing that the numbers of lineages in $\mathbf{n}_{\mathbf{x}}$ are observed at their respective interfaces in \mathbf{x} . For example, if $\mathbf{x} = (\underline{x}, \bar{y})$ and $\mathbf{n}_{\mathbf{x}} = (m, n)$, then $N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}$ is a shorthand for $N_{\underline{x}} = m, N_{\bar{y}} = n$. We use $R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}$ analogously to express the observation of the numbers of red lineages in $\mathbf{r}_{\mathbf{x}}$ 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 $\mathbf{r}_{\mathbf{L}(\mathbf{x})}$ 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:

$$\mathbf{F}_{\mathbf{x}}(\mathbf{n}_{\mathbf{x}}; \mathbf{r}_{\mathbf{x}}) = \mathbb{P}(R_{\mathbf{L}(\mathbf{x})} = \mathbf{r}_{\mathbf{L}(\mathbf{x})} \mid N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}, R_{\mathbf{x}} = \mathbf{r}_{\mathbf{x}}) \cdot \mathbb{P}(N_{\mathbf{x}} = \mathbf{n}_{\mathbf{x}}) \quad (2)$$

(see Fig. 2). These quantities are generalizations of similar quantities defined by Bryant et al. [1]. We will call them partial likelihoods, although, as noted by these authors, strictly speaking this is an abuse of language.

240
241
242

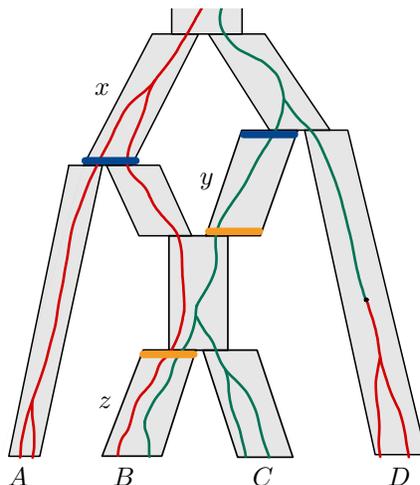


Fig 2. Illustration of the concepts and notation employed to describe likelihood computations. The species network topology is the same as that in Fig.1, but branches (populations) are now represented as grey parallelograms. A gene tree is drawn inside the species network (green and red lines). One mutation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

Computing partial likelihoods: the rules

243

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 the Supplementary Material.

244
245
246

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 $\mathbf{a} = (a_1, a_2, \dots, a_k)$ and $\mathbf{b} = (b_1, b_2, \dots, b_h)$, then $\mathbf{a}, \mathbf{b} = (a_1, \dots, a_k, b_1, \dots, b_h)$ and $\mathbf{a}, c = (a_1, a_2, \dots, a_k, c)$. 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 .

247
248
249
250
251
252
253
254

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

255

$$\mathbf{F}_{(x)}((n); (r)) = \mathbb{1}\{n = n_x\} \cdot \mathbb{1}\{r = r_x\}$$

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

256

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

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

257

258

259

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

$$\mathbf{F}_{\mathbf{x}, \mathbf{y}, \underline{z}}(\mathbf{n}_{\mathbf{x}}, \mathbf{n}_{\mathbf{y}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, \mathbf{r}_{\mathbf{y}}, r_{\underline{z}}) = \sum_{n_{\bar{x}}} \sum_{r_{\bar{x}}} \mathbf{F}_{\mathbf{x}, \bar{x}}(\mathbf{n}_{\mathbf{x}}, n_{\bar{x}}; \mathbf{r}_{\mathbf{x}}, r_{\bar{x}}) \mathbf{F}_{\mathbf{y}, \bar{y}}(\mathbf{n}_{\mathbf{y}}, n_{\underline{z}} - n_{\bar{x}}; \mathbf{r}_{\mathbf{y}}, r_{\underline{z}} - r_{\bar{x}}) \binom{n_{\bar{x}}}{r_{\bar{x}}} \binom{n_{\underline{z}} - n_{\bar{x}}}{r_{\underline{z}} - r_{\bar{x}}} \binom{n_{\underline{z}}}{r_{\underline{z}}}$$

The ranges of $n_{\bar{x}}$ and $r_{\bar{x}}$ in the summation terms are defined by

260

$$\max(0, n_{\underline{z}} - m_{\mathbf{y}}) \leq n_{\bar{x}} \leq \min(m_x, n_{\underline{z}}) \text{ and}$$

261

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

262

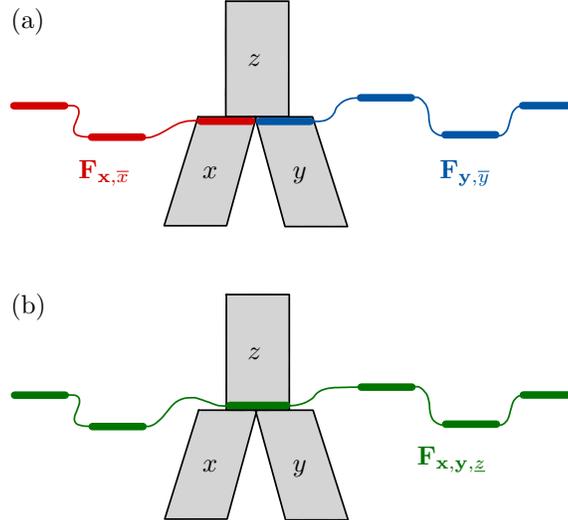


Fig 3. Illustration of Rule 2. Given (a) the partial likelihoods for the \mathbf{x}, \bar{x} (red) vector of population interfaces and the partial likelihoods for the \mathbf{y}, \bar{y} (blue) vector of population interfaces, Rule 2 allows us to compute the partial likelihoods for the (green) vector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

Rule 3: Let \mathbf{x}, \bar{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,

263

264

265

$$\mathbf{F}_{\mathbf{x}, \mathbf{y}, \underline{z}}(\mathbf{n}_{\mathbf{x}}, n_{\mathbf{y}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, r_{\mathbf{y}}, r_{\underline{z}}) = \mathbf{F}_{\mathbf{x}, \bar{x}}(\mathbf{n}_{\mathbf{x}}, n_{\underline{y}} + n_{\underline{z}}; \mathbf{r}_{\mathbf{x}}, r_{\underline{y}} + r_{\underline{z}}) \binom{n_{\underline{y}} + n_{\underline{z}}}{n_{\underline{y}}} \gamma_{\mathbf{y}}^{n_{\underline{y}}} \cdot \gamma_{\underline{z}}^{n_{\underline{z}}}$$

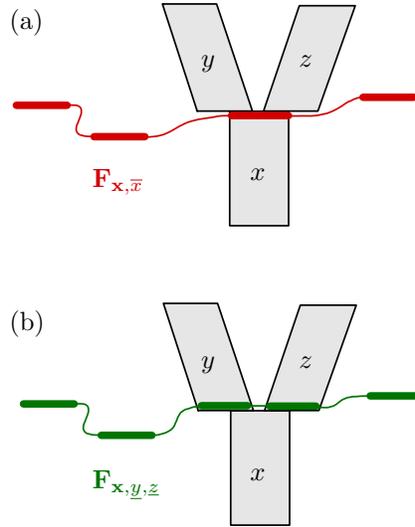


Fig 4. Illustration of Rule 3. Given (a) the partial likelihoods for the \mathbf{x}, \bar{x} (red) vector of population interfaces, Rule 3 allows us to compute the partial likelihoods for the (green) vector $\mathbf{x}, \underline{y}, \underline{z}$ (b).

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

$$\mathbf{F}_{\mathbf{z}, \underline{z}}(\mathbf{n}_{\mathbf{z}}, n_{\underline{z}}; \mathbf{r}_{\mathbf{z}}, r_{\underline{z}}) = \sum_{n_{\bar{x}}} \sum_{r_{\bar{x}}} \mathbf{F}_{\mathbf{z}, \bar{x}, \bar{y}}(\mathbf{n}_{\mathbf{z}}, n_{\bar{x}}, n_{\underline{z}} - n_{\bar{x}}; \mathbf{r}_{\mathbf{z}}, r_{\bar{x}}, r_{\underline{z}} - r_{\bar{x}}) \binom{n_{\bar{x}}}{r_{\bar{x}}} \binom{n_{\underline{z}} - n_{\bar{x}}}{r_{\underline{z}} - r_{\bar{x}}} \binom{n_{\underline{z}}}{r_{\underline{z}}}$$

The ranges of $n_{\bar{x}}$ and $r_{\bar{x}}$ 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 the Supplementary Material). 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 $\mathbf{F}_{\mathbf{x}}(\mathbf{n}_{\mathbf{x}}; \mathbf{r}_{\mathbf{x}})$ 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 ρ , 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:

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

where the conditional probabilities $\mathbb{P}(R_{\rho} = r | N_{\rho} = n)$ are obtained as described by Bryant et al. [1]. Note that the length of branch ρ does not play any role in the computation of the likelihood, so it is not identifiable.

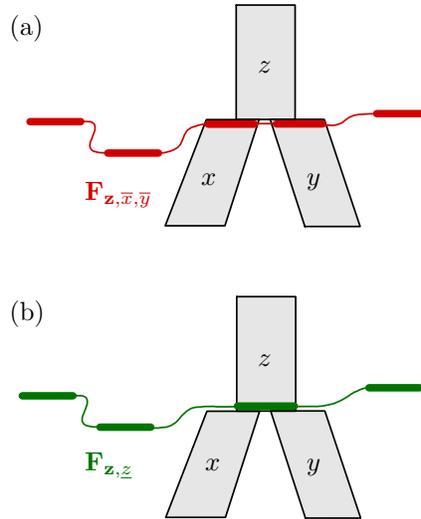


Fig 5. Illustration of Rule 4. Given (a) the partial likelihoods for the $\mathbf{z}, \bar{x}, \bar{y}$ (red) vector of population interfaces, Rule 4 allows us to compute the partial likelihoods for the (green) vector $\mathbf{z}, \underline{z}$ (b).

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 the Supplementary Material 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 [65], 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 (ρ) and the full likelihood of the network is computed from $\mathbf{F}_{(\rho)}$ using Equation (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 $\mathbf{F}_{(\bar{x})}$. 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 \bar{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 \bar{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 [2].

Time complexity of computing the likelihood

Our approach improves the running times by several orders of magnitude with respect to `MCMC_BiMarkers` [2]. 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}, \bar{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 Equation (3), and $\exp(\mathbb{Q}_x t_x)$ – is dominated by these terms, the total running time is $O(sn^{2\bar{K}+2})$, where \bar{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. [2]. 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. [2]). In Section 3 of the Supplementary Material, we show that this entails a total running time of $O(sn^{4\ell+4})$, where ℓ is the *level* of the network [34,66].

Thus, the improvement in running times with respect to the algorithm by Zhu et al. [2] relies on the fact that $2\bar{K} + 2 \ll 4\ell + 4$. One way of seeing this is to remark that, for any traversal of the network, $\bar{K} \leq \ell + 1$. We refer to Section 3 in the Supplementary Material for a proof of this result. Assuming that \bar{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. [2]. However, \bar{K} is potentially much smaller than the level ℓ , as depicted in Figure 6.

We call the minimum value of \bar{K} over all possible traversals of the network the *scanwidth* of the network [67]. 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 [67].

MCMC operators

SNAPPNET incorporates the MCMC operators of SPECIESNETWORK [55] to move through the network space, and also benefits from operators specific to the

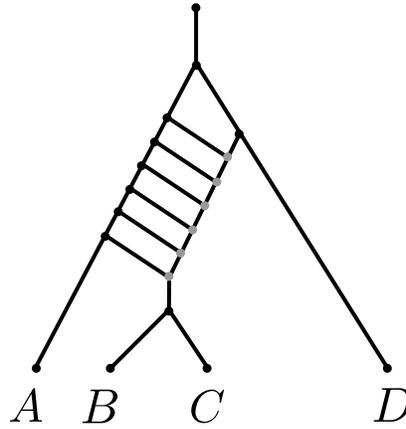


Fig 6. Example of a phylogenetic network where the level ℓ is equal to 6 (the reticulation nodes are depicted in grey), while $\bar{K} \in \{3, 4, 5, 6, 7\}$, depending on the traversal algorithm (not shown). A traversal ensuring that \bar{K} remains close to the lower end of this interval (the scanwidth 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.

mathematical model behind SNAPP [1] (e.g. population sizes, mutation rates, etc.). 358

In order to explore the network space, we used the following topological operators 359
 from SPECIESNETWORK: (a) *addReticulation* and (b) *deleteReticulation* add and delete 360
 reticulation nodes respectively, (c) *flipReticulation* flips the direction of a reticulation 361
 branch and finally (d) *relocateBranch* and (e) *relocateBranchNarrow* relocate either the 362
 source or the destination of random branch. The operators on gene trees from 363
 SPECIESNETWORK have been discarded since in SNAPPNET gene trees are integrated 364
 out. The following SNAPP operators acting on continuous parameters are incorporated 365
 within SNAPPNET: (a) *changeUAndV* changes the values of the instantaneous rates u 366
 and v , (b) *changeGamma* and (c) *changeAllGamma* scale a single population size or all 367
 population sizes, respectively. 368

Last, SNAPPNET takes also advantage of a few SPECIESNETWORK operators for 369
 continuous parameters: (a) *turnOverScale* and (b) *divrRateScale* allow to change 370
 respectively the hyperparameters r and d for the birth-hybridization process, (c) 371
inheritanceProbUniform and (d) *inheritanceProbRndWalk* transform the inheritance 372
 probability γ at a random reticulation node by drawing either a uniformly distributed 373
 number or by applying a uniform sliding window to the logit of γ , (e) *networkMultiplier* 374
 and (f) *originMultiplier* scale respectively the heights of all internal nodes or of the 375
 origin node, (g) *nodeUniform* and (h) *nodeSlider* move the height of a random node 376
 uniformly or using a sliding window. 377

In summary, SNAPPNET relies on 16 MCMC operators, described in SNAPPNET’s 378
 manual (<https://github.com/rabier/MySnappNet>). We refer to the original 379
 publications introducing these operators for more details [1, 55]. 380

Simulation study

Simulated data

We implemented a simulator called SIMSNAPPNET, an extension to networks of the SIMSNAPP software [1]. 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 [2]. 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 Figure 7. Networks A and B are rather simple networks that we wish our tool to be able to infer. They have been taken from [2] 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 [68] in Section 4 in the Supplementary Material.

We also studied networks C(3) and C(4), which are variants of network C (see Figure 8). Network C(k)—containing k reticulation nodes—is obtained by splitting species C into $k - 1$ species, named C_1, C_2, \dots, C_{k-1} , and by adding reticulations between them in the way depicted in Figure 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 the Supplementary Material.

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 [55]: $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1, 1)$, $\tau_0 \sim \mathcal{E}(10)$. This network prior enables to explore a large network space, while imposing more weights on networks with 1 or 2 reticulations (see Figure 1 of the Supplementary Material). 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 [2]. We refer to Figures 2 and 3 of the Supplementary Material for illustrations of the “bounded” prior.

MCMC convergence

To track the behaviour of the Bayesian algorithm, we used the Effective Sample Size (ESS) criterion [69]. 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 [70]. 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 [71], 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 [2], 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 [40, 72]. 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 [73] and [74]. We focused on 24 representative varieties (see Table 3 and Figure 13 in Supplementary Material) 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 4 and 5 in Supplementary Material). For each of the 12 chromosomes, we sampled 1k SNPs having only homozygous alleles. Following recommendations of [1], 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 [2] 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.

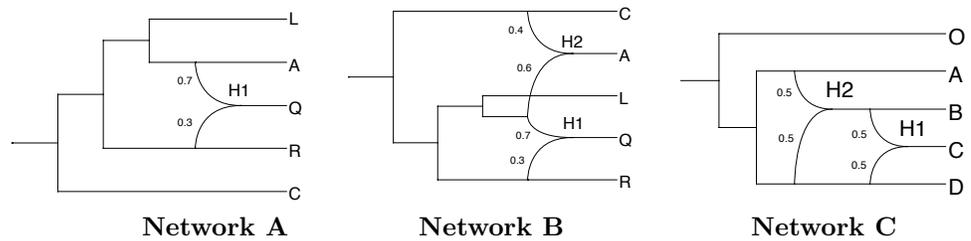


Fig 7. The three phylogenetic networks used for simulating data. Networks A and B are taken from [2]. Branch lengths are measured in units of expected number of mutations per site (i.e. substitutions per site). Displayed values represent inheritance probabilities.

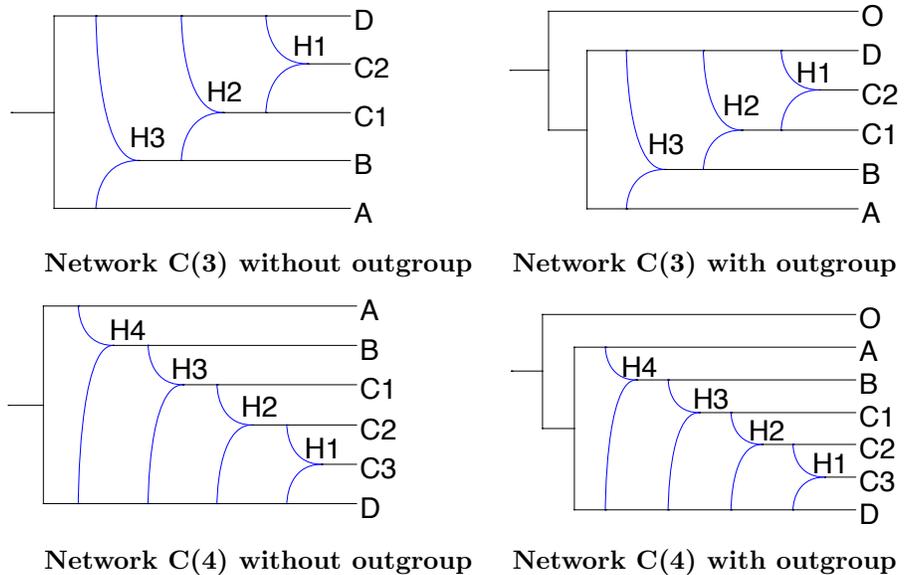


Fig 8. The networks from the C family, with either 3 or 4 reticulation nodes, and with or without outgroup O.

Results

Simulations

First, we compare the performances of `SNAPPNET` and `MCMC_BiMarkers` on data simulated with networks A and B (cf. Figure 7), already studied in [2], 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 [2], 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 [2], we considered u and v , the mutation rates, as known parameters. Indeed, `MCMC_BiMarkers` relies on the operators of [54] 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

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 Figure 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 $\text{ESS} > 200$; $d \sim \mathcal{E}(0.1)$, $r \sim \text{Beta}(1, 1)$, $\tau_0 \sim \mathcal{E}(10)$ for the network prior).

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 [2], 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 Figure 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 1 of the Supplementary Material. Similarly, on network B that hosts 2 reticulations, for $m = 1,000$ MCMC_BiMarkers almost always inferred a 1-reticulation network [2], 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.

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 1 in the Supplementary Material), whereas MCMC_BiMarkers mostly inferred networks [2]. 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 Figure 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 Figure 9). To sum up, SNAPPNET's ability to recover the correct network

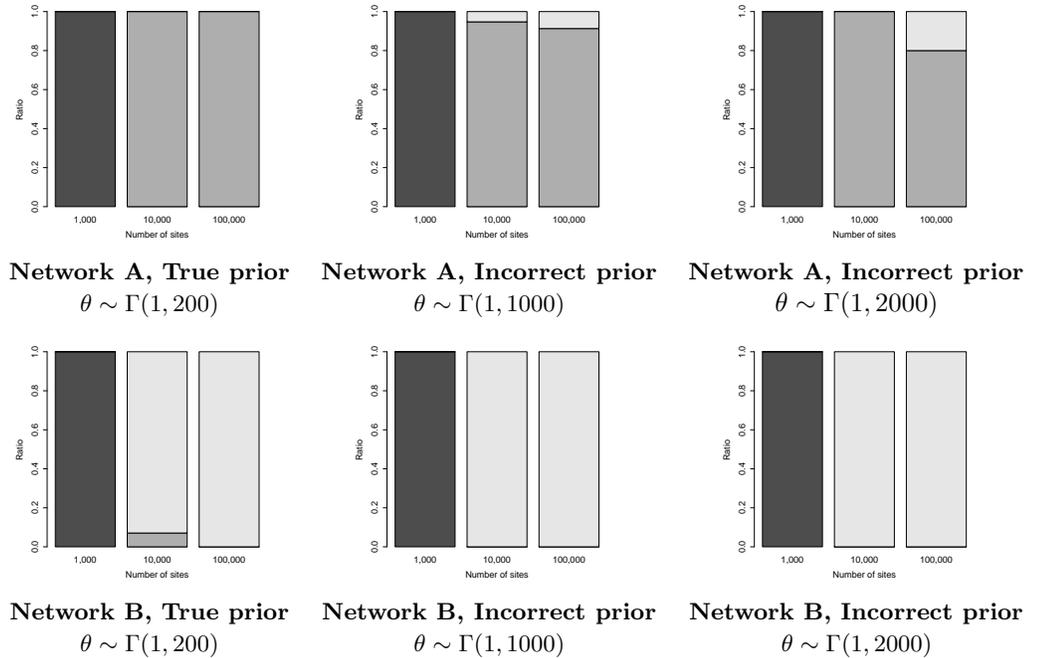


Fig 9. The ratio of trees (black), 1-reticulation networks (dark grey), 2-reticulations networks (light gray), sampled by SNAPPNET, under the different simulations settings studied in Table 1. Recall that networks A and B contain 1 and 2 reticulations, respectively.

topology did not really deteriorate with incorrect priors. 549

2) Ability to estimate continuous parameters for network A 550

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 [55], 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 [2]). Analogous results for networks B and C can be found in the Supplementary Material (Figures 4, 5, 6 and 7). 551-556

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 [2]) 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. 557-566

2a) Constant sites included in the analysis 567

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 567-570

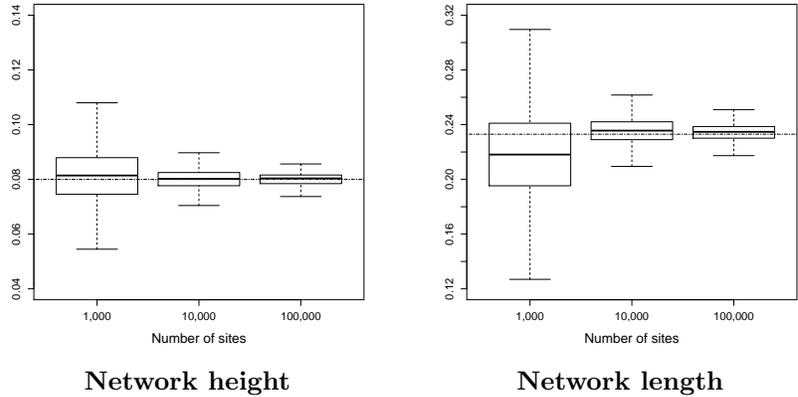


Fig 10. Estimated height and length for network A (see Figure 7), as a function of the number of sites. Heights and lengths are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. Two lineages per species were simulated. Constant sites are 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).

average given the parameters of the simulation, so that such simulated data sets only contained on average 220 variable sites.

572
573

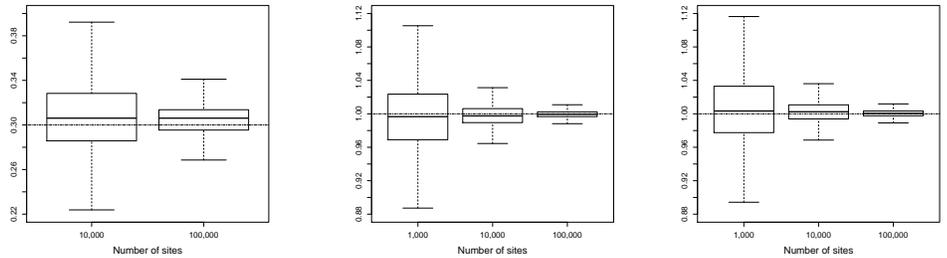


Fig 11. Estimated inheritance probability and instantaneous rates for network A (see Figure 7), as a function of the number of sites. True values are given by the dashed horizontal lines. Same framework as in Figure 10.

In order to limit the computational burden, the analysis for $m = 100,000$ relied only on 17 replicates with $ESS > 200$. Figure 10 reports on the estimated network height and the estimated network length. As expected, the accuracy increased with the number of sites. Figure 11 shows the same behaviour, regarding the inheritance probability γ , the rates u and v . Figure 12 is complementary to Figure 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. Figure 13 is dedicated to population sizes. For 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

574
575
576
577
578
579
580
581
582
583
584

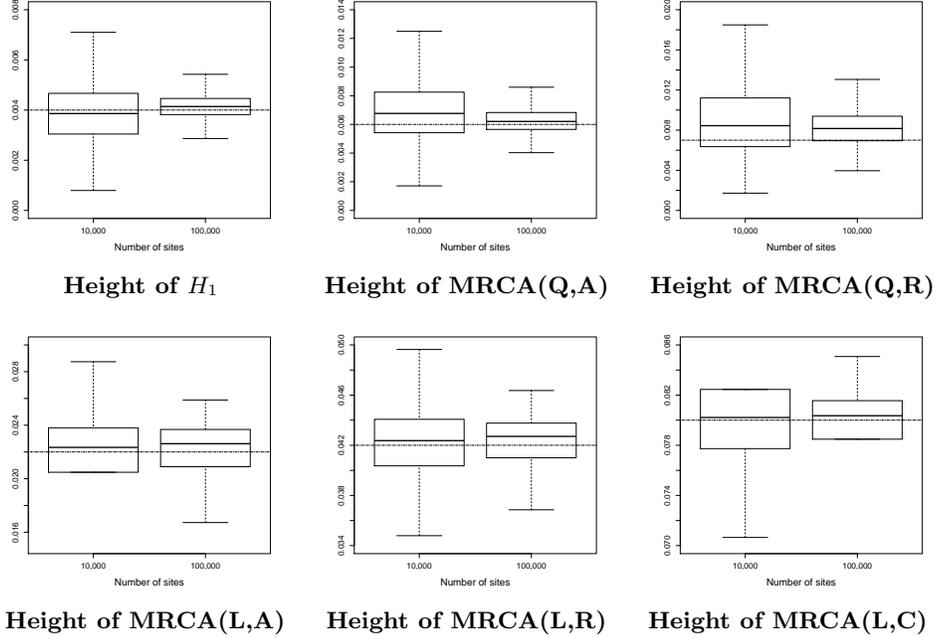


Fig 12. Estimated node heights of network A (see Figure 7), as a function of the number of sites. Heights are measured in units of expected number of mutations per site. True values are given by the dashed horizontal lines. Same framework as in Figure 10. The initials MRCA stand for “Most Recent Common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

MCMC_BiMarkers [2, Figure 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 $\mathbb{P}(X_1, \dots, X_m | \Psi) / \mathbb{P}(\text{“the } m \text{ sites are polymorphic”} | \Psi)$, due to Bayes’ rule.

Before focusing on continuous parameters, let us describe results regarding the topology. As mentioned in [2], 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 $\mathbb{P}(\text{“the } m \text{ sites are polymorphic”} | \Psi)$, 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. Figures 8, 9, 10 and 11 in the Supplementary Material illustrate results obtained from the experiment

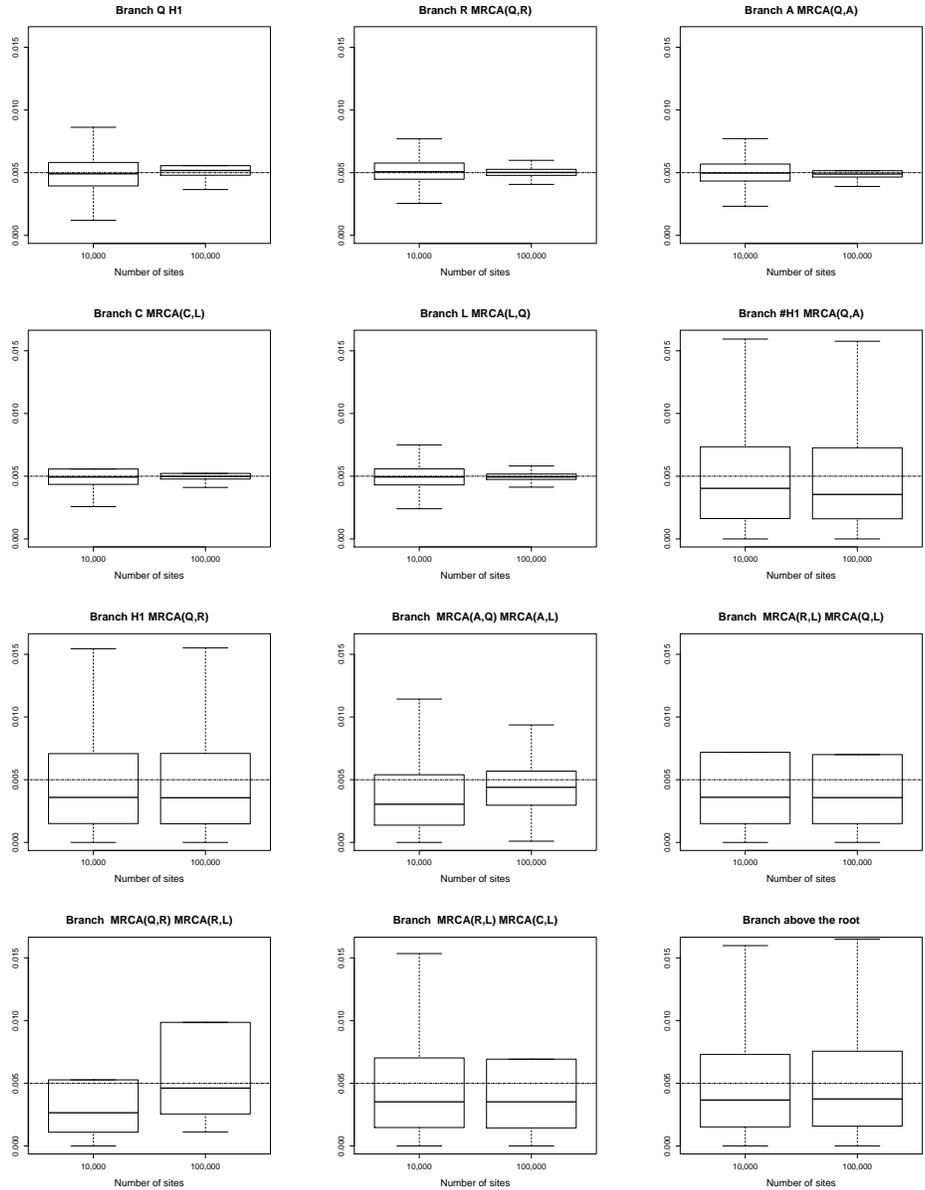


Fig 13. Estimated population sizes θ for each branch of network A (see Figure 7), as a function of the number of sites. True values are given by the dashed horizontal lines. Same framework as in Figure 10. The initials MRCA stand for “Most Recent Common Ancestor”.

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 Figure 12 in Supplementary Material). Surprisingly, the rates u and v were still very accurately

608
609
610
611
612
613
614

estimated.

615

Number of lineages for B and for C		Number of sites		
		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).

Number of lineages for B and for C		Number of sites		
		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}
	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}
	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{ESS} is the average ESS over the different replicates, and SE stands for the sampler efficiency.

Study of network C and its variants

616

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

617

1) Ability to recover the network topology

618

Tables 2 and 3 report the ability of SNAPPNET and MCMC_BiMarkers, respectively, 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.

619

620

621

622

623

624

In order to closely mimic what was done in [2] 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 [2]. Indeed, like SIMSNAPP, SIMSNAPPNET generates only count data (the number of alleles per site and per species). In contrast,

625

626

627

628

`simBiMarker` generates actual sequences, a prerequisite for running `MCMC_BiMarkers`. The commands used under the 4 lineages scenario are given in Section 5 of the Supplementary Material. 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.

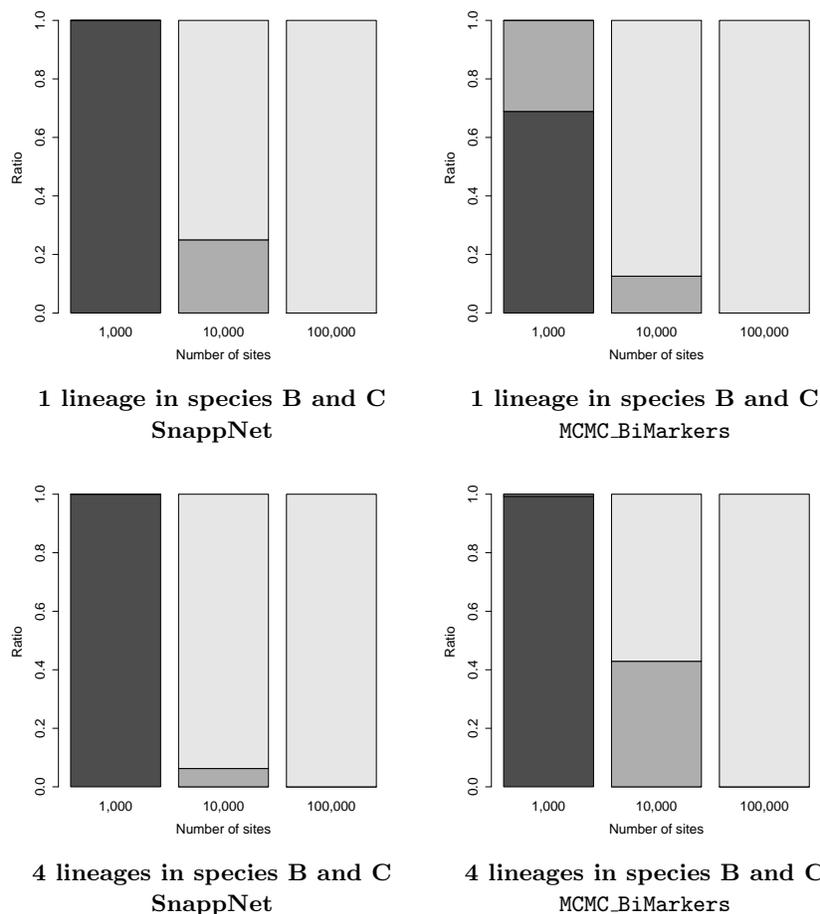


Fig 14. Frequency of trees (black), 1-reticulation networks (dark grey), 2-reticulations networks (light gray) sampled by `SNAPPNET` and `MCMC_BiMarkers`, when data were simulated from Network C (see Tables 2 and 3). Recall that network C contains 2 reticulations.

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. Figure 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 2 in the Supplementary Material). We note here that `SNAPPNET` was ran for a maximum 804,000 iterations for 10,000 sites, and a 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 Figure 7), with and without outgroup (i.e. the species O), and networks C(3) and C(4), again with and without outgroup (see Figure 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.

	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 ± 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.

The results are reported in Table 4. `SNAPPNET` managed to run for all the scenarios 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 than 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 the Supplementary Material 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) [74, 75]. Japonica and Indica each have several subgroups with geographical contrast (see [74] 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 [73], or to multiple parallel dynamics in Japonica, Indica and cAus [15, 29], depending on whether 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 [73] or as a secondary derivative from admixture between Japonica and a local wild rice close to cAus [76], or between Japonica and cAus with the contribution of one or several additional cryptic sources [77].

The most advanced studies of wild rice [73] 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 [74], we compared wild rices to cultivars on the basis of ca. 2.5 million SNPs (cf. Figure 13 in the Supplementary Material) 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, OrII 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, OrII), consistent with the classification by [73]. The 3 data sets we constructed are described in the Materials and methods.

In Figure 15, we report results for data set 1, which includes only one variety per subpopulation (cf. Table 4 in Supplementary Material). 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. 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)

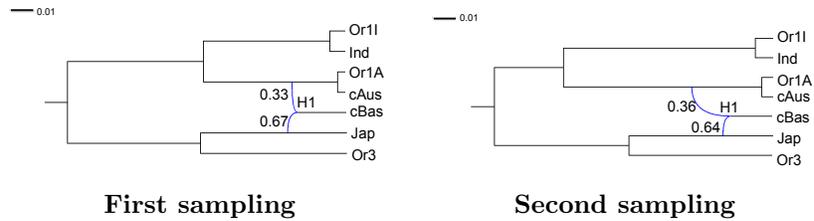


Fig 15. The two networks obtained for data set 1 with only one variety per subpopulation. Each network corresponds to the posterior mean of the distribution sampled by SNAPPNET. Inheritance probabilities are reported above reticulation edges and branch lengths are given in units of expected number of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

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. 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. Figure 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 5 in the Supplementary Material). In this case, large ESS values were observed when SNAPPNET

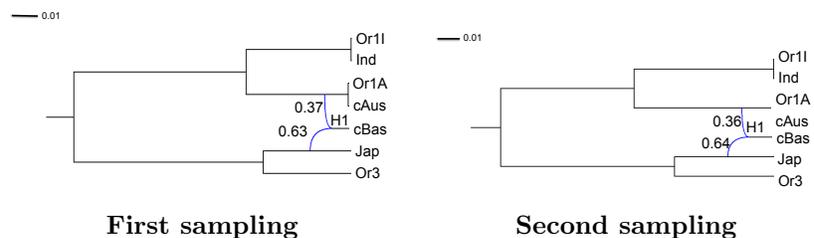


Fig 16. The two networks obtained for data set 2 with two varieties per subpopulation. Each network corresponds to the posterior mean of the distribution sampled by SNAPPNET. Inheritance probabilities are reported above reticulation edges and branch lengths are given in units of expected number of mutations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

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 Figure 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 (Figure 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.

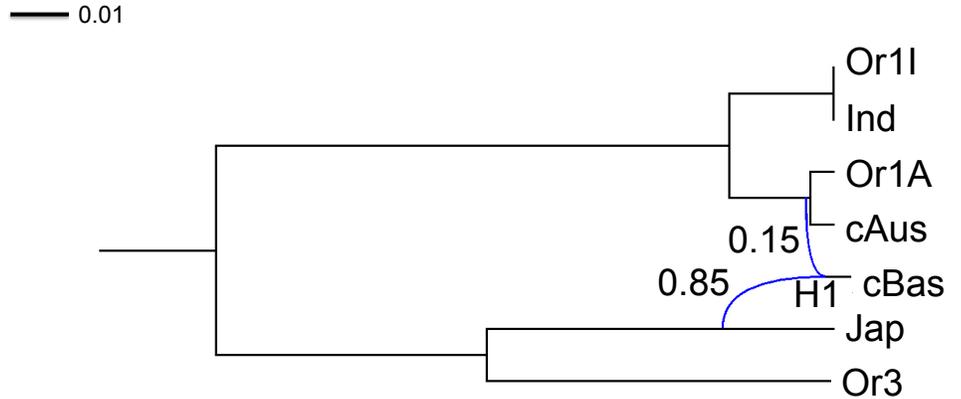


Fig 17. The MAP phylogenetic network obtained for data set 3 with two varieties per subpopulation. Inheritance probabilities are reported above reticulation edges and branch lengths are given in units of expected number of mutations per site (see the scale at the top left).

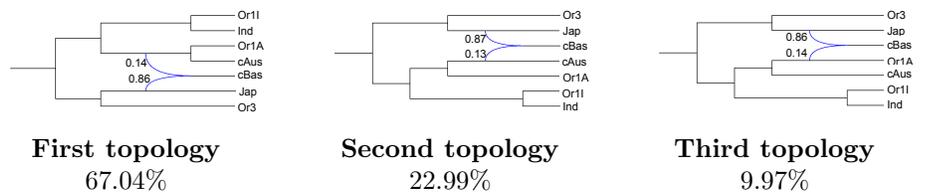


Fig 18. The three topologies sampled by SNAPPNET when data set 3 was considered. Reported inheritance probabilities for each topology are averages on sampled observations.

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 occurred 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 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. [2], 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 [2, 53, 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 [55].

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 [2] 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 [60]. 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 [2], or in other words the root of the blob to which r belongs (a *blob* is a maximal biconnected subgraph [66], see also the Supplementary Material). 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 $O(sn^{2\bar{K}+2})$, where \bar{K} is the maximum number of branches simultaneously considered in a partial likelihood. The interest in depending on \bar{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 \bar{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 [15, 78] and revealing a hybrid origin for the iconic varietal group cBasmati [76, 77]. 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 [79,80]. 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 [81] and BIC [82] 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 [40], 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 [83]).

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 [71]. 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 [84] or the path-lengths difference [85]) do not have unique, easy to compute, natural extensions for networks (see [40] 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 [2, 53, 55].

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. [55], 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 [86], 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” [17]. 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 [87], 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 [88].

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 [43, 62]. 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 [2] 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.

Acknowledgments

This work was supported by the French Agence Nationale de la Recherche, “Investissements d’Avenir” program, through the following projects: the project Genome Harvest (reference ID 1504-006, Labex Agro: ANR-10-LABX-0001-01), the project KIM Data & Life Sciences (I-SITE MUSE: ANR-16-IDEX-0006), the CoCoAlSeq project (ANR-19-CE45-0012) and the project AdaptGrass (reference ID170544IA, I-SITE MUSE: ANR-16-IDEX-0006). It was also funded by the CGIAR Research Program on Rice Agrifood Systems (RICE). This study has been realized with the support of the ATGC platform of the French National Institute of Bioinformatics, with the support of the High Performance Computing Platform MESO@LR, financed by the Occitanie / Pyrénées-Méditerranée Region, Montpellier Mediterranean Metropole and the University of Montpellier. This work also benefited from the Montpellier Bioinformatics Biodiversity platform supported by the LabEx CeMEB, an ANR “Investissements d’avenir” program (ANR-10-LABX-04-01). Last, it was supported by the CIRAD - UMR AGAP HPC Data Center of the South Green Bioinformatics platform (<http://www.south-green.fr/>).

References

1. 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.
2. 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.
3. Wen D, Yu Y, Zhu J, Nakhleh L. Inferring Phylogenetic Networks Using PhyloNet. *Systematic biology*. 2018;67(4):735–740.
 4. 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.
 5. 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.
 6. 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.
 7. 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.
 8. 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.
 9. Consortium IH, et al. The international HapMap project. *Nature*. 2003;426(6968):789.
 10. 3 RGP. The 3,000 rice genomes project. *GigaScience*. 2014;3(1):2047–217X.
 11. 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.
 12. 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.
 13. 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.
 14. 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.
 15. Civáň P, Craig H, Cox CJ, Brown TA. Three geographically separate domestications of Asian rice. *Nature plants*. 2015;1(11):15164.
 16. 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.
 17. Felsenstein J, Felsenstein J. *Inferring phylogenies*. vol. 2. Sinauer associates Sunderland, MA; 2004.
 18. Kingman JF. On the genealogy of large populations. *Journal of applied probability*. 1982;19(A):27–43.

19. 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.
20. Knowles LL, Kubatko LS. Estimating species trees: practical and theoretical aspects. John Wiley and Sons; 2011.
21. RoyChoudhury A, Felsenstein J, Thompson EA. A two-stage pruning algorithm for likelihood computation for a population tree. *Genetics*. 2008;.
22. 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.
23. Degnan JH, Rosenberg NA. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends in ecology & evolution*. 2009;24(6):332–340.
24. Maddison WP. Gene Trees in Species Trees. *Systematic Biology*. 1997 09;46(3):523–536. Available from: <https://doi.org/10.1093/sysbio/46.3.523>.
25. Mallet J. Hybrid speciation. *Nature*. 2007;446(7133):279.
26. Morales L, Dujon B. Evolutionary role of interspecies hybridization and genetic exchanges in yeasts. *Microbiology and Molecular Biology Reviews*. 2012;76(4):721–739.
27. 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.
28. Glemin S, Scornavacca C, Dainat J, Burgarella C, Viader V, Ardisson M, et al. Pervasive hybridizations in the history of wheat relatives. *bioRxiv*. 2018;p. 300848.
29. Civán P, Brown TA. Role of genetic introgression during the evolution of cultivated rice (*Oryza sativa* L.). *BMC evolutionary biology*. 2018;18(1):57.
30. 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.
31. Durantón 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.
32. Koonin EV, Makarova KS, Aravind L. Horizontal gene transfer in prokaryotes: quantification and classification. *Annual Reviews in Microbiology*. 2001;55(1):709–742.
33. Szöllösi 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.
34. Huson DH, Rupp R, Scornavacca C. Phylogenetic networks: concepts, algorithms and applications. Cambridge University Press; 2010.

35. Nakhleh L. Evolutionary phylogenetic networks: models and issues. In: Problem solving handbook in computational biology and bioinformatics. Springer; 2010. p. 125–158.
36. Morrison DA. Introduction to Phylogenetic Networks. RJR Productions; 2011.
37. Baroni M, Semple C, Steel M. A framework for representing reticulate evolution. *Annals of Combinatorics*. 2005;8(4):391–408.
38. Hudson RR. Properties of a neutral allele model with intragenic recombination. *Theoretical population biology*. 1983;23(2):183–201.
39. Huson DH, Scornavacca C. A survey of combinatorial methods for phylogenetic networks. *Genome biology and evolution*. 2011;3:23–35.
40. Degnan JH. Modeling hybridization under the network multispecies coalescent. *Systematic biology*. 2018;67(5):786–799.
41. 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.
42. 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.
43. Zhu S, Degnan JH. Displayed trees do not determine distinguishability under the network multispecies coalescent. *Systematic biology*. 2017;66(2):283–298.
44. 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>.
45. Kubatko LS. Identifying hybridization events in the presence of coalescence via model selection. *Systematic Biology*. 2009;58(5):478–488.
46. 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.
47. 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.
48. 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.
49. Yu Y, Ristic N, Nakhleh L; BioMed Central. Fast algorithms and heuristics for phylogenomics under ILS and hybridization. *BMC bioinformatics*. 2013;14(15):S6.
50. 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.
51. Yu Y, Nakhleh L. A maximum pseudo-likelihood approach for phylogenetic networks. *BMC genomics*. 2015;16(10):S10.

52. Solís-Lemus C, Ané C. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS Genetics*. 2016;12(3):e1005896.
53. Wen D, Yu Y, Nakhleh L. Bayesian inference of reticulate phylogenies under the multispecies network coalescent. *PLoS genetics*. 2016;12(5):e1006006.
54. Wen D, Nakhleh L. Coestimating reticulate phylogenies and gene trees from multilocus sequence data. *Systematic biology*. 2018;67(3):439–457.
55. 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.
56. 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.
57. Bayzid MS, Warnow T. Naive binning improves phylogenomic analyses. *Bioinformatics*. 2013;29(18):2277–2284.
58. 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.
59. 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.
60. Zhu J, Nakhleh L. Inference of species phylogenies from bi-allelic markers using pseudo-likelihood. *Bioinformatics*. 2018;34(13):i376–i385.
61. Pardi F, Scornavacca C. Reconstructible phylogenetic networks: do not distinguish the indistinguishable. *PLoS computational biology*. 2015;11(4):e1004135.
62. Cao Z, Liu X, Ogilvie HA, Yan Z, Nakhleh L. Practical aspects of phylogenetic network analysis using PhyloNet. *bioRxiv*. 2019;p. 746362.
63. Haldane J. The combination of linkage values and the calculation of distances between the loci of linked factors. *J Genet*. 1919;8(29):299–309.
64. Cavender JA. Taxonomy with confidence. *Mathematical biosciences*. 1978;40(3-4):271–280.
65. Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to Algorithms*, Third Edition. 3rd ed. The MIT Press; 2009.
66. 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.
67. Berry V, Scornavacca C, Weller M. Scanning Phylogenetic Networks is NP-hard. In: *SOFSEM*; 2020. In press.
68. Cardona G, Rosselló F, Valiente G. Extended Newick: it is time for a standard representation of phylogenetic networks. *BMC bioinformatics*. 2008;9(1):532.

69. Liu JS. Monte Carlo strategies in scientific computing. Springer Science & Business Media; 2008.
70. 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.
71. 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.
72. Nakhleh L. A metric on the space of reduced phylogenetic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2009;7(2):218–222.
73. 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.
74. 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.
75. Glaszmann JC. Isozymes and classification of Asian rice varieties. *Theoretical and Applied genetics*. 1987;74(1):21–30.
76. Civán P, Ali S, Batista-Navarro R, Drosou K, Ihejiro 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.
77. 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.
78. Civán P, Brown TA. Misconceptions regarding the role of introgression in the origin of *Oryza sativa* subsp. *indica*. *Frontiers in plant science*. 2018;9:1750.
79. 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.
80. Choi JY, Lye ZN, Groen SC, Dai X, Rughani P, Zaaier S, et al. Nanopore sequencing-based genome assembly and evolutionary genomics of circum-basmati rice. *Genome biology*. 2020;21(1):21.
81. Akaike H. Information theory and an extension of the maximum likelihood principle. In: *Selected papers of hirotugu akaike*. Springer; 1998. p. 199–213.
82. Schwarz G, et al. Estimating the dimension of a model. *The annals of statistics*. 1978;6(2):461–464.
83. Gusfield D. *ReCombinatorics: the algorithmics of ancestral recombination graphs and explicit phylogenetic networks*. MIT press; 2014.
84. Robinson DF, Foulds LR. Comparison of phylogenetic trees. *Mathematical biosciences*. 1981;53(1-2):131–147.
85. Steel MA, Penny D. Distributions of tree comparison metrics—some new results. *Systematic biology*. 1993;42(2):126–141.

86. 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.
87. Mueller NF, Bouckaert R. Adaptive parallel tempering for BEAST 2. *bioRxiv*. 2020;p. 603514.
88. 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.

Supporting information

S1. Supplementary material for the manuscript.