Bayesian Phylogenetic Inference using a Combinatorial Sequential Monte Carlo Method

Liangliang Wang   Alexandre Bouchard-Côté   Arnaud Doucet

Abstract

The application of Bayesian methods to large scale phylogenetics problems is increasingly limited by computational issues, motivating the development of methods that can complement existing Markov Chain Monte Carlo (MCMC) schemes. Sequential Monte Carlo (SMC) methods are approximate inference algorithms that have become very popular for time series models. Such methods have been recently developed to address phylogenetic inference problems but currently available techniques are only applicable to a restricted class of phylogenetic tree models compared to MCMC. In this paper, we propose an original Combinatorial SMC (CSMC) method to approximate posterior phylogenetic tree distributions which is applicable to a general class of models and can be easily combined with MCMC to infer evolutionary parameters. Our method only relies on the existence of a flexible partially ordered set structure and is more generally applicable to sampling problems on combinatorial spaces. We demonstrate that the proposed CSMC algorithm provides consistent estimates under weak assumptions, is computationally fast and is additionally easily parallelizable.

KEY WORDS: sequential Monte Carlo; particle Markov chain Monte Carlo; phylogenetics; Bayesian inference; poset.

1 Introduction

Bayesian statistics has formed the basis for many advanced models in phylogenetics, incorporating under a unified framework the numerous aspects of evolution—phylogeography, sequence evolution, alignment, molecular clocks (Lemey et al. 2009; Drummond and Suchard 2010; Huelsenbeck and Ronquist 2001; Ronquist and Huelsenbeck 2003; Ronquist et al. 2012; Suchard and Redelings 2006). However, a main challenge in Bayesian phylogenetics is the requirement to compute a posterior over a phylogenetic tree space. The exact calculation of this posterior involves summing over all possible trees, and for each tree, integrating over all possible combinations of branch lengths.

This challenging posterior computation is typically carried out by running MCMC algorithms for long periods (Yang and Rannala 1997; Larget and Simon 1999; Huelsenbeck and Ronquist 2001; Rannala and Yang 2003). Due to combinatorial constraints, the distribution on tree space is often a complex multimodal distribution (Lakner et al. 2008), and the main difficulty lies in the efficiency with which topology proposals sample the tree space. MCMC imposes relatively strict constraints on the types of proposals that can be used. More precisely, to alleviate the problem of a high rejection rate, only small moves are allowed in proposals, making it challenging to design fast mixing algorithms. With a few exceptions (Lakner et al. 2008; Höhna et al. 2008; Höhna and Drummond 2012), the proposals used by current phylogenetic MCMC samplers have remained largely unchanged in the past decade. Moreover, existing MCMC proposals are computationally expensive, with a computational cost dominated by a subset of the likelihood recursions that need to be recomputed each time local operations are applied.[1]

[1] For simple moves such as Nearest Neighbor Interchange (Lakner et al. 2008) the number of recursions to recomputed is proportional to the number of branches separating the two successive

SMC methods are another class of sampling algorithms which have become popular for state-space models (Doucet et al. 2001; Liu 2001) and are now increasingly used in more general settings (Del Moral, Doucet, and Jasra 2006). However, because of the intricacies of phylogenetic tree spaces, it is non-trivial to directly apply SMC methods to posterior tree inference, and previous work on applying SMC to phylogenetics has been limited in two important ways. First, SMC methods currently available (Teh et al. 2008; Görür and Teh 2009; Görür et al. 2012; Bouchard-Côté et al. 2012) are limited in the types of phylogenetic proposals they can use. Second, these methods do not provide a natural framework to handle non-clock trees and indeed, have never been applied to compute posteriors over such trees in previous work. This is an important limitation, as most current work in phylogenetics relies on non-clock tree models.

Our first contribution is to show how both of these limitations can be addressed using a new approach for building SMC phylogenetic tree inference algorithms. In our CSMC framework, the flexibility on the proposal distributions generalizes both MCMC methods and previous work on phylogenetic SMC. In particular, this flexibility makes it easy to construct non-clock tree proposals that are easy to parallelize, and that do not require expensive likelihood recalculations at each proposal step.

The proposed CSMC algorithm is motivated by a certain over-counting problem in sequentially constructing a non-clock phylogenetic tree. A conventional SMC algorithm applied to such a tree would favour the trees which can be constructed in multiple ways, whereas in our algorithm, a graded partially ordered set (poset) on an extended combinatorial space is used to compute correction terms and provides a consistent estimate of the posterior.

Our second contribution is a method to jointly infer the phylogenetic tree and the as-

operations. For more complicated moves such as Subtree Prune and Regraft, the number of recursions to recompute is proportional to the total number of branches in the tree.

sociated evolutionary parameters based on particle MCMC (Andrieu et al. 2010). We use the CSMC algorithm to design an efficient high dimensional proposal for MCMC updating jointly the tree and the evolutionary parameters. This is of significant interest since modelling uncertainty over evolutionary parameters is one of the key advantages of Bayesian phylogenetic methods over classical approaches.

2 Background and notation

2.1 Phylogenetic trees

Let X be a set of observed taxa, related through a phylogenetic tree t that we wish to estimate. A phylogenetic X-tree t represents the relationship among observed taxa via two objects: a tree topology and a set of branch lengths.

A *tree topology* is a connected acyclic graph, (V, E), where V is the set of vertices, and E is the set of edges. Vertices that have degree one are called *leaves*, representing the observed taxa; the other vertices are called *internal nodes*, denoting the unobserved taxa. In a phylogenetic rooted X-tree, there is a special vertex called 'root' that has degree two; the other internal nodes have three neighbours (the parent and two children). We regard the edges as being directed away from the root, describing the evolution of species originating from the root. In unrooted trees, this graph is undirected and does not contain a root node.

Branch lengths are positive real numbers, $b(e)$, associated with each edge $e \in E$. A branch length quantifies the intensity of the evolutionary changes between two nodes. In basic models, branch lengths are proportional to the expected number of mutations in the evolutionary history between two nodes. For all $v, v' \in V$, we define $b(v, v')$ as the sum of the branch lengths along the unique path between v and v'.

A tree is *ultrametric* if it can be rooted in a way that satisfies the following property: for any vertex $v \in V$, we have $b(v, x) = b(v, x')$ for all descendants x, x' of v, as illustrated in Figure 1 (a). This assumption implies a constant evolutionary rate along the paths from v to all its descendants. However it is well known that evolutionary rates vary substantially, for example because of unequal generation times. As a consequence, developing phylogenetic inference methods that do not make ultrametric assumptions is highly relevant to biologists. These general trees are called *non-clock trees*. We will denote the set of non-clock trees by $\mathcal{X} = \mathcal{X}(X)$. There has been work on generalizing ultrametric models to handle non-constant evolutionary rates (Thorne et al. 1998; Drummond and Suchard 2010), but due to the higher dimensional parameters of these models, non-clock models are still widely used.

2.2 Bayesian phylogenetic inference

Phylogenetic reconstruction is based on observed information $\mathcal{Y}$ located at the leaves X of phylogeny. For $X' \subset X$, we use the notation $\mathcal{Y}(X')$ for the subset of observations corresponding to a subset X' of the leaves. For simplicity, we assume that the observations take the form of a matrix $y_{x,s}$, where $x \in X$ and s denotes an aligned position on the genomes, called a site. See Morrison (2006) for an account of how this type of information is extracted from the raw biological sequences.

Our objective is to use $n = |X|$ observed biological sequences to estimate phylogenetic trees and parameters in the evolutionary model. In a Bayesian framework, we need to specify the prior distribution and likelihood function. Our tree and likelihood are parameterized by a vector of parameters, θ, equipped with a prior with density $p(\theta)$. For a tree $t \in \mathcal{X}$, the prior density given θ is denoted by $p(t|\theta)$. Branch lengths are here considered as being part of t, not part of θ. For example, a common prior over non-clock trees consists in a uniform distribution over topologies and a product of independent exponential distributions with rate λ_{bl} over the branch lengths. The probability of the observed data $\mathcal{Y}$ given parameters θ and tree t is $\mathbb{P}(\mathcal{Y}|\theta, t)$.

Bayesian inference relies on the joint posterior density,

$$p(\theta, t|\mathcal{Y}) = p(\theta|\mathcal{Y})p(t|\mathcal{Y}, \theta) = \frac{\mathbb{P}(\mathcal{Y}|\theta, t)p(t|\theta)p(\theta)}{\mathbb{P}(\mathcal{Y})}, \tag{1}$$

however computing the normalization, $\mathbb{P}(\mathcal{Y}) = \int\int \mathbb{P}(\mathcal{Y}|\theta, t)p(t|\theta)p(\theta)\,\mathrm{d}\theta\,\mathrm{d}t$, is often intractable—for example the total number of distinct labelled topologies of a rooted tree of n leaves is $(2n-3)!!$ (Semple and Steel 2003).

As it is standard in the phylogenetic literature, we assume that the sites (columns in the matrix $y_{x,s}$) are independent, and we use a continuous-time Markov chain to model the evolution of each site. Letting Q denote the rate matrix of the continuous-time Markov chain, and $\xi_{v,s}$, the state of the genome for the species $v \in V$ at site s, we write the evolutionary model along branch $e = (v \to v')$ as $\mathbb{P}(\xi_{v',s} = j|\xi_{v,s} = i) = (\exp(b(e)Q))_{i,j}$. If t is rooted, the full likelihood model, $\mathbb{P}(\mathcal{Y}|\theta, t)$, is described by a directed graphical model. This graphical model has a topology (V, E), a set of conditional probabilities given by $\mathbb{P}(\xi_{v',s} = j|\xi_{v,s} = i)$ as above, and a root distribution given by the stationary distribution of the continuous-time Markov chain with rate Q. Unrooted trees are approached by restricting the continuous-time Markov chain to be reversible, a common assumption in phylogenetics. In this case, all rootings keep the likelihood invariant, so $\mathbb{P}(\mathcal{Y}|\theta, t)$ can be computed by picking an arbitrary rooting.

In a Bayesian model, the rate matrix Q is obtained from a parametric function depending on the unknown parameters θ. For example, in the General Time Reversible (GTR) model (Tavaré 1986), θ_{GTR} includes the stationary state frequencies of the four nucleotides, $(\pi_A, \pi_C, \pi_G, \pi_T)$, as well as symmetric rates of change from nucleotide i to j, denoted γ_{ij}, for $i, j \in \{A, C, G, T\}$ such that $i < j$.

Two commonly used models to describe rate variation among sites in a sequence are based on a Gamma distributed rate variation across sites and a proportion of invariant sites. The shape parameter of this Gamma distribution is denoted by α and the proportion of invariant sites by p_0. In the most parameter-rich model considered in the present paper, the parameters are $\theta_{\mathrm{GTR}+\Gamma+\mathrm{I}} = (\theta_{\mathrm{GTR}}, \alpha, p_0)$. Many other parametric models exist, ranging from simple to very complex, but they can all be handled similarly within the Bayesian framework (Shapiro et al. 2006; Yang 2006).

3 Methodology

In this section, we introduce CSMC, an original algorithm to sample approximately from a target probability measure $\bar{\pi}$ on a combinatorial space $\mathcal{X}$ when one is only able to evaluate pointwise an unnormalized version π of $\bar{\pi}$. We denote by $\|\pi\|$ the integral of π over $\mathcal{X}$ so that $\bar{\pi} = \pi/\|\pi\|$. We then show how CSMC can be used to sample from the posterior of phylogenetic trees $\bar{\pi}(t) = p(t|\mathcal{Y}, \theta)$ and how it can be combined with MCMC so as to sample from $p(\theta, t|\mathcal{Y})$. In this paper, with a slight abuse of notation, we will not distinguish between a measure and its density.

3.1 Finite setup

To simplify the presentation, we first introduce the algorithm under the assumption that $\mathcal{X}$ is a finite but large combinatorial set (for example, a set of tree topologies). We will show in Section 3.5 how this can be relaxed to accommodate branch lengths in phylogenetics. At a high level, the main assumption on which our algorithm relies on is that any object t in the target sample space $\mathcal{X}$ can be constructed incrementally using a sequence of R intermediate objects.

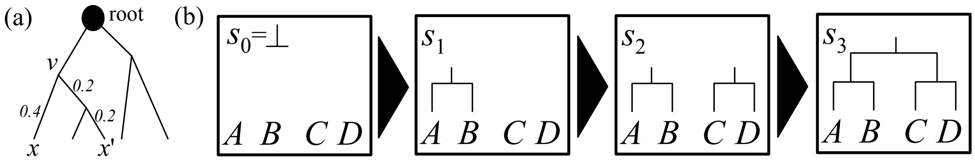


FIGURE 1: (a) An example of an ultrametric tree. (b) An example of the partial states of a discrete rooted X-tree.

In phylogenetics for example, the intermediate objects we will use are based on *forests*. As we do not consider here branch lengths, the term 'tree' is a short-hand for 'tree topology'. Consider for example a discrete rooted X-tree as in Figure 1 (b). Such a tree can be constructed by starting from the disconnected graph with vertices X, and by adding, at each step, one internal node $v \notin X$, and a pair of edges connecting v to a pair of roots from the previous generation. This is done such that the number of connected components decreases by one at every step (decreasing the number of connected components is equivalent to ensuring that no cycles are created). More precisely, we will build each rooted X-tree t by proposing a sequence of X-forests $s_0, s_1, \ldots, s_R$, where an X-forest $s_r = \{(t_i, X_i)\}$ is a collection of rooted X_i-trees t_i such that the disjoint union of leaves of the trees in the forest is equal to the original set of leaves, $\bigcup_i X_i = X$. The forest s_r is actually a tree if and only if $r = R$, where $R = |X| - 1$.

More generally, we will call the intermediate object s_r a *partial state of rank* r. Our terminology (partial states, rank) has an order theoretic motivation that will be described shortly. The set of all partial states of rank r will be denoted by $\mathcal{S}_r$ (continuing with the preceding example, the partial states of rank r are those composed of $|X| - r$ trees), and the set of partial states across all ranks will be denoted by $\mathcal{S} = \bigcup_r \mathcal{S}_r$. The sets of partial states considered in this section are assumed to satisfy the following three conditions: (1) The sets of partial states of different ranks are disjoint, i.e. $\mathcal{S}_r \cap \mathcal{S}_s = \varnothing$ for all $r \neq s$ (in phylogenetics, this holds since a forest with

r trees cannot be a forest with s trees when $r \neq s$). (2) The set of partial states of smallest rank has a single element denoted by $\perp$, i.e. $\mathcal{S}_0 = \{\perp\}$ (in phylogenetics, $\perp$ is the disconnected graph on X). (3) The set of partial states of rank R coincides with the target space, $\mathcal{S}_R = \mathcal{X}$ (in phylogenetics, at rank $R = |X| - 1$, forests have a single tree and are members of the target space $\mathcal{X}$). These conditions will be subsumed by the more general framework of Section 3.5, but the more concrete conditions above help understanding the poset framework.

In order to construct the CSMC algorithm, the user first needs to specify an *extension* of the measure π, defined only on the target space $\mathcal{X}$, into a measure over the larger space $\mathcal{S}$. The restriction of this extended measure to $\mathcal{X}$ should coincide with the target measure π. We abuse notation and use π for both the original target measure on $\mathcal{X}$ and its extension on $\mathcal{S}$.

In a Bayesian phylogenetic context, we have $\pi(t) = \pi_{\mathcal{Y}}(t) = \mathbb{P}(\mathcal{Y}|\theta, t)p(t|\theta)$, where we are assuming fixed parameters θ, an assumption we will relax in Section 3.7. A natural choice in non-clock models to obtain an extension of π into forests is to take a product over the trees in the forest s as follows:

$$\pi(s) = \prod_{(t_i, X_i) \in s} \pi_{\mathcal{Y}(X_i)}(t_i). \tag{2}$$

We call this choice of extension the *natural forest extension*, other choices are possible (see Section 5 for more examples).

3.2 CSMC methodology

In this section, we introduce the CSMC algorithm, a procedure that approximates the target probability measure $\bar{\pi}$ and its normalizing constant $\|\pi\|$ in R steps. At each step r, a list of K partial states is kept in memory. Each element of this list is called a *particle*, denoted $s_{r,1}, s_{r,2}, \ldots, s_{r,K} \in \mathcal{S}_r$. To each particle $s_{r,k}$ is associated a positive weight $w_{r,k}$. Refer to Algorithm 1 for an overview of the steps described in more detail in the following.

The general form of the algorithm has similarities with standard SMC algorithms, with the important exception of the weight updates, which needs to be altered to accommodate general combinatorial structures.

Given a list of weighted particles at rank $r \geq 1$, we construct a discrete positive measure:

$$\pi_{r,K}(s) = \|\pi_{r-1,K}\| \frac{1}{K} \sum_{k=1}^{K} w_{r,k} \delta_{s_{r,k}}(s), \quad \text{for all } s \in \mathcal{S}, \tag{3}$$

where δ_s is the Kronecker delta function. The algorithm constructs these measures recursively as follows.

Algorithm 1 : Combinatorial Sequential Monte Carlo (CSMC)

```
$\pi_{0,K} \leftarrow \delta_\perp$
for rank $r = 1, 2, \ldots, R$ do
    for all $k \in \{1, \ldots, K\}$ do
        sample $\tilde{s}_{r-1,k} \sim \bar{\pi}_{r-1,K}$
        sample $s_{r,k} \sim \nu^{+}_{\tilde{s}_{r-1,k}}$
        compute $w_{r,k} = w(\tilde{s}_{r-1,k}, s_{r,k})$ using Equation (4)
    end for
    construct $\pi_{r,K}$ using Equation (3)
end for
return $\pi_{R,K}$
```

The algorithm is initialized at rank $r = 0$ by initializing the list with K copies of the least partial state $\perp$. Given the empirical measure $\pi_{r-1,K}$ from the previous population of particles, a new list of particles and weights is created as follows at rank r.

First, we resample K times from the probability measure $\bar{\pi}_{r-1,K}$ and denote the sampled particles by $\tilde{s}_{r-1,1}, \tilde{s}_{r-1,2}, \ldots, \tilde{s}_{r-1,K}$. One can also optionally add an MCMC step at each rank $r = 1, 2, \cdots, R$ (after the resampling stage) as in the Resample-Move algorithm (Gilks and Berzuini 2001). Second, we grow each of the resampled particle, $\tilde{s}_{r-1,k}$, into a new particle of rank r, denoted by $s_{r,k}$ using a proposal distribution

$\nu_s^+ : \mathcal{S} \to [0,1]$. Given an initial partial state s and proposed partial state s', we denote the probability of proposing s' from s by $\nu_s^+(s')$. We assume that the successors proposed from a partial state of rank r will always have rank $r+1$; i.e. if $s \in \mathcal{S}_r$ and $\nu_s^+(s') > 0$, then $s' \in \mathcal{S}_{r+1}$. For example, when building discrete rooted X-trees, the proposal needs to select a pair of trees to merge. One simple choice is to pick a pair uniformly at random among the $\binom{|X|-r}{2}$ pairs; other choices are discussed in Section 5. Finally, we compute a weight for each of these new particles using the following formula:

$$w_{r,k} = w(\tilde{s}_{r-1,k}, s_{r,k}) = \frac{\pi(s_{r,k})}{\pi(\tilde{s}_{r-1,k})} \cdot \frac{\nu^-_{s_{r,k}}(\tilde{s}_{r-1,k})}{\nu^+_{\tilde{s}_{r-1,k}}(s_{r,k})}, \tag{4}$$

where ν_s^- is a probability distribution over $\mathcal{S}$ correcting an *overcounting* problem, detailed in Section 3.3. While the weight expression superficially looks like a Metropolis-Hastings ratio, the fundamental difference is that $\nu^+ \neq \nu^-$ in general. The overcounting correction is more closely related to the backward kernels of Del Moral et al. (2006), but due to the combinatorial nature of the space, the poset framework plays an instrumental role in constructing ν^- in the types of spaces we are interested in.

The algorithm returns at the final rank R a Monte Carlo approximation $\pi_{R,K}$ of π. In a phylogenetic context, its normalized version $\bar{\pi}_{R,K}$ approximates $\bar{\pi}(t) = p(t|\mathcal{Y}, \theta)$ while $\|\pi_{R,K}\|$ approximates the marginal likelihood $\mathbb{P}(\mathcal{Y}|\theta)$.

In the next section, we show how ν^- can be selected to guarantee convergence of $\bar{\pi}_{R,K}$ and $\|\pi_{R,K}\|$ to $\bar{\pi}$ and $\|\pi\|$ as $K \to \infty$. The precise meaning of convergence will be discussed in Sections 3.4 and 3.5.

3.3 Overcounting correction

The CSMC algorithm previously introduced is similar to standard SMC, with the exception of the extra overcounting correction ν^- in the weight update. Before giving further details on how to select ν^-, we first describe in the specific case of phylogenetic

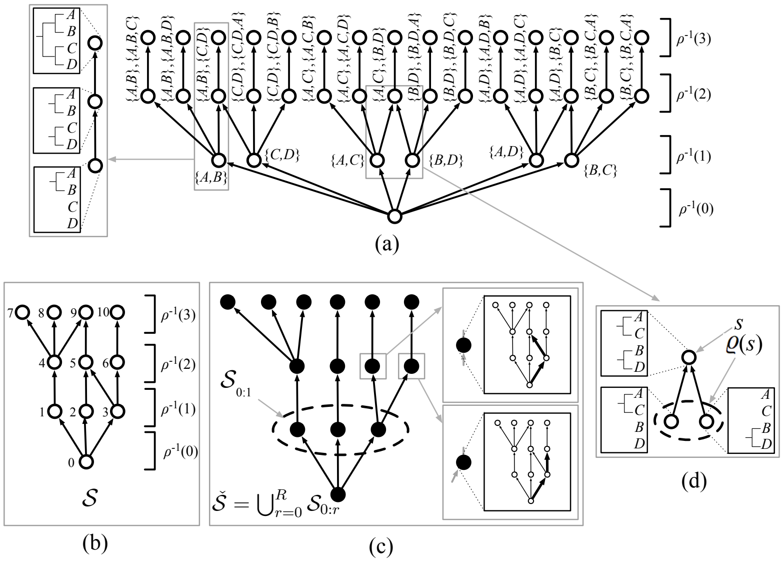


FIGURE 2: (a) All the different sequences of partial states (forests) leading to fully specified states (rooted X-trees). (b) An example of a simple cyclic poset. (c) An example of changing the simple cyclic poset in (b) to an acyclic case. (d) An example of the set of parents $\varrho(s)$ of a partial state s.

non-clock inference the problem that would arise if we were to omit this correction factor. To simplify the discussion, we start by considering a model where there are no observations.

In Figure 2 (a), we show all the different sequences of partial states (forests) leading to one of the $1 \cdot 3 \cdot 5 = 15$ fully specified states (rooted X-trees). An arrow between partial states s and s' means that s' can be obtained from s by one application of the proposal, i.e. that $\nu_s^+(s') > 0$.

A balanced binary tree on four taxa, for example one with rooted clades $\{A, B\}, \{C, D\}$, can be constructed in two distinct ways: either by first merging A and B, then C and D, or by first merging C and D, then A and B. On the other hand, an unbalanced tree on the same set of taxa can be constructed in only one way, for example the tree with clades $\{A, B\}, \{A, B, C\}$ can only be constructed by first proposing to merge A and B, then C and $\{A, B\}$. A consequence of this dichotomy is that under the uniform proposal, the expected fraction of particles with a balanced topology of each type is $2/18$, while it is 1/18 for unbalanced topologies (since there are 18 proposal paths, 2 for each balanced topology, 1 for each unbalanced one). Since we would like the probability of each topology to be close to 1/15, the naive estimate is therefore inconsistent.

In order to resolve this issue (by defining an appropriate overcounting function), it will be useful to formalize the graph shown in Figure 2 (b). This can be done by defining a *partial order* $\leq$ on $\mathcal{S}$. Recall that $(\mathcal{S}, \leq)$ is called a *partially ordered set*, or briefly a *poset*, if $\leq$ is a binary relation on $\mathcal{S}$ that is *reflexive, anti-symmetric*, and *transitive* (Stanley 1986). Also, for $s, s' \in \mathcal{S}$, we say that *s is covered by s'*, written $s \prec s'$, if $s \leq s'$ and there is no $z \in \mathcal{S}$ between s and s'. The covering relation determines the partial order in a finite ordered set, implying that in our combinatorial setup we can induce a poset on the extended space $\mathcal{S}$ by deeming that s' covers s if and only if $\nu_s^+(s') > 0$.

Recall that $\mathcal{S} = \bigcup_r \mathcal{S}_r$. Our poset $(\mathcal{S}, \leq)$ is equipped with an extra structure called a *rank* ρ: a function from $\mathcal{S}$ to $\{0, 1, \cdots, R\}$ such that $\rho(s_0) = 0$ if s_0 is a minimal element of the poset, and $\rho(s') = \rho(s) + 1$ if $s \prec s'$ in $\mathcal{S}$. With these definitions, graphs such as Figure 2 (b) can then be regarded as the *Hasse diagram* corresponding to this induced poset, namely a graph where the set of vertices is $\mathcal{S}$, and there is an edge between s and s' whenever $s \prec s'$.

In previous work (Bouchard-Côté et al. 2012), the overcounting problem has been avoided by forbidding proposals ν^+ that induce a cyclic Hasse diagram. In the CSMC algorithm, ν^- is used to avoid this artificial restriction. We present the solution when $\mathcal{S}$ is a finite space in this section. The finite assumption is lifted in Section 3.5.

In order to have consistency as in Proposition 2, the only requirement on ν^- is:

Assumption 1. *For all $s, s' \in \mathcal{S}$, $\nu_s^+(s') = 0$ implies $\nu_{s'}^-(s) = 0$.*

We now give an example where Assumption 1 is satisfied. Let $\varrho(s)$ denote the set of possible parents of a partial state s and $|\varrho(s)|$ its cardinality. When $|\varrho(s')|$ is finite then selecting $\nu^-_{s'}(s) = |\varrho(s')|^{-1} \times \mathbf{1}[\nu^+_s(s') > 0]$ ensures Assumption 1 holds. As $\sum_s \mathbf{1}[\nu^+_s(s') > 0] = |\{s : \nu^+_s(s') > 0\}| = |\varrho(s')|$, this choice of overcounting correction $\nu^-_{s'}(s)$ is indeed a probability measure for any s'. In phylogenetics, $|\varrho(s)|$ is equal to the number of nontrivial trees in the forest s, where a tree is said to be trivial if it has a single leaf in it (see Figure 2 (d)), i.e. $|\varrho(s')| = \sum_{(t_i,X_i)\in s'} \mathbf{1}[|X_i| > 1]$.

While Assumption 1 is weak, one can select ν^- so as to minimize the variance of the weights appearing in the CSMC algorithm by generalizing Proposition 3.1 in Del Moral et al. (2006).

3.4 Analysis in the finite case

To motivate further the overcounting correction, we sketch in this section an elementary proof that CSMC estimates converge in L^2 norm. We write $\mu\phi$ as a short-hand for the integral of a function ϕ with respect to a measure μ.

Formally, we will prove the following result, where for simplicity, in this paper → denotes convergence as the number of particles $K \to \infty$ in L^2 unless stated otherwise.

Proposition 2. *Under Assumption 1, we have $\pi_{R,K}\phi \to \pi\phi$ for all $\phi : \mathcal{X} \to \mathbb{R}$.*

This implies that $\bar{\pi}_{R,K}\phi \to \bar{\pi}\phi$ and $\|\pi_{R,K}\| \to \|\pi\|$ where $\|\pi_{R,K}\| = \prod_{r=1}^{R} \left(\frac{1}{K}\sum_{k=1}^{K} w_{r,k}\right)$. In our Bayesian context, this means that our estimates of the posterior expectations and of the marginal likelihood are convergent.

Proposition 2 can be established in two steps. First, previous SMC results apply directly when the induced Hasse diagram is acyclic. Second, we show that in the cyclic case, we can construct a certain distribution $\check{\pi}$ and proposal $\check{\nu}^+$ on a larger space $\check{\mathcal{S}}$ with the following properties: (1) The target distribution π can be obtained from $\check{\pi}$ by marginalization; (2) The induced Hasse diagram is acyclic, so the algorithm on $\check{\mathcal{S}}$ is consistent by the first step of the proof; (3) The proposal steps and weight updates of a standard SMC algorithm on $\check{\mathcal{S}}$ are equivalent to the CSMC algorithm. Hence Proposition 2 follows from standard SMC results.

As described above, let us start by assuming the poset is acyclic. In this case, we claim that we can invoke Proposition 4 of Bouchard-Côté et al. (2012). First, the boundedness assumptions required in Proposition 4 are automatically satisfied since here $|\mathcal{S}| < \infty$. Second, the connectedness assumption made in this previous work, Assumption 2b (which states that the Hasse diagram, viewed as an acyclic graph, needs to be connected), can be shown to hold using the following argument: assume on the contrary that there is a connected component $\mathcal{C} \subset \mathcal{S}$ in the Hasse diagram that does not contain $\perp$, and let s be a minimal element, where $s \in \mathcal{C}$ by finiteness. Since $\{s' : \nu_s^-(s') > 0\} \subset \rho^{-1}(\rho(s) - 1)$, we have a contradiction. Therefore there can be only one connected component.

We present the reduction to the acyclic case. Let $\mathcal{S}_{0:r} = \mathcal{S}_0 \times \mathcal{S}_1 \times \cdots \times \mathcal{S}_r$, the set of paths of length $r + 1$ in $\mathcal{S}$. We will view the algorithm as incrementally building partial states over a larger space, with $\check{s}_0 \in \mathcal{S}_0, \check{s}_1 \in \mathcal{S}_{0:1}, \check{s}_2 \in \mathcal{S}_{0:2}, \ldots, \check{s}_R \in \mathcal{S}_{0:R}$. In other words, instead of viewing the algorithm as operating over $\mathcal{S} = \bigcup_{r=0}^{R} \mathcal{S}_r$, we will view it as operating over $\check{\mathcal{S}} = \bigcup_{r=0}^{R} \mathcal{S}_{0:r}$.

Let us start by introducing a new measure $\check{\pi}$ on $\check{\mathcal{S}}$. Let $\check{s}$ be an element in $\check{\mathcal{S}}$, i.e. a sequence of forests, say of length $r + 1$, $\check{s} = \check{s}_r = (s_0, s_1, \ldots, s_r) \in \mathcal{S}_{0:r}$. Following Del Moral et al. (2006), we define the new unnormalized measure by a product $\check{\pi}(\check{s}_r) = \pi(s_r) \prod_{j=1}^{r} \nu_{s_j}^-(s_{j-1})$. Since the ν_s^- are assumed to be probability distributions, marginalization over $s_0, s_1, \ldots, s_{r-1}$ recovers the original measure π.

The proposal over $\check{\mathcal{S}}$ creates an identical copy of the sequence of forests, and adds to it a new element by sampling from the proposal density ν^+. With this definition, given an element $\check{s} \in \check{\mathcal{S}}$, there can be only one predecessor $\varrho(\check{s})$, namely the prefix of the sequence with the last element removed. As a simple example, Figure 2 (c) shows the acyclic poset over the extended space for the simple finite cyclic poset in Figure 2 (b).

Finally, standard SMC operating on this extended space can be seen to be equivalent to CSMC, since the weight updates simplify to:

$$\frac{\check{\pi}(\check{s}_r)}{\nu^+_{\check{s}_{r-1}}(\check{s}_r)\check{\pi}(\check{s}_{r-1})} = \frac{\pi(s_r)\prod_{j=1}^{r}\nu^-_{s_j}(s_{j-1})}{\nu^+_{\check{s}_{r-1}}(\check{s}_r)\pi(s_{r-1})\prod_{j=1}^{r-1}\nu^-_{s_j}(s_{j-1})} = \frac{\pi(s_r)\nu^-_{s_r}(s_{r-1})}{\pi(s_{r-1})\nu^+_{s_{r-1}}(s_r)}.$$

This completes the proof in the finite cyclic case.

3.5 General setup

In this section, we extend the result of the previous section to more general spaces. As before, π denotes the unnormalized target measure, but in this section we do not restrict π to be defined over a discrete space. More precisely, let $\mathcal{F}_{\mathcal{X}}$ denote a sigma-algebra on $\mathcal{X}$, and let $\pi : \mathcal{F}_{\mathcal{X}} \to [0, \infty)$. We assume that the user has provided an extension $\pi : \mathcal{F}_{\mathcal{S}} \to [0, \infty)$, and a pair of forward and backward probability kernels $\nu^+, \nu^- : \mathcal{S} \times \mathcal{F}_{\mathcal{S}} \to [0, 1]$.

Next, we define the following measures on the product space $\mathcal{S} \times \mathcal{S}$: $\tau^+(A \times B) = \pi_A(\nu^+(B)) = \int \pi_A(dx)\nu^+_x(B)$, $\tau^-(A \times B) = \pi_B(\nu^-(A)) = \int \pi_B(dx)\nu^-_x(A)$, where $A, B \in \mathcal{F}_{\mathcal{S}}$, and for any measure λ and measurable A, $\lambda_A(B)$ denotes $\lambda(A \cap B)$.

We will make two assumptions.

Assumption 3. *We have $\tau^- \ll \tau^+$.*

This assumption implies by the Radon-Nikodym theorem the existence of a derivative $\tau^-/\tau^+ : \mathcal{S} \times \mathcal{S} \to [0, \infty)$. We also assume that there is a version $w = \tau^-/\tau^+$ such that:

Assumption 4. *There is a ranked poset $(\mathcal{S}, \prec, \rho)$ such that s' covers s if and only if $w(s, s') > 0$. Using the notation π_r as a shorthand for $\pi_{\rho^{-1}(r)}$, the Hasse diagram* *induced by $\prec$ is (a) connected and (b) there is an R such that $\pi_r = \pi$ for $r \geq R$, and that π_r is a Dirac delta for $r \leq 0$.*

To get a compact notation for CSMC, we introduce the following Monte Carlo *propagation operator*:

$$(\operatorname{prop}_K \lambda)\phi = \|\lambda\| \left(\frac{1}{K} \sum_{k=1}^{K} w(S_k, S'_k)\phi(S'_k) \right),$$

where $S_k \sim \bar{\lambda}$, $S'_k | S_k \sim \nu^{+}_{S_k}(\cdot)$, independently, $\lambda : \mathcal{F}_{\mathcal{S}} \to [0, \infty)$ is an arbitrary positive measure on the poset, and $\phi : \mathcal{S} \to \mathbb{R}$ is a test function. We remind the reader that these operators are random. We have dropped the dependence on S_k, S'_k to simplify the notation.

The propagation operator incorporates both a multinomial resampling step and a proposal step in one equation. If we denote the composition of these operators by $\operatorname{prop}^2_K \pi_0 = (\operatorname{prop}_K(\operatorname{prop}_K \pi_0))$, where $\pi_0 = \delta_\perp$, then the full CSMC algorithm can be summarized by $\pi_{R,K} = \operatorname{prop}^R_K \pi_0$. With this notation, we can write the main result as:

Proposition 5. *Under Assumptions 3 and 4, and if $\phi : \mathcal{X} \to \mathbb{R}$ is measurable, $|\phi| \leq C_1$ and $w \leq C_2$ for some C_1 and C_2, then for all $r \in \{1, 2, \ldots, R\}$, $\pi_{r,K}\phi \to \pi_r\phi$. In particular, $\pi_{R,K}\phi \to \pi\phi$.*

The proof and supporting lemmas can be found in the appendix.

3.6 A concrete non-clock phylogenetic tree example

We now apply the theory of the last section to construct a detailed example of a CSMC algorithm handling non-clock trees with continuous branch lengths. More precisely we will describe the poset, the extended distribution on this poset, the proposal, and the overcounting correction.

The poset we will use is defined over *rooted non-clock forests* $s = \{(t_i, X_i)\}$, which we formally define as a set of rooted non-clock X_i-trees t_i. The poset structure defined over this set $\mathcal{S}$ is induced by forest inclusion: a forest s is deemed to precede another forest s' if the topology of s is a subset of the topology of s', and the branch lengths of the shared edges match in s and s'.

Note that the extended distribution over $\mathcal{S}$ defined by Equation (2) can accommodate non-clock trees with branch lengths, for example by setting its constituents $\mathbb{P}(\mathcal{Y}|\theta, t)$ and $p(t|\theta)$ to those described in Section 2.2.

To specify a proposal over rooted non-clock forests, three elements need to be sampled: the pair of trees to merge, the length of the added branch b', and the new position of the root. To pick the pair of trees to merge, the cheapest option is to pick the pair of trees in s uniformly at random among the $\binom{|s|}{2}$ pairs.

For the proposal over branch lengths, we need to consider two subcases, depending on the number of trees in the forest. If there are exactly two trees in the forest before applying the proposal (in other words, if this is the last iteration of CSMC), we propose a single length b'_1 distributed according to an exponential distribution with parameter λ_{bl} (see Section 2.2). Otherwise if there are more than two trees, we propose two independent branch lengths, b'_1, b'_2, each with rate λ_{bl}. The density of the proposal ν^+ is therefore given by:

$$\nu_s^+(s') = \binom{|s|}{2}^{-1} \lambda_{\mathrm{bl}} \exp(-b'_1 \lambda_{\mathrm{bl}}) \Big(\mathbf{1}[|s| = 2] + \lambda_{\mathrm{bl}} \exp(-b'_2 \lambda_{\mathrm{bl}}) \mathbf{1}[|s| > 2] \Big)$$

For the overcounting correction, we can still apply the choice of ν^- discussed in the finite setup even though $\mathcal{S}$ is continuous. Indeed, the way we construct the partial order ensures $|\varrho(s')| < \infty$.

Putting everything together, writing (without loss of generality)

$$s' = s \cup \{(t_{\mathrm{m}}, X_{\mathrm{m}})\} \backslash \{(t_1, X_1), (t_2, X_2)\},$$

for some merged subtree $(t_{\mathrm{m}}, X_{\mathrm{m}})$ connecting the subtrees (t_1, X_1) and (t_2, X_2), we get the weight update:

$$w(s, s') = \frac{\mathbb{P}(\mathcal{Y}(X_{\mathrm{m}})|\theta, t_{\mathrm{m}})}{\mathbb{P}(\mathcal{Y}(X_1)|\theta, t_1)\mathbb{P}(\mathcal{Y}(X_2)|\theta, t_2)} \frac{1}{\sum_{(t_i, X_i) \in s'} \mathbf{1}[|X_i| > 1]}.$$

3.7 Particle Markov chain Monte Carlo

We have seen in the previous sections how CSMC can be used to obtain approximations $\hat{p}(t|\theta, \mathcal{Y})$ and $\hat{\mathbb{P}}(\mathcal{Y}|\theta)$ of both $p(t|\theta, \mathcal{Y})$ and $\mathbb{P}(\mathcal{Y}|\theta)$. However, as discussed in Section 2.2, in most realistic scenarios the evolutionary parameters θ are unknown and we are interested in sampling from $p(\theta, t|\mathcal{Y})$.

Designing efficient MCMC methods in this context is challenging. Standard strategies consist of updating the parameters given the tree and updating the tree given the parameters. Even when θ is known, it is difficult to sample from $p(t|\mathcal{Y}, \theta)$ and this was the main motivation for the introduction of the CSMC method. Moreover, even if it were possible to sample efficiently from $p(t|\mathcal{Y}, \theta)$, such a Gibbs type strategy would mix slowly when the tree and parameters are strongly correlated under the posterior.

We propose here an alternative algorithm where we update jointly the parameter and the tree. This algorithm can be thought of an approximation of the 'ideal' marginal Metropolis-Hastings algorithm targeting $p(\theta, t|\mathcal{Y})$ using a proposal $q_{\mathrm{mmh}}((\theta, t), (\theta^*, t^*))$ $= q_{\mathrm{param}}(\theta, \theta^*)p(t^*|\mathcal{Y}, \theta^*)$, where $q_{\mathrm{param}}(\theta, \cdot)$ is a proposal distribution to propose a new parameter from the current value θ. The terminology "marginal" stems from the fact that this is somewhat equivalent to integrating out t as the resulting acceptance ratio is independent of t. Unfortunately this algorithm cannot be implemented as it requires being able to sample from $p(t|\mathcal{Y}, \theta)$ and the acceptance ratio is dependent on $\mathbb{P}(\mathcal{Y}|\theta)$.

Algorithm 2 : Particle marginal Metropolis-Hastings

```
Initialization, $i = 0$,
        set $\theta(0)$ arbitrarily and
        run the CSMC algorithm targeting $p(t|\mathcal{Y}, \theta(0))$, sample $t(0) \sim \hat{p}(\cdot|\mathcal{Y}, \theta(0))$ and
        let $\hat{\mathbb{P}}(\mathcal{Y}|\theta(0))$ denote the marginal likelihood estimate.
for iteration $i \geq 1$ do
        sample $\theta^* \sim q_{\text{param}}(\theta(i-1), \cdot)$,
        run the CSMC algorithm targeting $p(t|\mathcal{Y}, \theta^*)$, sample $t^* \sim \hat{p}(\cdot|\mathcal{Y}, \theta^*)$ and
        let $\hat{\mathbb{P}}(\mathcal{Y}|\theta^*)$ denote the marginal likelihood estimate. With probability

        $$\min\left(1, \frac{\hat{\mathbb{P}}(\mathcal{Y}|\theta^*)p(\theta^*)}{\hat{\mathbb{P}}(\mathcal{Y}|\theta(i-1))p(\theta(i-1))} \frac{q_{\text{param}}(\theta^*, \theta(i-1))}{q_{\text{param}}(\theta(i-1), \theta^*)}\right), \qquad (5)$$

        set $\theta(i) = \theta^*$, $t(i) = t^*$, and $\hat{\mathbb{P}}(\mathcal{Y}|\theta(i)) = \hat{\mathbb{P}}(\mathcal{Y}|\theta^*)$; otherwise set $\theta(i) = \theta(i-1)$, $t(i) = t(i-1)$
        and $\hat{\mathbb{P}}(\mathcal{Y}|\theta(i)) = \hat{\mathbb{P}}(\mathcal{Y}|\theta(i-1))$.
end for
```

The main idea behind the particle marginal Metropolis-Hastings (Andrieu et al. 2010) is to substitute the CSMC estimates $\hat{p}(t|\theta, \mathcal{Y})$ and $\hat{\mathbb{P}}(\mathcal{Y}|\theta)$ for $p(t|\mathcal{Y}, \theta)$ and $\mathbb{P}(\mathcal{Y}|\theta)$. In our context, this algorithm takes the form described in Algorithm 2. The key result of Andrieu et al. (2010), which can be straightforwardly adapted here, states that whatever being the number K of particles used in the CSMC, the particle MCMC kernel admits $p(\theta, t|\mathcal{Y})$ as invariant distribution. However, the choice of K will affect the performance of the algorithm. Doucet et al. (2015) give theory showing that one should select K such that the standard deviation of the log-likelihood estimate is around one so as to minimize the asymptotic variance of the resulting particle MCMC estimates for fixed computational efforts.

4 Numerical examples

We checked the correctness of our implementation of the CSMC algorithm (within particle MCMC) using the joint distribution testing methodology of Geweke (2004) (see Supplementary Document) before giving the following numerical examples. In this section, we used the commonly adopted four rate category discrete approximation

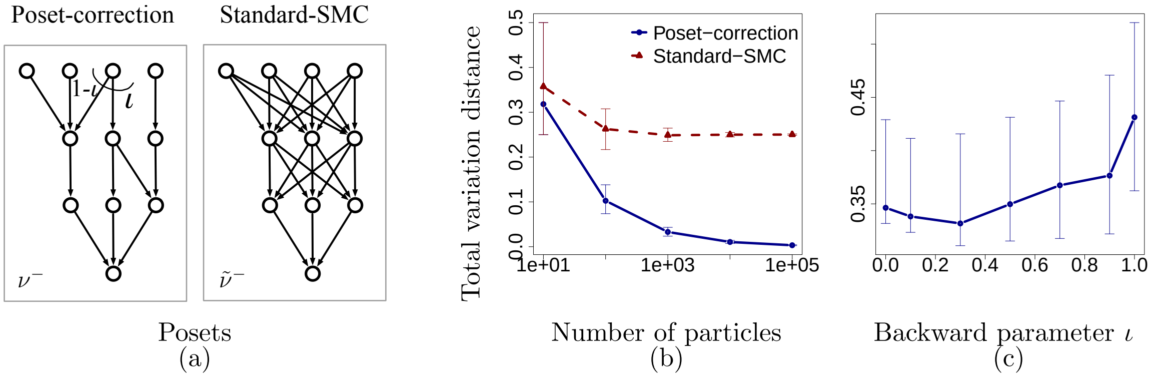


FIGURE 3: (a) The support of the backward kernels used in the synthetic poset experiments. (b) Total variation distance of the particle approximation to the exact target distribution using the two algorithms as the number of particles K increases. (c) The total variation distance as a function of the parameter ι for $K = 10$.

to approximate a gamma rate distribution (Yang 2006). The Java software allowing to reproduce all the experiments in this section is available at `http://people.stat.sfu.ca/~lwa68/csmcphylo/CSMC-Phylo.html`.

4.1 Synthetic posets

We start with an illustration of the effect that a lack of appropriate correction can have on the approximation in cyclic posets. We use a synthetic poset with a small cardinality $|\mathcal{S}| < \infty$ so that the target distribution can be computed exactly. The poset has a support as in Figure 2 (b), with each value on the support generated from independent uniform distributions, and then normalized. [2] Here both the exact solution and the approximations are low-dimensional multinomial distributions, so the total variation distance can be computed efficiently.

First, we use this simple test case to show that Assumption 3 on ν^- ensures consistency of the estimates (Proposition 5). Two SMC algorithms are compared: the first one, 'Poset-correction' is a CSMC algorithm using a backward kernel satisfying

[2] The values of π, ν^+, ν^- can be downloaded from `http://people.stat.sfu.ca/~lwa68/csmcphylo/syntheticPoset`.

Assumption 3, while the second one, 'Standard-SMC', does not. The latter can be equivalently viewed as using a backward proposal proportional to $\tilde{\nu}^{-}_{s'}(s) = 1$ on all s such that $\rho(s') = \rho(s) + 1$, in which case the weight update reduces to a standard SMC weight update. It can be checked easily that this choice violates Assumption 3. The supports of the two backward kernels are shown in Figure 3 (a).

In Figure 3 (b), we compare the performance of the two algorithms as the number of particles increases. The performance of the algorithms is measured using the total variation distance, and the experiment for each number of particle is repeated 1000 times using different random seeds. The results show that only the algorithm satisfying Assumption 3 gives an approximation with a total variation distance going to zero as the number of particles K increases.

We did a second experiment to give an example where cycles in posets are beneficial. In this experiment, we fix the structure of the backward kernel as in the 'Poset-correction', with the exception of one of the backward transition, which we parameterize with a number $\iota \in [0, 1]$ shown in Figure 3 (a). When $\iota \in \{0, 1\}$, this effectively corresponds to removing a cycle in the Hasse diagram of the poset. We show in Figure 3 (c) the total variation distance as a function of this parameter ι for $K = 10$. It can be seen that the best performance is attained away from the points $\{0, 1\}$, demonstrating that cycles can indeed be beneficial.

4.2 Phylogenetic experiments

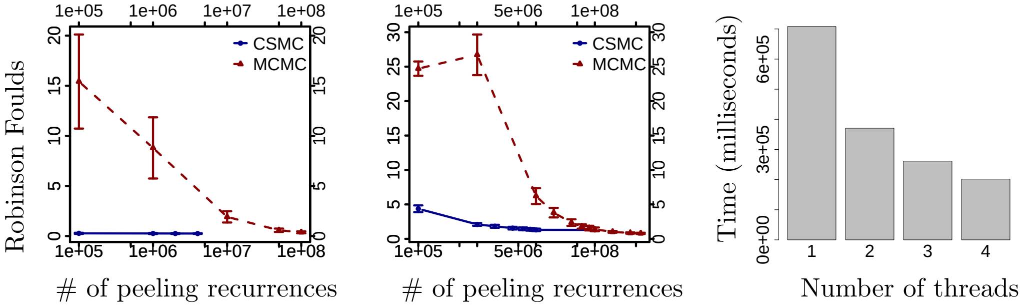
Reconstruction of synthetic phylogenies. In this study, we evaluate the quality of phylogenetic trees reconstructed using the proposed CSMC method. In each experiment, we summarize the posterior tree distribution using the majority-rule consensus tree (Felsenstein 2003), and we calculate the tree distance between the majority-rule consensus trees and the true trees using the Robinson Foulds metric (Robinson and Foulds 1981). Smaller tree distances reflect better tree reconstructions.

We simulated 1000 ultrametric trees of 100 taxa, assuming the waiting time between two coalescent events was exponentially distributed with rate 10. The non-clock trees were obtained by perturbing the branch lengths of ultrametric trees. Specifically, we modified a branch of length b by adding to it a noise randomly sampled from Unif$(-.3b, .3b)$. For each tree, we generated 10 datasets, each consisting of 100 DNA sequences of length 2000, using the continuous-time Markov chain that is parameterized by the GTR+Γ+I model with $(\pi_A, \pi_C, \pi_G, \pi_T) = (0.3, 0.2, 0.2, 0.3)$, $(\gamma_{AC}, \gamma_{AG}, \gamma_{AT}, \gamma_{CG}, \gamma_{CT}, \gamma_{GT}) = (0.26, 0.18, 0.17, 0.15, 0.11, 0.13)$, $\alpha = 0.5$, and $p_0 = 0$.

The CSMC algorithm was applied to each of these datasets. For comparison, we also used a popular Bayesian phylogeny software package, MrBayes, which provides an efficient implementation of MCMC for a range of tree models (Huelsenbeck and Ronquist 2001; Ronquist et al. 2012). In this set of experiments, we fixed the parameters to the true values and focused on comparing their performance in phylogenetic tree estimation (results with unknown parameters are presented further in this section). Priors for the branch lengths were exponential distributions with rate 10. We set a uniform prior over tree topologies.

In both the CSMC and MCMC algorithms, the computational bottleneck is the peeling recurrence (Felsenstein 1981)—the sum-product or belief propagation recurrence, specialized to phylogenetics—, which needs to be computed at each speciation event. The running time of each recurrence call is proportional to the number of sites times the squares of the number of characters.[3] Since our algorithms and MrBayes are implemented in different programming languages (Java versus C), we first report running

[3] This can be accelerated by parallelization (Ayres et al. 2012), but this is orthogonal as it is possible to do so with both CSMC and MCMC.



(a)

(b)

(c)

FIGURE 4: The mean Robinson Foulds metric (standard deviation) versus the # of peeling recurrences in log scale averaged over 1000 datasets simulated from ultrametric trees (a) and non-clock trees (b). (c) Computing time in milliseconds versus different number of threads.

times here as the number of times the peeling recurrence is calculated—but we also report wall clock times below. Figure 4 (a) and (b) show the mean and standard deviation of the Robinson Foulds metric versus the number of peeling recurrences in log scale using the datasets simulated from ultrametric trees and non-clock trees, respectively. In this setup, CSMC generally outperformed MrBayes in terms of providing a higher mean accuracy as well as a smaller variance for any given computational budget. More specifically, for ultrametric trees, CSMC can achieve a Robinson Foulds metric that is very close to zero with the number of particles as small as 1000 (equivalently 10^5 peeling recurrences) in about 3.6 minutes. In contrast, MrBayes used more than 80 minutes to reach similar performance. For non-clock trees, CSMC outperformed MrBayes when the number of peeling recurrences is smaller than about 10^8; using a larger number of iterations (in MCMC) and particles (in CSMC), CSMC and MrBayes performed similarly for the simulated datasets.

Gains from parallelization. To illustrate the computational gains achievable by parallelizing CSMC, we ran CSMC on two simulated datasets from an ultrametric tree and a non-clock tree, each containing 30 observed taxa and 2000 sites. We ran our algorithm using 100,000 particles, varying the number of threads used on a 2.40GHz

	π_A	π_C	π_G	γ_{AC}	γ_{AG}	γ_{AT}	γ_{CG}	γ_{CT}	α
True	0.3	0.2	0.2	0.26	0.18	0.17	0.15	0.11	0.5
pMCMC	.28(.01)	.21(.01)	.22(.01)	.31(.02)	.18(.02)	.17(.02)	.13(.02)	.12(.01)	.48(.04)
MrBayes	.28(.01)	.21(.01)	.22(.01)	.30(.02)	.18(.02)	.17(.02)	.13(.02)	.12(.01)	.49(.05)

Table 1: True values of the evolutionary parameters, and posterior means and standard deviations obtained using MrBayes and particle MCMC. We omit the parameters π_T, γ_{GT}, which are deterministic functions of those shown here (via simplex and rate normalization constraints).

Intel Xeon 16-cores E7330 architecture. Figure 4 (c) shows the computing time in milliseconds versus different number of threads for the two datasets. The results show that notable speed gains can even be made by adding a small number of additional cores. In this experiment, only the proposal was parallelized. The resampling step, which can be expensive when a large number of particles is used, could also be parallelized (Lee and Whiteley 2014).

Estimation of evolutionary parameters. We generated a random ultrametric tree of 50 taxa and simulated sequences of length 1000 from the same GTR+Γ model as the previous experiment, but holding out all GTR+Γ parameters this time. We used 10,000 particles in the CSMC algorithm, and ran 10,000 MCMC iterations (total running time: 3h19m). We used the default setting for MrBayes and ran 10^7 MCMC iterations (total running time: 5h30m). We used trace plots to ensure that the number of iterations was sufficient for both MrBayes and particle MCMC. Table 1 shows the true values and the posterior means and standard deviations of the evolutionary parameters obtained using MrBayes and particle MCMC. Figure 5 shows the histograms of the parameter posteriors and the log-likelihood of the posterior trees obtained using particle MCMC (first row) and MrBayes (second row). While the parameter posteriors are similar, particle MCMC generally samples higher likelihoods, with an average log-likelihood of -5960 (2.4) versus -5969 (6.4) for MrBayes. The cause of the difference is most likely that the posterior distribution of trees is multimodal and that MCMC can be

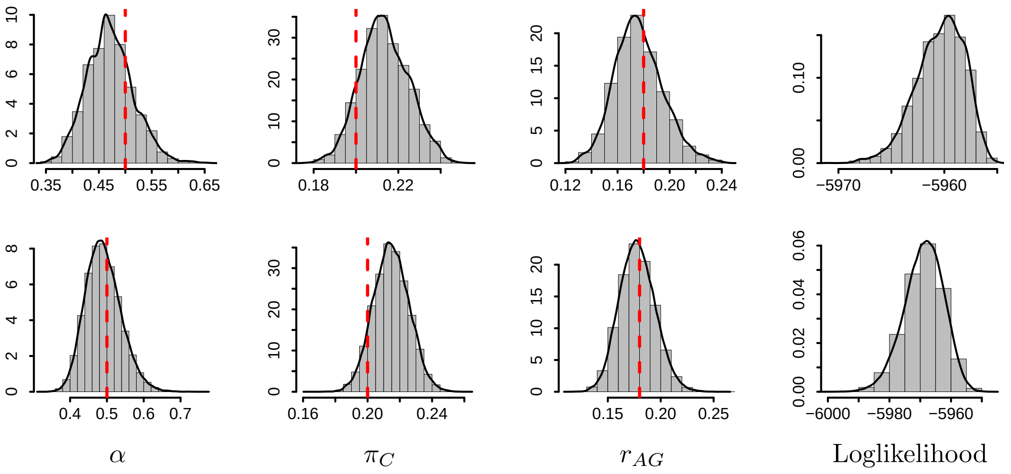


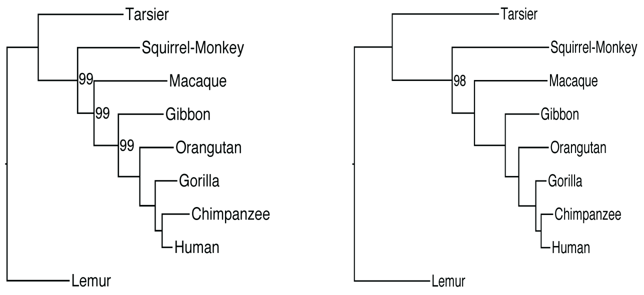
FIGURE 5: Histograms of a representative subset of the parameter posterior distributions, and of the log-likelihood of the posterior trees obtained using particle MCMC (first row, computed in 3h19m) and MrBayes (second row, computed in 5h30m). Dashed vertical lines represent the true values of the parameters.

more easily trapped in a local maximum. To verify this, we initialized MrBayes with the tree obtained from particle MCMC at the last iteration ($t(i_{\text{last}})$ in the notation of Algorithm 2, where i_{last} denotes the last particle MCMC iteration) and ran MrBayes for 10^7 iterations. With this initialization, the average log-likelihood of MrBayes was -5962 (5.4), which is closer to the result from particle MCMC.

Primate dataset. To illustrate our method, we analyzed a set of mitochondrial DNA sequences for the protein coding regions of nine primates (Brown et al. 1982) with a GTR+Γ model. Each DNA sequence has 888 sites. To select an efficient trade-off between the number of particles K and the number of MCMC iterations, we follow Doucet et al. (2015), which prescribes selecting K such that the standard deviation of the log-likelihood estimate is around one. Let $\sigma(\hat{\theta}; K)$ denote the standard deviation of the log-likelihood estimate obtained from CSMC using K particles, where $\hat{\theta}$ is the posterior mean of the evolutionary parameters from a short particle MCMC chain using a large number of particles ($50,000$ in this example). Table 2 shows that $\sigma(\hat{\theta}; K)$

K	1,000	10,000	11,000	12,000	15,000	50,000
$\sigma(\hat{\theta}; K)$	12.798	1.356	1.189	0.998	0.993	0.553

Table 2: The standard deviations of the log-likelihood estimates obtained using CSMC run on the primate dataset.



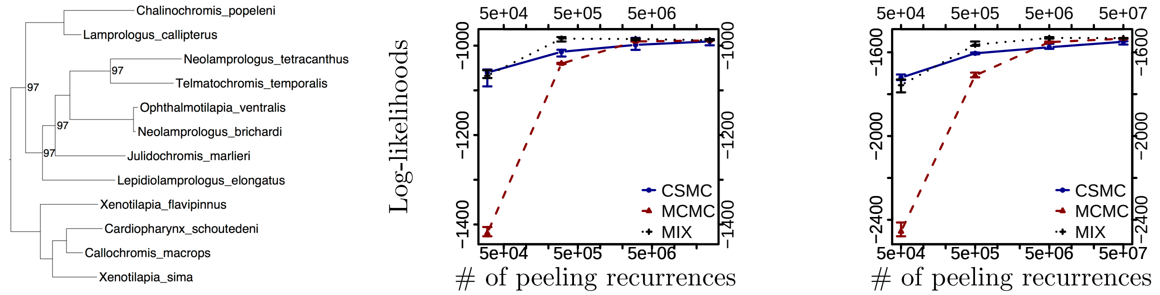
Particle MCMC MrBayes

Figure 6: The majority-rule consensus trees for the primate dataset estimated by the particle MCMC and MrBayes. The numbers on the trees represent the clade posterior probabilities (number 100 is omitted).

decreases as the number of K increases, having $\sigma(\hat{\theta}; 12,000)$ close to one. We ran the particle MCMC using 12,000 particles for 10,000 iterations, which took 56 minutes. For comparison, we ran MrBayes for 10^7 iterations, which took 55 minutes. The mean (standard deviation) log-likelihoods are -5038 (2.6) using the particle MCMC and -5040 (3.8) using MrBayes. Figure 6 depicts the majority-rule consensus trees with log-likelihoods -5655 and -5873, obtained by the particle MCMC and MrBayes, respectively.

Cichlid Fishes. We analyzed aligned protein coding mitochondrial DNA sequences obtained from 12 species from two tribes (*Ectodini* and *Lamprologini*) of African cichlid fish (Kocher et al. 1995; Cheon and Liang 2008). Each DNA sequence consists of 1047 sites. Since about a half of the sites have identical nucleotides, we used the GTR+Γ+I model, where there is one parameter, p_0, to consider the proportion of the invariant sites. We used 20,000 particles in the CSMC algorithm at each iteration of the particle MCMC algorithm, and ran it for 10,000 iterations. Figure 7(a) shows the estimated majority-rule consensus trees and the clade posterior probabilities.

Chloroplast dataset. In this section, we investigated the performance of a simpler strategy mixing CSMC and MCMC for reconstructing phylogenetic trees: running CSMC for $\eta\%$ of the budget followed by MCMC for the remaining $(1-\eta)\%$ of the budget. We analyzed two randomly selected sub-datasets of 30 and 50 ribosomal RNA sequences obtained from the chloroplast of 199 plant species (Cannone et al. 2002) using a simpler model: Kimura's two parameter (K2P) model (Kimura 1980). The only unknown parameter, the transition/transversion rate, was approximated by its posterior mean obtained from a pilot run of MrBayes. We applied to this dataset CSMC, MrBayes, and a simple combination of the two methods (MIX), which consists in running CSMC with a number of particles calibrated for 30% of the allowed computational budget, followed by MCMC for the rest of the allowed time, but initialized by sampling from the CSMC approximation. Figure 7(b,c) shows the log-likelihood of the majority-rule consensus trees of using CSMC, MrBayes, and MIX versus the number of peeling recurrences. The three methods obtained similar majority-rule consensus trees given a sufficiently large budget. On a limited computational budget, CSMC outperformed MCMC by a large margin. Moreover, the MIX strategy outperformed both CSMC and MCMC for all computational budgets considered (except for CSMC performing slightly better than MIX on the smallest budget considered).



(a) (b) (c)

Figure 7: (a) The majority-rule consensus trees for cichlid fish estimated by particle MCMC. The numbers on the trees represent the clade posterior probabilities. (b) Log-likelihood of the majority-rule consensus trees using CSMC, MrBayes, and a combination strategy (MIX, see text), as a function of the number of peeling recurrences, on the Chloroplast RNA datasets of 30 taxa, and (c), 50 taxa.

5 Conclusion and discussion

We have presented an original SMC methodology that provides a complementary approach to MCMC for Bayesian phylogenetic inference. While SMC methods and related approaches such as sequential importance sampling and approximate Bayesian computation are already popular in population genetics, they have not yet become widely used in phylogenetics. This is because a richer class of trees is needed in phylogenetics models, whereas related SMC work on population genetics typically focuses on coalescent models (Griffiths and Tavaré 1996; Liu 2001; Beaumont et al. 2002; Marjoram et al. 2002; Iorio and Griffiths 2004; Paul et al. 2011). In contrast to previous phylogenetic SMC methods (Teh et al. 2008; Görür and Teh 2009; Tom et al. 2010; Bouchard-Côté et al. 2012), our proposed CSMC methodology is not restricted to ultrametric trees and can handle the full range of combinatorial structures required by modern phylogenetic models. CSMC can additionally be easily combined with MCMC through particle MCMC so as to estimate evolutionary parameters. We have demonstrated numerically that in many Bayesian phylogenetic scenarios, CSMC can efficiently complement MCMC methods in terms of computation speed and estimation accuracy.

We have also shown that our method benefits from parallel architectures. While this is also true to some extent with parallel Metropolis coupled MCMC (Altekar et al. 2004), there are important differences in how CSMC and the parallel Metropolis coupled MCMC sampler make use of parallelization. In the parallel Metropolis coupled MCMC sampler, the additional gains from parallelism can quickly decrease as more chains are added, because many swaps are needed to get from the most heated chain to the main chain. In contrast, the CSMC sampler has a structure well-adapted to parallelization.

There are a range of possible ways to further improve the current methodology. Many of the sophisticated strategies developed for standard SMC methods directly apply in our context, including using improved resampling schemes or additional local MCMC steps; see Doucet and Johansen (2011) for a recent review.

Our framework also specifies weak order-theoretic conditions for choosing a proposal distribution and an extension of π, while preserving asymptotic guarantees. Since samplers use a finite number of particles in practice, these choices obviously impact the algorithm performance. For example, it might be possible to develop sampling analogues of informed search techniques (Russell and Norvig 2009). Additionally, there are various ways to define the sequence of partial states as long as the objects obtained at the last step of CSMC are in the target space. This could lead to new algorithms.

While we have concentrated on the computational gains in our experiments, there are also modelling motivations behind CSMC. For example, some evolutionary models have been put forward recently that take into account the evolving structure of molecules (Nasrallah et al. 2011). Since the likelihood is intractable in all of these models, a sampler would therefore need to augment the space with the state of the string at intermediate points of phylogeny. Unfortunately, because it is non-trivial to propose new values for these latent strings in an MCMC sampler, these more accurate models have not yet been used for phylogenetic inference; their use has been limited to forward data simulation. It is simpler to propose values for these latent strings with CSMC, where the task reduces to an end-point conditioned simulation problem (Hobolth and Stone 2009).

Finally, although the proposed method is motivated by Bayesian phylogenetic inference, CSMC has potential applications outside of this domain, for example in the field of multiple sequence alignment (Edgar and Batzoglou 2006). Multiple sequence alignment methods can be broadly categorized into heuristic sequential methods (Larkin et al. 2007), or model-based approaches (Holmes and Bruno 2001; Redelings and Suchard 2005). Despite the theoretical and practical limitations of heuristic alignment methods over model-based methods, the former are often preferred by practitioners in practice, mostly on computational grounds. By proposing alignments sequentially, CSMC provides a promising framework to bring the benefits of model-based alignment models to a wider range of situations.

6 Acknowledgements

The authors would like to thank the editor, the associate editor, the reviewers and Jens Lagergren from KTH Royal Institute of Technology for their constructive comments which helped improve the paper significantly. The authors would like to thank Westgrid for the computing support. This research was supported by grants from the Natural Science and Engineering Research Council of Canada to Liangliang Wang and Alexandre Bouchard-Côté. Arnaud Doucet's research was partly funded by the Engineering and Physical Sciences Research Councils under grants EP/K000276/1 and

EP/K009850/1.

Appendix: Proof of L^2 convergence

We present a simple proof of Proposition 5 showing L^2 convergence of the CSMC estimate. There are many sharper convergence results available for standard SMC (Del Moral 2004) that could be adapted to our setup. However the main point here is to illustrate how the conditions on the poset are used in the proof. We assume throughout this section that the assumptions of Proposition 5 hold.

Recall also that we are assuming multinomial resampling at every step, and that this step is included in the definition of the operator prop_K.

We start by introducing a series of lemmas.

Lemma 6. *For any positive measure λ with $\|\lambda\| < \infty$, we have $\mathbb{E}[(\mathrm{prop}_K \lambda)\phi] = (\mathrm{prop}\,\lambda)\phi$, where $(\mathrm{prop}\,\lambda)\phi = \int \lambda(\,\mathrm{d}x) \int \nu_x^+(\,\mathrm{d}y) w(x,y)\phi(y)$, and $\mathrm{prop}_K \lambda$ is defined in Section 3.5.*

Proof. By linearity:

$$\begin{aligned}\mathbb{E}[(\mathrm{prop}_K \lambda)\phi] &= \|\lambda\| \mathbb{E}[w(S_1, S'_1)\phi(S'_1)] \\ &= \|\lambda\| \int \bar{\lambda}(\,\mathrm{d}x) \int \nu_x^+(\,\mathrm{d}y) w(x,y)\phi(y) = (\mathrm{prop}\,\lambda)\phi\end{aligned}$$

□

Lemma 7. *For any positive measure λ with $\|\lambda\| < \infty$, we have $\mathbb{E}\left[(\mathrm{prop}_K \lambda)\phi - (\mathrm{prop}\,\lambda)\phi\right]^2 \le \frac{(C_1C_2)^2\|\lambda\|^2}{K}$, so $(\mathrm{prop}_K \lambda)\phi \xrightarrow{\mathbf{L}^2} (\mathrm{prop}\,\lambda)\phi$.*

Proof. Using independence of (S_k, S'_k) and $(S_{k'}, S'_{k'})$, $k \neq k'$ in the definition of prop_K, we have:

$$\mathbb{E}\left[(\mathrm{prop}_K \lambda)\phi - (\mathrm{prop}\,\lambda)\phi\right]^2 = \tfrac{\|\lambda\|^2}{K} \mathbf{V}ar[w(S_k, S'_k)\phi(S'_k)] \le \tfrac{(C_1C_2)^2\|\lambda\|^2}{K}.$$

□

Corollary 8. *We have $\mathbb{E}\left[\pi_{r,K}\phi - (\mathrm{prop}\,\pi_{r-1,K})\phi\right]^2 \to 0$.*

Proof. Since $\|\pi_{r,K}\| \le C_2^r < \infty$,

$$\mathbb{E}\left[\pi_{r,K}\phi - (\mathrm{prop}\,\pi_{r-1,K})\phi\right]^2 = \mathbb{E}\left\{\mathbb{E}\left[(\pi_{r,K}\phi - (\mathrm{prop}\,\pi_{r-1,K})\phi)^2 \,\middle|\pi_{r-1,K}\right]\right\} \le \mathbb{E}\left[\frac{\|\pi_{r-1,K}\|^2 C_1^2 C_2^2}{K}\right] \to 0.$$

□

Lemma 9. *For all r, $(\mathrm{prop}\, \pi_r)\phi = \pi_{r+1}\phi$.*

Proof. By definition, we have for all bounded measurable $f : \mathcal{S}^2 \to \mathbb{R}$, $\int\int \pi(\,\mathrm{d}x)\nu_x^+(\,\mathrm{d}y) f(x,y) = \int\int \tau^+(\,\mathrm{d}x,\,\mathrm{d}y) f(x,y)$. Similarly, $\int\int \pi(\,\mathrm{d}y)\nu_y^-(\,\mathrm{d}x) f(x,y) = \int\int \tau^-(\,\mathrm{d}x,\,\mathrm{d}y) f(x,y)$. Using these identities and basic properties of the Radon-Nikodym derivative $w = \,\mathrm{d}\tau^-/\,\mathrm{d}\tau^+$:

$$\begin{aligned}
(\mathrm{prop}\, \pi_r)\phi &= \int \mathbf{1}[\rho(x) = r]\pi(\,\mathrm{d}x) \int \nu_x^+(\,\mathrm{d}y) w(x,y)\phi(y) \\
&= \int\int \mathbf{1}[\rho(y) = r+1]\pi(\,\mathrm{d}x)\nu_x^+(\,\mathrm{d}y) w(x,y)\phi(y) \\
&= \int\int \mathbf{1}[\rho(y) = r+1]\tau^+(\,\mathrm{d}x,\,\mathrm{d}y) w(x,y)\phi(y) \\
&= \int\int \mathbf{1}[\rho(y) = r+1]\tau^-(\,\mathrm{d}x,\,\mathrm{d}y)\phi(y) \\
&= \int\int \mathbf{1}[\rho(y) = r+1]\pi(\,\mathrm{d}y)\nu_y^-(\,\mathrm{d}x)\phi(y) = \int \mathbf{1}[\rho(y) = r+1]\pi(\,\mathrm{d}y)\phi(y) \int \nu_y^-(\,\mathrm{d}x) = \pi_{r+1}\phi.
\end{aligned}$$

Here to change the indicator $\mathbf{1}[\rho(x) = r]$ into $\mathbf{1}[\rho(y) = r+1]$, we have used the definition of the poset and the fact that its Hasse diagram is connected. □

Lemma 10. *If for all bounded measurable ϕ, $\pi_{r,K}\phi \xrightarrow{\mathbf{L}^2} \pi_r\phi$, then we also have $(\mathrm{prop}\, \pi_{r,K})\phi \xrightarrow{\mathbf{L}^2} (\mathrm{prop}\, \pi_r)\phi$. Moreover, by Lemma 9 the right-hand side of the last equation is equal to $\pi_{r+1}\phi$.*

Proof. Let ϕ be a bounded function, so that there exists a new constant C such that $|\phi| < C$. Let $\tilde{\phi}(x) = \int_A \nu_x^+(\,\mathrm{d}y) w(x,y)\phi(y)$. Since $w < C_2$, $|\tilde{\phi}'| < CC_2$, we can use the test function $\tilde{\phi}$ in $\pi_{r,K}\phi \xrightarrow{\mathbf{L}^2} \pi_r\phi$ to obtain $(\mathrm{prop}\, \pi_{r,K})\phi \xrightarrow{\mathbf{L}^2} (\mathrm{prop}\, \pi_r)\phi$. □

Proof. (of the main proposition) We proceed by induction, showing for $r \geq 0$, and for all bounded ϕ, we have $\pi_{r,K}\phi \xrightarrow{\mathbf{L}^2} \pi_r\phi$. The base case is trivial, since $\pi_{0,K}$ and π_0 are equal to a Dirac delta on the same atom. To prove the induction hypothesis, we first decompose the L^2 norm using Minkowski inequality, and control each term separately:

$$\mathbb{E}^{1/2}\left[\pi_{r+1,K}\phi - \pi_{r+1}\phi\right]^2 \leq \mathbb{E}^{1/2}\left[\pi_{r+1,K}\phi - (\mathrm{prop}\, \pi_{r,K})\phi\right]^2 + \mathbb{E}^{1/2}\left[(\mathrm{prop}\, \pi_{r,K})\phi - \pi_{r+1}\phi\right]^2.$$

But by Corollary 8, the first term goes to zero; and by Lemma 10 and the induction hypothesis, the second term also goes to zero. □

References

Altekar, G., S. Dwarkadas, J. Huelsenbeck, and F. Ronquist (2004). Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. *Bioinformatics 20*, 407–415.

Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society B 72*(3), 269–342.

Ayres, D. L., A. Darling, D. J. Zwickl, P. Beerli, M. T. Holder, P. O. Lewis, J. P. Huelsenbeck, F. Ronquist, D. L. Swofford, M. P. Cummings, A. Rambaut, and M. A. Suchard (2012). Beagle: An application programming interface and high-performance computing library for statistical phylogenetics. *Systematic Biology 61*(1), 170–173.

Beaumont, M. A., W. Zhang, and D. J. Balding (2002). Approximate Bayesian computation in population genetics. *Genetics 162*, 2025–2035.

Bouchard-Côté, A., S. Sankararaman, and M. I. Jordan (2012). Phylogenetic inference via sequential Monte Carlo. *Systematic Biology 61*, 579–593.

Brown, W., E. Prager, A. Wang, and A. Wilson (1982). Mitochondrial DNA sequences of primates: Tempo and mode of evolution. *Journal of Molecular Evolution 18*(4), 225–239.

Cannone, J., S. Subramanian, M. Schnare, J. Collett, L. D'Souza, Y. Du, B. Feng, N. Lin, L. Madabusi, K. Muller, N. Pande, Z. Shang, N. Yu, and R. Gutell (2002). The comparative RNA web (CRW) site: An online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics 3*(1), 15.

Cheon, S. and F. Liang (2008). Phylogenetic tree construction using sequential stochastic approximation Monte Carlo. *Biosystems 91*(1), 94–107.

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications.* New York: Springer.

Del Moral, P., A. Doucet, and A. Jasra (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society B 68*(3), 411–436.

Doucet, A., N. de Freitas, and N. Gordon (2001). *Sequential Monte Carlo methods in practice.* Springer.

Doucet, A. and A. M. Johansen (2011). A tutorial on particle filtering and smoothing: fifteen years later. In *Handbook of Nonlinear Filtering.* Cambridge University Press.

Doucet, A., M. Pitt, G. Deligiannidis, and R. Kohn (2015). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*. To appear.

Drummond, A. and M. Suchard (2010). Bayesian random local clocks, or one rate to rule them all. *BMC biology 8*(1), 114.

Edgar, R. C. and S. Batzoglou (2006). Multiple sequence alignment. *Current Opinion in Structural Biology 16*(3), 368–373.

Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol. 17*, 368–376.

Felsenstein, J. (2003). *Inferring phylogenies.* Sinauer Associates.

Geweke, J. (2004). Getting it right. *Journal of the American Statistical Association 99*(467), 799–804.

Gilks, W. R. and C. Berzuini (2001). Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society B 63*(1), 127–146.

Görür, D., L. Boyles, and M. Welling (2012). Scalable inference on Kingman's coalescent using pair similarity. *Journal of Machine Learning Research 22*, 440–448.

Görür, D. and Y. W. Teh (2009). An efficient sequential Monte Carlo algorithm for coalescent clustering. In *Advances in Neural Information Processing Systems (NIPS).*

Griffiths, R. and S. Tavaré (1996). Monte Carlo inference methods in population genetics. *Math. Comput. Modelling 23*, 141–158.

Hobolth, A. and E. A. Stone (2009). Simulation from endpoint-conditioned, continuous-time Markov chains on a finite state space, with applications to molecular evolution. *Annals of Applied Statistics 3*(3), 1204–1231.

Höhna, S., M. Defoin-Platel, and A. Drummond (2008). Clock-constrained tree proposal operators in Bayesian phylogenetic inference. In *8th IEEE International Conference on BioInformatics and BioEngineering*, pp. 1–7.

Höhna, S. and A. J. Drummond (2012). Guided tree topology proposals for Bayesian phylogenetic inference. *Syst. Biol. 61*(1), 1–11.

Holmes, I. and W. J. Bruno (2001). Evolutionary hmms: a Bayesian approach to multiple alignment. *Bioinformatics 17*(9), 803–820.

Huelsenbeck, J. P. and F. Ronquist (2001). MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics 17*(8), 754–755.

Iorio, M. D. and R. C. Griffiths (2004). Importance sampling on coalescent histories. *Adv. Appl. Prob. 36*, 417–433.

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol. 16*, 111–120.

Kocher, T. D., J. A. Conroy, K. R. McKaye, J. R. Stauffer, and S. F. Lockwood (1995). Evolution of nadh dehydrogenase subunit 2 in east african cichlid fish. *Molecular phylogenetics and evolution 4*(4), 420–432.

Lakner, C., P. van der Mark, J. P. Huelsenbeck, B. Larget, and F. Ronquist (2008). Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Syst. Biol. 57*(1), 86–103.

Larget, B. and D. Simon (1999). Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. *Mol. Biol. Evol. 16*, 750–759.

Larkin, M., G. Blackshields, N. Brown, R. Chenna, P. McGettigan, H. McWilliam, F. Valentin, I. Wallace, A. Wilm, R. Lopez, J. Thompson, T. Gibson, and D. Higgins (2007). Clustal W and Clustal X version 2.0. *Bioinformatics 23*(21), 2947–2948.

Lee, A. and N. Whiteley (2014). Forest resampling for distributed sequential Monte Carlo. *arXiv:stat. 1406.6010.*

Lemey, P., A. Rambaut, A. J. Drummond, and M. A. Suchard (2009). Bayesian phylogeography finds its roots. *PLoS Computational Biology 5*(9), e1000520.

Liu, J. S. (2001). *Monte Carlo Strategies in Scientific Computing.* Springer.

Marjoram, P., J. Molitor, V. Plagnol, and S. Tavaré (2002). Markov chain Monte Carlo without likelihoods. *Proc. Nat. Acad. Sci. 100*, 15324–15328.

Morrison, D. A. (2006). Multiple sequence alignment for phylogenetic purposes. *Australian Systematic Botany 19*(6), 479–539.

Nasrallah, C. A., D. H. Mathews, and J. P. Huelsenbeck (2011). Quantifying the impact of dependent evolution among sites in phylogenetic inference. *Syst. Biol. 60*(1), 60–73.

Paul, J. S., M. Steinrücken, and Y. S. Song (2011). An accurate sequentially Markov conditional sampling distribution for the coalescent with recombination. *Genetics 187*, 1115–1128.

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

Redelings, B. D. and M. A. Suchard (2005). Joint Bayesian estimation of alignment and phylogeny. *Syst. Biol. 54*(3), 401–418.

Robinson, D. and L. Foulds (1981). Comparison of phylogenetic trees. *Mathematical Biosciences 53*, 131–147.

Ronquist, F. and J. P. Huelsenbeck (2003). Mrbayes 3: Bayesian phylogenetic inference under mixed models. *Bioinformatics 19*(12), 1572–1574.

Ronquist, F., M. Teslenko, P. van der Mark, D. L. Ayres, A. Darling, S. Hohna, B. Larget, L. Liu, M. A. Suchard, and J. P. Huelsenbeck (2012). MrBayes 3.2: Efficient Bayesian phylogenetic inference and model choice across a large model space. *Syst. Biol. 61*, 539–542.

Russell, S. and P. Norvig (2009). *Artificial Intelligence: A Modern Approach.* Prentice Hall.

Semple, C. and M. Steel (2003). *Phylogenetics.* Oxford.

Shapiro, B., A. Rambaut, and A. J. Drummond (2006). Choosing appropriate substitution models for the phylogenetic analysis of protein-coding sequences. *Mol. Biol. Evol. 23*(1), 7–9.

Stanley, R. P. (1986). *Enumerative Combinatorics. Volume I.* Cambridge University Press.

Suchard, M. A. and B. D. Redelings (2006). Bali-phy: simultaneous Bayesian inference of alignment and phylogeny. *Bioinformatics 22*(16), 2047–2048.

Tavaré, S. (1986). Some probabilistic and statistical problems in the analysis of DNA sequences. *Lectures on Mathematics in the Life Sciences 17*, 56–86.

Teh, Y. W., H. Daumé III, and D. M. Roy (2008). Bayesian agglomerative clustering with coalescents. In *Advances in Neural Information Processing Systems (NIPS).*

Thorne, J. L., H. Kishino, and I. S. Painter (1998). Estimating the rate of evolution of the rate of molecular evolution. *Mol. Biol. Evol. 15*(12), 1647–1657.

Tom, J. A., J. S. Sinsheimer, and M. A. Suchard (2010). Reuse, recycle, reweigh: Combating influenza through efficient sequential Bayesian computation for massive data. *Annals of Applied Statistics 4*, 1722–1748.

Yang, Z. (2006). *Computational Molecular Evolution.* Oxford University Press.

Yang, Z. and B. Rannala (1997). Bayesian phylogenetic inference using DNA sequences: a Markov chain Monte Carlo method. *Mol. Biol. Evol. 14*, 717–724.