# Efficient Continuous-Time Markov Chain Estimation

**Monir Hajiaghayi** MONIRH@CS.UBC.CA
Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

**Bonnie Kirkpatrick** BBKIRK@CS.MIAMI.EDU
Department of Computer Science, University of Miami, Coral Gables, FL 33124, United States

**Liangliang Wang** LIANGLIANG_WANG@SFU.CA
Department of Statistical and Actuarial Sciences, Simon Fraser University, Burnaby, BC V5A 1S6, Canada

**Alexandre Bouchard-Côté** BOUCHARD@STAT.UBC.CA
Statistics Department, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

## Abstract

Many problems of practical interest rely on Continuous-time Markov chains (CTMCs) defined over combinatorial state spaces, rendering the computation of transition probabilities, and hence probabilistic inference, difficult or impossible with existing methods. For problems with countably infinite states, where classical methods such as matrix exponentiation are not applicable, the main alternative has been particle Markov chain Monte Carlo methods imputing both the holding times and sequences of visited states. We propose a particle-based Monte Carlo approach where the holding times are marginalized analytically. We demonstrate that in a range of realistic inferential setups, our scheme dramatically reduces the variance of the Monte Carlo approximation and yields more accurate parameter posterior approximations given a fixed computational budget. These experiments are performed on both synthetic and real datasets, drawing from two important examples of CTMCs having combinatorial state spaces: string-valued mutation models in phylogenetics and nucleic acid folding pathways.

## 1. Introduction

Continuous-time Markov chains (CTMCs) play a central role in applications as diverse as queueing theory, phylogenetics, genetics, and models of chemical interactions (Huelsenbeck & Ronquist, 2001; Munsky & Khammash, 2006). The process can be thought of as a timed random walk on a directed graph where the countable, but potentially infinite, set of graph nodes are the values that the process can take on. There are probabilities of transition associated with the edges of the graph, and the holding time, or length of time between two transitions, is exponentially distributed with a rate depending on the current node. A path simulated from this random process is an ordered list of the nodes visited and the times at which they are reached.

In leveraging the modelling capabilities of CTMCs, the bottleneck is typically the computation of the *transition probabilities*: the conditional probability that a trajectory ends in a given end state, given a start state and a time interval. This computation involves the marginalization over the uncountable set of end-point conditioned paths. Although we focus on the Bayesian framework in this work, where the transition probabilities appear in Metropolis-Hastings ratios, the same bottleneck is present in the frequentist framework, where transition probabilities are required for likelihood evaluation. When the state space is small, exact marginalization can be done analytically via the matrix exponential. Unfortunately, this approach is not directly applicable to infinite state spaces, and is not computationally feasible in large state spaces because of the cubic running time of matrix exponentiation.

We propose an efficient Monte Carlo method to approach inference in CTMCs with weak assumptions on the state space. Our method can approximate transition probabilities as well as estimate CTMC parameters for this general class of processes. More precisely, we are interested in countably infinite state space CTMCs that satisfy the following two criteria. First, we require the construction of a certain type of potential on the state space. We describe this potential in more detail in Section 2, and show in Section 3

*Proceedings of the $31^{st}$ International Conference on Machine Learning*, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

that such potentials can be easily constructed even for complex models. Second, the CTMC should be explosion-free to avoid pathologies (i.e., we require that there is a finite number of transitions with probability one in any bounded time interval).

In contrast, classical uniformization methods assume that there is a fixed bound on all the rates (Grassmann, 1977), a much stronger condition than our explosion-free assumption. For example, in the first of the two application domains that we investigated, inference in string-valued CTMCs for phylogenetics, the models are explosion-free but do not have a fixed bound on the rates. Other approaches, based on performing Markov chain Monte Carlo (MCMC) with auxiliary variables, relax the bounded rate assumption (Rao & Teh, 2011; 2012), but they have a running time that depends linearly on the size of the state space in the sparse case and quadratically in the dense case.

Particle-based methods offer an interesting complementary approach, as they have a time complexity per particle that depends on the imputed *number of transitions between the two end points* instead of on the size of the state space.

In the simplest case, one can implement this idea using a proposal distribution equal to the generative process over paths initialized at the start point. The weight of a particle is then equal to one if the end point of the generated path coincides with observed end point, and zero otherwise. We call this proposal the *forward sampling* proposal. This idea can be turned into a consistent estimator of posterior distributions over parameters using pseudo-marginal methods (Beaumont, 2003; Andrieu & Roberts, 2009) (or in more complicated setups, particle MCMC methods (Andrieu et al., 2010)).

Unfortunately, the forward sampling method has two serious limitations. First, the requirement of imputing waiting times between each transition means that the proposal distribution is defined over a potentially high-dimensional continuous space. This implies that large numbers of particles are required in practice. Second, in problems where each state has a large number of successors, the probability of reaching the end state can become extremely small, which for example further inflates the number of particles required to obtain non degenerate Metropolis-Hastings ratios in particle MCMC (Andrieu et al., 2010) algorithms.

End point informed proposals over transitions and waiting times have been developed in previous work (Fan & Shelton, 2008), but this previous work is tailored to dynamic Bayesian models rather than to the combinatorial problems studied here. Our method greatly simplifies the development of end point informed proposals by marginalizing all continuous variables. There has also been work on related end-point conditioning problems in the rare event simulation literature (Juneja & Shahabuddin, 2006), but this previous work has focused on the discrete-time setting.

## 2. Methodology

For expositional purposes, we start by describing the simplest setup in which our method can be applied: computing the probability that a CTMC with known rate parameters occupies state $y \in \mathcal{X}$ at time $T$ given that it occupies state $x \in \mathcal{X}$ at time 0, where $\mathcal{X}$ is a countable set of states. The main contributions of this paper can be understood in this simple setup. We then show that our method can be extended to certain types of partial or noisy observations, to more than two observations organized as a time series or a tree (branching process), and to situations where some or all the parameters of the CTMC are unknown.

**Notation.** Let $\nu(x, y)$ denote the transition probability from state $x \in \mathcal{X}$ to state $y \in \mathcal{X}$ given that a state jump occurs (i.e. $\sum_{y:y\neq x} \nu(x, y) = 1, \nu(x, x) = 0$). Let $\lambda(x)$ denote the rate of the exponentially-distributed holding time at state $x$ ($\lambda : \mathcal{X} \to [0, \infty)$).[1] We only require efficient point-wise evaluation of $\lambda(\cdot), \nu(\cdot, \cdot)$ and efficient simulation from $\nu(x, \cdot)$ for all $x \in \mathcal{X}$. We start by assuming that $\nu$ and $\lambda$ are fixed, and discuss their estimation afterward. We define some notation for paths sampled from this process. Let $X_1, X_2, \dots$ denote the list of visited states with $X_i \neq X_{i+1}$, called the *jump chain*, and $H_1, H_2, \dots$, the list of corresponding *holding times*. The model is characterized by the following distributions: $X_{i+1}|X_i \sim \nu(X_i, \cdot)$, $H_i|X_i \sim F(\lambda(X_i))$, where $F(\lambda)$ is the exponential distribution CDF with rate $\lambda$. Given a start state $X_1 = x$, we denote by $\mathbb{P}_x$ the probability distribution induced by this model. Finally, we denote by $N$ the number of states visited, counting multiplicities, in the interval $[0, T]$, i.e. $(N = n) = (\sum_{i=1}^{n-1} H_i \leq T < \sum_{i=1}^{n} H_i)$.

**Overview of the inference method.** Using the simple setup introduced above, the problem we try to solve is to approximate $\mathbb{P}_x(X_N = y)$, which we approach using an importance sampling method. Each proposed particle consists of a sequence (a list of variable finite length) of states, $x^* = (x_1, \dots, x_n) \in \mathcal{X}^*$, starting at $x$ and ending at $y$. In other words, we marginalize the holding times, hence avoiding the difficulties involved with sequentially proposing times constrained to sum to the time $T$ between the end points.

Concretely, our method is based on the following elementary property, proved in the Supplement:

**Proposition 1.** *If we let* $\pi(x^*) = \gamma(x^*)/\mathbb{P}_x(X_N = y)$,

[1] Note that this is a reparameterization of the standard rate matrix $q_{x,y}$, with $q_{x,x} = -\lambda(x)$, and $q_{x,y} = \lambda(x)\nu(x, y)$ for $x \neq y$.

*where,*

$$\gamma(x^*) = \mathbf{1}(x_n = y)\left(\prod_{i=1}^{n-1} \nu(x_i, x_{i+1})\right) \times \tag{1}$$
$$\mathbb{P}\left(\sum_{i=1}^{n-1} H_i \le T < \sum_{i=1}^{n} H_i \middle| X^* = x^*\right),$$

*where the $H_i$'s are sampled according to $F(\lambda(X_i))$ independently given $X^* = (X_1, \cdots, X_N)$ and where $n = |x^*|$, then $\pi$ is a normalized probability mass function.*

As our notation for $\gamma, \pi$ suggests, we use this result as follows (see Algorithm 1 in the Supplement for details). First, we define an importance sampling algorithm that targets the unnormalized density $\gamma(x^*)$ via a proposal $\tilde{\mathbb{P}}(X^* = x^*)$. Let us denote the $k$-th particle produced by this algorithm by $x^*(k) \in \mathcal{X}^*$, $k \in \{1, \ldots, K\}$, where the number of particles $K$ is an approximation accuracy parameter. Each of the $K$ particles is sampled independently according to the proposal $\tilde{\mathbb{P}}$. Second, we exploit the fact that the sample average of the unnormalized importance weights $w(x^*(k)) = \gamma(x^*(k))/\tilde{\mathbb{P}}(X^* = x^*(k))$ generated by this algorithm provide a consistent estimator for the normalizer of $\gamma$. Finally, by Proposition 1, this normalizer coincides with the quantity of interest here, $\mathbb{P}_x(X_N = y)$. The only formal requirement on the proposal is that $\mathbb{P}_x(X^* = x^*) > 0$ should imply $\tilde{\mathbb{P}}(X^* = x^*) > 0$. However, to render this algorithm practical, we need to show that it is possible to define efficient proposals, in particular proposals such that $\mathbb{P}_x(X^* = x) > 0$ if and only if $\tilde{\mathbb{P}}(X^* = x^*) > 0$ (in order to avoid particles of zero weight). We also need to show that $\gamma$ can be evaluated point-wise efficiently, which we establish in Proposition 2.

**Proposal distributions.** Our proposal distribution is based on the idea of simulating from the jump chain, i.e. of sequentially sampling from $\nu$ until $y$ is reached. However this idea needs to be modified for two reasons. First, (1) since the state is countably infinite in the general case, there is a potentially positive probability that the jump chain sampling procedure will never hit $y$. Even when the state is finite, it may take an unreasonably large number of steps to reach $y$. Second, (2) forward jump chain sampling, assigns zero probability to paths visiting $y$ more than once.

We address (1) by using a user-specified *potential* $\rho^y : \mathcal{X} \to \mathbb{N}$ centred at the target state $y$ (see Supplement for the conditions we impose on $\rho^y$). For example we used the Levenshtein (i.e., minimum number of insertion, deletion, and substitution required to change one string into another) and Hamming distances for the string evolution and RNA kinetics applications respectively. Informally, the fact that this distance favors states which are closer to $y$ is all that we need to bias the sampling of our new jump process towards visiting $y$.

How do we bias the proposal sampling of the next state? Let $D(x) \subset \mathcal{X}$ be the set of states that decrease the potential from $x$. The proposed jump-chain transitions are chosen with probability

$$\tilde{\mathbb{P}}(X_{i+1} = x_{i+1} | X_i = x_i) = \tag{2}$$
$$(\alpha^y_{x_i})\left(\frac{\nu(x_i, x_{i+1})\mathbf{1}\{x_{i+1} \in D(x_i)\}}{\sum_{x'_{i+1} \in D(x_i)} \nu(x_i, x'_{i+1})}\right)$$
$$+ (1 - \alpha^y_{x_i})\left(\frac{\nu(x_i, x_{i+1})(1 - \mathbf{1}\{x_{i+1} \in D(x_i)\})}{\sum_{x'_{i+1} \notin D(x_i)} \nu(x_i, x'_{i+1})}\right).$$

We show in the Supplement that under weak conditions, we will hit target $y$ in finite time with probability one if we pick $\alpha^y_x = \max\{\alpha, \sum_{x'_{i+1} \in D(x_i)} \nu(x_i, x'_{i+1})\}$. Here $\alpha > 1/2$ is a tuning parameter. We discuss the sensitivity of this parameter, as well as strategies for setting it in Section 3.2.

Point (2) can be easily addressed by simulating a geometrically-distributed number of excursions where the first excursion starts at $x$, and the others at $y$, and each excursion ends at $y$. We let $\beta$ denote the parameter of this geometric distribution, a tuning parameter, which we also discuss at the end of Section 3.2.

**Analytic jump integration.** In this section, we describe how the unnormalized density $\gamma(x^*)$ defined in Equation (1) can be evaluated efficiently for any given path $x^* \in \mathcal{X}^*$.

It is enough to show that we can compute the following integral for $H_i | X^* \sim F(\lambda(X_i))$ independently conditionally on $X^*$:

$$\mathbb{P}\left(\sum_{i=1}^{n-1} H_i \le T < \sum_{i=1}^{n} H_i \middle| X^* = x^*\right) = \tag{3}$$
$$\int \cdots \int_{h_i > 0 : \sum_{i=1}^{n} h_i = T} g(h_1, h_2, \ldots, h_n)\, \mathrm{d}h_1\, \mathrm{d}h_2 \ldots \mathrm{d}h_n,$$

where $\quad g(h_1, h_2, \ldots, h_n) =$

$$\left\{\prod_{i=1}^{n-1} f(h_i; \lambda(x_i))\right\}(1 - F(h_n; \lambda(x_n))),$$

and where $f$ is the exponential density function. Unfortunately, there is no efficient closed form for this high-dimensional integral, except for special cases (for example, if all rates are equal) (Akkouchi, 2008). This integral is related to those needed for computing convolutions of non-identical independent exponential random variables. While there exists a rich literature on numerical approximations to these convolutions, these methods either add assumptions on the rate multiplicities (e.g. $|\{\lambda(x_1), \ldots, \lambda(x_N)\}| = |(\lambda(x_1), \ldots, \lambda(x_N))|$), or are computationally intractable (Amari & Misra, 1997).

We propose to do this integration using the construction of an auxiliary, finite state CTMC with a $n+1$ by $n+1$ rate matrix $\check{Q}$ (to be defined shortly). The states of $\check{Q}$ correspond to the states visited in the path $(x_1, x_2, \ldots, x_n)$ with multiplicities plus an extra state $s_{n+1}$. All off-diagonal entries of $\check{Q}$ are set to zero with the exception of transitions going from $x_i$ to $x_{i+1}$, for $i \in \{1, \dots, n\}$. More specifically, $\check{Q}$ is

$$\begin{bmatrix} -\lambda(x_1) & \lambda(x_1) & 0 & \cdots & 0 & 0 \\ 0 & -\lambda(x_2) & \lambda(x_2) & \cdots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \cdots & -\lambda(x_n) & \lambda(x_n) \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}. \quad (4)$$

This construction is motivated by the following property which is proven in the Supplement:

**Proposition 2.** *For any finite proposed path $(x_1, x_2, \dots, x_n)$, if $\check{Q}$ is defined as in Equation (4), then*

$$\big(\exp(T\check{Q})\big)_{1,n} = \mathbb{P}\left(\sum_{i=1}^{n-1} H_i \le T < \sum_{i=1}^{n} H_i \,\middle|\, X^* = x^*\right) \quad (5)$$

*where* $\exp(A)$ *denotes the matrix exponential of $A$.*[2]

**Trees and sequences of observation.** We have assumed so far that the observations take the form of a single branch with the state fully observed at each end point. To approach more general types of observations, for example a series of partially observed states, or a phylogenetic tree with observed leaves, our method can be generalized by replacing the importance sampling algorithm by a sequential Monte Carlo (SMC) algorithm. We focus on the tree case in this work which we describe in detail in Section 3, but we outline here how certain partially observed sequences can also be approached to start with something simpler.

Consider a setup where the observation at time $T_i$ is a set $A_i \subset \mathcal{X}$ (i.e. we condition on $(X(T_i) \in A_i; i \in \{1, \dots, m\})$ which arises for example in (Saeedi & Bouchard-Côté, 2011)). In this case, the importance sampling algorithm described in the previous section can be used at each iteration, with the main difference being that the potential $\rho$ is modified to compute a distance to a set $A_i$ rather than a distance to a single point $y$. See Algorithm 5 in the Supplement for details.

For other setups, the construction of the potential is more problem-specific. One limitation of our method arises when the observations are only weakly informative of the hidden state. We leave these difficult instances for future work and reiterate that many interesting and challenging problems fall within the reach of our method (for example, the computational biology problems presented in the next section).

**Parameter estimation.** So far, we have assumed that the parameters $\nu$ and $\lambda$ governing the dynamics of the process are known. We now consider the case where we have a parametric family with unknown parameter $\theta \in \Theta$ for the jump transition probabilities $\nu_\theta$ and for the holding time mean function $\lambda_\theta$. We denote by $\mathbb{P}_{x,\theta}$ the induced distribution on paths and by $p$ a prior density on $\theta$. To approximate the posterior distribution on $\theta$, we use pseudo-marginal methods (Beaumont, 2003; Andrieu & Roberts, 2009) in the fixed end-point setup and particle MCMC methods (Andrieu et al., 2010) in the sequences and trees setup. While our algorithm can be combined with many variants of these pseudo-marginal and particle MCMC methods, in this section, for simplicity we describe the grouped independence Metropolis-Hastings (GIMH) approach.

At each MCMC iteration $t$, the algorithm keeps in memory a pair $x^{(t)} = (\theta^{(t)}, \hat{Z}^{(t)}_{\theta^{(t)}})$ containing a current parameter $\theta^{(t)}$ and an approximation $\hat{Z}^{(t)}_{\theta^{(t)}}$ of the marginal probability of the observations[3] $\mathcal{Y}$ given $\theta^{(t)}$, $\hat{Z}^{(t)}_{\theta^{(t)}} \approx \mathbb{P}_{\theta^{(t)}}(\mathcal{Y})$. This approximation is obtained from the algorithm described in the previous subsections. Even though this approximation is inexact for a finite number of particles, the GIMH sampler is still guaranteed to converge to the correct stationary distribution (Andrieu et al., 2010).

The algorithm requires the specification of a proposal density on parameter $q(\theta'|\theta)$. At the beginning of each MCMC iteration, we start by proposing a parameter $\theta^*$ from this proposal $q$. We then use the estimate $\hat{Z}_{\theta^*}$ of $\mathbb{P}_{\theta^*}(\mathcal{Y})$ given by the average of the weights $w(x^{*(t)}(k))$ to form the ratio $r(\theta^{(t)}, \theta^*)$, below, where $k$ is the index for particles. We accept $(\theta^*, \hat{Z}_{\theta^*})$, or remain as before, according to a Bernoulli distribution with probability $\min\{1, r(\theta^{(t)}, \theta^*)\}$ where

$$r(\theta^{(t)}, \theta^*) = \frac{p(\theta^*)}{p(\theta^{(t)})} \frac{\hat{Z}_{\theta^*}}{\hat{Z}^{(t)}_{\theta^{(t)}}} \frac{q(\theta^{(t)}|\theta^*)}{q(\theta^*|\theta^{(t)})}.$$

See Algorithm 4 in the Supplement for details.

## 3. Numerical examples

### 3.1. String-valued evolutionary models

Molecular evolutionary models, central ingredients of modern phylogenetics, describe how biomolecular sequences (RNA, DNA, or proteins) evolve over time via a CTMC where jumps are character substitutions, insertions and deletions (indel), and states are biomolecular sequences. Previous work focused on the relatively restricted range of evolutionary phenomena for which computing marginal probabilities of the form $\mathbb{P}_x(X_N = y)$ can

[2] Multiplicities of the rates in $\check{Q}$ greater than one will break diagonalization-based methods of solving $\exp(T\check{Q})$, but other efficient matrix exponentiation methods such as the squaring and scaling method are still applicable in these cases.

[3] For example, in the single branch setting, $\mathcal{Y} = (X_1 = x, X_N = y)$.

be done exactly.

In particular, we are not aware of existing methods for doing Bayesian inference over context-dependent indel models, i.e. models where insertions and deletions can depend on flanking characters. Modelling the context of indels is important because of a phenomenon called *slipped strand mispairing* (SSM), a well known explanation for the evolution of repeated sequences (Morrison, 2009; Hickey & Blanchette, 2011; Arribas-Gil & Matias, 2012). For example, if a DNA string contains a substring of *"TATATA"*, the non-uniform error distribution in DNA replication is likely to lead to a long insertion of extra *"TA"* repeats.

**Model.** In order to describe our SSM-aware model, it is enough to describe its behavior on a single branch of a tree, say of length $T$. Each marginal variable $X_t$ is assumed to have the countably infinite domain of all possible molecular sequences. We define $\lambda(x)$, as a function of the mutation rate per base $\theta_{\text{sub}}$, the global point insertion (i.e. insertion of a single nucleotide) rate $\lambda_{\text{pt}}$, the point deletion rate per base $\mu_{\text{pt}}$, the global SSM insertion rate $\lambda_{\text{SSM}}$ (which copies a substring of length up to three to the right of that substring), and the SSM deletion rate per valid SSM deletion location $\mu_{\text{SSM}}$ (deletion of a substring of length up to three at the right of an identical substring):

$$\lambda(x) = m(x)\theta_{\text{sub}} + \lambda_{\text{pt}} + m(x)\mu_{\text{pt}} + \lambda_{\text{SSM}} + k(x)\mu_{\text{SSM}}$$

where $m(x)$ is the length of the string $x$ and $k(x)$ is the number of valid SSM deletion locations in $x$. We denote these evolutionary parameters by $\theta = (\theta_{\text{sub}}, \lambda_{\text{pt}}, \mu_{\text{pt}}, \lambda_{\text{SSM}}, \mu_{\text{SSM}})$. The jump transition probabilities from $x$ to $x'$ are obtained by normalizing each of the above rates. For example the probability of deleting the first character given that there is a change from sequence $x$ is $\mu_{\text{pt}}/\lambda(x)$. Note that since the total insertion rate does not depend on the length of the string, the process is explosion-free for all $\theta$. At the same time, there is no fixed bound on the deletion rate, ruling out classical methods such as uniformization or matrix exponentiation.

**Validation on a special case.** Before moving on to more complex experiments, we started with a special case of our model where the true posterior can be computed numerically. This is possible by picking a single branch, and setting $\lambda_{\text{SSM}} = \mu_{\text{SSM}} = 0$, in which case the process reduces to a process for which analytic calculation of $\mathbb{P}_{x,\theta}(X_N = y)$ is tractable (Bouchard-Côté & Jordan, 2012). We fixed the substitution parameter $\theta_{\text{sub}}$, and computed as a reference the posterior by numerical integration on $\lambda_{\text{pt}}, \mu_{\text{pt}}$ truncated to $[0, 3]^2$ and using $100^2$ bins.

We generated 200 pairs of sequences along with their sequence alignments[4], with $T = 3/10, \lambda = \lambda_{\text{pt}} = 2, \mu = \mu_{\text{pt}} = 1/2$ and held out the mutations and the true value of parameters $\lambda$ and $\mu$. We put an exponential prior with rate $1.0$ on each parameter. We approximate the posterior using our method, initializing the parameters to $\lambda = \mu = 1$, using $\alpha = 2/3, \beta = 19/20$, 64 particles, and a proposal $q$ over parameters given by the multiplicative proposal of Lakner et al. (Lakner et al., 2008). We show the results of $\lambda$ in Figure 1a and in the Supplement Figure: results of parameter $\mu$. In both cases the posterior approximation is shown to closely mirror the numerical approximation. The evolution of the Monte Carlo quartiles computed on the prefixes of Monte Carlo samples also shows that the convergence is rapid (Figure 1b).

Next, we compared the performance of a GIMH algorithm computing $\hat{Z}_{\theta(t)}$ using our method, with a GIMH algorithm computing $\hat{Z}^{(t)}_{\theta(t)}$ using forward sampling. We performed this comparison by computing the Effective Sample Size (ESS) after a fixed computational budget (3 days). For the parameter $\lambda$, our method achieves an ESS of 1782.7 versus 44.6 for the forward sampling GIMH method; for the parameter $\mu$, our method achieves an ESS of 6761.2 versus 90.2 for the forward sampling GIMH method. In those experiments, we used 100 particles per MCMC step, but we also tried different values and observed the same large gap favoring our method (see Supplement Figure: Varying the number of particles per MCMC step).

We also generated three datasets based on branch lengths from $\{0.15, 0.3, 0.6\}$, each containing 10 pairs of sequences $(x, y)$ along with their sequence alignments and estimated the transition probability $\mathbb{P}_x(Y_N = y)$ using our method (denoted Time-Integrated Path Sampling, TIPS), and using forward simulation (denoted forward sampling, FS). We compared the two methods in Figure 1c by looking at the absolute log error of the estimate $\hat{p}$, error($\hat{p}$) $= |\log \hat{p} - \log \mathbb{P}_x(X_N = y)|$. We performed this experiment with a range of numbers of particles, $\{2^1, 2^2, \ldots, 2^{20}\}$ and plotted the relative errors as a function of the wall clock time needed for each approximation method. We also computed the variances of the importance weights for specific alignments and compared these variances for FS and TIPS (see Figure 1d). We observed that the variances were consistently two orders of magnitudes lower with our method compared to FS.

**Tree inference via SMC.** We now consider the general case, where inference is on a phylogenetic tree, and the SSM parameters are non-zero. To do this, we use existing SMC algorithms for phylogenetic trees (Teh et al., 2008; Bouchard-Côté et al., 2012; Wang, 2012), calling our algorithm at each proposal step. We review phylogenetic inference in the Supplement where we also give in Algorithm 6 the details of how we combined our method with phyloge-

[4] A sequence alignment is a graph over the observed nucleotides linking the nucleotides that have a common ancestor.

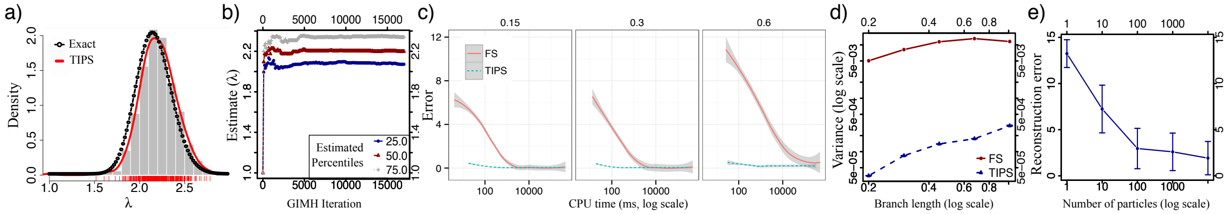


*Figure 1.* a) Validation of the posterior estimate on the Poisson Indel Process dataset. The histogram and the density estimate in red is obtained from 35,000 GIMH iterations; the black curve is obtained by numerical integration. The generating value is $\lambda_{\text{pt}} = 2$. b) Convergence of percentiles computed from prefixes of the GIMH output. c) Relative errors on the transition probabilities for branch lengths from $\{0.15, 0.3, 0.6\}$. d) Estimated variance of the weights. e) Reconstruction error on tree distances.

netic SMC.

To evaluate our method, we sampled 10 random trees from the coalescent on 10 leaves, along each of which we simulated 5 sets of molecular sequences according to our evolutionary model. We used the following parameters: SSM length=3, $\theta_{sub} = 0.03$, $\lambda_{pt} = 0.05$, $\mu_{pt} = 0.2$, $\lambda_{SSM} = 2.0$, and $\mu_{SSM} = 2.0$. One subset of simulated data is shown in the Supplement Figure: Sequence Simulation. The unaligned sequences on leaves are used for tree reconstruction using our method. We summarized the posterior over trees using a consensus tree optimizing the posterior expected pairwise distances (Felsenstein, 1981). Figure 1e shows tree distances using the partition metric (Felsenstein, 2003) between generated trees and consensus trees reconstructed using our evolutionary model. The tree distance decreases as the number of particles increases, and a reasonable accuracy is obtained with only 100 particles, suggesting that it is possible to reconstruct phylogenies from noisy data generated by complex evolutionary mechanisms.

### 3.2. RNA folding pathways

Nucleic acid folding pathways predict how RNA and DNA molecules fold in on themselves via intra-molecular interactions. The state space of our stochastic process that describes folding is the set of all folds, or secondary structures, of the nucleic acid molecule which is a combinatorial object. For RNA molecules, the secondary structure is the primary determiner of RNA function. For DNA its fold can help determine gene transcription rates. Understanding the folding pathways can be useful for designing nano-scale machines that have potential health applications (Venkataraman et al., 2010). For these reasons, it is often useful in applications to get an accurate estimate of the probability that a nucleic acid molecule beginning in one secondary structure, $x$, will transition in the given time, $T$, to a target structure, $y$. This is called the transition probability, and it is typically computed by either solving a system of linear differential equations or by computing a matrix exponential of a large matrix. Here, we will use our method (denoted as TIPS) to approximate these transition probabilities.

**Model.** An RNA fold can be characterized by a set of base pairs, either C-G, A-U, or G-U, each of which specifies the sequence positions of the two bases involved in the pairing. We will default the discussion to RNA sequences where we are interested in pseudo-knot-free RNA structures. These secondary structures can be represented as a planar circle graph with the sequence arrayed along a circle and non-crossing arcs between positions of the sequence which are base paired. Here, we will use structure to mean secondary structure. The folding of a molecule into secondary structures happens in a dynamic fashion.

In the pathway model we consider, successive structures $X_i$ and $X_{i+1}$ must differ by exactly one base pair. Let $X_1 = x$ and $X_N = y$ where $x$ is the given start structure and $y$ is the given final structure. See for example Figure 5 of the Supplement, where a folding path is given for a short RNA (holding times not shown) with $x$ being the unfolded state and $y$ being the Minimum Free Energy (MFE) structure.

To formalize the folding pathway, we need to introduce the generator matrix, $Q$. This matrix contains an entry for every possible pair of secondary structures. The Kawasaki rule gives the rate of the probabilistic process moving from structure $x$ to structure $x'$ as $\lambda(x)\nu(x, x') = \exp\left(E(x) - E(x')\right)/(kT)$ if $x' \in R(x)$, and zero otherwise where $E(x)$ is the energy of structure $x$, $R(x)$ is the set of secondary structures within one base pair of structure $x$ and $k$ is the Boltzmann constant. When given a nucleic acid sequence of $m$ bases, there are at most $O(3^m)$ secondary structures that can be created from it, making the size of the generator matrix exponential in the sequence length. This model was described by Flamm et al. (Flamm et al., 2000).

**Results.** In this section, we compare the accuracy of the transition probability estimates given by our method (TIPS) to those obtained by forward sampling method (FS) which is still widely used in the field of RNA folding pathways (Flamm et al., 2000; Schaeffer, 2012). We used the RNA molecules shown in Supplement Table: Biological RNA Sequences.

For each method (TIPS and FS) and molecule, we first approximated the probability $\mathbb{P}_x(X_N = y)$ that beginning in its unfolded structure $x$, the molecule would end, after folding time $T$, in its MFE structure $y$. We then computed, as a reference, the probability of this transition using an expensive matrix exponential. Computing the matrix exponential on the full state space was only possible for the RNAs of no more than 12 nucleotides. For the longer RNAs, we restricted the state space to a connected subset $S$ of secondary structures (Kirkpatrick et al., 2013). While our method scales to longer RNAs, we wanted to be able to compare against forward sampling and to the true value obtained by matrix exponentiation.

We ran the experiments with a range of number of particles, $\{5^1, 5^2, \cdots, 5^6\}$, for 30 replicates on folding times from $\{0.125, 0.25, \cdots, 8\}$. Here, similarly to the previous example, we compare the performance of the two methods by looking at the absolute log error of the estimate $\hat{p}$ (i.e., error($\hat{p}$) $= |\log \hat{p} - \log \mathbb{P}_x(X_N = y)|$) over all replicates. The parameters used for the TIPS method are as follows: $\alpha = \frac{2}{3}$ and $\beta = \max(0.25, 1 - \frac{T}{16})$ where $T$ is the specified folding time interval.

Figures 2a, 2d show the performance of the FS and TIPS methods on selective folding times, $\{0.25, 1, 4\}$. Figures 2b, 2e show the CPU times (in milliseconds) corresponding to the minimum number of particles required to satisfy the certain accuracy level, $I = \{\hat{p} : \text{error}(\hat{p}) < 1.0\}$ on all the folding times. Supplement Figure: Performance vs. folding time shows similar plots for two other RNA molecules.

The variances of FS and TIPS weights, for $5^6 = 15625$ particles, are also computed and compared on different folding times (see Figures 2c, 2f).

The graphs show that our novel method (TIPS) outperforms FS in estimating the probability of transition from $x$ to $y$ in shorter folding times, since it needs many fewer particles (and correspondingly faster CPU times) than FS to be able to precisely estimate the probability. For instance, for the RNA21 molecule with folding time 0.25, FS cannot satisfy the accuracy level $I$, given above, even with 15625 particles, however TIPS only needs 5 particles with 16 ms of CPU time to satisfy the same accuracy level. Similarly, the variance of our method is smaller by a larger margin (note that the variance is shown in log scale in Figures 2c, 2f).

For longer folding times in Figure 2, the performance of the TIPS and FS methods would be comparable (in terms of the obtained errors and CUP times) slightly in favour of forward sampling. For example, for the HIV23 molecule with folding time 4.0, TIPS and FS require 5 and 25 particles, and CPU times, 12 ms and 5 ms, respectively to satisfy $I$.

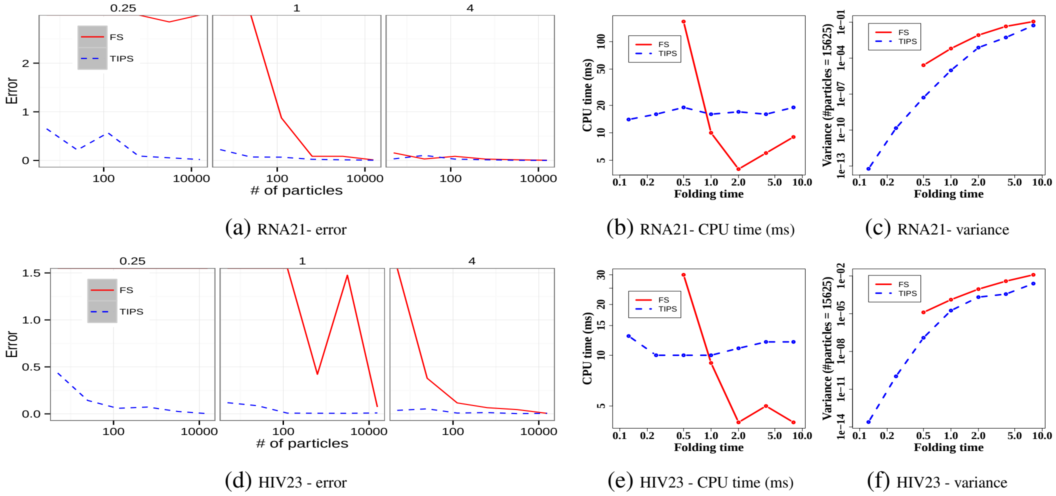
One caveat of these results is that in contrast to the phylogenetic setup, where TIPS was not sensitive to a range of values of the tuning parameters $\alpha, \beta$, it was more sensitive to these tuning parameters in the RNA setup. See Supplement Figure: Tuning parameter $\alpha$. We believe that the behavior of our method is more sensitive to $\alpha, \beta$ in the RNA case because the sampled jump chains are typically longer. Intuitively, for longer folding times, the transition probabilities are more influenced by the low probability paths, as these low probability paths comprise a greater percent of all possible paths. This means that any setting of $\alpha$ that heavily biases the sampled paths to be from the region just around $x$ and $y$ will need to sample a large number of paths in order to approximate the contribution of paths with a low probability. This situation is analogous to the well-known problems in importance sampling of mismatches between the proposal and actual distributions. Similar sampling considerations apply to parameter $\beta$ which controls the number of excursions from $y$. If $\beta$ is too restrictive, again, paths will be sampled that do not well reflect the actual probability of excursions. Parameter tuning is therefore an important area of future work. It might be possible to use some automated tuners (Hutter et al., 2009; Wang et al., 2013) or to approach the problem by essentially creating mixtures of proposals each with its own tuning parameters.

At the same time, note that the reason why FS can still perform reasonably well for longer folding times is that we picked the final end point to be the MFE, which has high probability under the stationary distribution. For low probability targets, FS will often fail to produce even a single hitting trajectory, whereas each trajectory sampled by our method will hit the target by construction.

## 4. Conclusion

We have presented an efficient method for approximating transition probabilities and posterior distributions over parameters in countably infinite CTMCs. We have demonstrated on real RNA molecules that our method is competitive with existing methods for estimating the transition probabilities which marginalize over folding pathways and provide a model for the kinetics of a single strand of RNA interacting chemically with itself. We have also shown, using a realistic, context-dependent indel evolutionary process, that the posterior distributions approximated by our method were accurate in this setting.

What makes our method particularly attractive in large or countably infinite state space CTMCs is that our method's running time per particle is independent of the size of the state space. The running time does depend cubically on the



(a) RNA21- error

(b) RNA21- CPU time (ms)

(c) RNA21- variance

(d) HIV23 - error

(e) HIV23 - CPU time (ms)

(f) HIV23 - variance

*Figure 2.* Performance of our method (TIPS) and forward sampling (FS) on RNA21 and HIV23 molecules with their *subset* state space. The relative errors of the estimates vs. folding times, $\{0.25, 1, 4\}$, are shown (left) along with the CPU times corresponding to the minimum number of particles required to satisfy the accuracy level $I$ in milliseconds (middle) and the variance of TIPS and FS estimations (right) on folding times, $\{0.125, 0.25, \cdots, 8\}$.

number of imputed jumps, so we expect that our method will be most effective when the typical number of transitions between two observations or imputed latent state is moderate (no more than approximately a thousand with current architectures). The distribution of the jump chain should also be reasonably concentrated to ensure that the sampler can proceed with a moderate number of particles. We have shown two realistic examples where these conditions are empirically met.

## Acknowledgment

This work was partially funded by an NSERC Discovery Grant and a Google Faculty Award. Computing was supported by WestGrid.

## References

Akkouchi, M. . On the convolution of exponential distributions. *Journal of the Chungcheong Mathematical Society*, 21(4):502–510, 2008.

Amari, S. and Misra, R. . Closed-form expressions for distribution of sum of exponential random variables. *IEEE Transactions on Reliability*, 46(4):519–522, 1997.

Andrieu, C. and Roberts, G. O. . The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

Andrieu, C. , Doucet, A. , and Holenstein, R. . Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc. B*, 72(3):269–342, 2010.

Arribas-Gil, A. and Matias, C. . A context dependent pair hidden Markov model for statistical alignment. *Statistical Applications in Genetics and Molecular Biology*, 11(1):1–29, 2012.

Beaumont, M. A. . Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–1160, 2003.

Bouchard-Côté, A. and Jordan, M. I. . Evolutionary inference via the Poisson indel process. *Proc. Natl. Acad. Sci.*, 10.1073/pnas.1220450110, 2012.

Bouchard-Côté, A. , Sankararaman, S. , and Jordan, M. I. . Phylogenetic inference via sequential Monte Carlo. *Syst. Biol.*, 61: 579–593, 2012.

Fan, Y. and Shelton, C. . Sampling for approximate inference in continuous time Bayesian networks. In *Tenth International Symposium on Artificial Intelligence and Mathematics*, 2008.

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

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

Flamm, C. , Fontana, W. , Hofacker, I. , and Schuster, P. . RNA folding at elementary step resolution. *RNA*, 6:325–338, 2000.

Grassmann, W. K. . Transient solutions in Markovian queueing systems. *Computers and Operations Research*, 4:47–100, 1977.

Hickey, G. and Blanchette, M. . A probabilistic model for sequence alignment with context-sensitive indels. *Journal of Computational Biology*, 18(11):1449–1464, 2011. doi: doi: 10.1089/cmb.2011.0157.

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

Hutter, F. , Hoos, H. H. , Leyton-Brown, K. , and Stützle, T. . ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306, October 2009.

Juneja, S. and Shahabuddin, P. . *Handbooks in Operations Research and Management Science*, volume 13, chapter Rare-Event Simulation Techniques: An Introduction and Recent Advances, pp. 291–350. Elsevier, 2006.

Kirkpatrick, B. , Hajiaghayi, M. , and Condon, A. . A new model for approximating RNA folding trajectories and population kinetics. *Computational Science & Discovery*, 6, January 2013.

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

Morrison, D. A. . Why would phylogeneticists ignore computerized sequence alignment? *Syst. Biol.*, 58(1):150–158, 2009.

Munsky, B. and Khammash, M. . The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, 124(4):044104–1 – 044104–13, 2006.

Rao, V. and Teh, Y. W. . Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pp. 619–626, Corvallis, Oregon, 2011. AUAI Press.

Rao, V. and Teh, Y. W. . MCMC for continuous-time discrete-state systems. *NIPS*, 2012.

Saeedi, A. and Bouchard-Côté, A. . Priors over Recurrent Continuous Time Processes. *NIPS*, 24:2052–2060, 2011.

Schaeffer, J. M. . The multistrand simulator: Stochastic simulation of the kinetics of multiple interacting dna strands. Master's thesis, California Institute of Technology, 2012.

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

Venkataraman, S. , Dirks, R. M. , Ueda, C. T. , and Pierce, N. A. . Selective cell death mediated by small conditional RNAs. *Proc. Natl. Acad. Sci.*, 107(39):16777–16782, 2010. doi: 10.1073/pnas.1006377107. URL `http://www.pnas.org/content/107/39/16777.abstract`.

Wang, L. . *Bayesian Phylogenetic Inference via Monte Carlo Methods*. PhD thesis, The University Of British Columbia, August 2012.

Wang, Z. , Mohamed, S. , and de Freitas, N. . Adaptive Hamiltonian and Riemann Monte Carlo samplers. In *International Conference on Machine Learning (ICML)*, 2013.