Supplementary Document

Monir Hajiaghayi
Department of Computer Science
University of British Columbia
monirh@cs.ubc.ca

Bonnie Kirkpatrick
Department of Computer Science
University of Miami
bbkirk@cs.miami.edu

Liangliang Wang
Department of Statistical and Actuarial Sciences
Simon Fraser University
liangliang_wang@sfu.ca

Alexandre Bouchard-Côté
Statistics Department
University of British Columbia
bouchard@stat.ubc.ca

This Supplement contains the proofs and pseudo-codes of the methods referenced in the **Methodology** section of the paper, as well as supplemental figures and tables for the **Numerical examples** section of the paper. The proofs involve the basic properties of our CTMC approach, and provide rigorous justification for the algorithms presented in the paper. The pseudo-codes illustrate the overviews of the steps of different methods used in the manuscript. The supplemental figures and tables also provide more extensive justification for the practical applicability of our method to both the phylogenetic and RNA settings.

1 Methodology

We prove here the results stated in the methodology section of the main paper.

1.1 Proposition 1

Proof. We have, for any $x^* = (x_1, x_2, \ldots, x_n) \in \mathcal{X}^*$,

$$\begin{aligned}
\mathbf{1}(x_n = y) &\frac{\left(\prod_{i=1}^{n-1} \nu(x_i, x_{i+1})\right) \mathbb{P}\left(\sum_{i=1}^{n-1} H_i \leq T < \sum_{i=1}^{n} H_i \middle| X^* = x^*\right)}{\mathbb{P}_x(X_N = y)} \\
&= \frac{\mathbf{1}(x_n = y)\mathbb{P}_x(X_1 = x_1, \ldots, X_n = x_n)}{\mathbb{P}_x(X_N = y)} \\
&\quad \times \mathbb{P}\left(\sum_{i=1}^{n-1} H_i \leq T < \sum_{i=1}^{n} H_i \middle| X_1 = x_1, \ldots, X_n = x_n\right) \\
&= \mathbf{1}(x_n = y)\frac{1}{\mathbb{P}_x(X_N = y)} \\
&\quad \times \mathbb{P}_x\left(\sum_{i=1}^{n-1} H_i \leq T < \sum_{i=1}^{n} H_i, X_1 = x_1, \ldots, X_n = x_n\right) \\
&= \mathbb{P}_x(X^* = x^* | X_N = y).
\end{aligned}$$

Since the right hand side is a conditional distribution,

$$\pi(x^*) = \mathbb{P}_x(X^* = x^* | X_N = y),$$

is indeed a normalized probability mass function. □

1.2 Proposal distributions

We show that the proposal defined in Equation (2) of the main paper hits the target end point y with probability one under the following assumptions:

1. The potential $\rho^y(x)$ takes the value zero if and only if $x = y$.
2. The potential always changes by one in absolute value for all proposed states:

$$\tilde{\mathbb{P}}_x\left(|\rho^y(X_2) - \rho^y(x)| = 1\right) = 1.$$

3. For all states $x \neq y$, there is always a way to propose a state that results in a decrease in potential:

$$\tilde{\mathbb{P}}_x(\rho^y(X_2) < \rho^y(x)) > 0 \quad \text{for all } x \in \mathcal{X}, x \neq y.$$

To simplify the notation, we will drop the y superscript for the remaining of this section.

To prove that the process always hits y, it is enough to show that the sequence $\rho(X_n)$ is a supermartingale, which in our case reduces to showing that $\mathbb{E}[\rho(X_2)|X_1] \leq \rho(X_1)$.

Note that the last condition ensures that the normalizer $\sum_{x_2' \in D(X_1)} \nu(X_1, x_2')$ is always positive, hence our expression of the proposal is always well defined. Note that technically, we should also require $\mathbb{P}_x(\rho^y(X_2) < \rho^y(x)) > 0$ to ensure that the second normalizer, $\sum_{x_2' \notin D(X_1)} \nu(X_1, x_2')$, is also positive, but if this is not the case, the proposal can always be replaced by ν in these cases without changing the conclusion of the result proven here.

Using the second condition, we have:

$$\begin{aligned}
\mathbb{E}[\rho(X_2)|X_1] &= \alpha_{X_1}(\rho(X_1) - 1) + (1 - \alpha_{X_1})(\rho(X_1) + 1) \\
&= 1 - 2\alpha_{X_1} + \rho(X_1) \\
&\leq \rho(X_1).
\end{aligned}$$

Finally, since the supermartingale $\rho(X_n)$ is non-negative, $\tilde{\mathbb{P}}(N < \infty) = 1$, we conclude that the process always hits y.

1.3 Proposition 2

Let $\check{X}_1, \check{X}_2, \ldots$ and $\check{H}_1, \check{H}_2, \ldots$ denote the states and holding times respectively of a CTMC with rate matrix $\check{Q}$. The states take values in $\{1, 2, \ldots, n+1\}$, and we let $\check{\mathbb{P}}_1$ denote the path probabilities under this process conditioned on starting at $X_1 = 1$. Let $\check{N}$ be defined similarly to N (the random number of states visited):

$$\begin{aligned}
(\check{N} = n) &= \left(\sum_{i=1}^{n-1} \check{H}_i \leq T < \sum_{i=1}^{n} \check{H}_i \right) \\
&= \left\{ \omega \in \check{\Omega} : \sum_{i=1}^{n-1} \check{H}_i(\omega) \leq T < \sum_{i=1}^{n} \check{H}_i(\omega) \right\}.
\end{aligned}$$

Here, $\check{\Omega}$ is an auxiliary probability space used to define the above random variables:

$$\begin{aligned}
\check{X}_i &: \check{\Omega} \to \mathcal{X} \\
\check{H}_i &: \check{\Omega} \to [0, \infty).
\end{aligned}$$

Proof. For all $i \in \{2, \dots, n+1\}$, only state $i-1$ has a positive rate of transitioning to state i, therefore $(\check{X}_i = j) \subset (\check{X}_{i-1} = j-1)$ for all j. Applying this inductively yields:

$$\begin{aligned}
\left(\exp(T\check{Q})\right)_{1,n} &= \check{\mathbb{P}}_1\left(\check{X}_{\check{N}} = n\right) \\
&= \check{\mathbb{P}}_1\left(\check{X}_{\check{N}} = n, \check{X}_{\check{N}-1} = n-1\right) \\
&\vdots \\
&= \check{\mathbb{P}}_1\left(\check{X}_{\check{N}} = n, \check{X}_{\check{N}-1} = n-1, \dots, \check{X}_1 = n - \check{N} + 1\right) \\
&= \check{\mathbb{P}}_1\left(\check{N} = n, \check{X}_1 = 1, \check{X}_2 = 2, \dots, \check{X}_n = n\right) \\
&= \check{\mathbb{P}}_1\left(\check{N} = n\right) \check{\mathbb{P}}_1\left(\check{X}_1 = 1, \check{X}_2 = 2, \dots, \check{X}_n = n | \check{N} = n\right) \\
&= \check{\mathbb{P}}_1\left(\check{N} = n\right) \prod_{i=2}^{n} \check{\mathbb{P}}(\check{X}_i = i | \check{X}_{i-1} = i-1) \\
&= \check{\mathbb{P}}_1\left(\check{N} = n\right) \\
&= \int \int \cdots \int_{h_i > 0 : h_1 + h_2 + \cdots + h_n = T} g(h_1, h_2, \dots, h_n)\, \mathrm{d}h_1\, \mathrm{d}h_2 \dots\, \mathrm{d}h_n.
\end{aligned}$$

□

In this part, for further clarity, we give the pseudo-codes of the proposed algorithms.

Our novel method (denoted as Time Integrated Path Sampling, TIPS) is demonstrated in Algorithm 1. This method uses the propose method introduced in Algorithm 2 in order to sample each particle, consisting of a sequence of states starting at x and ending at the target, y. The propose method also employs Algorithm 3 as a part of its structure to sample the particles hitting the target.

Algorithm 1 : **TIPS**(x, y, T)

$s \leftarrow 0$
for $k = 1, 2, \dots, K$ **do**
  $(L, \tilde{p}, p) \leftarrow$ **propose**$(x, \{y\})$
  $\check{Q} \leftarrow \check{Q}(L)$ {See Section 'Analytic jump integration'}
  $n = |L|$ {The length of the list of states L}
  $s \leftarrow s + p \times (\exp(T\check{Q}))_{1,n} / \tilde{p}$
end for
return s/K

Algorithm 2 : **propose**(x, A)

$(L, \tilde{p}, p) \leftarrow$**proposeHittingPath**$(x, A,$ false$)$
$n \sim$ Geo$(\cdot, \beta)$ {Geometric with support 1, 2, ...}
$\tilde{p} \leftarrow p \times$ Geo$(n; \beta)$ {Multiply by geometric probability mass function}
for $i = 2, 3, \dots, n$ **do**
  $x' \leftarrow$ **last**(L) {Last state visited in the list L}
  $(L', \tilde{p}', p') \leftarrow$**proposeHittingPath**$(x', A,$ true$)$
  $\tilde{p} \leftarrow \tilde{p} \times \tilde{p}'$
  $p \leftarrow p \times p'$
  $L \leftarrow L \circ L'$ {Concatenation of the two lists}
end for
return $(L, \tilde{p}, p)$

Algorithm 3 : **proposeHittingPath**(x, A, b)

$p \leftarrow 1$
$\tilde{p} \leftarrow 1$
$L \leftarrow$ **list**(x) {Creates a new list containing the point x}
for $i = 1, 2, \ldots$ **do**
  if $x \in A$ and (not(b) or $i > 1$) **then**
    return $(L, \tilde{p}, p)$
  end if
  $x'|x \sim \tilde{\mathbb{P}}(\cdot|X_{i-1} = x)$
  $\tilde{p} \leftarrow \tilde{p} \times \tilde{\mathbb{P}}(X_i = x'|X_{i-1} = x)$
  $p \leftarrow p \times \nu(x, x')$
  $L \leftarrow L \circ x'$
  $x \leftarrow x'$
end for

An overview of the parameter estimation method explained in the manuscript is also shown in Algorithm 4. In this algorithm, statio() computes the stationary probability mass function, for example Poisson in the Poisson Indel Process example Bouchard-Côté & Jordan (2012).

Algorithm 4 : **TIPS-parameters**$(\{x_i, y_i, T_i\})$

Initialization:
    Choose initial parameter $\theta^{(1)}$
    $z \leftarrow 0$
for $(t = 2, \cdots, T)$ **do**
  $\theta^* \sim q(\cdot|\theta^{(t-1)})$
  $z^* \leftarrow 1$
  for all data index i **do**
    $z^* \leftarrow z^* \times$ **TIPS**$(x_i, y_i, T_i) \times$ **statio**(x_i)
  end for
  $r \leftarrow \frac{p(\theta^*)}{p(\theta^{(t-1)})} \frac{z^*}{z} \frac{q(\theta^{(t-1)}|\theta^*)}{q(\theta^*|\theta^{(t-1)})}$
  Sample $u \sim \text{Uniform}(0, 1)$
  if $u < \min\{1, r\}$ **then**
    $z \leftarrow z^*$
    $\theta^{(t)} \leftarrow \theta^*$
  else
    $\theta^{(t)} \leftarrow \theta^{(t-1)}$
  end if
end for

Moreover, Algorithm 5, extends Saeedi & Bouchard-Côté (2011) and demonstrate the revised sequential Monte Carlo (SMC) method for approaching more general types of observations, for example a series of partially observed states, or a phylogenetic tree with observed leaves. Note that this algorithm is amenable to parallelization Lee et al. (2010); Jun et al. (2012).

2 Numerical examples

In this section we have figures and tables for both the phylogenetic and RNA settings. These give more detail on the results that we have obtained from applying our method to these CTMCs.

Algorithm 5 : **TIPS-SMC**(A_i, T_i)

Initialization:
  $w_{0,k} \leftarrow 1/K$
  $x_{0,k} \leftarrow$ nil
for $g = 1, 2, \ldots, G$ **do**
  for $k = 1, 2, \ldots, K$ **do**
    $(L, \tilde{p}, p) \leftarrow$ **propose**$(x_{g-1,k}, A_g)$
    $\check{Q} \leftarrow \check{Q}(L)$ {See Section 'Analytic jump integration'}
    $n = |L|$ {The length of the list of states L}
    $w_{g,k} \leftarrow w_{g-1,k} \times p \times (\exp(T_g \check{Q}))_{1,n}/\tilde{p}$
    $x_{g,k} \leftarrow$ **last**(L)
  end for
  if ESS$(w_{g,\cdot}) <$ threshold **then**
    $(w_{g,\cdot}, x_{g,\cdot}) \leftarrow$ **resample**$(w_{g,\cdot}, x_{g,\cdot})$ {Perform SMC resampling step}
  end if
end for

2.1 Phylogenetics

2.1.1 Validation

We generated 200 pair-wise alignments 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 λ and μ. We approximate the posterior using our method. We show the results of λ in the paper and μ in Figure 1. 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 show that the convergence is rapid in this case.

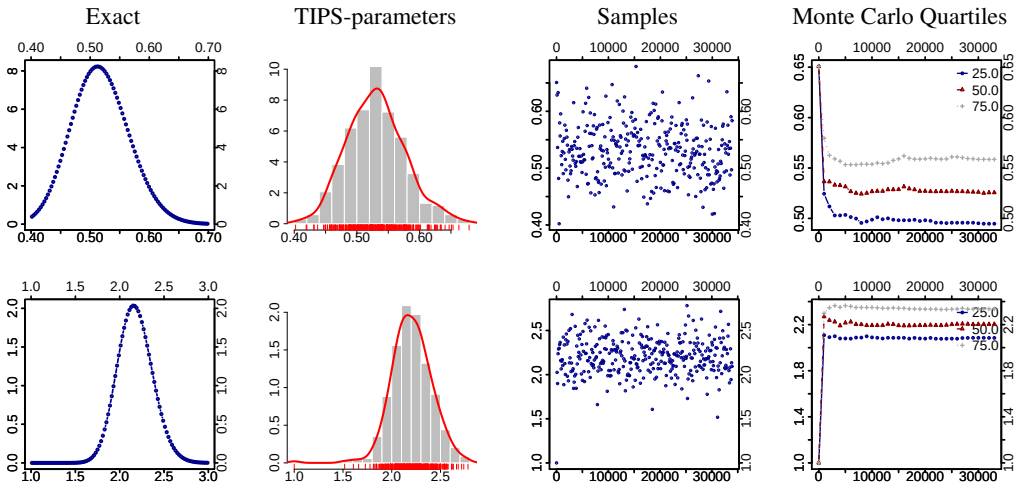


Figure 1: **Results of parameter** μ (1^{st} row) in the validation of the implementation using the Poisson Indel Process (PIP). From left to right: posterior obtained via numerical methods; approximate posterior using our method with 64 particles and 35,000 GIMH iterations; sampled parameter values; convergence of the quantile computed from the GIMH output. For further clarity, the results of parameter λ (2^{nd} row) are also shown.

		ESS	
Parameter	N. Particles	FS	TIPS
λ	10	175.9	1,556.2
	100	44.6	**7,082.7**
	1000	25.6	927.8
	10000	35.2	147.5
	100000	12.2	12.4
μ	10	11.3	774.1
	100	90.2	**6,761.2**
	1000	31.0	903.5
	10000	48.3	128.9
	100000	16.7	12.0

Table 1: **Varying the number of particles per MCMC step.** We show the effective sample size obtained in a fixed computational budget (wall clock time of 3 days), comparing a GIMH based on our method (TIPS), compared to the forward sampling method (FS). Refer to Section 3 of the main paper for details.

2.1.2 Tree inference

We start by introducing some notation for data on a phylogenetic tree τ. Let v_0 denote the root, and $V(\tau)$ the other nodes. Let $\varrho(v)$ denote the parent of the node $v \in V(\tau)$. Let $X^{*(v)} = (X_1^{(v)}, \cdots, X_{N_v}^{(v)})$ denote the sequence of molecular strings that evolve from $\varrho(v)$ to v. The string $X_{N_v}^{(v)}$ at Node v is also denoted by $X^{(v)}$ for simplicity. Note that only the strings at the leaves are observed, denoted $\mathcal{Y}$. Denote all unobserved strings by $X^*(\tau) = \{X^{*(v)} : v \in V(\tau)\}\backslash\mathcal{Y}$, where $\backslash$ is the set difference symbol. The probability of $\mathcal{Y}$ and $X^*(\tau) = x^*(\tau)$ given τ, θ is

$$\mathbb{P}_\theta\left(\mathcal{Y}, X^*(\tau) = x^*(\tau)\middle|\tau\right) =$$
$$\mathbb{P}\left(X^{*(v_0)} = x^{*(v_0)}\right) \prod_{v \in V(\tau)} \mathbb{P}_{x^{(\varrho(v))},\theta}\left(X^{*(v)} = x^{*(v)}\right).$$

We use an improper uniform distribution over the strings as the distribution for the root sequence.

In the Bayesian framework, we aim at the posterior on τ, $X^*(\tau)$, θ given $\mathcal{Y}$, which has a density proportional to $\gamma(\tau, x^*(\tau), \theta) = \mathbb{P}_\theta(\mathcal{Y}, X^*(\tau) = x^*(\tau)|\tau)p(\tau)p(\theta)$, where $p(\tau)$ is a prior for τ.

For fixed evolutionary parameters, we use the framework of Wang (2012) to estimate the posterior of τ. In this framework, we let the r-th partial state s_r be a forest that includes the forest topology and the associated branch lengths, denoted τ_r, as well as the unobserved strings at the root of each tree in that forest, $x^*(\tau_r)$; i.e. $s_r = (\tau_r, x^*(\tau_r))$. We used the following sequence of intermediate distributions over forests: $\gamma(s_r) = \prod_{\tau_i \in s_r} \gamma(\tau_i, x^*(\tau_i)))$. In the weight update step, besides proposing two branch lengths and randomly choosing a pair of trees from the current forest to merge (as in Wang (2012)), we use the proposal distribution described in the Methodology Section of the main paper for proposing the hidden strings, with the only difference that a root sequence is selected uniformly among the intermediate strings on the proposed path. Algorithm 6 shows the component of the algorithm not present in previous work: proposing a root sequence and two discrete paths L_{left} and L_{right} linking this root to its two children. We assume that the pair of sequences to merge, x_{left} and x_{right}, as well as the branch lengths connecting each to the newly formed root, $T_{\text{left}}, T_{\text{right}}$, have been picked by a standard phylogenetic SMC proposal, with proposal density $\tilde{p}_{\text{tree}}$. These become inputs to Algorithm 6, which returns $(L_{\text{left}}, L_{\text{right}}, \tilde{p}, p)$. From this output, we compute two auxiliary rate matrices, $\check{Q}_{\text{left}} = \check{Q}(L_{\text{left}})$ and $\check{Q}_{\text{right}} = \check{Q}(L_{\text{right}})$, one for each newly formed branch, using the same method as before.

Given all this information, we update the particle weights as follows:[1]

$$w_{g,k} = w_{g-1,k} \frac{p \text{ statio}(x_{\text{left}})}{\tilde{p}\, \tilde{p}_{\text{tree}}} \left(\exp(T_{\text{left}} \check{Q}_{\text{left}})\right)_{1,|L_{\text{left}}|} \left(\exp(T_{\text{right}} \check{Q}_{\text{right}})\right)_{1,|L_{\text{right}}|}.$$

Algorithm 6 : **phylo-sequence-proposal**$(x_{\text{left}}, T_{\text{left}}, x_{\text{right}}, T_{\text{right}})$

$(L, \tilde{p}, p) \leftarrow$ **propose**$(x_{\text{left}}, x_{\text{right}}, T_{\text{left}} + T_{\text{right}})$
$i \sim \text{Unif}\{1, 2, \ldots, |L|\}$
$\tilde{p} \leftarrow \tilde{p} \times 1/|L|$
$L_{\text{left}} \leftarrow$ **reverse**(**sublist**$(L, 1, i)$)
$L_{\text{right}} \leftarrow$ **sublist**$(L, i, |L|)$
return $(L_{\text{left}}, L_{\text{right}}, \tilde{p}, p)$

We simulated subsets of molecular sequences with different random seeds according to our evolutionary model. The parameters are: SSM length=3, $\theta_{sub} = 0.03$, $\lambda_{pt} = 0.05$, $\mu_{pt} = 0.2$, $\lambda_{SSM} = 2.0$, and $\mu_{SSM} = 2.0$. A subset of the data is shown in the Figure 2.

```
internal_0|CA--G---C---A--G-------------TG--A---
internal_1|-GAG-C---G-G------AA----GA----TGC-TGC
internal_2|--AG-CAG--CC------CG--C-GAC---TG-----
internal_3|-GAG-C---G-G------AA----GA----TGC----
internal_4|-GAG-C---G-G------AA----GA----TGC----
internal_5|-GAG-C---G-G------AA----GA----TGC----
internal_6|-GAG-C---G-G------AA----GA----TGC----
internal_7|-TAG-C---G-C------CA--C-GAC---TGC----
internal_8|ATAG-C---G------C----A----G-C-GGCA---
leaf_0    |CA--G---C---A--G----C--A---G-TG--A---
leaf_1    |-GAG-C---G-G------AA----GT----TGC-TGC
leaf_2    |-GAG-C---G-G------AA----GA----TGC-TGC
leaf_3    |-GAG-C---G-G------AA----GA----TGC----
leaf_4    |CA--G---C---A--G-------------TG--A---
leaf_5    |-GAT-C---G-G------AA----GA----TGC----
leaf_6    |--AG-CAG--CC------CG--C-GAC---TG-----
leaf_7    |-GAG-C---G-G------AA----GA----TGC----
leaf_8    |--AG-CAG--CC-GC--CCG--C-GAC---CG-----
leaf_9    |-GAT-----G-G------GA----GT-----GC----
```

Figure 2: **Sequence Simulation.** One subset of simulated sequence data for the experiments on trees via SMC.

2.2 RNA Folding Pathways

To compare our method (denoted as TIPS) to that of forward sampling (denoted as FS), we first obtained an absolute accuracy by computing the matrix exponential. Then we computed the absolute log error estimate $\hat{p}$ (i.e., $\text{error}(\hat{p}) = |\log \hat{p} - \log \mathbb{P}_x(X_N = y)|$) of our method and forward sampling on the RNA molecules shown in Table 2. These RNA sequences are short, due to the limit in the size of matrix computed using the matrix exponential. The state spaces for the first two RNA sequences were sufficiently small for computation of the matrix exponential. The complete state space of the last two RNAs were too large for the matrix exponential, so we sampled a subset of the state space.

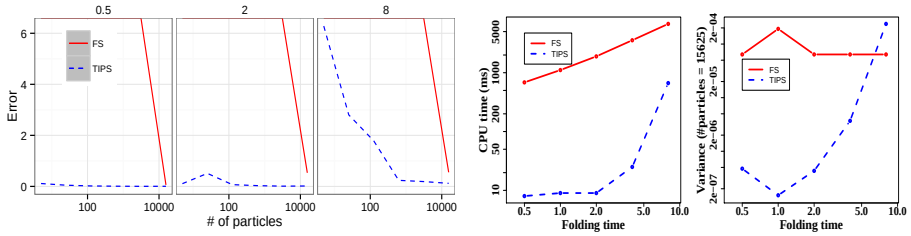
Figures 3a, 3d show the performance of the FS and TIPS methods, for two short molecules 1AFX and 1XV6, on selective folding times, $\{0.5, 2, 8\}$. Figures 3b, 3e show the CPU times (in milliseconds) corresponding to the minimum number of particles required to satisfy the certain accuracy level, $I = \{\hat{p} : \text{error}(\hat{p}) < 1.0\}$ on different folding times including the selective ones.

[1]Note that this formula assumes that the process is reversible, but can be extended to the non-reversible case.

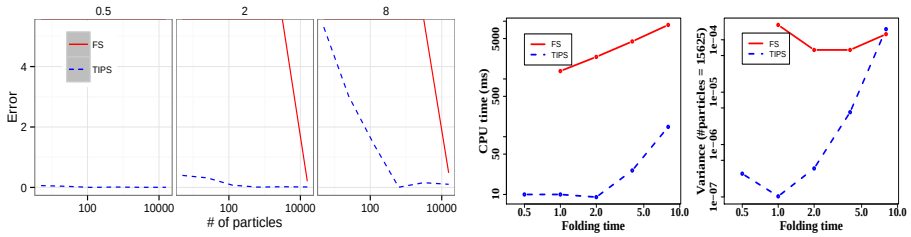
Sequence	Length	$\lvert\mathcal{X}\rvert$	$\lvert S\rvert$
1AFX	12	70	-
1XV6	12	48	-
RNA21	21	~ 1100	657
HIV	23	~ 1500	266

Table 2: **Biological RNA Sequences.**

The variances of FS and TIPS weights, for $5^6 = 15625$ particles, are also computed and compared on different folding times (see Figures 3c, 3f).



(a) 1AFX- error (b) 1AFX- CPU time (ms) (c) 1AFX- variance



(d) 1XV6 - error (e) 1XV6 - CPU time (ms) (f) 1XV6 - variance

Figure 3: Performance of our method (TIPS) and forward sampling (FS) on 1AFX and 1XV6 molecules with their *whole* state space. The relative errors of the estimates vs. folding times, $\{0.5,2,8\}$ 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, $\{0.5, 1, \cdots, 8\}$.

Our method has two main tuning parameters, a geometric parameter, β, over the number of repeated excursions from y to itself, and a parameter, α, weighting the probability of the steps decreasing the value of the potential. We found that the accuracy of the sampler in the case of RNA folding pathways was sensitive to the setting of the parameters (see Figure 4). Parameter tuning is an important area of future work.

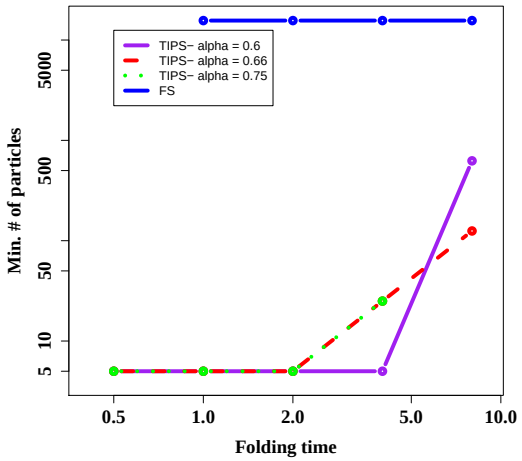


Figure 4: **Tuning parameter α.** Performance of our method (TIPS) using different values of α compared to forward sampling (FS) for estimating the folding pathway of the 1XV6 molecule on its *whole* state space.

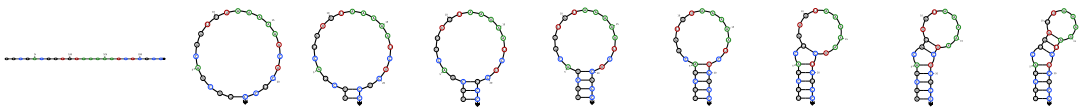


Figure 5: An example of a sampled folding pathway for the HIV23 molecule with $T = 0.125$.

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.

Jun, S.-H. , Wang, L. , and Bouchard-Côté, A. . Entangled Monte Carlo, 2012.

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.

Lee, A. , Yau, C. , Giles, M. B. , Doucet, A. , and Holmes, C. C. . On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 19(4):769–789, 2010.

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.