Title
An efficient Bayesian inference framework for coalescent-based nonparametric phylodynamics

Permalink
https://escholarship.org/uc/item/6p23s3h0

Journal
Bioinformatics, 31(20)

ISSN
1367-4803

Authors
Lan, S
Palacios, JA
Karcher, M
et al.

Publication Date
2015-03-25

DOI
10.1093/bioinformatics/btv378

Peer reviewed
An Efficient Bayesian Inference Framework for Coalescent-Based Nonparametric Phylodynamics

Shiwei Lan 1*, Julia A. Palacios 2,3, Michael Karcher 4, Vladimir N. Minin 4,5 and Babak Shahbaba 6*

1Department of Statistics, University of Warwick, Coventry CV4 7AL.
2Department of Ecology and Evolutionary Biology, Brown University.
3Department of Organismic and Evolutionary Biology, Harvard University.
4Department of Statistics, University of Washington.
5Department of Biology, University of Washington.
6Department of Statistics, University of California, Irvine.

ABSTRACT
Phylodynamics focuses on the problem of reconstructing past population size dynamics from current genetic samples taken from the population of interest. This technique has been extensively used in many areas of biology, but is particularly useful for studying the spread of quickly evolving infectious diseases agents, e.g., influenza virus. Phylodynamics inference uses a coalescent model that defines a probability density for the genealogy of randomly sampled individuals from the population. When we assume that such a genealogy is known, the coalescent model, equipped with a Gaussian process prior on population size trajectory, allows for nonparametric Bayesian estimation of population size dynamics. While this approach is quite powerful, large data sets collected during infectious disease surveillance challenge the state-of-the-art of Bayesian phylodynamics and demand computationally more efficient inference framework. To satisfy this demand, we provide a computationally efficient Bayesian inference framework based on Hamiltonian Monte Carlo for coalescent process models. Moreover, we show that by splitting the Hamiltonian function we can further improve the efficiency of this approach. Using several simulated and real datasets, we show that our method provides accurate estimates of population size dynamics and is substantially faster than alternative methods based on elliptical slice sampler and Metropolis-adjusted Langevin algorithm.

1 INTRODUCTION
Population genetics theory states that changes in population size affect genetic diversity, leaving a trace of these changes in individuals’ genomes. The field of phylodynamics relies on this theory to reconstruct past population size dynamics from current genetic data. In recent years, phylodynamic inference has become an essential tool in areas like ecology and epidemiology. For example, a study of human influenza A virus from sequences sampled in both hemispheres pointed to a source-sink dynamics of the influenza evolution [Rambaut et al., 2008].

Phylodynamics connects population dynamics and genetic data using coalescent-based methods [Griffiths and Tavare, 1994; Kuhner et al., 1998; Drummond et al., 2002; Strimmer and Pybus, 2001; Drummond et al., 2005; Opgen-Rhein et al., 2005; Heled and Drummond, 2008; Minin et al., 2008; Palacios and Minin, 2013]. Typically, phylodynamics relies on Kingman’s coalescent model, which is a probability model that describes formation of genealogical relationships of a random sample of molecular sequences. The coalescent model is parameterized in terms of the effective population size, an indicator of genetic diversity [Kingman, 1982].

While recent studies have shown promising results in alleviating computational difficulties of phylodynamic inference [Palacios and Minin, 2012, 2013], existing methods still lack the level of computational efficiency required to realize the full potential of phylodynamics: developing surveillance programs that can operate similarly to weather monitoring stations allowing public health workers to predict disease dynamics in order to optimally allocate limited resources in time and space. To achieve this goal, we present an accurate and computationally efficient inference method for modeling population dynamics given a genealogy. More specifically, we concentrate on a class of Bayesian nonparametric methods based on Gaussian processes [Minin et al., 2008; Gill et al., 2013; Palacios and Minin, 2013]. Following Palacios and Minin [2012] and Gill et al. [2013], we assume a log-Gaussian process prior on the effective population size. As a result, the estimation of effective population size trajectory becomes similar to the estimation of intensity of a log-Gaussian Cox process [LGCP; Möller et al., 1998], which is extremely challenging since the likelihood evaluation becomes intractable: it involves integration over an infinite-dimensional random function. We resolve the intractability in likelihood evaluation by discretizing the integration interval with a regular grid to approximate the likelihood and the corresponding score function.

For phylodynamic inference, we propose a computationally efficient Markov chain Monte Carlo (MCMC) algorithm using Hamiltonian Monte Carlo [HMC; Duane et al., 1987; Neal, 2010] and one of its variation, called Split HMC [Leimkuhler and Reich, 2004; Neal, 2010; Shahbaba et al., 2013], which speeds-

* to whom correspondence should be addressed

© 2014.
up standard HMC’s convergence. Our proposed algorithm has several advantages. First, it updates all model parameters jointly to avoid poor MCMC convergence and slow mixing rates when there are strong dependencies among model parameters [Knorr-Held and Rue, 2002]. Second, unlike a recently proposed Integrated Nested Laplace Approximation [INLA Rue et al., 2009, Palacios and Minin, 2012] method, our approach can be extended to a more realistic setting where instead of a genealogy of the sampled individuals, we observe their genetic data that indirectly inform us about genealogical relationships. Third, we show that our method is up to an order of magnitude more efficient than MCMC algorithms, such as Metropolis-Adjusted Langevin algorithm [MALA; Roberts and Tweedie, 1996], adaptive MALA [aMALA; Knorr-Held and Rue, 2002], and Elliptical Slice Sampler [ES2; Murray et al., 2010], frequently used in phylodynamics. Finally, although in this paper we focus on phylodynamic studies, our proposed methodology can be easily applied to more general point process models.

The remainder of the paper is organized as follows. In Section 2, we provide a brief overview of the coalescent model and HMC algorithms. Section 3 presents the details of our proposed sampling methods. Experimental results based on simulated and real data are provided in Section 4. Section 5 is devoted to discussion and future directions.

2 PRELIMINARIES

2.1 Coalescent

Assume that a genealogy with time measured in units of generations is available. The coalescent model allows us to trace the ancestry of a random sample of n genomic sequences as tree: two sequences or lineages merge into a common ancestor as we go back in time. Those “merging” times are called coalescent times. The coalescent with variable population size can be viewed as an inhomogeneous Markov death process that starts with n lineages at present time, \( t_0 = 0 \), and decreases by one at each of the consequent coalescent times, \( t_{n-1} < \cdots < t_1 \), until reaching their most recent common ancestor [Griffiths and Tavaré, 1994].

Suppose we observe a genealogy of n individuals sampled at time 0. Under the standard (isochronous) coalescent model, the coalescent times \( t_n = 0 < t_{n-1} < \cdots < t_1 \), conditioned on the effective population size trajectory, \( N_e(t) \), have the density

\[
P(t_1, \ldots, t_n \mid N_e(t)) = \prod_{k=2}^{n} P(t_{k-1} \mid t_k, N_e(t)) \]

where \( C_k = \frac{t_k}{N_e(t_{k-1})} \) and \( I_k = \{t_k, t_{k-1}\} \). Note that the larger the population size, the longer it takes for two lineages to coalesce. Further, the larger the number of lineages, the faster two of them meet their common ancestor [Palacios and Minin, 2012, 2013].

For rapidly evolving organisms, we may have different sampling times. When this is the case, the standard coalescent model can be generalized to account for such heterochronous sampling [Rodrigo and Felsenstein, 1999]. Under the heterochronous coalescent the number of lineages change at both coalescent times and sampling times. Let \( \{t_k\}_{k=1}^{n} \) denote the coalescent times as before, but now let \( s_m = 0 < s_{m-1} < \cdots < s_1 \) denote sampling times of \( n_m, \ldots, n_1 \) sequences respectively, where \( \sum_{j=1}^{m} n_j = n \). Further, let \( s \) and \( n \) denote the vectors of sampling times \{s_j\}_{j=1}^{m} \) and numbers of sequences \{n_j\}_{j=1}^{m} \) sampled at these times, respectively. Then the coalescent likelihood of a single genealogy becomes

\[
P(t_1, \ldots, t_n \mid s, n, N_e(t)) = \prod_{k=2}^{n} C_{s,k} \exp \left\{ - \int_{I_{s,k}} \frac{C_{s,k}}{N_e(t)} \, dt - \sum_{j \geq 1} \int_{I_{s,k}} \frac{C_{s,k}}{N_e(t)} \, dt \right\} \]

where the coalescent factor \( C_{s,k} = \left( t_{s,k} \right) \) depends on the number of lineages \( I_{s,k} \) in the interval \( I_{s,k} \) defined by coalescent times and sampling times. For \( k = 2, \ldots, n \), we denote half-open intervals that end with a coalescent event by

\[
I_{0,k} = \{ \max\{t_k, s_j\}, t_{k-1}\},
\]

for \( s_j < t_{k-1} \) and half-open intervals that end with a sampling event by \( (i > 0) \)

\[
I_{s, k} = \{ \max\{t_k, s_j+i\}, s_{j+i-1}\},
\]

for \( t_k < s_{j+i-1} \leq s_j < t_{k-1} \). In density \( \{I_{s,k}\}_{k=0}^{m-1} \) and \( m-1 \) intervals \( \{I_{s,k}\}_{k=0}^{m} \) for all i, k. Note that only those intervals satisfying \( I_{s,k} \subset \{t_k, t_{k-1}\} \) are non-empty. See Figure 1 for more details.

We can think of isochronous coalescence as a special case of heterochronous coalescence when \( m = 1 \).}
We assume the following log-Gaussian Process prior on the effective population size, \(N_e(t)\):

\[
N_e(t) = \exp[f(t)], \quad f(t) \sim \mathcal{GP}(0, C(\theta)),
\]

(5)

where \(\mathcal{GP}(0, C(\theta))\) denotes a Gaussian process with mean function \(0\) and covariance function \(C(\theta)\). A priori, \(N_e(t)\) is a log-Gaussian Process.

For computational convenience, we use Brownian motion, which is a special case of GP, as our prior for \(f(t)\). We define the covariance function as \(C(\kappa) = \frac{1}{2} C_{BM}\), where the precision parameter \(\kappa\) has a Gamma(\(\alpha, \beta\)) prior. For ascending times \(0 = x_0 < x_1 < x_2 < \cdots < x_{D-1}\), the \((i, j)\)-th element of \(C_{BM}\) is set to \(\min(x_i, x_j)\). This way, we reduce the computational complexity of inverting the covariance matrix from \(\mathcal{O}(n^3)\) to \(\mathcal{O}(n)\) since the inverse covariance matrix is tri-diagonal [Rue and Held, 2005] [Palacios and Minin, 2013] with elements

\[
C_{BM}^{-1}(i, j) = \begin{cases} 
\frac{1-x_i}{1-x_i-x_j} + \frac{1}{x_i}, & \text{if } i = j, \\
-\frac{1}{|x_i-x_j|}, & \text{if } |i-j| = 1, \\
0, & \text{otherwise}.
\end{cases}
\]

In practice we modify the \((1, 1)\) element, \(1/(x_2 - x_1) + 1/x_1\), to be \(1/(x_2 - x_1)\) and denote the resulting covariance matrix as \(C_{BM}\) to indicate that it comes from an intrinsic autoregression [Besag and Kooperberg, 1995] [Knorr-Held and Rue, 2002]. Note that \(C_{BM}^{-1}\) is degenerate in 1 eigen direction so we add a small number, \(\epsilon\), to the diagonal elements of \(C_{BM}\) to make it invertible when \(C_{BM}\) is needed.

### 2.2 HMC

Hamiltonian Monte Carlo [Duane et al., 1987] [Neal, 2010] is a Metropolis sampling algorithm that suppresses the random walk behavior by proposing states that are distant from the current state, but nevertheless accepts new proposals with high probability. These distant proposals are found by numerically simulating Hamiltonian dynamics, whose state space consists of position, denoted by the vector \(\theta\), and momentum, denoted by the vector \(p\). The objective is to sample from the continuous probability distribution with the density function \(p(\theta)\). It is common to assume \(p \sim N(M, \Sigma)\), where \(M\) is a symmetric, positive-definite matrix known as the mass matrix, often set to the identity matrix \(I\) for convenience.

In this simulation of Hamiltonian dynamics, the potential energy, \(U(\theta)\), is defined as the negative log density of \(\theta\) (plus any constant); the kinetic energy, \(K(p)\) for momentum variable \(p\), is set to be the negative log density of \(p\) (plus any constant). Then the total energy of the system, the Hamiltonian function, is defined as their sum: \(H(\theta, p) = U(\theta) + K(p)\). Then the system of \((\theta, p)\) evolves according to the following set of Hamilton’s equations:

\[
\dot{\theta} = \nabla_p H(\theta, p) = M^{-1} p, \\
\dot{p} = -\nabla_\theta H(\theta, p) = -\nabla_\theta U(\theta).
\]

(6)

In practice, we use a numerical method called leapfrog to approximate the Hamilton’s equations [Neal, 2010] when the analytical solution is not available. We numerically solve the system for \(L\) steps, with some step size, \(\epsilon\), to propose a new state in the Metropolis algorithm, and accept or reject it according to the Metropolis acceptance probability. [See Neal, 2010 for more discussions].

### 3 METHOD

#### 3.1 Discretization

As discussed above, the likelihood function (2) is intractable in general. We can, however, approximate the likelihood using discretization. To this end, we use a fine regular grid, \(x = \{x_d\}_{d=1}^D\), over the observation window and approximate \(N_e(t)\) by a piece-wise constant function as follows:

\[
N_e(t) \approx \sum_{d=1}^{D-1} \exp[f(x_d^*)] I_{d} \{x_d, x_{d+1}\}, \quad x_d^* = \frac{x_d + x_{d+1}}{2}.
\]

(7)

Note that the regular grid \(x\) does not coincide with the sampling coalescent times, except for the first sampling time \(s_m = x_1\) and the last coalescent time \(t_D = x_D\). To rewrite (7) using the approximation (2), we sort all the time points \(\{t, s, x\}\) to create new \(D + m + n - 4\) half-open intervals \(\{I_{\alpha}\}\) with either coalescent time points, sampling time points, or grid time points as the end points (See Figure [1]).

For each \(\alpha \in \{1, \cdots, D + m + n - 4\}\), there exists some \(i, k\) and \(d\) such that \(I_{\alpha} = I_{i, k} \cap (x_d, x_{d+1}]\). Each integral in density (2) can be approximated as a sum:

\[
\int_{I_{i, k}} C_{i, k} \frac{dt}{N_e(t)} \approx \sum_{I_{i, k} \subseteq I_{i, k}} C_{i, k} \exp[-f(x_d^*)] \Delta_d,
\]

where \(\Delta_d := x_{d+1} - x_d\). This way, the likelihood of coalescent times (2) can be rewritten as a product of the following terms:

\[
\frac{C_{i, k} \exp[f(x_d^*)]}{\exp[f(x_d^*)]} y_{\alpha} \exp \left(-\frac{C_{i, k} \Delta_d}{\exp[f(x_d^*)]} \right),
\]

(8)

where \(y_{\alpha}\) is set to 1 if \(I_{i, k}\) ends with a coalescent time, and to 0 otherwise. This happens to be proportional to the probability mass of a Poisson random variable \(y_{\alpha}\) with intensity \(\lambda_{\alpha} := C_{i, k} \Delta_d \exp[-f(x_d^*)]\). Therefore, the likelihood of coalescent times (2) can be approximated as follows:

\[
P[t_1, \ldots, t_n | N_e(t)] \approx \prod_{\alpha=1}^{D+m+n-4} P[y_{\alpha} | N_e(t)]
\]

\[
= \prod_{d=1}^{D-1} \prod_{I_{i, k} \subseteq I_{i, k} \cap (x_d, x_{d+1}]} \left\{ \frac{C_{i, k}}{\exp[f(x_d^*)]} y_{\alpha} \exp \left(-\frac{C_{i, k} \Delta_d}{\exp[f(x_d^*)]} \right) \right\},
\]

(9)

where the coalescent factor \(C_{i, k}\) on each interval \(I_{i, k}\) is determined by the number of lineages \(l_{i, k}\) in \(I_{i, k}\).

#### 3.2 Sampling methods

Denote \(f := \{f(x_d^*)\}_{d=1}^{D-1}\). Our model can be summarized as

\[
y_{\alpha} \sim \text{Poisson}[\lambda_{\alpha}(f)],
\]

\[
f|\kappa \sim \mathcal{N}\left(0, \frac{1}{\kappa} C_{IN}\right),
\]

(10)

\[
\kappa \sim \text{Gamma}(\alpha, \beta).
\]
Transforming the coalescent times, sampling times and grid points into \( \{y_0, C_{i,k}, \Delta x\} \), we condition on these data to generate posterior samples for \( f = \log N_c(x^*) \) and \( \kappa \), where \( x^* = \{x^*_d\}\) is the set of the middle points in (7). We use these posterior samples to make inference about \( N_c(t) \).

For sampling \( f \) using HMC, we use \( \varphi \) to compute the discretized log-likelihood
\[
l = - \sum_{d=1}^{D-1} \sum_{t_d \in \{x^*_d, x^*_{d+1}\}} \{y_0 f(x^*_d) + C_{i,k} \Delta c \exp[-f(x^*_d)]\}
\]
and the corresponding gradient (score function)
\[
s_d = - \sum_{t^*_d \in \{x^*_d, x^*_{d+1}\}} \{y_0 - C_{i,k} \Delta c \exp[-f(x^*_d)]\}.
\]

Because the prior on \( \kappa \) is conditionally conjugate, we could directly sample from its full conditional posterior distribution,
\[
\kappa|\sim \text{Gamma}(\alpha + (D - 1)/2, \beta + f^T C^{-1}_{in} f/2).
\]

However, updating \( f \) and \( \kappa \) separately is not recommended in general because of their strong interdependency [Knorr-Held and Rue, 2002]. Large value of precision \( \kappa \) strictly confines the variation of \( f \), rendering slow movement in the space occupied by \( f \). Therefore, we update \( \{f, \kappa\} \) jointly in MCMC sampling algorithms. In practice, it is better to sample \( \theta := (f, \tau) \), where \( \tau = \log(\kappa) \) is in the same scale as \( f = \log N_c(x^*) \). Note that the log-likelihood of \( \theta \) is the same as that of \( f \) because equation (2) does not involve \( \tau \). The log-density prior on \( \theta \) is defined as follows:
\[
\log P(\theta) \propto ((D - 1)/2 + \alpha - 1)\tau - (f^T C^{-1}_{in} f/2 + \beta e^\tau).
\]

### 3.3 Speed up by splitting Hamiltonian

Splitting the Hamiltonian is a technique used to speed up HMC [Leimkuhler and Reich, 2004; Neal [2010]; Shahbaba et al., 2013]. The underlying idea is to divide the total Hamiltonian into several terms, such that the dynamics associated with some of these terms can be solved analytically. For these parts, typically quadratic forms, the simulation of the dynamics does not introduce a discretization error, allowing for faster movements in the parameter space.

We split the Hamiltonian \( H(\theta, p) = U(\theta) + K(p) \) as follows:
\[
\begin{align*}
H(\theta, p) &= \frac{-l - ((D - 1)/2 + \alpha - 1)\tau + \beta e^\tau}{2} + \\
f^T C_{in}^{-1} f e^\tau + p^T p &= \frac{l - ((D - 1)/2 + \alpha - 1)\tau + \beta e^\tau}{2}. \quad (13)
\end{align*}
\]

We further split the middle part (which is the dominant part) into two dynamics involving \( f \) and \( \tau \) respectively,
\[
\begin{cases}
\dot{f} = -p_{-D}
\quad \dot{\tau} &= f_{-D},
\end{cases}
\]
\[
\begin{cases}
p_{-D} = -C_{in}^{-1} f e^\tau.
\end{cases}
\]

Using the spectral decomposition \( C_{in}^{-1} = Q A Q^{-1} \) and denoting \( f^* := \sqrt{\kappa} e^{\tau/2} Q^{-1} f \) and \( p^*_{-D} := Q^{-1} p_{-D} \), we can analytically solve the dynamics (14a) as follows [Lan, 2013] (more details are in the appendix):
\[
\begin{bmatrix}
\dot{f}(t) \\
\dot{p}_{-D}(t)
\end{bmatrix}
= \begin{bmatrix}
\cos(\sqrt{\kappa} e^{\tau/2} t) & \sin(\sqrt{\kappa} e^{\tau/2} t) \\
-\sin(\sqrt{\kappa} e^{\tau/2} t) & \cos(\sqrt{\kappa} e^{\tau/2} t)
\end{bmatrix}
\begin{bmatrix}
f^*(0) \\
p^*_{-D}(0)
\end{bmatrix}.
\]

We then use the standard leapfrog method to solve the dynamics (14b) and the residual dynamics in (13) in a symmetric way. Note that we only need to diagonalize \( C_{in}^{-1} \), once before the sampling, and calculate \( f^T C_{in}^{-1} f e^\tau = f^* T ; \) therefore, the overall computational complexity of the integrator is \( O(D^2) \).

### 4 EXPERIMENTS

We illustrate the advantages of our HMC-based methods using three simulation studies and apply these methods to analysis of a real dataset. We evaluate our methods by comparing them to INLA in terms of accuracy and to several sampling algorithms, MALA, aMALA, and ES2, in terms of sampling efficiency. We define sampling efficiency as time-normalized effective sample size (ESS). Given \( B \) MCMC samples for each parameter, we calculate the corresponding ESS \( ESS = B[1 + 2\Sigma_{k=1}^K(\gamma(k))]^{-1} \), where \( \Sigma_{k=1}^K(\gamma(k)) \) is the sum of \( K \) monotone sample autocorrelations [Geyer, 1992]. We use the minimum ESS normalized by the CPU time, \( s \) (in seconds), as the overall measure of efficiency: \( \min(ESS)/s \).

We tune the stepsize and number of leapfrog steps for our HMC-based algorithm such that their overall acceptance probabilities are in a reasonable range (close to 0.70). Since MALA [Roberts and Tweedie, 1996] and aMALA [Knorr-Held and Rue, 2002] can be viewed as HMC with one leapfrog step for numerically solving Hamiltonian dynamics, we implement MALA and aMALA proposals using our HMC framework. MALA, aMALA, and HMC-based methods update \( f \) and \( \tau \) jointly. aMALA uses a joint block-update method designed for GMRF models: it first generates a proposal \( \kappa|\kappa \) from some symmetric distribution independently of \( f \), and then updates \( f^*|f, \kappa |\) based on a local Laplace approximation. Finally, \( f^*|f, \kappa \) is either accepted or rejected. It can be shown that aMALA is equivalent to a form of Riemannian MALA [Roberts and Stramer, 2002; Girolami and Calderhead, 2011] also see Appendix B]. In addition, aMALA closely resembles the most frequently used MCMC algorithm in Gaussian process-based phylogenetics [Minn et al., 2008; Gill et al., 2013].

ES2 [Murray et al., 2010] is a commonly used sampling algorithm designed for models with Gaussian process priors. It was also adopted by Palacios and Minin [2013] for phylogenetic inference. ES2 implementation relies on the assumption that the target distribution is approximately normal. This, of course, is not a suitable assumption for the joint distribution of \( (f, \tau) \). Therefore, we alternate the updates \( f|\kappa \) and \( \kappa|f \) when using ES2.

#### 4.1 Simulations

We simulate three genealogies relating \( n = 50 \) individuals with the following true trajectories:

1. **logistic trajectory:**
   \[
   N_c(t) = \begin{cases}
   10 + 1000\exp(-t) & , t \ mod \ 12 \leq 6, \\
   1000\exp(-t) & , t \ mod \ 12 > 6; \\
   \end{cases}
   \]

2. **exponential growth:**
   \[
   N_c(t) = 1000\exp(-t);
   \]

3. **boumibust:**
   \[
   N_c(t) = \begin{cases}
   1000\exp(-t + 2), & t \leq 2, \\
   1000\exp(-t + 2), & t > 2.
   \end{cases}
   \]
Efficient Bayesian Phylodynamics

Table 1. Sampling efficiency in modeling simulated population trajectories. The true population trajectories are: I) logistic, II) exponential growth, and III) boombust respectively. AP is the acceptance probability. s/iter is the seconds per sampling iteration. “spdup” is the speedup of efficiency measurement minESS/s using ES^2 as baseline.

Table 2. Demographic parameters (nu and nu') and their uncertainties for the three simulations.

4.2 Human Influenza A in New York

Next, we analyze real data based on a genealogy estimated from 288 H3N2 sequences sampled in New York state from January 2001 to March 2005 in order to estimate population size dynamics of human influenza A in New York [Palacios and Minin, 2012, 2013]. The key feature of the influenza A virus epidemic in temperate regions like New York are the epidemic peaks during winters followed by strong bottlenecks at the end of the winter season. We use 120 grid points in the likelihood approximation. Figure 4 shows that with $C_{-1}^{BM}$, MCMC algorithms identify such peak-bottleneck pattern more clearly than INLA. However, their results based on intrinsic precision matrix, $C_{-1}^{in}$, are quite comparable to that of INLA. In
Fig. 3. Trace plots of log-likelihoods for different sampling algorithms based on a simulated coalescent model with logistic population trajectory. splitHMC converges the fastest.

Table 2 we can see that the speedup by HMC and splitHMC over other MCMC methods is substantial.

<table>
<thead>
<tr>
<th>Method</th>
<th>AP</th>
<th>ε/λer</th>
<th>minESS(f)/s</th>
<th>spdup(f)</th>
<th>ESS(τ)/s</th>
<th>spdup(τ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES²</td>
<td>1.00</td>
<td>1.92E-03</td>
<td>0.34</td>
<td>1.00</td>
<td>0.27</td>
<td>1.00</td>
</tr>
<tr>
<td>MALA</td>
<td>0.78</td>
<td>2.12E-03</td>
<td>0.29</td>
<td>0.86</td>
<td>0.60</td>
<td>2.20</td>
</tr>
<tr>
<td>aMALA</td>
<td>0.77</td>
<td>8.48E-03</td>
<td>0.002</td>
<td>0.01</td>
<td>0.33</td>
<td>1.23</td>
</tr>
<tr>
<td>HMC</td>
<td>0.71</td>
<td>1.74E-02</td>
<td>1.69</td>
<td>4.98</td>
<td>0.57</td>
<td>2.12</td>
</tr>
<tr>
<td>splitHMC</td>
<td>0.75</td>
<td>1.11E-02</td>
<td>2.90</td>
<td>8.53</td>
<td>1.06</td>
<td>3.92</td>
</tr>
</tbody>
</table>

Table 2. Sampling efficiency of MCMC algorithms in influenza data.

5 DISCUSSION

In this paper, we have proposed new HMC-based sampling algorithms for phylodynamic inference, which are substantially more efficient than existing methods. Although we have focused on phylodynamics, the methodology presented here can be applied to general LGCP models.

There are several possible future directions. One possibility is to use ES² as a proposal generating mechanism in updating f as opposed to using it for sampling from the posterior distribution. Finding a good proposal for f(τ), however, remains challenging.

An important extension of the methods presented here is to allow for genealogical uncertainty. The MCMC methods analyzed here can be incorporated into a hierarchical framework to infer population size trajectories from sequence data directly. In contrast, INLA cannot be adapted easily to perform inference from sequence data. This greatly limits its generalizability.

Fig. 4. Population dynamics of influenza A in New York (2001-2005): shaded region is the 95% credible interval calculated with samples given by splitHMC. Results using C_{BM} (upper) vs results using C_{vrs} (lower).

APPENDIX A: SPLITHMC

Here, we show how to solve Hamiltonian dynamics defined by (13) using the "splitting" strategy. Denote z := (f, p), M := C_{vrs}^{-1} e^τ, and A := \begin{bmatrix} 0 & I \\ -M & 0 \end{bmatrix}. The dynamics (14) can be written as

\[ \dot{z} = Az. \]  

We then have the analytical solution to (13) as

\[ z(t) = e^{A^t} z(0), \]  

where e^{A^t} is a matrix defined as e^{A^t} := \sum_{i=0}^t \frac{A^i}{i!}, which in turn can be written as follows:

\[ e^{A^t} = \begin{bmatrix} I - \frac{t^2}{2!} M + \frac{t^4}{4!} M^2 + \cdots & \cdots \\ -Mt + \frac{t^3}{3!} M^2 - \frac{t^5}{5!} M^3 + \cdots & \cdots \\ M^{\frac{3}{2}} \sin(\sqrt{Mt}) & \cos(\sqrt{Mt}) \\ M^{\frac{3}{2}} \sin(\sqrt{Mt}) & \cos(\sqrt{Mt}) \\ 0 & I \end{bmatrix} \]

For positive definite matrix M, we can use the spectral decomposition M = QDQ^{-1}, where Q is orthogonal matrix, i.e.
\[ Q^{-1} = Q^T. \text{ Therefore we have} \]
\[
\begin{bmatrix}
\cos(\sqrt{D}t) & \sin(\sqrt{D}t) \\
-sin(\sqrt{D}t) & \cos(\sqrt{D}t)
\end{bmatrix}
\]
\[ \begin{bmatrix} 0 & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} D \frac{\tau}{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\tau}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ Q & 1 \end{bmatrix}.
\]

In practice, we only need to diagonalize \( C^{-1}_n \) once: \( C^{-1}_n = QAQ^{-1} \), then \( D = \text{Ac} \). If we let \( f^* := \sqrt{\text{Ac}^2}Q^{-1}f, p_{D} := Q^{-1}p_{-D}, \) we have the following solution \[ (17) \]
\[
\begin{bmatrix} f^*(t) \\ p_{-D}^*(t) \end{bmatrix} = \begin{bmatrix} \cos(\sqrt{\text{Ac}^2}t) & \sin(\sqrt{\text{Ac}^2}t) \\ -\sin(\sqrt{\text{Ac}^2}t) & \cos(\sqrt{\text{Ac}^2}t) \end{bmatrix} \begin{bmatrix} f^*(0) \\ p_{-D}(0) \end{bmatrix}.
\]

We then apply leapfrog method to the remaining dynamics. Algorithm 1 summarizes these steps.

**Algorithm 1** splitHMC for the coalescent model (splitHMC)

- Initialize \( \theta^{(1)} \) at current \( \theta = (\theta, \tau) \)
- Sample a new momentum value \( p^{(1)} \sim \mathcal{N}(0, 1) \)
- Calculate \( H(\theta^{(1)}, p^{(1)}) = U(\theta^{(1)}) + K(p^{(1)}) \) according to \[ (3) \]
- for \( \ell = 1 \) to \( \ell \) do
  - \( p_{D}^{(\ell-1/2)} = p_{D}^{(\ell)} + \epsilon/2p_{D}^{(\ell-1)}T^{(\ell)}/2 \)
  - \( \tau^{(\ell-1/2)} = \tau^{(\ell)} + \epsilon/2p_{-D}^{(\ell-1/2)}T^{(\ell)}/2 \)
  - \( f^{(*)}_{-D}^{(\ell+1/2)} = \begin{bmatrix} \cos(\sqrt{\text{Ac}^2}T^{(\ell+1)/2}) & \sin(\sqrt{\text{Ac}^2}T^{(\ell+1)/2}) \\ \sin(\sqrt{\text{Ac}^2}T^{(\ell+1)/2}) & \cos(\sqrt{\text{Ac}^2}T^{(\ell+1)/2}) \end{bmatrix} \begin{bmatrix} f^{(*)}_{-D}^{(\ell)} \\ p_{-D}^{(\ell)} \end{bmatrix} \)
  - \( f^{(*)}_{D}^{(\ell+1/2)} = \begin{bmatrix} \cos(\sqrt{\text{Ac}^2}T^{(\ell+1)/2}) & \sin(\sqrt{\text{Ac}^2}T^{(\ell+1)/2}) \\ \sin(\sqrt{\text{Ac}^2}T^{(\ell+1)/2}) & \cos(\sqrt{\text{Ac}^2}T^{(\ell+1)/2}) \end{bmatrix} \begin{bmatrix} f^{(*)}_{D}^{(\ell)} \\ p_{D}^{(\ell)} \end{bmatrix} \)
  - \( \tau^{(\ell+1)} = \tau^{(\ell+1/2)} + \epsilon/2p_{-D}^{(\ell+1/2)}T^{(\ell+1)/2} \)
  - \( p_{D}^{(\ell+1)} = p_{D}^{(\ell+1/2)} - \epsilon/2f^{(*)}_{D}^{(\ell+1/2)}T^{(\ell+1)/2} \)
  - \( p^{(\ell+1)} = p^{(\ell+1/2)} + \epsilon/2((D - 1)/2 + \alpha - 1) - \exp(\tau^{(\ell+1)}) \)
- end for

Calculate \( H(\theta^{(\ell+1)}, p^{(\ell+1)}) = U(\theta^{(\ell+1)}) + K(p^{(\ell+1)}) \) according to \[ (3) \]
Calculate the acceptance probability \( \alpha = \min\{1, \exp[-H(\theta^{(\ell+1)}, p^{(\ell+1)})] + H(\theta^{(\ell)}, p^{(\ell)})]\}

Accept or reject the proposal according to \( \alpha \) for the next state \( \theta' \)

**APPENDIX B: ADAPTIVE MALA**

We now show that the joint block updating in [Knorr-Held and Rue 2002] can be recognized as an adaptive MALA algorithm. First, we sample \( \kappa' | \kappa \sim p(\kappa' | \kappa) \propto \frac{1}{\kappa' + c} \) on \( [\kappa/c, \kappa c] \) for some \( c > 1 \) controlling the step size of \( \kappa \). Denote \( w := \{C_{i,k} \Delta x_i\}_{1}^{D+m+n-4} \)

**Algorithm 2** Adaptive MALA (aMALA)

Given current state \( \theta = (\theta, \kappa) \) calculate potential energy \( U(\theta) \)

repeat
- \( z \sim \text{Unif}[1/c, c] \), \( u \sim \text{Unif}[0, 1] \)
- unless \( u < e^{\frac{e}{z} \kappa} \)
- update precision parameter \( \kappa' = \kappa z \)
- Sample momentum \( p \sim \mathcal{N}(0, \kappa^{-1}) \)
- Calculate log of proposal density \( \log p(f' | \kappa, \kappa') = -\frac{1}{2}p^T G(f, \kappa) p + \frac{1}{2} \log \det G(f, \kappa) \)
- update momentum \( p \leftarrow p - \epsilon/2G(f, \kappa)^{-1} \nabla U(f, \kappa) \)
- update latent variables \( f' := f + \epsilon p \)
- Calculate log of reverse proposal density \( \log p(f', \kappa) = -\frac{1}{2}p^{T}G(f', \kappa)p + \frac{1}{2} \log \det G(f', \kappa) \)
- Calculate new potential energy \( U(\theta') \)

Accept/reject the proposal according to \( \log \alpha = -U(\theta') + U(\theta) - \log p(f' | \kappa, \kappa) + \log p(f | \kappa, \kappa) \) for the next state \( \theta' \)

and use the following Taylor expansion for \( \log p(f | \kappa) \) about \( f' \):
\[
\log p(f | \kappa) = -y^T f - w^T \exp(-f) - \frac{1}{2}f^T \kappa C_{in}^{-1} f
\]
\[
\approx -y^T f - (w \exp(-f'))^T [1 - (f - f') + (f - f')^2]/2 - \frac{1}{2}f^T \kappa C_{in}^{-1} f
\]
\[
= (-y + w \exp(-f'))^T f + \frac{1}{2}f^T \kappa C_{in}^{-1} + \text{diag}(w \exp(-f')) f
\]
\[
= b^T f - \frac{1}{2}f^T G f,
\]
where \( b(f') := -y + w \exp(-f')(1 + f') \), \( G(f, \kappa) := \kappa C_{in}^{-1} + \text{diag}(w \exp(-f')) \), Setting \( f \) to the current state, \( f' \), and propose \( f' | f, \kappa' \) from the following Gaussian distribution:
\[
f' | f, \kappa' \sim N(\mu, \Sigma),
\]
with
\[
\mu = G(f, \kappa')^{-1} b(f) = f + G(f, \kappa'^{-1}) \nabla f \log p(f | \kappa') \text{ and } \Sigma = G(f, \kappa'^{-1}),
\]
which has the same form as Langevin dynamical proposals. Interestingly, \( G(f, \kappa) \) is exactly the (observed) Fisher information. That is, this approach is equivalent to Riemannian MALA [Girolami and Calderhead 2011].

Finally, \( \theta' = (f', \kappa') \) is jointly accepted with the following probability:
\[
\alpha = \min\{1, p(\theta' | D) p(\kappa' | \kappa) p(f' | f', \kappa') \}
\]
\[
= \min\{1, p(\theta' | D) p(f' | f', \kappa') \}
\]
where \( p(\kappa' | \kappa) \) is a symmetric proposal. Algorithm 2 summarizes the steps for adaptive MALA.
REFERENCES


Shiwei Lan. Advanced bayesian computational methods through geometric techniques, 2013. Copyright - Copyright ProQuest, UMI Dissertations Publishing 2013; Last updated - 2014-02-13; First page - n/a; M3: Ph.D.


