

UCLA

UCLA Previously Published Works

Title

Gradients Do Grow on Trees: A Linear-Time $O(N)$ -Dimensional Gradient for Statistical Phylogenetics.

Permalink

<https://escholarship.org/uc/item/1fm0b35t>

Journal

Molecular Biology and Evolution, 37(10)

ISSN

0737-4038

Authors

Ji, Xiang
Zhang, Zhenyu
Holbrook, Andrew
et al.

Publication Date


2020-10-01

DOI

10.1093/molbev/msaa130

Peer reviewed

Gradients Do Grow on Trees: A Linear-Time $O(N)$ -Dimensional Gradient for Statistical Phylogenetics

Xiang Ji ^{†,1} Zhenyu Zhang,² Andrew Holbrook,² Akihiko Nishimura,³ Guy Baele,⁴ Andrew Rambaut,⁵ Philippe Lemey,⁴ and Marc A. Suchard^{*,1,2,6}

¹Department of Biomathematics, David Geffen School of Medicine, University of California Los Angeles, Los Angeles, CA

²Department of Biostatistics, Fielding School of Public Health, University of California Los Angeles, Los Angeles, CA

³Department of Biostatistics, Bloomberg School of Public Health, Johns Hopkins University, Baltimore, MD

⁴Department of Microbiology, Immunology and Transplantation, Rega Institute, KU Leuven, Leuven, Belgium

⁵Institute of Evolutionary Biology, Centre for Immunology, Infection and Evolution, University of Edinburgh, Edinburgh, United Kingdom

⁶Department of Human Genetics, David Geffen School of Medicine, University of California Los Angeles, Los Angeles, CA

*Corresponding author: E-mail: msuchard@ucla.edu.

[†]Present address: Department of Mathematics, School of Science & Engineering, Tulane University, New Orleans, LA

Associate editor: Jeffrey Townsend

Abstract

Calculation of the log-likelihood stands as the computational bottleneck for many statistical phylogenetic algorithms. Even worse is its gradient evaluation, often used to target regions of high probability. Order $O(N)$ -dimensional gradient calculations based on the standard pruning algorithm require $O(N^2)$ operations, where N is the number of sampled molecular sequences. With the advent of high-throughput sequencing, recent phylogenetic studies have analyzed hundreds to thousands of sequences, with an apparent trend toward even larger data sets as a result of advancing technology. Such large-scale analyses challenge phylogenetic reconstruction by requiring inference on larger sets of process parameters to model the increasing data heterogeneity. To make these analyses tractable, we present a linear-time algorithm for $O(N)$ -dimensional gradient evaluation and apply it to general continuous-time Markov processes of sequence substitution on a phylogenetic tree without a need to assume either stationarity or reversibility. We apply this approach to learn the branch-specific evolutionary rates of three pathogenic viruses: West Nile virus, Dengue virus, and Lassa virus. Our proposed algorithm significantly improves inference efficiency with a 126- to 234-fold increase in maximum-likelihood optimization and a 16- to 33-fold computational performance increase in a Bayesian framework.

Key words: linear-time gradient algorithm, random-effects molecular clock model, Bayesian inference, maximum likelihood.

Introduction

Advances in the portability, accuracy, and cost-efficiency of genome sequencing technology (Quick et al. 2016) are generating genetic data at an ever-increasing pace, overwhelming many key computational tools for molecular analysis. The enormity of modern data sets presents a general challenge in molecular evolution, but the problem is particularly pressing in infectious disease research.

The ability to collect and sequence pathogen genomes in real time requires the development of novel statistical methods that are able to process the sequences in a timely manner and produce interpretable results to inform national public health organizations, rather than act as a bottleneck to the epidemiological response workflow. Coupling such methods with highly efficient computing is key to rapid dissemination of outbreak analysis results to make global health decisions focused on intervention strategies and disease control. Molecular phylogenetics has

become an essential analytical tool for understanding the complex patterns in which rapidly evolving pathogens propagate throughout and between countries, owing to the complex travel and transportation patterns evinced by modern economies (Pybus et al. 2015), along with other factors such as increased global population and urbanization (Bloom et al. 2017). Of the statistical paradigms employed in this domain, likelihood-based inference is by far the most dominant because of its ability to incorporate complex statistical models while offering accurate tree reconstruction under a wide range of evolutionary scenarios (see, e.g., Ogden and Rosenberg 2006). These likelihood-based approaches require repeated evaluation of the observed data likelihood function and its gradient and therefore computational performance is heavily dependent on data scale. As a result, and yet despite their lower accuracy, faster heuristics often substitute for likelihood-based methods in scenarios where a timely response is essential.

Felsenstein's pruning algorithm (Felsenstein 1973, 1981) makes the observed data likelihood in phylogenetics computationally tractable. The observed molecular sequences at the tips evolve on the phylogenetic tree according to a continuous-time Markov chain (CTMC) with discrete states. The pruning algorithm marginalizes over all possible latent states of the CTMC at internal nodes and calculates the probability of the observed sequence data through a post-order tree traversal, that visits all nodes once in a descendant-to-parent fashion that works its way up to the root starting from the tips. This traversal requires $\mathcal{O}(N)$ operations for each likelihood evaluation, where N is the number of sampled molecular sequences. For a CTMC with discrete states, one can calculate the first derivative of the likelihood by substituting the transition probability matrix with its derivative matrix into the pruning algorithm (Kishino et al. 1990; Adachi and Hasegawa 1996; Yang 2000; Bryant et al. 2005; Kenney and Gu 2012). This pruning-based gradient calculation requires the same computational effort as the likelihood evaluation for a parameter on a given branch, i.e., $\mathcal{O}(N)$, but costs $\mathcal{O}(N^2)$ operations to calculate with respect to (w.r.t.) parameters pertaining to all branches. Both maximum-likelihood and Bayesian inference are popular frameworks for inferring the phylogeny and its related evolutionary parameters, requiring the same observed data likelihood to be estimated w.r.t. the parameter space. Parameters of interest include the topology of the evolutionary tree, branch lengths, parameters within the infinitesimal generator matrix that describes the CTMC as well as mixture model parameters that describe evolutionary processes such as among-site rate heterogeneity (Yang 1994) and varying rates between partitions (Yang 1996; Shapiro et al. 2006).

Owing to the complexity of the phylogenetic likelihood surface (see, e.g., Sanderson et al. 2015), maximum-likelihood frameworks employ nonlinear optimization to find the maximum-likelihood estimate (MLE) for model parameters. Importantly, the computations required to find the MLE differ greatly between parameters, as certain "local" parameters—often specific to a single branch or a subset of branches—only require a (small) part of the likelihood function to be re-evaluated whereas other "global" parameters—typically the parameters of the CTMC process—require a complete re-evaluation. In addition to the global optimization routine that re-evaluates the complete likelihood when proposing new parameter values, maximum-likelihood software packages such as RAxML (Stamatakis et al. 2005) and GARLI (Zwickl 2006) incorporate a local optimization routine that only optimizes a few branch-specific parameters—for example, in the vicinity of a recent topological change—while keeping all other parameters fixed. Although both applications adopt pruning-based algorithms for gradient calculations, the computational cost of local optimization routines is roughly only $\mathcal{O}(N)$, which they achieve by optimizing only $\mathcal{O}(1)$ number of parameters, for example, the three branch lengths connecting the internal node that is the target of a tree rearrangement operation. An additional advantage of such local routines is the possibility to perform multiple

evaluations of branch-specific derivatives in parallel, conditional on the remainder of the tree not changing.

Bayesian phylogenetic inference packages combine prior knowledge with the (observed data) likelihood into a joint density proportional to the posterior and, as such, attempt to estimate posterior distributions for all parameters of interest. Despite its great success for incorporating complex statistical models (see, e.g., Huelsenbeck et al. 2001), Bayesian phylogenetic inference remains computationally intensive. The computational cost of the gradient evaluation prevents Bayesian phylogenetics from benefiting from more efficient gradient-based samplers, such as the Hamiltonian Monte Carlo (HMC) sampler (Neal 2011). In summary, both maximum-likelihood and Bayesian implementations of phylogenetic modeling stand to benefit from faster calculations of the gradient.

We here propose an $\mathcal{O}(N)$ algorithm for calculating the gradient w.r.t. all branch-specific parameters by complementing the postorder traversal in the pruning algorithm with its corresponding preorder traversal. The algorithm thus extends the pioneering work of Schadt et al. (1998) to general CTMCs (homogeneous or not) while not assuming stationarity or reversibility. We apply our proposed algorithm to study the evolutionary rates of viral sequences that we model with a random-effects clock model that combines both fixed- and random-effects when accommodating evolutionary rate variation (Bletsa et al. 2019). We show that the proposed approach significantly improves inference efficiency of the branch-specific evolutionary rates under both maximum-likelihood and Bayesian frameworks.

New Approach

In this section, we define necessary notation for deriving the gradient algorithm. We then illustrate the likelihood calculation through the postorder traversal as in the pruning algorithm and the update of the postorder partial likelihood vectors. We derive a new partial likelihood vector at each node and its update through a preorder traversal. We expand the likelihood at any node as the inner product of its post- and preorder partial likelihood vectors. Finally, we derive the $\mathcal{O}(N)$ -dimensional gradient using the two partial likelihood vectors at all nodes.

Notation

Consider a phylogeny \mathcal{F} with N tips and $N-1$ internal nodes. Assume that the root node is on the top and the tip nodes are at the bottom of \mathcal{F} . We denote the tip nodes with numbers $1, 2, \dots, N$ and the internal nodes with numbers $N+1, N+2, \dots, 2N-1$ where the root node is fixed at $2N-1$. Any branch on \mathcal{F} connects a parent node to its child node where the parent node is closer to the root. We denote $\text{pa}(i)$ as the parent node of node i . We refer to a branch by the number of the child node it connects. On \mathcal{F} , we model the sites in the sequence alignment as independent and identically distributed such that they arise from conditionally independent CTMCs acting along each branch. Depending on the state space of the CTMCs, a site can be a single (nucleotide) column or multiple consecutive columns that contain a

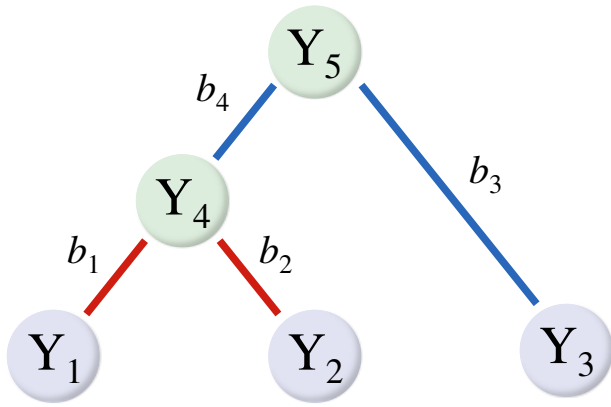


FIG. 1. Schematic of a 3-taxon tree. The observed data at a site $\mathbf{Y} = (Y_1, Y_2, Y_3)'$ are character states at the tips of the tree. The latent states Y_4 and Y_5 are at internal nodes of the tree. We divide the observed data \mathbf{Y} into two disjoint sets with $\mathbf{Y}_{[4]} = \{Y_1, Y_2\}$ and $\mathbf{Y}_{[4]} = \{Y_3\}$ to help set up the corresponding post- and preorder partial likelihood vectors at internal node 4. We further color the branches to show the update of the two partial likelihood vectors at internal node 4 such that red branches correspond to the update of the postorder partial likelihood vector and blue branches correspond to the update of the preorder partial likelihood vector. RR

codon (or encode for an amino acid) or even the entire sequence.

Suppose we have observed (at tips) and latent (at internal nodes) discrete evolutionary characters Y_i for $i = 1, \dots, 2N - 1$ at a site. Character Y_i has m possible states (e.g., $m = 4$ for nucleotide substitution models, $m = 20$ for amino acid substitution models and $m = 61$ for codon substitution models that exclude the stop-codons). Let b_i denote the branch length of branch i . Let r_i denote the evolutionary rate on branch i and t_i denote the real time of node i . Then $b_i = r_i(t_i - t_{pa(i)})$. For branch i with CTMC infinitesimal rate matrix \mathbf{Q}_i , the transition probability matrix is $\mathbf{P}_i = e^{\mathbf{Q}_i b_i}$. Let $\boldsymbol{\pi} = [\mathbb{P}(Y_{2N-1} = 1), \mathbb{P}(Y_{2N-1} = 2), \dots, \mathbb{P}(Y_{2N-1} = m)]'$ denote the state distribution at the root node (not necessarily the stationary distribution of the CTMCs).

The evolutionary rates and chronological times appear implicitly in the likelihood function through the branch lengths. This implicitness poses an inference challenge for molecular dating, also known as divergence time estimation. Having samples with different sampling times, such as serially sampled viral sequences or fossil information, supplements additional time anchors for calibration. Improvement on characterizing the other confounding factor, the evolutionary rates, relies on the development of more biologically plausible clock models that describe the rate changes on the tree. However, such models come at the cost of having to infer many highly correlated parameters that can be computationally demanding for large data sets (see Inferring Evolutionary Rate Variation section for more detail).

To set up the post- and preorder partial likelihood vectors, we further divide the observed characters $\mathbf{Y} = \{Y_i, 1 \leq i \leq N\}$ at tips into two disjoint sets w.r.t. any node in \mathcal{F} . Let $\mathbf{Y}_{[i]}$ denote the observed characters at the

tip nodes descendant of node i . Let $\mathbf{Y}_{[i]} = \mathbf{Y} \setminus \mathbf{Y}_{[i]}$ denote the observed characters at the tip nodes not descendant from node i . Finally, let $\phi = \{\mathcal{F}, r_i, b_i, t_i, \mathbf{Q}_i; \forall i\}$ collect all model parameters. The length m postorder partial likelihood vector \mathbf{p}_i of node i at a site has the j th element being $(\mathbf{p}_i)_j = \mathbb{P}(\mathbf{Y}_{[i]} | Y_i = j)$. When i is a tip node, $\mathbb{P}(\mathbf{Y}_{[i]} | Y_i = j) = \mathbf{1}_{\{Y_i=j\}}$ for $j = 1, 2, \dots, m$. For partially observed and missing data at the tip node, one can modify the postorder partial likelihood vector to reflect this information (Felsenstein 1981). Similarly, the preorder partial likelihood vector \mathbf{q}_i of node i has the j th element being $(\mathbf{q}_i)_j = \mathbb{P}(Y_i = j, \mathbf{Y}_{[i]})$. For the root node, $\mathbf{Y}_{[2N-1]} = \emptyset$, and the preorder partial likelihood vector is the same as the state distribution (i.e., $\mathbf{q}_{2N-1} = \boldsymbol{\pi}$).

Likelihood

The likelihood is the marginal probability of the observed discrete characters at the tip nodes that sums over all possible latent characters at the internal nodes:

$$\begin{aligned} \mathbb{P}(\mathbf{Y}) &= \sum_{\mathbf{Y}_{N+1}} \sum_{\mathbf{Y}_{N+2}} \dots \sum_{\mathbf{Y}_{2N-1}} \mathbb{P}(\mathbf{Y}, \mathbf{y}) \text{ and} \\ \mathbb{P}(\mathbf{Y}, \mathbf{y}) &= \mathbb{P}(\mathbf{Y}_{2N-1}) \prod_{j=1}^{2N-2} \mathbb{P}(\mathbf{Y}_j | \mathbf{Y}_{pa(j)}), \end{aligned} \quad (1)$$

where the summation at internal nodes are w.r.t. all possible latent states. We omit the conditioning on the parameters ϕ above and in later derivations to save space. We use the example phylogenetic tree in figure 1 with three tip nodes and two internal nodes to demonstrate the likelihood calculation. The observed data (at a site) in figure 1 are $\mathbf{Y} = \{Y_1, Y_2, Y_3\}$. And, one obtains the likelihood of the observed data by marginalizing over $\mathbf{y} = \{Y_4, Y_5\}$.

Postorder Traversal

The pruning algorithm is a dynamic programming algorithm that calculates equation (1) through postorder traversal (Felsenstein 1973, 1981). The postorder traversal visits every node on the tree in a descendent node first fashion. For example, two possible postorder traversals for the example tree in figure 1 are $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ or $1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5$. Using the latter, the decomposition:

$$\begin{aligned} \mathbb{P}(\mathbf{Y}) &= \sum_{\mathbf{Y}_5} \mathbb{P}(\mathbf{Y}_5) [\sum_{\mathbf{Y}_4} \mathbb{P}(\mathbf{Y}_4 | \mathbf{Y}_5) \mathbb{P}(\mathbf{Y}_1 | \mathbf{Y}_4) \mathbb{P}(\mathbf{Y}_2 | \mathbf{Y}_4)] \\ &\quad \mathbb{P}(\mathbf{Y}_3 | \mathbf{Y}_5) \end{aligned} \quad (2)$$

shows how the pruning algorithm separates the grand sum in equation (1) into intermediate steps at the internal nodes for the example phylogenetic tree. With the postorder partial likelihood vector and the transition probability matrices, the matrix-vector representation of equation (2) is:

$$\mathbb{P}(\mathbf{Y}) = \boldsymbol{\pi}' [\mathbf{P}_4 (\mathbf{P}_1 \mathbf{p}_1 \circ \mathbf{P}_2 \mathbf{p}_2) \circ \mathbf{P}_3 \mathbf{p}_3], \quad (3)$$

where \circ denotes the element-wise multiplication.

Only postorder partial likelihood vectors at the tip nodes appear explicitly in equation (3). The recursive update for the postorder partial likelihood vector \mathbf{p}_k at internal node k given

the postorder partial likelihood vectors \mathbf{p}_i and \mathbf{p}_j at its two descendent nodes i and j (i.e., $\text{pa}(i) = \text{pa}(j) = k$) is implicit in equation (3):

$$\mathbf{p}_k = \mathbf{P}_i \mathbf{p}_i \circ \mathbf{P}_j \mathbf{p}_j. \quad (4)$$

Again, for the update of the postorder partial likelihood vector at internal node 4 in figure 1, $k = 4$, $i = 1$, $j = 2$, and $\mathbf{p}_4 = \mathbb{P}(\mathbf{Y}_{[4]} | \mathbf{Y}_4) = \mathbf{P}_1 \mathbf{p}_1 \circ \mathbf{P}_2 \mathbf{p}_2$. We color the branches relevant to this update red.

The postorder traversal updates all postorder partial likelihood vectors up to the root node. At the end of the traversal, the likelihood is just the inner product of the state distribution vector with the postorder partial likelihood vector at the root node.

$$\begin{aligned} \mathbb{P}(\mathbf{Y}) &= \sum_{j=1}^m [\mathbb{P}(Y_{2N-1} = j) \mathbb{P}(\mathbf{Y}_{[2N-1]} | Y_{2N-1} = j)] \\ &= \boldsymbol{\pi}' \mathbf{p}_{2N-1}. \end{aligned} \quad (5)$$

In the next section, we expand the likelihood as the inner product at any node of its post- and preorder partial likelihood vectors. In fact, this expansion is obvious for the root node because the preorder partial likelihood vector at the root node is just the state distribution vector and equation (5) becomes $\mathbb{P}(\mathbf{Y}) = \mathbf{q}'_{2N-1} \mathbf{p}_{2N-1}$. Further, the expansion enables us to derive the linear-time algorithm that calculates all branch-specific derivatives at once.

Preorder Traversal

The preorder traversal starts from the root node, where $\mathbf{q}_{2N-1} = \boldsymbol{\pi}$, and updates all remaining preorder partial likelihood vectors by visiting them in the reverse order of the postorder traversal. Assume that we have calculated all postorder partial likelihood vectors and consider recursively internal node k with its two immediate descendent nodes i and j . The preorder partial likelihood vector for descendent node i falls out as:

$$\begin{aligned} \mathbb{P}(Y_i, \mathbf{Y}_{[i]}) &= \sum_{Y_k} \mathbb{P}(Y_i, Y_k, \mathbf{Y}_{[k]}, \mathbf{Y}_{[j]}) \\ &= \sum_{Y_k} \mathbb{P}(Y_i | Y_k) \mathbb{P}(\mathbf{Y}_{[j]} | Y_k) \mathbb{P}(Y_k, \mathbf{Y}_{[k]}) \\ &= \sum_{Y_k} \mathbb{P}(Y_i | Y_k) [\sum_{Y_j} \mathbb{P}(\mathbf{Y}_{[j]} | Y_j) \mathbb{P}(Y_j | Y_k)] \mathbb{P}(Y_k, \mathbf{Y}_{[k]}), \end{aligned} \quad (6)$$

since $\mathbb{P}(\mathbf{Y}_{[j]} | Y_j)$ and $\mathbb{P}(Y_k, \mathbf{Y}_{[k]})$ are already known. The matrix-vector representation of equation (6) is:

$$\mathbf{q}_i = \mathbf{P}'_i [\mathbf{q}_k \circ (\mathbf{P}_j \mathbf{p}_j)]. \quad (7)$$

The derivation of the preorder partial likelihood vector for node j is similar. Use figure 1 as an example and consider the update of the preorder partial likelihood vector at internal node 4. Then $i = 4$, $j = 3$, $k = 5$, and $\mathbf{q}_4 = \mathbf{P}'_4 [\mathbf{q}_5 \circ (\mathbf{P}_3 \mathbf{p}_3)]$. We color the branches relevant in this update blue.

For gradient calculations, it becomes useful to rewrite the likelihood as the inner product at any node of its post- and preorder partial likelihood vectors. For node k , we have:

$$\begin{aligned} \mathbb{P}(\mathbf{Y}) &= \sum_{Y_k} \mathbb{P}(Y_k, \mathbf{Y}_{[k]}, \mathbf{Y}_{[k]}) \\ &= \sum_{Y_k} \mathbb{P}(\mathbf{Y}_{[k]} | Y_k) \mathbb{P}(Y_k, \mathbf{Y}_{[k]}) \\ &= \mathbf{P}'_k \mathbf{q}_k. \end{aligned} \quad (8)$$

In the next section, we derive the derivative of the log-likelihood w.r.t. any one branch-specific parameter based on equation (8). In this manner, the new algorithm calculates the gradient of the log-likelihood w.r.t. all branch-specific parameters at once using $\mathcal{O}(N)$ operations.

Gradient

To ease presentation, we use only the matrix-vector forms for derivation in this section. The scalar forms are similar to those of the previous sections. With the likelihood expanded at node i as in equation (8), we derive the gradient vector of the log-likelihood w.r.t. the branch lengths that has the i th element being the partial derivative of the log-likelihood w.r.t. b_i :

$$\begin{aligned} \frac{\partial}{\partial b_i} \mathbb{P}(\mathbf{Y}) &= \frac{\partial}{\partial b_i} [\mathbf{p}'_i \mathbf{q}_i] / \mathbb{P}(\mathbf{Y}) \\ &= \mathbf{p}'_i \frac{\partial \mathbf{q}_i}{\partial b_i} / \mathbb{P}(\mathbf{Y}) \\ &= \mathbf{q}'_i \mathbf{Q}_i \mathbf{p}_i / \mathbb{P}(\mathbf{Y}), \end{aligned} \quad (9)$$

where the third equality follows the fact that the partial derivative of the preorder partial likelihood vector \mathbf{q}_i w.r.t. the branch length b_i is:

$$\begin{aligned} \frac{\partial \mathbf{q}_i}{\partial b_i} &= \frac{\partial}{\partial b_i} \{ \mathbf{P}'_i [\mathbf{q}_k \circ (\mathbf{P}_j \mathbf{p}_j)] \} \\ &= \left(\frac{\partial}{\partial b_i} e^{\mathbf{Q}_i b_i} \right)' [\mathbf{q}_k \circ (\mathbf{P}_j \mathbf{p}_j)] \\ &= (e^{\mathbf{Q}_i b_i} \mathbf{Q}_i)' [\mathbf{q}_k \circ (\mathbf{P}_j \mathbf{p}_j)] \\ &= \mathbf{Q}'_i \mathbf{q}_i. \end{aligned} \quad (10)$$

Likelihood and Gradient with Substitution Rate Heterogeneity

Equation (9) assumes homogeneous substitution rate across sites. A popular approach to model the substitution rate heterogeneity across sites is by using a hidden Markov model where one models the substitution rate as the discrete hidden state with multiple rate categories (Yang 1994). For discrete rate category l with rate γ_l , the transition probability matrix for branch k of rate category l is $\mathbf{P}_{k|\gamma_l} = e^{\mathbf{Q}_k b_k \gamma_l}$. As in hidden Markov models, the likelihood becomes the weighted sum of the conditional likelihood of each rate category that marginalizes over all possible hidden states:

$$\begin{aligned} \mathbb{P}(\mathbf{Y}) &= \sum_{\gamma_l} \mathbb{P}(\mathbf{Y} | \gamma_l) \mathbb{P}(\gamma_l) \\ &= \sum_{\gamma_l} \mathbf{p}'_{k|\gamma_l} \mathbf{q}_{k|\gamma_l} \mathbb{P}(\gamma_l), \end{aligned} \quad (11)$$

where $\mathbf{p}_{k|\gamma_l}$ and $\mathbf{q}_{k|\gamma_l}$ are the corresponding post- and preorder partial likelihood vectors at node k for rate category l .

Their updates are the same as in the rate homogeneous case by substituting $\mathbf{P}_{k|\gamma_i}$ for \mathbf{P}_k . Similarly, the numerator and denominator of equation (9) become weighted sums in the rate heterogeneous case:

$$\frac{\partial}{\partial b_i} \mathbb{P}(\mathbf{Y}) = \sum_{\gamma_i} \gamma_i \mathbf{p}'_{i|\gamma_i} \mathbf{Q}'_i \mathbf{q}_{i|\gamma_i} \mathbb{P}(\gamma_i) / \mathbb{P}(\mathbf{Y}). \quad (12)$$

Equations (10) and (12) show that we only need the post- and preorder partial likelihood vectors \mathbf{p}_i , \mathbf{q}_i and the infinitesimal rate matrix \mathbf{Q}_i at node i for calculating the partial derivative of branch i . In fact, we can calculate these matrix-vector multiplications and vector-vector inner products together with the update of the preorder partial likelihood vectors in the preorder traversal. This action gives us the gradient vector of all partial derivatives w.r.t. branch 1, 2, ..., $2N - 2$ in one single preorder traversal.

Diagonal Elements of the Hessian Matrix

We derive the diagonal elements of the Hessian matrix w.r.t. the log-likelihood to use it later for preconditioning in Hamiltonian Monte Carlo Sampling section. The second-order derivative of the preorder partial likelihood vector is similar to that of its gradient by substituting \mathbf{Q} with \mathbf{Q}^2 in equation (10). Without loss of generality, we illustrate the derivation with the likelihood function in equation (11) where rate homogeneity is its special case with one rate category:

$$\frac{\partial^2}{\partial b_i^2} \mathbb{P}(\mathbf{Y}) = \sum_{\gamma_i} \gamma_i^2 \mathbf{p}'_{i|\gamma_i} (\mathbf{Q}_i^2)' \mathbf{q}_{i|\gamma_i} \mathbb{P}(\gamma_i) / \mathbb{P}(\mathbf{Y}) - \left[\frac{\partial}{\partial b_i} \mathbb{P}(\mathbf{Y}) \right]^2. \quad (13)$$

Applications

We show that our gradient-based approach significantly improves computational efficiency when drawing inference with applications in nonlinear optimization under a maximum-likelihood framework and through HMC sampling under a Bayesian framework.

Nonlinear Optimization

Nonlinear optimization is essential to obtain MLEs in statistical phylogenetics. The parameters include, but are not limited to, branch lengths and substitution rates. GARLI (Zwickl 2006) and RAxML (Stamatakis et al. 2005) employ a number of optimization algorithms such as the Newton-Raphson method and Brent's method for various situations. RAxML can also optionally use the quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno, known as the BFGS algorithm (see, e.g., Dennis and Schnabel 1996), to optimize substitution rate parameters. The unconstrained optimization of an objective function over a set of real parameters is formulated as: $\min_{\mathbf{x}} f(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^n$ is a real vector with length $n \geq 1$. In maximum-likelihood inference, the objective function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is the negative log-likelihood.

The past few decades have witnessed the development of a collection of optimization algorithms (see Nocedal and Wright 2006; Lange 2013 for details). Here, we revisit the

BFGS algorithm and its limited-memory variant (L-BFGS). We then apply the L-BFGS algorithm for obtaining the MLE. All positive parameters in the model are log-transformed into unconstrained parameter spaces.

Like other iterative optimization algorithms, the BFGS algorithm starts at an initial position \mathbf{x}_0 in the parameter space and then iteratively generates a sequence of positions $\{\mathbf{x}_k\}_{k=0}^{\infty}$. The BFGS algorithm is a line search method that minimizes the objective function in each iteration along one specified direction $\delta_k: \min_{\alpha_k > 0} f(\mathbf{x}_k + \alpha_k \delta_k)$ and the iteration continues at $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \delta_k$ until iterates make no more fruitful progress, reach a solution point within a certain error tolerance or max out in number of iterations. Let $\mathbf{s}_k = \alpha_k \delta_k$ be the increment vector in the parameter space of iteration k , $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ be the gradient vector of iteration k , and $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$ be the difference between the gradient vector of iteration $k + 1$ and the gradient vector of the previous iteration k . BFGS determines the line search direction similarly to that of the Newton method except that one approximates the inverse of the Hessian matrix $(\nabla^2 f(\mathbf{x}_k))^{-1}$ by \mathbf{H}_k :

$$\delta_k = -\mathbf{H}_k \mathbf{g}_k$$

$$\mathbf{H}_{k+1} = (\mathbf{I} - \rho_k \mathbf{s}_k \mathbf{y}'_k) \mathbf{H}_k (\mathbf{I} - \rho_k \mathbf{y}_k \mathbf{s}'_k) + \rho_k \mathbf{s}_k \mathbf{s}'_k, \quad (14)$$

where $\rho_k = \frac{1}{\mathbf{y}'_k \mathbf{s}_k}$ and equation (14) satisfies the secant condition $\mathbf{H}_{k+1} \mathbf{y}_k = \mathbf{s}_k$. BFGS starts with an "initial" approximate of the inverse Hessian matrix (i.e., $\mathbf{H}_0 = \mathbf{H}_{\text{init}}$) and updates the \mathbf{H} matrix at each iteration. Alternatively, the L-BFGS algorithm "remembers" only the most recent m iterations such that it initializes $\mathbf{H}_{k+1-m} = \mathbf{H}_{\text{init}}$ and applies equation (14) m times to get \mathbf{H}_{k+1} for the next iteration. A typical choice of the initial matrix \mathbf{H}_{init} is the product of a scalar constant with the identity matrix (see Nocedal and Wright 2006; Lange 2013 for choices of the scalar). Therefore, L-BFGS approximates the Hessian matrix with local curvature information.

Hamiltonian Monte Carlo Sampling

The proposed linear-time gradient algorithm also enables efficient inference under a Bayesian framework through HMC sampling. HMC is a state-of-the-art Markov chain Monte Carlo (MCMC) method that exploits numerical solutions of Hamiltonian dynamics (Neal 2011). Given a parameter of interest θ with the posterior density $\pi(\theta)$, HMC introduces an auxiliary parameter \mathbf{p} and samples from the product density $\pi(\theta, \mathbf{p}) = \pi(\theta)\pi(\mathbf{p})$. The parameter \mathbf{p} typically follows a multivariate normal distribution $\mathbf{p} \sim \mathcal{N}(0, \mathbf{M})$ whose covariance matrix \mathbf{M} is referred to as the "mass matrix." The basic version of HMC sets the mass matrix to the identity matrix, but we discuss a judicious choice in the next section.

Due to the physical laws that motivate HMC, one refers to θ as the "position" variable and \mathbf{p} as the "momentum" variable. One then sets the "potential energy" to the negative log posterior density $U(\theta) = -\log(\pi(\theta))$ and the "kinetic energy" to $K(\mathbf{p}) = \mathbf{p}' \mathbf{M}^{-1} \mathbf{p} / 2$. The sum of the potential and kinetic energy forms the Hamiltonian function $H(\theta, \mathbf{p}) = U(\theta) + K(\mathbf{p})$. From the current state (θ_0, \mathbf{p}_0) , HMC generates a Metropolis proposal (Metropolis et al. 1953) by

simulating Hamiltonian dynamics in the space (θ, \mathbf{p}) that evolves according to the differential equation:

$$\begin{aligned} \frac{d\mathbf{p}}{dt} &= -\nabla U(\theta) = \nabla \log \pi(\theta) \\ \frac{d\theta}{dt} &= \nabla K(\mathbf{p}) = \mathbf{M}^{-1} \mathbf{p}. \end{aligned} \tag{15}$$

The popular “leapfrog” method (Neal 2011) numerically approximates a solution to equation (15). Each leapfrog step of size ϵ follows the trajectory:

$$\begin{aligned} \mathbf{p}_{t+\epsilon/2} &= \mathbf{p}_t + \frac{\epsilon}{2} \nabla \log \pi(\theta_t) \\ \theta_{t+\epsilon} &= \theta_t + \epsilon \mathbf{M}^{-1} \mathbf{p}_{t+\epsilon/2} \\ \mathbf{p}_{t+\epsilon} &= \mathbf{p}_{t+\epsilon/2} + \frac{\epsilon}{2} \nabla \log \pi(\theta_{t+\epsilon}). \end{aligned} \tag{16}$$

We need n leapfrog steps, and hence $n + 1$ gradient evaluations, to simulate the dynamics from time $t = 0$ to $t = n\epsilon$. Such an HMC proposal can have small correlation with the current state, yet be accepted with high probability (Neal 2011). In particular, HMC promises better scalability in the number of parameters (Beskos et al. 2013) and enjoys wide-ranging successes as one of the most reliable MCMC approaches in general settings (Gelman et al. 2013; Kruschke 2014; Monnahan et al. 2017).

Preconditioning with Adaptive Mass Matrix Informed by the Diagonal Hessian

Geometric structure of the posterior distribution significantly affects the computational efficiency of HMC. For example, when the scales of the posterior distribution vary among individual parameters, failing to account for such structure may reduce the efficiency of HMC (Neal 2011; Carpenter et al. 2017). We can adapt HMC for such structure by modifying the dynamics in equation (15) via an appropriately chosen mass matrix \mathbf{M} . Replacing the standard identity matrix with a nonidentity one is equivalent to “preconditioning” the posterior distribution via parameter transformation (Neal 2011; Livingstone and Girolami 2014; Nishimura and Dunson 2016).

Practitioners often choose a mass matrix that approximates the inverse of the posterior covariance matrix of θ (Carpenter et al. 2017) or the negative Hessian of the posterior distribution (Girolami and Calderhead 2011). These two approaches yield similar mass matrices when the posterior distribution is approximately Gaussian. For more complex distributions, however, the Hessian better accounts for the underlying geometry (Girolami and Calderhead 2011) and is further supported by the linear stability analysis of the leapfrog integrator (Hairer et al. 2006). Despite its theoretical advantages, a major practical issue with a Hessian-based approach is the obligate use of a θ -dependent mass matrix $\mathbf{M} = \mathbf{M}(\theta)$. The corresponding dynamics require computationally demanding numerical integrators, each step of which requires several iterations of evaluating and inverting the mass matrix (Girolami and Calderhead 2011).

To incorporate information from the Hessian without excessive computational burden, we adaptively tune \mathbf{M} to estimate the expected Hessian averaged over the posterior distribution. We further restrict \mathbf{M} to remain diagonal and hence approximate the diagonals of the expected Hessian only. This restriction is commonly imposed to regularize the estimate, and a diagonal matrix alone can greatly enhance sampling efficiency of HMC in many situations (Salvatier et al. 2016; Carpenter et al. 2017). In addition, we only update the diagonal mass matrix every $k = 10$ HMC iterations so that the cost of computing the expected Hessian diagonals remains negligible. More precisely, from the first s HMC iterations, we compute:

$$\begin{aligned} H_{ii}^{(s)} &= \frac{1}{\lfloor s/k \rfloor} \sum_{s:s/k \in \mathbb{Z}^+} -\frac{\partial^2}{\partial^2 \theta_i} \log \pi(\theta) \Big|_{\theta = \theta^{(s)}} \\ &\approx \mathbb{E}_{\pi(\theta)} \left[-\frac{\partial^2}{\partial^2 \theta_i} \log \pi(\theta) \right]. \end{aligned} \tag{17}$$

The $(s + 1)^{th}$ iteration then updates the mass matrix with appropriate lower and upper thresholds to make sure that it remains positive-definite and numerically stable:

$$M_{ii}^{(s+1)} = \begin{cases} m_{\min} & \text{if } H_{ii} < m_{\min} \\ m_{\max} & \text{if } H_{ii} > m_{\max} \\ H_{ii}^{(s)} & \text{otherwise} \end{cases} \tag{18}$$

for $0 < m_{\min} < m_{\max}$. The above procedure ensures “vanishing adaptation” $H_{ii}^{(s+1)} - H_{ii}^{(s)} = \mathcal{O}(s^{-1})$ such that HMC remains ergodic despite the adaptation (Andrieu and Thoms 2008).

Inferring Evolutionary Rate Variation

Until the development of the first molecular clock model in the 1960s (Zuckerkandl and Pauling 1962, 1965), our understanding of evolutionary time scale derived mostly from fossil records, because evolutionary rate and time are confounded when comparing homologous DNA sequences. Molecular clock models provide means to anchor the evolutionary time so that chronological events can be estimated.

Molecular Clock Models

In its simplest and earliest form, the molecular clock model assumes a constant evolutionary rate across the tree (Zuckerkandl and Pauling 1962). Researchers often refer to this model as the “strict” clock model. Over the past few decades, researchers have developed a variety of clock models to accommodate the inadequacy of ignoring rate variation among lineages of the strict clock model (see Kumar 2005; Ho and Duchêne 2014 for extensive reviews). One way to characterize a molecular clock model is by the number of unique branch-specific evolutionary rates. The strict clock model assumes rate homogeneity among all branches. Multi-rate clock models relax the homogeneity assumption by assigning branches to rate categories. Branches in the same category

Table 1. Maximum-Likelihood Estimate (MLE) Inference Efficiency Using Two Optimization Methods: Our Proposed Gradient Method (Analytic) and a Central Finite Difference Numerical Scheme (Numeric).

Example	No. Rates	Analytic		Numeric		Speedup	
		Time(s)	Iterations	Time(s)	Iterations	Per Iteration	Total
WNV	206	0.3	12	59.3	20	126.2×	210.4×
LASV	420	1.2	10	369.1	19	168.8×	320.6×
DENV	702	19.1	90	4,827.9	97	234.8×	253.1×

NOTE.—For each example and method, we report the total time to complete MLE inference, as well as the number of iterations required for optimization on an Intel Core i7-2600 quad-core processor running at 3.40 GHz. Our proposed method yields a minimum 200-fold increase in performance across the entire inference, which averages out to a minimum 126-fold performance increase per iteration.

share the same evolutionary rate. The number of categories is usually >1 but smaller than the total number of branches (Hasegawa et al. 1989; Huelsenbeck et al. 2000; Yoder and Yang 2000; Drummond and Suchard 2010). Relaxed molecular clock models contain the highest possible number of unique branch-specific rates where each branch evolves at its own rate. There are two major classes of relaxed molecular clock models, autocorrelated and uncorrelated clock models. The major difference between the two classes is their assumption about the causation of the rate variation. Autocorrelated relaxed clock models assume that evolutionary rate undergoes a diffusion process from the root node to successive branches (Thorne et al. 1998; Kishino et al. 2001; Aris-Brosou and Yang 2002), whereas uncorrelated clock models make no assumption of rate correlation among branches (Drummond et al. 2006; Rannala and Yang 2007; Lemey et al. 2010). A recent addition to the growing list of clock models consists of a mixed relaxed clock model that combines the merits of autocorrelated and uncorrelated relaxed clocks (Lartillot et al. 2016).

Application of relaxed clock models inevitably leads to higher dimensional parameter spaces. However, the computational efficiency of existing methods limits our ability to draw likelihood-based inference from these high-dimensional evolutionary models, a problem that is exacerbated in large data sets. We show that our new gradient algorithm ameliorates this difficulty through applications in gradient-based optimization methods and HMC sampling. Specifically, we demonstrate marked improvement on computational efficiency for inferring the evolutionary rates of three viruses as described in Materials and Methods under a random-effects relaxed clock model.

Random-Effects Relaxed Clock Models

The random-effects relaxed clock model combines a strict clock and an uncorrelated relaxed clock model. We model the evolutionary rate r_i of branch i as the product of a global tree-wise mean parameter μ and a branch-specific random effect ϵ_i . We model the random effect ϵ_i 's as independent and identically distributed from a lognormal distribution such that ϵ_i has mean 1 and variance ψ^2 under a hierarchical model where ψ is the scale parameter. We note that the popular uncorrelated relaxed clock model is a special case of this clock model and will hence also benefit from the improvements in this manuscript.

Priors

We assign a conditional reference prior to the global tree-wise mean parameter μ (Ferreira and Suchard 2008) and an exponential prior with mean $\frac{1}{3}$ to the scale parameter ψ . We use the same substitution models as in each example's original study (Pybus et al. 2012; Nunes et al. 2014; Andersen et al. 2015).

Results

We present the computational efficiency improvements conferred by our linear-time gradient algorithm for inferring the branch-specific evolutionary rates.

Optimization

We obtain MLEs of the branch-specific random effects conditional on all other parameters via the L-BFGS algorithm for all three viral data sets. In computing these MLEs, we compare the performance of our analytic gradient method with an often-used central finite difference scheme. The numerical scheme calculates the partial derivative of one branch-specific rate through two likelihood evaluations and has a complexity of $\mathcal{O}(N^2)$ for the gradient w.r.t. all rates. On the other hand, our analytic approach scales $\mathcal{O}(N)$ (see New Approach section). Table 1 shows a summary of the comparison, illustrating the immense performance increase across the three data sets of our analytic method. Averaged over each iteration of the MLE estimation process, the analytic method outperforms the finite difference scheme by a factor of 126- to 235-fold, leading to a total real-time speedup of 210- to 321-fold.

Posterior Inference

We infer the posterior distribution of all evolutionary rates using three different MCMC transition kernels in BEAST (Suchard et al. 2018) using BEAGLE (Ayres et al. 2019). The first transition kernel is the univariate transition kernel that Pybus et al. (2012) formerly employed, which we will refer to as “Univariate.” “Univariate” updates propose new values for one rate r_i at a time whereas the HMC transition kernels propose new values for all $2N - 2$ rates simultaneously. We consider two mass matrix choices for HMC. “Vanilla” HMC (vHMC) employs an identity matrix and “preconditioned” HMC (pHMC) employs an adaptive diagonal matrix informed by the Hessian.

We compare the efficiency of these three transition kernels through their effective sample size (ESS) per unit time for

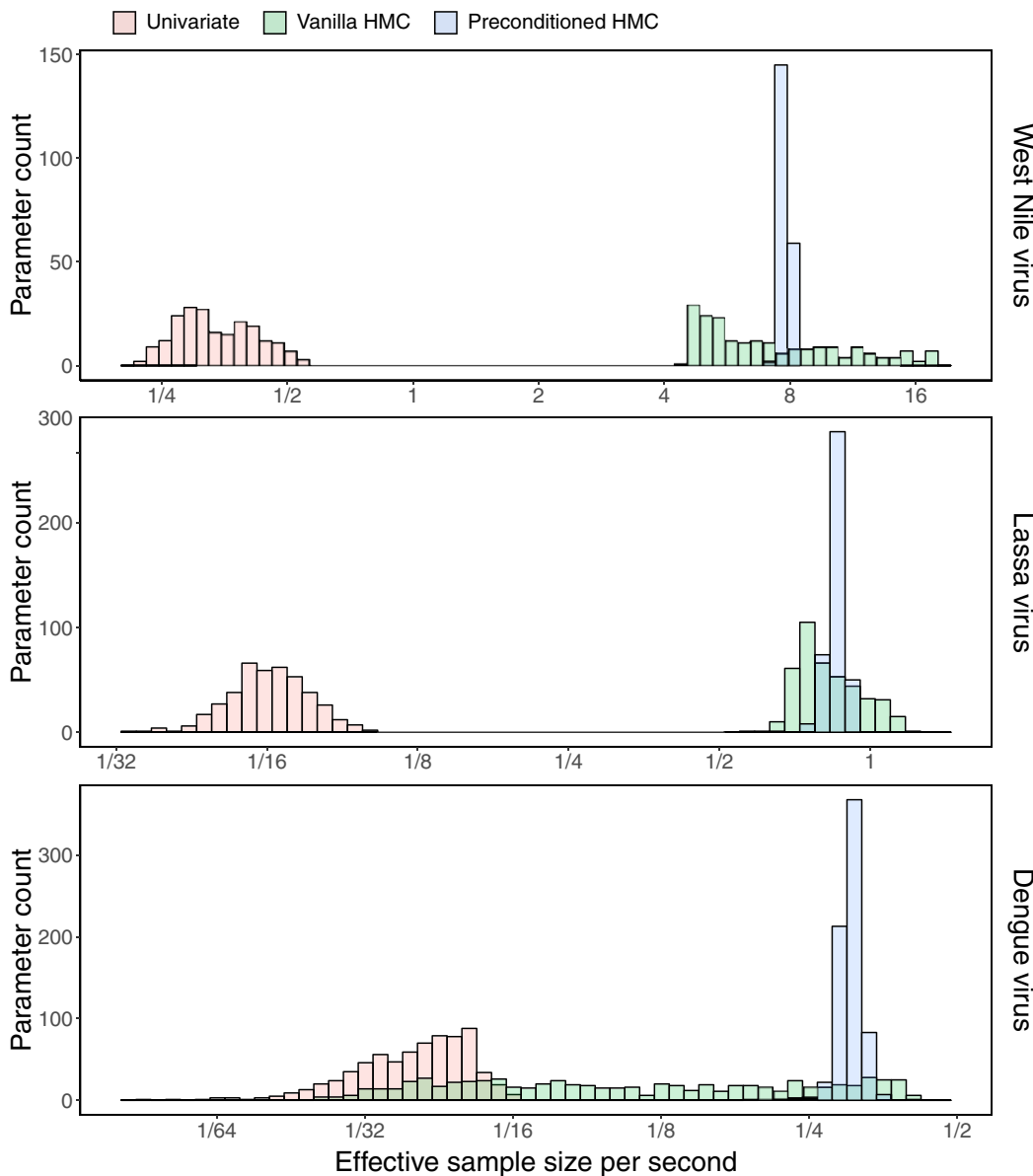


Fig. 2. Posterior sampling efficiency on all branch-specific evolutionary rate for the WNV, LASV, and DENV examples. We bin parameters by their ESS/s values. The three transition kernels employed in the MCMC are color-coded: a univariate transition kernel, a “vanilla” HMC transition kernel with an identity mass matrix, and a “preconditioned” HMC transition kernel with an adaptive mass matrix informed by the diagonal elements of the Hessian matrix.

estimating all branch-specific evolutionary rates. For each analysis, we fix the number of MCMC iterations such that they run for approximately the same time, that is, 100,000 iterations for both HMC kernels compared with 15 million iterations for the univariate kernel when analyzing the West Nile virus (WNV) data set, 50,000 iterations for both HMC kernels compared with 20 million iterations for the univariate kernel when analyzing the Lassa virus (LASV) data set, and 20,000 iterations for both HMC kernels compared with 7.5 million iterations for the univariate kernel when analyzing the Dengue virus (DENV) data set.

Figure 2 illustrates the rate estimates binned by their ESS per second for the three virus data sets, and table 2 reports the relative increase in ESS per second of the two HMC samplers compared with the univariate kernel over

all branch-specific evolutionary rates. Compared with the univariate kernel, the vHMC sampler achieves a 2.2- to 20.9-fold speedup, whereas the pHMC sampler achieves a 16.4- to 33.9-fold speedup in terms of the minimum ESS per unit time. The vHMC sampler achieves a 2.5- to 19.8-fold speedup in terms of the median ESS per unit time, whereas the pHMC sampler achieves a 7.4- to 23.9-fold speedup. The unusual spread of the ESS per second distribution for the vHMC sampler under the DENV example is likely attributable to large variation among the scales of the branch-specific evolutionary rates as discussed in more detail in Discussion. The more uniform sampling efficiency of the pHMC sampler arises from the accommodation of the variability in scales among the rates in the mass matrix.

We use BEAST (Suchard et al. 2018) in combination with BEAGLE (Ayres et al. 2019) to infer the branch-specific evolutionary rates of the three virus examples described in Materials and Methods under a random-effects relaxed clock model. The BEAST analyses comprise 20 million MCMC iterations for the WNV data set, 10 million iterations for the

LASV data set, and 60 million iterations for the DENV data set, to achieve sufficiently high ESS values for all branch-specific evolutionary rates, as assessed using Tracer (Rambaut et al. 2018). In accompanying inferred phylogeny figures, we color the branches according to their inferred posterior mean branch-specific evolutionary rate. The range of colors reflects the high variation of rates in all three virus examples.

Table 2. Relative Speedup in Terms of Effective Sample Size Per Second (ESS/s) of Our “Vanilla” HMC (vHMC) and “Preconditioned” HMC (pHMC) Transition Kernels Over a Univariate (univariate) Transition Kernel, for All Three Virus Data Sets.

		ESS/s			Speedup	
		Univariate	vHMC	pHMC	vHMC	pHMC
WNV	Minimum	0.215	4.483	7.271	20.9×	33.9×
	Median	0.326	6.446	7.793	19.8×	23.9×
LASV	Minimum	0.033	0.552	0.656	16.7×	19.8×
	Median	0.063	0.797	0.858	12.6×	13.6×
DENV	Minimum	0.011	0.025	0.187	2.2×	16.4×
	Median	0.041	0.101	0.304	2.5×	7.4×

NOTE.—We report speedup with respect to the minimum and median ESS/s across parameters for each example and method.

West Nile Virus

Our analysis estimates the tree-wise (fixed-effect) mean rate μ with posterior mean 5.67 (95% Bayesian credible interval: 5.04, 6.30) $\times 10^{-4}$ substitutions per site per year and an estimated variability characterized by the scale parameter ψ of the lognormal distributed branch-specific random effects with posterior mean 0.33 (0.21, 0.46) similar to previous estimates (Pybus et al. 2012). Figure 3 shows the maximum clade credible evolutionary tree of the WNV example. Our analysis discriminates the NY99 lineage as defined in Davis et al. (2005). The NY99 lineage is basal to all other genomes congruent with the American epidemic likely to result from the introduction of a single highly pathogenic lineage.

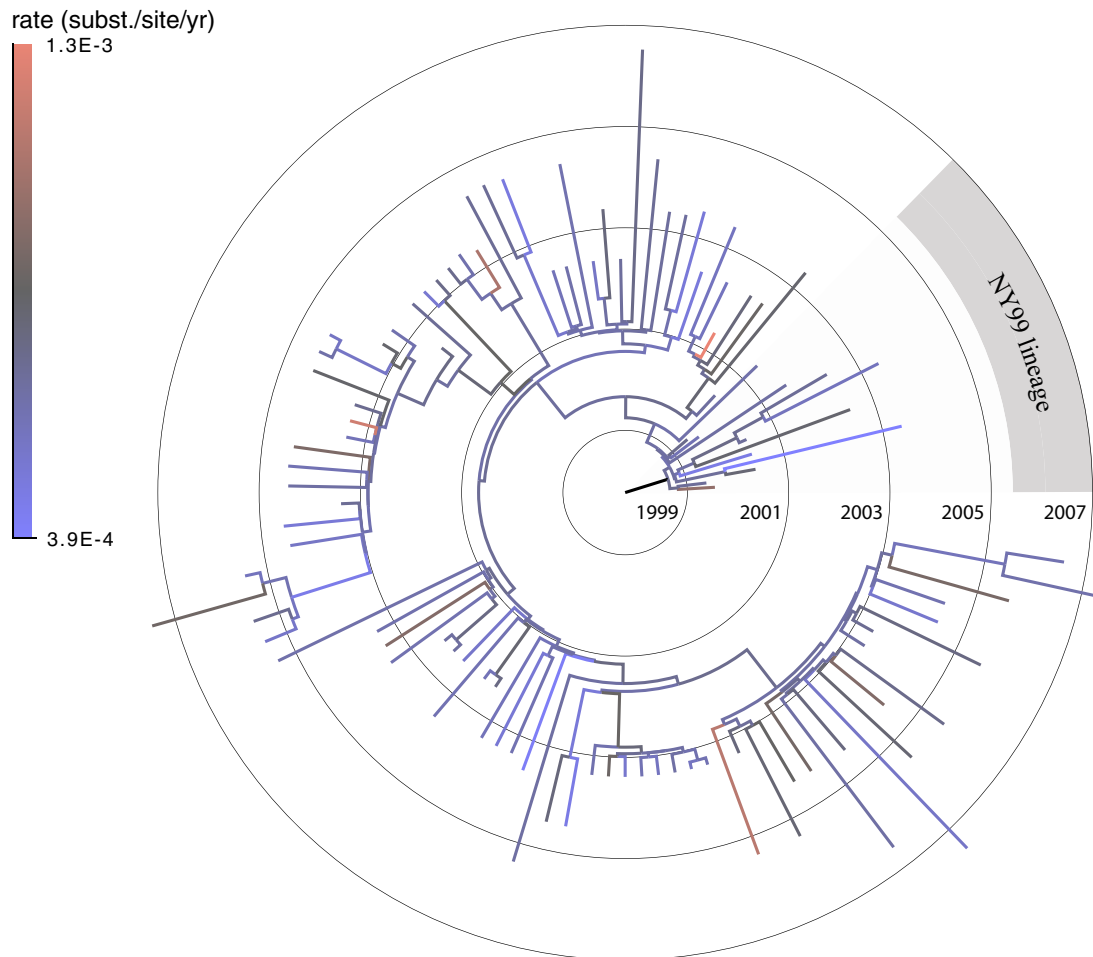


Fig. 3. Maximum clade credible tree of the WNV example. The data set consists of 104 sequences of the WNV. Branches are color-coded by the posterior means of the branch-specific evolutionary rates. The concentric circles indicate the time scale with the year numbers. The gray sector in the outer ring indicates the same 13 samples of the NY99 lineage as identified in the original study.

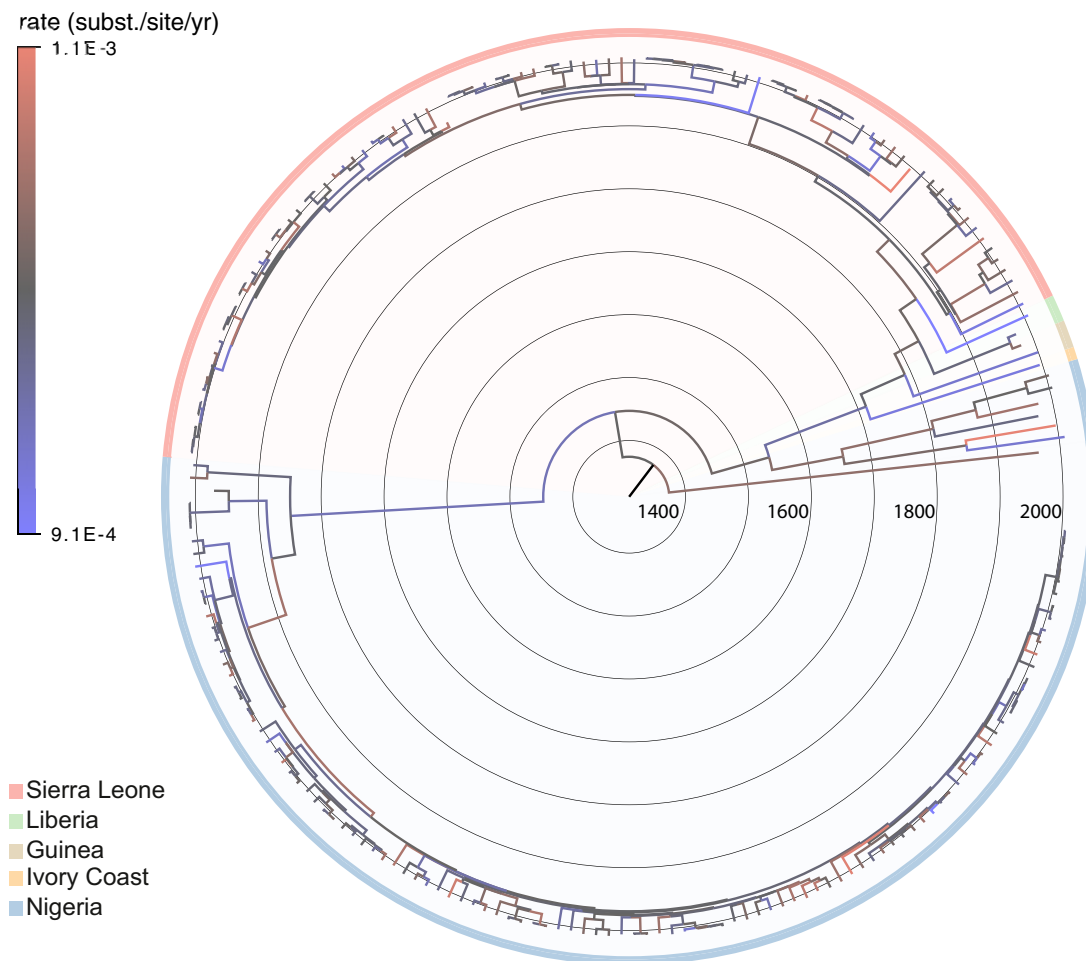


FIG. 4. Maximum clade credible tree of the LASV example. The data set consists of 211 sequences of the S segment of the LASV. Branches are color-coded by the posterior means of the branch-specific evolutionary rates according to the color bar on the top left. The concentric circles indicate the time scale with the year numbers. The outer ring indicates the geographic locations of the samples by the color code on the bottom left.

Lassa Virus

Our analysis estimates $\mu = 1.00 (0.97, 1.10) \times 10^{-3}$ substitutions per site per year for the S segment of LASV similar to previous estimates (Andersen et al. 2015; Kafetzopoulou et al. 2019), with more rate variability ($\psi = 0.088[0.029, 0.142]$) as compared with WNV. Figure 4 shows the maximum clade credible evolutionary tree of the LASV example. Our result agrees with LASV being a long-standing human pathogen that likely originated in modern-day Nigeria more than a thousand years ago and spread into neighboring West African countries within the last several hundred years (Andersen et al. 2015; Kafetzopoulou et al. 2019).

Dengue Virus

Our analysis estimates $\mu = 4.75 (4.05, 5.33) \times 10^{-4}$ substitutions per site per year for serotype 3 of DENV similar to previous estimates (Allicock et al. 2012; Nunes et al. 2014), with the largest rate variability of all examples analyzed here ($\psi = 1.26[1.06, 1.45]$). Figure 5 shows the maximum clade credible evolutionary tree of the DENV example. We identify the same two Brazilian lineages as in Nunes et al. (2014), and both lineages appear to originate from the Caribbean.

Discussion

We presented a new algorithm for evaluating the gradient of the phylogenetic model likelihood w.r.t. branch-specific parameters. Our approach achieves linear complexity in the number of sequences by complementing the postorder traversal in Felsenstein's pruning algorithm (Felsenstein 1973, 1981) with its reverse preorder traversal. The two traversals together complete Baum's forward-backward algorithm (Baum 1972). Schadt et al. (1998) previously employed the forward-backward algorithm to calculate the likelihood and its gradient w.r.t. the relatively small number of parameters that characterize a generalized Kimura (1980) CTMC. On the other hand, pruning-only-based gradient algorithms have made improvements over the past few years that scale $\mathcal{O}(N h)$ instead of $\mathcal{O}(N^2)$ where h is the total level of the tree (Kenney and Gu 2012). However, in many phylogenetic problems with nonneutral evolutionary processes, h is often much closer to N than $\log N$. Careful reuse of some computations when properly rerooting the tree can further accelerate the pruning-based gradient method. Unfortunately, rerooting the tree requires the CTMC to be time-reversible and at stationarity. The assumptions of reversibility and stationarity can be biologically unreasonable but are often kept for simplicity and

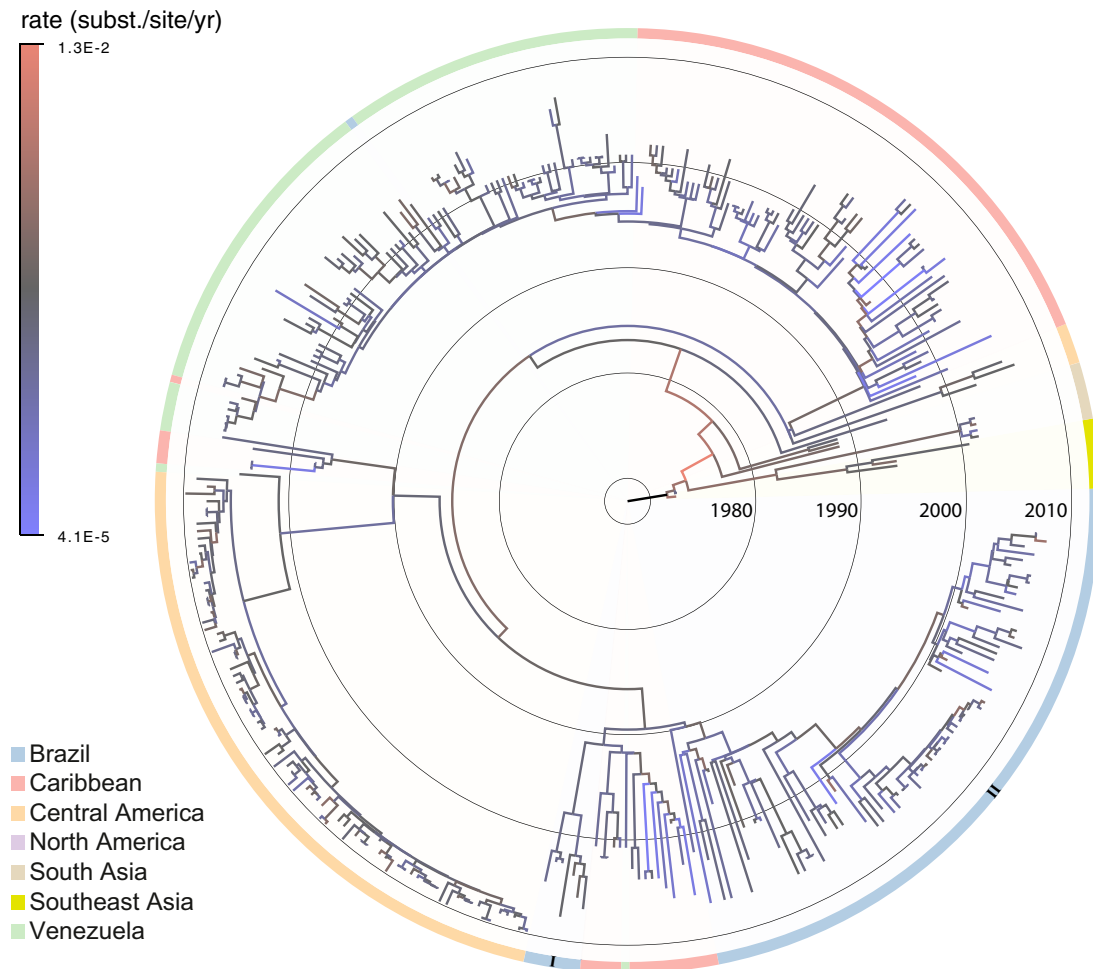


Fig. 5. Maximum clade credible tree of the DENV example. The data set consists of 352 sequences of the serotype 3 of the DENV. Branches are color-coded by the posterior means of the branch-specific evolutionary rates according to the color bar on the top left. The concentric circles indicate the time scale with the year numbers. The outer ring indicates the geographic locations of the samples by the color code on the bottom left. “I” and “II” indicate the two Brazilian lineages as in the original study.

computational tractability. Our linear-time gradient algorithm extends the approach in Schadt et al. (1998) to general CTMCs. Our algorithm does not require any model assumptions on stationarity or reversibility and can be applied to both homogeneous and nonhomogeneous Markov processes.

Our algorithm calculates the likelihood and its gradient w.r.t. all branch-specific parameters through the postorder and the complementary preorder traversal. One essential benefit of the proposed algorithm is that it calculates the gradient w.r.t. a collection of branch-specific parameters (e.g., evolutionary rate and time parameters) at the same time with no additional cost for caching. However, the computational load is not identical for the two traversals. For example, the postorder traversal calculates the transition probabilities at all branches that can be reused in the preorder traversal (see eqs. 9 and 10). Moreover, the preorder traversal updates approximately twice as many partial likelihood vectors as the postorder traversal. This difference is due to the additional preorder partial likelihood vectors at the tip nodes together with the post- and preorder partial likelihood vectors at the internal nodes.

Interestingly, one can also use the post- and preorder partial likelihood vectors to obtain the gradient w.r.t. any (possibly tree-wise) parameter θ that characterizes \mathbf{Q}_i . To accomplish this task, we first substitute $\mathbf{Q}_i \rightarrow \mathbf{P}_i^{-1} \frac{\partial \mathbf{P}_i}{\partial \theta}$ in equations (9) and (12) (see, e.g., Kalbfleisch and Lawless 1985 for obtaining the partial differential matrices). We then sum these contributions up over all branches. For $\theta = \pi$, the stationary distribution, an additional gradient contribution may arise at the root node. Depending on the dimensionality of θ , however, computing numerical gradient approximations through multiple likelihood evaluations may be faster.

Through our three example data sets, we illustrate the use of our gradient algorithm in both maximum-likelihood and Bayesian analyses. We show that our new algorithm can considerably accelerate inference in both frameworks. In the maximum-likelihood analyses, we compare the performance of the L-BFGS optimization method using our gradient algorithm with the same optimizer but using a central finite difference numerical gradient algorithm. We choose this numerical scheme for two reasons. One is that the central scheme has only roughly twice the computational cost as

pruning-based analytical gradient methods. The other reason is to investigate the influence of numerical error in optimization. The observed per-iteration speedup with our gradient algorithm increases with increasing number of sequences in the data set. This finding is consistent with our gradient algorithm being a linear-time algorithm in the number of sequences as opposed to quadratic pruning-based algorithms. We also observe slightly more iterations in the optimization with the numeric gradient than with the proposed analytic gradient method. Moreover, for all three data sets, the optimization with our analytic gradient method ends with slightly higher log-likelihood values at the fifth digit after the decimal point with the same stopping criteria. The ℓ^2 -norm of the gradient when the optimization stops is higher with the numerical method suggesting early termination due to numerical trouble. Numerical error builds up from the matrix exponential calculations and propagates along the tree.

A caveat of our optimization comparison is that we do not compare with other widely used optimization criteria. For example, GARLI (Zwickl 2006) and RAxML (Stamatakis et al. 2005) incorporate local optimization routines in addition to global optimization. The purpose of local optimization is partly to avoid the computational burden of optimizing all branches simultaneously, especially after a topological rearrangement. For time-reversible models at stationarity, with properly rerooting the tree, the branch lengths in the vicinity of a topological rearrangement can be efficiently optimized via the Newton–Raphson method incorporating both the gradient and the Hessian information for one branch at a time. However, such optimization strategy is only efficient for optimization over a limited number of parameters, because the computational complexity for evaluating the Hessian matrix increases quadratically with the number of parameters.

In the Bayesian analyses, our linear-time gradient algorithm allows efficient sampling of all branch-specific evolutionary rates from their posterior density using HMC. The vanilla HMC sampler gains a 2.2- to 20.9-fold increase in learning the branch-specific rates with the minimum ESS per unit time criterion. The preconditioning improves the efficiency of HMC with a 16.4- to 33.9-fold increase. The computational cost for evaluating the diagonal entries of the Hessian matrix is almost the same as the gradient (see eq. 13). In fact, the first term is nearly identical to the gradient in equation (12) except for replacing the infinitesimal matrix \mathbf{Q}_i and the discrete rate γ_i by their quadratic forms. The second term in equation (13) reuses the gradient evaluated at the current position from the cached values for updating the momentum (see eq. 16). Moreover, we update the adaptive preconditioning mass matrix every ten iterations of the HMC sampler. This adaptation limits the additional computational cost in evaluating the diagonal of the Hessian matrix.

We observe an inverse correlation between the variability of the scales among the branch-specific evolutionary rates and the spread of ESS per second for the “vanilla” HMC sampler as shown in figure 2. Specifically, using the standard deviation (SD) of the marginal posterior distribution as a qualitative measure for the scale, the WNV, LASV, and

DENV examples return a variance across the SDs of all branch-specific evolutionary rates as 0.014, 0.006, and 0.036 and the ratio between the maximum and the minimum of the SDs being 2.2, 1.7, and 17.8, respectively. The branch-specific evolutionary rates of the DENV example exhibit the highest variability among the three data sets and the “vanilla” HMC sampler performs the worst for this data set. As discussed in Hamiltonian Monte Carlo Sampling section, not accounting for high variability among the scales of the parameters reduces the efficiency of the “vanilla” HMC sampler. Preconditioning improves the inadequate performance of the “vanilla” HMC sampler via the adaptive mass matrix informed by the diagonal elements of the Hessian. The mass matrix incorporates the variation in scales among the branch-specific evolutionary rates with a negligible cost of additional computation.

Finally, although our examples jointly infer topology, branch-specific rates and other model parameters, we report efficiency gains while conditioning on a single topology to avoid identifiability issues that arise across the rates when the topology changes. Common across Bayesian phylogenetics, our Metropolis-with-Gibbs (Tierney 1994; Andrieu et al. 2003) inference strategy cycles between sampling the topology, the rates and then the other models, each from their respective full conditional distributions. As expected, sampling the high-dimensional rates remains rate-limiting, so their efficiency gain is the most germane. We expect, however, that increased sampling efficiency conditional on one topology also helps us explore topology space by decreasing autocorrelation along the Metropolis-with-Gibbs cycle, but this requires future work to justify more fully.

Materials and Methods

Implementation

We have implemented a central processing unit (CPU) version of the algorithm in this manuscript within the development branch of the software package BEAGLE (Ayres et al. 2019). We employ these extensions within the development branch of BEAST (Suchard et al. 2018) for the demonstrations in this manuscript. We provide instructions and the BEAST XML files for reproducing these analyses on Github at https://github.com/suchard-group/hmc_clock_manuscript_supplement.

Emerging Viral Sequences

We examine the molecular evolution of WNV in North America (1999–2007), the S segment of LASV in West Africa (2008–2013) and serotype 3 of DENV in Brazil (1964–2010) (Pybus et al. 2012; Nunes et al. 2014; Andersen et al. 2015). In all three virus data sets, phylogenetic analyses have revealed a high variation of the evolutionary rates across branches in the underlying phylogeny.

West Nile Virus

WNV is a mosquito-borne RNA virus with birds as the primary host. The first detected case in United States was in New York City in August 1999, and the virus reached the American west coast by 2004. In total, human infections resulted in over

1,200 deaths. The data consist of 104 full genomes, with a total alignment length of 11,029 nucleotides, and were collected from infected human plasma samples from 2003 to 2007 as well as near-complete genomes obtained from GenBank (Pybus et al. 2012).

Lassa Virus

Every year, LASV is responsible for thousands of deaths and tens-of-thousands of hospitalizations (Andersen et al. 2015). Although many LASV infections are subclinical, they can also lead to Lassa fever, a hemorrhagic fever similar to that caused by Ebola virus. Perhaps less well-known than Ebola viral disease, Lassa fever can nonetheless lead to over 50% fatality rates among hospitalized patients. Unlike Ebola virus, which passes directly between humans, LASV circulates in a rodent (*Mastomys natalensis*) reservoir and mainly infects humans through contact with rodent excreta. LASV is a single-stranded RNA virus with a genome consisting of two segments: the L segment is 7.3 kilobase pairs (kb) long; the S segment is 3.4 kb long. In this manuscript, we use the S segment of the LASV sequence data set of Andersen et al. (2015) that consists of 211 samples obtained at clinics in both Sierra Leone and Nigeria, rodents in the field, laboratory isolates, and previously sequenced genomes.

Dengue Virus

Worldwide, DENV infects close to 400 million people and causes >25,000 deaths annually. Much like the LASV, DENV can also lead to hemorrhagic fever that is often referred to as “breakbone fever” on account of the severe joint and muscle pain it causes. DENV is endemic to the tropics and subtropics, with mosquitoes transmitting the virus between humans. Nunes et al. (2014) selected 352 serotype 3 DENV (DENV-3) sequences from a total of 639 complete DENV genomes based on genetic diversity and maximization of the sampling interval. The sample collection ranged from 1964 to 2010 within a total of 31 distinct countries in Southeast Asia, North America, Central America, the Caribbean, and South American countries.

Acknowledgments

We thank Jeffrey Thorne for thoughtful comments. The research leading to these results has received funding from the European Research Council under the European Union’s Horizon 2020 research and innovation program (Grant Agreement No. 725422—ReservoirDOCS). The Artic Network receives funding from the Wellcome Trust through project 206298/Z/17/Z. M.A.S. and X.J. are partially supported by NSF Grant DMS 1264153 and NIH Grants R01 AI107034 and U19 AI135995. A.H. acknowledges support by NIH-NIAID grant K25AI153816. G.B. acknowledges support from the Interne Fondsen KU Leuven/Internal Funds KU Leuven under grant agreement C14/18/094. P.L. acknowledges support by the Research Foundation—Flanders (“Fonds voor Wetenschappelijk Onderzoek—Vlaanderen,” G066215N, G0D5117N, and G0B9317N).

References

- Adachi J, Hasegawa M. 1996. MOLPHY version 2.3: programs for molecular phylogenetics based on maximum likelihood. Tokyo: Institute of Statistical Mathematics.
- Allcock OM, Lemey P, Tatem AJ, Pybus OG, Bennett SN, Mueller BA, Suchard MA, Foster JE, Rambaut A, Carrington CV. 2012. Phylogeography and population dynamics of dengue viruses in the Americas. *Mol Biol Evol.* 29(6):1533–1543.
- Andersen KG, Shapiro BJ, Matranga CB, Sealfon R, Lin AE, Moses LM, Folarin OA, Goba A, Ochia I, Ehiane PE, et al. 2015. Clinical sequencing uncovers origins and evolution of Lassa virus. *Cell* 162(4):738–750.
- Andrieu C, De Freitas N, Doucet A, Jordan MI. 2003. An introduction to MCMC for machine learning. *Mach Learn.* 50(1/2):5–43.
- Andrieu C, Thoms J. 2008. A tutorial on adaptive MCMC. *Stat Comput.* 18(4):343–373.
- Aris-Brosou S, Yang Z. 2002. Effects of models of rate evolution on estimation of divergence dates with special reference to the meta-zoan 18S ribosomal RNA phylogeny. *Syst Biol.* 51(5):703–714.
- Ayres DL, Cummings MP, Baele G, Darling AE, Lewis PO, Swofford DL, Huelsenbeck JP, Lemey P, Rambaut A, Suchard MA. 2019. BEAGLE 3: improved performance, scaling and usability for a high-performance computing library for statistical phylogenetics. *Syst Biol.* 68(6):1052–1061.
- Baum L. 1972. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities* 3:1–8.
- Beskos A, Pillai N, Roberts G, Sanz-Serna J-M, Stuart A. 2013. Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli* 19(5A):1501–1534.
- Bletsa M, Suchard MA, Ji X, Gryseels S, Vrancken B, Baele G, Worobey M, Lemey P. 2019. Divergence dating using mixed effects clock modelling: an application to HIV-1. *Virus Evol.* 5(2):vez036.
- Bloom DE, Black S, Rappuoli R. 2017. Emerging infectious diseases: a proactive approach. *Proc Natl Acad Sci U S A.* 114(16):4055–4059.
- Bryant D, Galtier N, Poursat M-A. 2005. Likelihood calculation in molecular phylogenetics. *Math Evol Phylogeny.* 33–62.
- Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A. 2017. Stan: A probabilistic programming language. *J Stat Softw.* 76(1):1–32.
- Davis CT, Ebel GD, Lanciotti RS, Brault AC, Guzman H, Siirin M, Lambert A, Parsons RE, Beasley DW, Novak RJ, et al. 2005. Phylogenetic analysis of North American West Nile virus isolates, 2001–2004: evidence for the emergence of a dominant genotype. *Virology* 342(2):252–265.
- Dennis JE Jr, Schnabel RB. 1996. Numerical methods for unconstrained optimization and nonlinear equations. *Classics Appl. Math.* 16. Philadelphia: SIAM.
- Drummond AJ, Ho SY, Phillips MJ, Rambaut A. 2006. Relaxed phylogenetics and dating with confidence. *PLoS Biol.* 4(5):e88.
- Drummond AJ, Suchard MA. 2010. Bayesian random local clocks, or one rate to rule them all. *BMC Biol.* 8(1):114.
- Felsenstein J. 1973. Maximum likelihood and minimum-steps methods for estimating evolutionary trees from data on discrete characters. *Syst Biol.* 22(3):240–249.
- Felsenstein J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol.* 17(6):368–376.
- Ferreira MA, Suchard MA. 2008. Bayesian analysis of elapsed times in continuous-time Markov chains. *Can J Stat.* 36(3):355–368.
- Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB. 2013. Bayesian data analysis. 3rd ed. New York: Chapman and Hall/CRC.
- Girolami M, Calderhead B. 2011. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J R Stat Soc B.* 73(2):123–214.
- Hairer E, Lubich C, Wanner G. 2006. Geometric numerical integration: structure-preserving algorithms for ordinary differential equations. Springer Ser. Comput. Math. 31. Berlin: Springer-Verlag.
- Hasegawa M, Kishino H, Yano T-A. 1989. Estimation of branching dates among primates by molecular clocks of nuclear DNA which slowed down in Hominoidea. *J Hum Evol.* 18(5):461–476.

- Ho SY, Duchêne S. 2014. Molecular-clock methods for estimating evolutionary rates and timescales. *Mol Ecol*. 23(24):5947–5965.
- Huelsenbeck JP, Larget B, Swofford D. 2000. A compound Poisson process for relaxing the molecular clock. *Genetics* 154(4):1879–1892.
- Huelsenbeck JP, Ronquist F, Nielsen R, Bollback JP. 2001. Bayesian inference of phylogeny and its impact on evolutionary biology. *Science* 294(5550):2310–2314.
- Kafetzopoulou L, Pullan S, Lemey P, Suchard M, Ehichioya D, Pahlmann M, Thielebein A, Hinzmann J, Oestereich L, Wozniak D, et al. 2019. Metagenomic sequencing at the epicenter of the Nigeria 2018 Lassa fever outbreak. *Science* 363(6422):74–77.
- Kalbfleisch J, Lawless JF. 1985. The analysis of panel data under a Markov assumption. *J Am Stat Assoc*. 80(392):863–871.
- Kenney T, Gu H. 2012. Hessian calculation for phylogenetic likelihood based on the pruning algorithm and its applications. *Stat Appl Genet Mol Biol*. 11(4):1–46.
- Kimura M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J Mol Evol*. 16(2):111–120.
- Kishino H, Miyata T, Hasegawa M. 1990. Maximum likelihood inference of protein phylogeny and the origin of chloroplasts. *J Mol Evol*. 31(2):151–160.
- Kishino H, Thorne JL, Bruno WJ. 2001. Performance of a divergence time estimation method under a probabilistic model of rate evolution. *Mol Biol Evol*. 18(3):352–361.
- Kruschke J. 2014. Doing Bayesian data analysis. A tutorial with R, JAGS, and Stan. 2nd ed. New York: Academic Press.
- Kumar S. 2005. Molecular clocks: four decades of evolution. *Nat Rev Genet*. 6(8):654–662.
- Lange K. 2013. Optimization. Springer Texts in Statistics. New York: Springer.
- Lartillot N, Phillips MJ, Ronquist F. 2016. A mixed relaxed clock model. *Phil Trans R Soc B*. 371(1699):20150132.
- Lemey P, Rambaut A, Welch JJ, Suchard MA. 2010. Phylogeography takes a relaxed random walk in continuous space and time. *Mol Biol Evol*. 27(8):1877–1885.
- Livingstone S, Girolami M. 2014. Information-geometric Markov chain Monte Carlo methods using diffusions. *Entropy* 16(6):3074–3102.
- Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. 1953. Equation of state calculations by fast computing machines. *J Chem Phys*. 21(6):1087–1092.
- Monnahan CC, Thorson JT, Branch TA. 2017. Faster estimation of Bayesian models in ecology using Hamiltonian Monte Carlo. *Methods Ecol Evol*. 8(3):339–348.
- Neal RM. 2011. MCMC using Hamiltonian dynamics. In: Brooks S, Gelman A, Jones G, Meng XL, editors. Handbook of Markov Chain Monte Carlo. Oxford: Taylor & Francis Group.
- Nishimura A, Dunson D. 2016. Geometrically tempered Hamiltonian Monte Carlo. *arXiv: 1604.00872*.
- Nocedal J, Wright S. 2006. Numerical optimization. 2nd ed. Springer Science & Business Media. New York: Springer.
- Nunes MR, Palacios G, Faria NR, Sousa EC Jr, Pantoja JA, Rodrigues SG, Carvalho VL, Medeiros DB, Savji N, Baele G, et al. 2014. Air travel is associated with intracontinental spread of dengue virus serotypes 1–3 in Brazil. *PLoS Negl Trop Dis*. 8(4):e2769.
- Ogden TH, Rosenberg MS. 2006. Multiple sequence alignment accuracy and phylogenetic inference. *Syst Biol*. 55(2):314–328.
- Pybus OG, Suchard MA, Lemey P, Bernardin FJ, Rambaut A, Crawford FW, Gray RR, Arinaminpathy N, Stramer SL, Busch MP, et al. 2012. Unifying the spatial epidemiology and molecular evolution of emerging epidemics. *Proc Natl Acad Sci U S A*. 109(37):15066–15071.
- Pybus OG, Tatem AJ, Lemey P. 2015. Virus evolution and transmission in an ever more connected world. *Proc R Soc B*. 282(1821):20142878.
- Quick J, Loman NJ, Duraffour S, Simpson JT, Severi E, Cowley L, Bore JA, Koundouno R, Dudas G, Mikhail A, et al. 2016. Real-time, portable genome sequencing for Ebola surveillance. *Nature* 530(7589):228–232.
- Rambaut A, Drummond AJ, Xie D, Baele G, Suchard MA. 2018. Posterior summarization in Bayesian phylogenetics using Tracer 1.7. *Syst Biol*. 67(5):901–904.
- Rannala B, Yang Z. 2007. Inferring speciation times under an episodic molecular clock. *Syst Biol*. 56(3):453–466.
- Salvatier J, Wiecki TV, Fonnesbeck C. 2016. Probabilistic programming in Python using PyMC3. *PeerJ Comput Sci*. 2:e55.
- Sanderson MJ, McMahon MM, Stamatakis A, Zwickl DJ, Steel M. 2015. Impacts of terraces on phylogenetic inference. *Syst Biol*. 64(5):709–726.
- Schadt EE, Sinsheimer JS, Lange K. 1998. Computational advances in maximum likelihood methods for molecular phylogeny. *Genome Res*. 8(3):222–233.
- Shapiro B, Rambaut A, Drummond AJ. 2006. Choosing appropriate substitution models for the phylogenetic analysis of protein-coding sequences. *Mol Biol Evol*. 23(1):7–9.
- Stamatakis A, Ludwig T, Meier H. 2005. RAxML-iii: a fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics* 21(4):456–463.
- Suchard MA, Lemey P, Baele G, Ayres DL, Drummond AJ, Rambaut A. 2018. Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. *Virus Evol*. 4(1):vey016.
- Thorne JL, Kishino H, Painter IS. 1998. Estimating the rate of evolution of the rate of molecular evolution. *Mol Biol Evol*. 15(12):1647–1657.
- Tierney L. 1994. Markov chains for exploring posterior distributions. *Ann Statist*. 22(4):1701–1728.
- Yang Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. *J Mol Evol*. 39(3):306–314.
- Yang Z. 1996. Maximum-likelihood models for combined analyses of multiple sequence data. *J Mol Evol*. 42(5):587–596.
- Yang Z. 2000. Maximum likelihood estimation on large phylogenies and analysis of adaptive evolution in human influenza virus A. *J Mol Evol*. 51(5):423–432.
- Yoder AD, Yang Z. 2000. Estimation of primate speciation dates using local molecular clocks. *Mol Biol Evol*. 17(7):1081–1090.
- Zuckerandl E, Pauling L. 1965. Evolutionary divergence and convergence in proteins. In: Bryson V, Vogel HJ, editors. Evolving genes and proteins. New York: Elsevier. p. 97–166.
- Zuckerandl E, Pauling LB. 1962. Molecular disease, evolution and genic heterogeneity. In: Kasha M, Pullman B, editors. Horizons in biochemistry. New York: Academic Press. p. 189–225.
- Zwickl DJ. 2006. Genetic algorithm approaches for the phylogenetic analysis of large biological sequence data sets under the maximum likelihood criterion [Ph.D. thesis]. Austin (TX): The University of Texas.