Log Transformation Improves Dating of Phylogenies

Uyen Mai[1] and Siavash Mirarab[2*]

[1] *Department of Computer Science and Engineering, UC San Diego, CA 92093, USA*
[2] *Department of Electrical and Computer Engineering, UC San Diego, CA 92093, USA*
[*] Corresponding author: smirarab@ucsd.edu

Abstract

Phylogenetic trees can only be inferred from sequence data with branch lengths estimated in the unit of the expected number of substitutions. The inferred trees give an incomplete view of evolutionary histories since no timing information can be inferred from these trees, but many applications of phylogenies require trees that have branch lengths in the unit of time. Many methods have been developed to translate the inferred edge lengths from substitution unit into the unit of time using calibration points, but none is universally accepted as they are challenged in both scalability and accuracy under complex models. Here, we introduce a new method that formulates dating as a non-convex optimization problem, where the variance of log-transformed rate multipliers are minimized across the tree. On simulated and real data, we show that our method, wLogDate, is often more accurate than alternatives and is more robust to model assumptions.

Keywords: Divergence time estimation, phylogenetic dating, non-convex optimization

1 Introduction

Phylogenetic inference from sequence data fundamentally cannot reveal exact timing of events unless paired with external information. Sequence data gives us estimates of the level of divergence, which is a function of both time and mutation rates. As such, the inferred phylogenies measure branch lengths in the unit of expected number of substitutions per site. Dating a phylogeny is the process of translating substitution branch lengths to unit of time. While dating is a difficult process [1], it is crucial for understanding evolutionary processes [2, 3] and is a fundamental need in many clinical applications of phylogenetics and phylodynamics [4].

Dating requires soft or hard constraints for the timing of *some* nodes of the tree, leaving the timing of the remaining nodes to be inferred. Timing information is often represented in the form of calibration points (e.g., obtained by carbon-dating relevant fossils [5]) and sampling times for leaves of the tree (relevant for fast-evolving viruses and bacteria).

A large number of computational methods for dating phylogenies are available [1]. A main point of differentiation between these methods is the clock model they assume; i.e., a model for how rates of mutation can vary with time [6]. Some rely on strict molecular clock [7] where rates are assumed to be close to fixed [e.g., 8–10] while others allow rates to be drawn independently from a stationary distribution [11–13]. In addition, there are methods that model evolution of rates with time [14] and allow correlated rates across branches [15–20] or even genes [21].

Another distinction between methods is the use of parametric models [22]. Most dating methods rely on an explicit parametric statistical models and formulate dating as estimating a set of parameters in a maximum likelihood (ML) or Bayesian inference framework. Other methods rely on optimization problems that may have model assumptions but do not compute likelihood under an explicit statistical model. When the assumed parametric models are close to the reality, we expect them to outperform non-parametric models. However, parametric methods can be sensitive to model deviations, a problem that non-parametric methods promise to fix.

In this paper, we introduce a new method of dating phylogenies based on a non-parametric approach. Our method, called LogDate, is based on a simple idea. We define rates necessary

to obtain time unit branch lengths as the product of a single overall rate and a set of rate multiplicators, one per branch. We then seek to find the overall rate and all rate multiplicators such that the log-transformed rate multiplicators have the minimum variance. This formulation gives us a constrained optimization problem, which although not convex, can be solved in a scalable fashion using the standard approaches such as sequential least squares programming. Formulation of dating as an optimization problem is not new [8, 9]. Our main contribution here is the realization that log-transforming rate multipliers can produce more accurate dates. While the method is non-parametric, it corresponds to the ML estimate under specific parameters of a statistical model. Finally, in extensive simulation studies and on real data we show that LogDate has higher accuracy in inferring node times compared to alternative methods, including some that rely on time-consuming Bayesian inference.

2 Methods

2.1 Notations

For a binary tree T with n leaves, we let $[0 \dots n-2]$ uniquely index internal nodes (root is assigned 0) and let $[n-1 \dots 2n-2]$ index leaves. Let $par(i)$ denote the parent of node i and let $c_l(i)$ and $c_r(i)$ denote the left and right children of node i, respectively. We refer to the edge incident to node i (i.e. the edge between $par(i)$ and i) as e_i. We can measure branch lengths ot T in two units. We let t_i denote the divergence time of node i; i.e. the time when species i diverged into $c_l(i)$ and $c_r(i)$. Then, $\tau_i = t_i - t_{par(i)}$ is the length of the edge e_i in time unit; let $\tau = [\tau_1, \tau_2, \dots, \tau_{2n-2}]$. We then let b_i be the length of e_i measured in the unit of the expected number of substitutions incurred along edge e_i and define $\mathbf{b} = \{b_1 \dots b_{2n-1}\}$. Note that we can write $b_i = \tau_i \times \mu_i$ where μ_i is the rate of mutation along the branch e_i. Finally, note that models of sequence evolution depend on b_i and thus, μ_i or τ_i are not identifiable. Moreover, $\mathbf{b}$ is not observable directly. Instead, sequences with length s evolve down the tree, from which we can estimate $\hat{\mathbf{b}} = \{\hat{b}_1, \hat{b}_2, \dots, \hat{b}_{2n-1}\}$ using standard approaches (e.g., ML).

2.2 Formulating dating as an optimization problem

Given inferred substitution unit branch lengths $\hat{\mathbf{b}}$ and some external information on the divergence time of some nodes, the goal of phylogenetic dating is to find τ. Let $\mathbf{f} = [f_1 \dots f_{2n-2}]$ be unknown variables where each $f_i = \tau_i/\hat{b}_i$. Our goal of finding τ is identical to finding $\mathbf{f}$. Assuming elements of $\mathbf{f}$ are distributed around an overall rate μ, we define $\nu_i = f_i\mu = \mu\tau_i/\hat{b}_i$. We formulate dating as a constrained optimization problem on $2n-1$ variables $\mathbf{x} = [\nu_1, \nu_2, \dots, \nu_{2n-2}, \mu]$. Calibration and sampling points define linear constraints on $\mathbf{x}$ (Section 2.2.1). Because constraints do not fully determine τ, we need extra assumptions on $\mathbf{x}$, which would be motivated by properties of the distribution of f_i. The choice of the objective function and the optimization strategies distinguish different dating approaches and are our main contributions in this paper.

2.2.1 Constructing linear constraints

For each edge e_i we need to find the scaling factor f_i such that $\tau_i = f_i\hat{b}_i$ and for any pair of nodes (i, j) associated with sampling times (t_i, t_j), the following constraint $\psi(i, j)$ must be satisfied

$$\psi(i,j) : \mu(t_j - t_i) = \sum_{k \in P(m,j)} \nu_k \hat{b}_k - \sum_{k \in P(i,m)} \nu_k \hat{b}_k \tag{1}$$

where m is the LCA of i and j and $P(u, v)$ is the path connecting u and v. Thus, given k time points, $k(k-2)/2$ constraints must hold. However, these constraints are not independent of each other; in fact, only $k-1$ of these constraints imply all others. Let t_0 be the *unknown* time at the root of the tree. For k calibration points $t_1, \dots, t_k$, we can setup k constraints of the form:

$$C_i : \mu(t_i - t_0) = \sum_{k \in P(0,i)} \nu_k \hat{b}_k \tag{2}$$

Now note that for any pair (i, j), the linear constraint given in Eq. 1 can be derived by $C_j - C_i$. Finally, we can remove t_0 from the set of constraints by subtracting C_1 from each of the other constraints $C_2, \ldots, C_k$. This gives us the final $k-1$ linear constraints on ν, which we denote as Ψ. Algorithm 1 (in Appendix S1) gives a simple way to setup these constraints in linear time.

2.2.2 Optimization Criteria

The choice of the optimization criteria depends on our assumption on the distribution of f_i, which is a combination of the distribution of μ_i and the random process that generates $\hat{b}_i$ from b_i. In traditional strict-clock model [7] (e.g., as used by LSD [8] and LF [9]), a constant evolutionary rate is assumed throughout the tree ($\forall_i \mu_i = \mu$). Under this model, the distribution of f_i is determined by deviations of $\hat{b}_i$ from b_i.

In LF [9], the number of substitutions *per sequence site* on a branch i is modelled by a Poisson distribution with mean $\lambda = \mu\tau_i$. The total *expected* number of substitutions on branch i is $sb_i = s\mu\tau_i$ and the total *observed* number of substitutions can be modeled as $s\hat{b}_i \sim Poisson(s\mu\tau_i)$. Thus, the log likelihood function can be written as:

$$\log L^P = \sum_{i=1}^{2n-2} \Big(s\hat{b}_i \log(s\mu\tau_i) - s\mu\tau_i - \log((s\hat{b}_i)!) \Big) = \sum_{i=1}^{2n-2} \Big(s\hat{b}_i \log(s\hat{b}_i) - \log((s\hat{b}_i)!) \Big) + \sum_{i=1}^{2n-2} s\hat{b}_i \big(\log \nu_i - \nu_i \big) \ .$$

Given s and $\hat{b}_i$, we can find $\mathbf{x}$ by maximizing $\log L^P$ w.r.t. the constraints. Thus, we seek:

$$\mathbf{x}^* = \arg\min_{\nu} \sum_{i=1}^{2n-2} \hat{b}_i \big(\nu_i - \log \nu_i \big) \qquad \text{w.r.t. } \Psi \text{ constraints.} \tag{3}$$

In LSD [8], $\hat{b}_i$ is assumed to follow a Gaussian model $\hat{b}_i \sim Gaussian(\mu\tau_i, \sigma_i^2)$ and the variance is approximated by $\hat{b}_i/s$ (the method includes further smoothing strategies omitted here). The log-likelihood function under this Gaussian model can therefore be written as follow:

$$\log L^G = \sum_{i=1}^{2n-2} -\frac{1}{\sigma_i^2} (\hat{b}_i - \mu\tau_i)^2 \approx \sum_{i=1}^{2n-2} -\frac{s}{\hat{b}_i} (\hat{b}_i - \mu\tau_i)^2 = \sum_{i=1}^{2n-2} -s\hat{b}_i (1 - \nu_i)^2$$

Thus, the ML estimate can be formulated as:

$$\mathbf{x}^* = \arg\min_{\nu} \sum_{i=1}^{2n-2} \hat{b}_i (1 - \nu_i)^2 \qquad \text{w.r.t. } \Psi \text{ constraints.} \tag{4}$$

A nice property of both LF and LSD formulations is their convexity. Langley and Fitch proved that the LF method's negative log-likelihood function is convex and thus the local minimum is also the global minimum [9]. Our constraint-based formulation of LF also can be easily proved convex by showing its Hessian matrix (Eq. 3) is positive definite. The objective function of LSD is further simplified thanks to the Gaussian approximation. To *et al.* pointed out their objective function is a weighted-least-squares [8]. Using our formulation, we also see that Eq. 4 together with the calibration constraints form a standard convex quadratic optimization problem with a unique analytical solution. To *et al.* also proposed a (quasi)linear-time algorithm to efficiently solve the LSD problem by exploiting the structure of the tree.

2.3 LogDate Method

LF and LSD only seek to model the Poisson-like errors in $\hat{\mathbf{b}}$ and ignore true rate heterogeneity. Strict-clock assumption is now believed to be unrealistic in many settings [23–25], motivating relaxed clocks, typically by assuming that μ_is are drawn i.i.d. from some distribution [11–13]. Most methods rely on presumed parametric distributions (typically, LogNormal, Exponential, or Gamma) and estimate parameters using ML [12], MAP [13], or MCMC [11, 26].

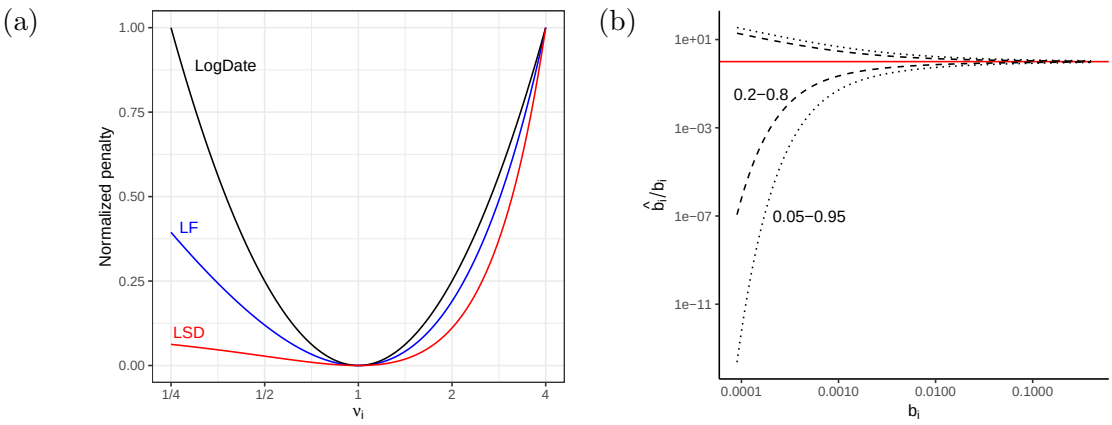

Figure 1: (a) The penalty associated to multiplying a single edge i with multiplier ν_i in LSD, LF, and LogDate approaches, as shown in Equations 3, 4, and 5. To allow comparison, we normalize the penalty to be zero at $\nu = 1$ and to be 1 at $\nu = 4$. (b). The confidence interval of the ratio between estimated and true branch length using the Poisson model. The confidence interval of Poisson follows $1/2\chi^2_{2sb_i}$ on the lower side and $1/2\chi^2_{2sb_i+2}$ on the upper side and is drawn for $\alpha/2 = 0.0.5$ and $\alpha/2 = 0.2$ to get 0.2–0.8 and 0.05–0.095 intervals for $0.0001 \leq b_i \leq 0.4$.

Our goal is to develop a non-parametric method that does not assume clock but also avoids arbitrary assumptions on the model that generates the rate multipliers. Instead, we follow a parsimony principle: we assume that given two solutions of $\mathbf{x}$ both satisfying the calibration constraints, the solution with less variability in f_i values is more preferable. Thus, by parsimony, we prefer solutions that minimize deviations from a strict clock.

A natural way to formalize the parsimony principle is to minimize the variance of f_i, which is the same as finding μ and all ν_i such that ν_i is centered at 1 and minimize $\sum_{i=1}^{2n-2}(\nu_i - 1)^2$. Interestingly, the ML function used by LSD (Eq. 4) is a weighted version of this approach. However, the minimum variance principle results in a fundamental asymmetry: multiplying or dividing the rate of a branch by the same factor is penalized differently (Fig 1a); e.g., the penalty for $\nu_i = 4$ is more than ten times larger than $\nu_i = 1/4$. The LF model is more symmetrical than LSD but remains asymmetrical (Fig 1a). This property results from the asymmetric distribution of the Poisson distribution around its mean, especially for small mean, in log scale. For example, for small b_i, the lower limit of the confidence interval of $\hat{b}_i/b_i$ is further from the mean on a log scale than the upper limit (Fig 1b). Thus, a very small $\hat{b}_i/b_i$, which is within the realm of possible outcomes, is not penalized heavily. In contrast, going above the mean rate is heavily penalized.

Without a model of rate variation, we assert that the penalty for multiplying a branch by a factor of ν should be no different than dividing the branch by ν. Note that this assertion is only with respect to true variations of the mutation rate and not branch length estimation error. This simple property (we call it *symmetry of ratios*) is not offered by minimization of variance (note that LF and LSD do not seek to model true rate variation; they target branch length estimation error only). To ensure the symmetry of ratios, we propose taking the logarithm of the multipliers ν_i before minimizing their variance. This simple log-transformation is the main insight behind our method.

2.3.1 LogDate optimization function

We formulate the LogDate problem as follows. Given $\hat{\mathbf{b}}$ and the set of calibration constraints described in section 2.2.1, we seek to infer $\mathbf{x} = [\nu_1, \nu_2, \ldots, \nu_{2n-2}, \mu]$ to minimize

$$\mathbf{x}^* = \arg\min_{\nu} \sum_{1}^{2n-1} \log^2(\nu_i) \qquad \text{w.r.t. } \Psi \text{ constraints.} \tag{5}$$

This objective function satisfies the symmetry of ratio property that we desire (Fig. 1a). Since ν_i values are multipliers of rates around μ, it is also easy to see that assuming μ is the mean rate, this optimization problem minimizes the variance of the log-transformed rate multipliers (around their mean 1). Note that the objective function only depends on ν_i and μ is included only in the constraints. This setting reduces the complexity of the objective function and speeds up the optimization search. Since the values of ν_i close to 1 are preferred in Eq. 5, the optimal solution would push μ to the mean rate. In addition, it is straightforward to prove that if ν_i rate multipliers are drawn from a LogNormal distribution with the symmetry of ratios (that is, with any parameters that result in mode 1), the LogDate optimization problem is identical to finding the ML solution (Claim 1 in Appendix S1).

The vanilla LogDate formulation, however, has a limitation: by allowing rates to vary freely in a multiplicative way, it fails to model the Poisson process that governs branch length estimation error ($\hat{b}_i/b_i$). Following the Poisson model, i.e. $s\hat{b}_i \sim Poisson(s\mu\tau_i)$, the variance of $\hat{b}_i$ can be computed by noting that $var(s\hat{b}_i) = s\mu\tau_i$ and thus $var(\hat{b}_i) = \mu\tau_i/s = b_i/s$. As Figure 1b shows, when b_i is small, $\hat{b}_i$ can easily vary by several orders of magnitude around b_i. Furthermore, these variances are not symmetrical; drawing values several factors smaller than the mean is more likely than drawing values above the mean by the same factor. This lack of symmetry in the Poisson process is not captured in our LogDate formulation.

Since the true branch length b_i is unknown, a common practice is to use the estimated $\hat{b}_i$ in place of b_i to estimate its variance as $\hat{b}_i/s$. This explains why both LF and LSD objective functions (Eqs. 3 and 4) put a weight of $\hat{b}_i$ to each term of ν_i. Following the same strategy, we propose weighting each $\log^2(\nu_i)$ term by w_i in a way that reduces the penalty and allows more deviations in the log space if the branch is small (and is thus subject to higher error). Since we log transform ν_i, explicitly computing weights in a way that is guaranteed to cancel out the Poisson variations is challenging. However, since we know that the weights should reflect the variance of b_i in logarithmic scale, we conjecture that w_i should be monotonically increasing with $\hat{b}_i$ (Fig. 1b). Using $w_i = \sqrt{\hat{b}_i}$ as weights works well in our preliminary experiments. The shortest branches, however, require even more care.

When the branch is very short, for a limited-size alignment, with high probability, we will see zero mutations. Thus, $\hat{b}_i$ can vary by several orders of magnitude around b_i (Figure 1b). For these no-event branches, tree estimation tools report arbitrary small lengths. Therefore, using $\hat{b}_i$ in place of b_i to estimate its variance is not effective for very small branches. To deal with this challenge, the r8s [27] implementation of LF collapses all branches less than one mutation (i.e. collapsing all $\hat{b}_i$ where $s\hat{b}_i < 1$). In LSD, To *et. al.* proposed adding a smoothing constant c/s to each $\hat{b}_i$ to estimate b_i variance, where c is a tuning parameter that the user can adjust. Following a similar strategy, we propose adding a small constant $\tilde{b}$ to each $\hat{b}_i$. We choose $\tilde{b}$ to be the maximum branch length that produces no mutations with probability at least $1-\alpha$ where $\alpha \in [0,1]$ is a tuning parameter. Assuming $s\hat{b} \sim Poisson(sb)$, then it is easy to show that the maximum $\tilde{b}$ that satisfies $Pr(s\hat{b} = 0 | b = \tilde{b}) \geq 1-\alpha$ is given by $\tilde{b} = -\log(1-\alpha)/s$. By default, we set $\alpha = 0.01$ but allow the user to adjust this parameter. Thus, we define the weighted LogDate (wLogDate) as follows.

$$\mathbf{x}^* = \arg\min_{\nu} \sum_{1}^{2n-1} \sqrt{\hat{b}_i + \tilde{b}} \log^2(\nu_i) \qquad \text{w.r.t. } \Psi \text{ constraints} \tag{6}$$

2.3.2 Solving the optimization problem

LogDate problems (Eq. 5 and Eq. 6) are non-convex, and hence solving them is non-trivial. The problem is convex if $0 \leq \nu_i \leq e$. For small clock deviation and small estimation error in $\hat{b}_i$, the ν_i values should be small so that the problem becomes convex (or close) with one local minimum. However, as $\nu_i \leq e$ is not guaranteed, we have to rely on gradient-based numerical

optimization methods to search for multiple local minima and select the best solution we can find. To search for local minima, we use the Scipy solver with `trust-constr` [28] method. To help the solver work efficiently, we incorporate three techniques that we next describe.

Computing Jacobian and Hessian matrices. Computing the Jacobian and Hessian matrices analytically speeds-up the search. By taking the partial derivative of each ν_i, we can compute the Jacobian, J, of Eq. 6. Also, since Eq. 6 is separable, its Hessian H is a $(2n-2) \times (2n-2)$ diagonal matrix. Simple derivations give us:

$$J = \left[2\sqrt{\hat{b}_1}\frac{\log \nu_1}{\nu_1}, \ldots, 2\sqrt{\hat{b}_{2n-2}}\frac{\log \nu_{2n-2}}{\nu_{2n-2}}\right]^T \qquad \text{and} \qquad H_{ii} = 2\sqrt{\hat{b}_i}\frac{1-\log \nu_i}{\nu_i^2}\ .$$

Sparse matrix representation. Because the Hessian matrix is diagonal, it is sufficient to only store its diagonal elements to save space and computational costs. In addition, the calibration constraints matrix is also highly sparse. We can prove that the upper bound for the number of non-zero elements in our $(n-1) \times (2n-1)$ matrix is $O(n \log n)$. In fact, if the tree is either caterpillar or balanced, the number of non-zeroes reduced to $\Theta(n)$. Thus, in our implementation, we use sparse matrix representation made available by the Scipy package.

Producing multiple feasible initial points. Numerical optimization needs initial values and since our optimization problem is non-convex, we need to provide multiple initials. Providing initial points that are feasible (i.e. satisfied the calibration constraints) helps the SciPy solver work efficiently. Below we describe a heuristic strategy to find multiple initial points given sampling times $t_1, t_2, \ldots, t_n$ of all the leaves (as is common in phylodynamics).

We first describe the process to get a single initial point. First, we compute the root age t_0 and μ using root-to-tip regression (RTT) [10]. Next, we scale all branches of T to conform with Ψ as follow: let $m = \arg\min_i t_i$ (breaking ties arbitrarily). Let $d(r, i)$ denote the distance from the root r to node i and $P(r, m)$ denote the path from r to m. For each node i in $P(r, m)$, we set $\tau_i = \hat{b}_i(t_m - t_0)/d(r, m)$. Then going upward from m to r following $P(m, r)$, for each edge (i, j) we compute $t_j = t_i - \tau_i$ and recursively apply the process on the clade i. At the root, we set t_m to the second oldest (second minimum) sampling time and repeat the process on a new path until all leaves are processed. Since RTT gives us μ, to find ν we simply set $\nu_i = {}^{\mu\tau_i}/_{\hat{b}_i}$.

To find multiple initial points, we repeatedly apply RTT to a set of randomly selected clades of T and scale each clade using the aforementioned strategy. Specifically, we randomly select a set S of some internal nodes in the tree and add the root to S. Then, by a post-order traversal, we visit each $u \in S$ and date the clade u using the scaling strategy described above. We then remove the entire clade u from the tree but keep the node u as a leaf (note that the age of u is already computed) and repeat the process for the next node in S. The root will be the last node to be visited. After visiting the root, we have all the τ_i for all i. To finalize the process, we set μ to the mutation rate given by RTT at the root and for each branch i, compute ν_i as ${}^{\mu\tau_i}/_{\hat{b}_i}$. In a tree of n leaves, we have $2^{(n-1)} - 1$ ways to select the initial non-empty set S, giving us a way to select many feasible $\mathbf{x}$ as initial points.

2.4 Experiments

2.4.1 Datasets

Simulated data. To *et al.* [8] provides a simulated dataset of HIV *env* gene. Their time trees were generated based on a birth-death model with periodic sampling times. There are four tree models, namely D995_11_10 (M1), D995_3_25 (M2), D750_11_10 (M3), and D750_3_25 (M4), each of which has 100 replicates for a total of 400 different tree topologies. M1 and M2 simulate intra-host HIV evolution and are ladder-like while M3 and M4 simulate inter-host evolution and are balanced. Also, M4 has much higher root-to-tip distance (mean: 57) compared to M1–M3

(22, 33, and 29). Starting from conditions simulated by To *et al.*, we use the provided time tree to simulate the clock deviations. Using an uncorrelated model of the rates, we draw each rate from one of three different distributions, each of which is centered at the value $\mu = 0.006$ as in the original study. Thus, we set each μ_i to $x_i\mu$ where x_i is drawn from one of three distributions: LogNormal (mean:1.0, std: 0.4), Gamma ($\alpha = \beta = 6.05$), and Exponential ($\lambda = 1$). Sequences of length 1000 were simulated for each of the model conditions using SeqGen [29] under the same model and the same settings as the original study [8].

H1N1 2009 pandemic. We re-analyze the H1N1 biological data provided by To *et al.* [8] which includes 892 H1N1pdm09 sequences collected worldwide between 13 March 2009 and 9 June 2011. We reuse the estimated PhyML [30] trees, 100 bootstrap replicates, and all the results of the dating methods other than LogDate that are provided by To *et al.*

HIV. We study a dataset of 926 HIV-1 subtype B *pol* sequences obtained in San Diego between 1996 and 2018. We use IQTree [31] to infer a tree under the GTR+Γ model, root the tree on 22 outgroups, then remove the outgroups. Because of the size, we could not run BEAST.

West African Ebola epidemic. We study the dataset of Zaire Ebola virus from Africa, which includes 1,610 near-full length genomes sampled between 17 March 2014 and 24 October 2015. The data was collected and analyzed by Dudas *et al.* [32] using BEAST and re-analyzed by Volz *et al.* using IQTree [31] to estimate the ML tree and *treedater* [12] to infer node ages. We run LSD, LF, and wLogDate on the IQTree from a previous paper [12] and use the BEAST trees from another paper [32], which include 1000 sampled trees (BEAST-1000) and the Maximum clade credibility tree (BEAST-MCC). To root the IQTree, we search for the rooting position that minimizes the triplet distance [33] between the IQTree and the BEAST-MCC tree.

2.4.2 Methods Compared

We compare wLogDate to three other methods: LSD [8], LF [9] implemented in r8s [27], and BEAST [26]. We used phyML [30] to re-estimate branch lengths of the true tree topology in substitution unit and used each of the three methods wLogDate, LSD (in the rooted and constrained mode), and LF to infer the time tree. wLogDate was run with 10 feasible starting points produced . For the Bayesian method BEAST, we also fixed the true rooted tree topology and only inferred node ages. Following the settings in the original study [8], we ran BEAST using HKY+Γ8 and coalescent with constant population size tree prior. We used two clock models on the rate parameter: the strict-clock (i.e. fixed rate) and the LogNormal distribution. For the strict-clock prior, we set clock rate prior to a uniform distribution between 0 and 1. For the LogNormal prior, we set the ucld.mean prior to a uniform distribution between 0 and 1, and ucld.stdev prior to an exponential distribution with parameter $1/3$ (default value). For all BEAST runs, we set the length of the MCMC chain to 10^7 generations, burn-in to 10%, and sampling to every 10^4 generations (all settings identidcal to the original study [8]).

2.4.3 Error Measurement

On the simulated datasets, where the ground truth is known, we compare the accuracy of the methods using several metrics. We compute the root mean squared error (RMSE) measured between true and estimated vector of times (τ) and normalize it by tree height. We also rank methods by RMSE rounded to two decimal digits (to avoid different ranks when errors are similar). Besides the exact timing of nodes, the two parameters that are most important to users of dating are the time to Most Recent Common Ancestor (tMRCA) of all nodes and the mutation rate. We plot the distribution of both. Finally, on real data, we also show lineage-through-time (LTT) plots [34], which trace the number of lineages at any point in time.

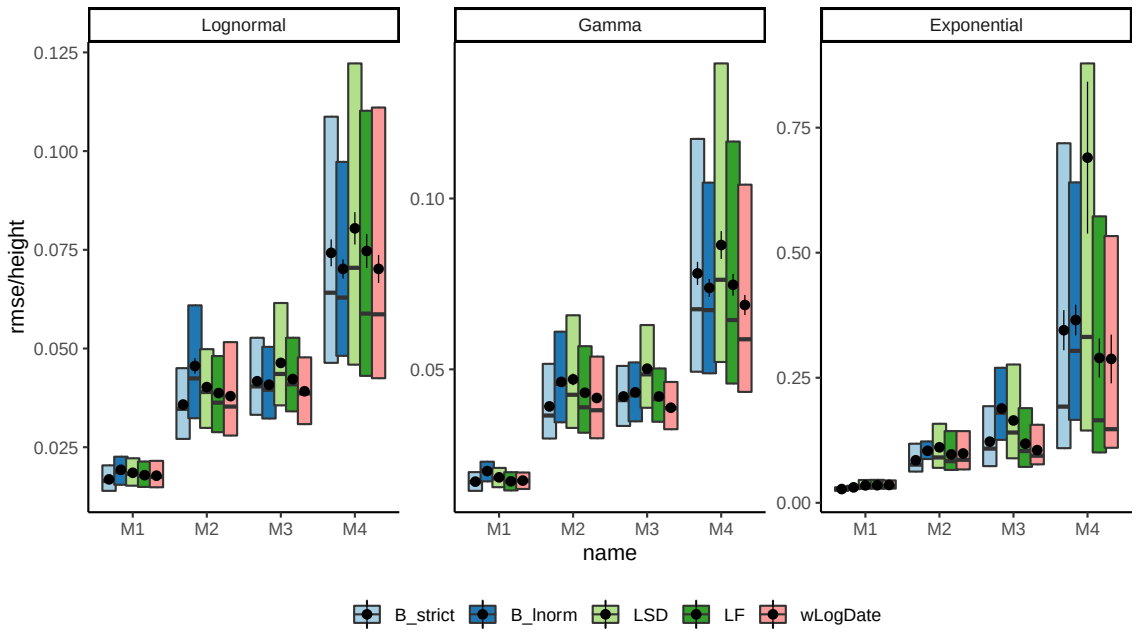

Figure 2: Distributions of RMSE normalized by the tree height for internal node ages inferred by all methods on model trees M1–M4, each with clock models Lognorm, Gamma, and Exponential. Boxes show median, 10% and 90% quantiles; dots and error bars show mean and standard error (100 replicates).

Tree model	Clock model	BEAST lognorm	BEAST strict-clock	LF	LSD	wLogDate
M4	LogNormal	**1**	3	4	5	**1**
	Gamma	2	4	3	5	**1**
	Exponential	4	3	2	5	**1**
M3	LogNormal	2	3	3	5	**1**
	Gamma	4	2	2	5	**1**
	Exponential	5	3	2	4	**1**
M2	LogNormal	5	**1**	3	4	2
	Gamma	4	**1**	3	5	2
	Exponential	4	**1**	2	5	3
M1	LogNormal	4	**1**	2	4	2
	Gamma	5	**1**	**1**	4	**1**
	Exponential	2	**1**	3	3	5
Average rank		3.5	2	2.5	4.5	**1.75**

Table 1: Ranking of the dating methods under different model conditions. For each model condition, the averaged RMSE of all internal node ages is computed and ranked among the dating methods (rounded to two decimal digits). Best method shown in bold.

3 Results

3.1 Simulated data

We first evaluate the convergence of the ScipPy optimization method across 10 starting points (Fig. S1). LogDate and wLogDate converge to a stable result after 50–200 iterations, depending on the model condition. Convergence seems easier when rates are Gamma or LogDate and harder when the rates are Exponential. We then compared LogDate and wLogDate (Fig. S2). The wLogDate method is never worse than LogDate, and under some conditions (e.g., M4 with the Exponential rate variation), weighting results in dramatic improvements. Thus, in the rest of the paper, we focus on wLogDate, designated as the default way of using the method.

Measured by RMSE, the accuracy of all methods varies substantially across model trees (M1 – M4) and models of rate variation (Fig. 2). Comparing methods, for many conditions, wLogDate has the lowest error, and in many others, it is ranked second best (Table 1). Across all conditions, wLogDate has a mean rank of 1.75, followed by BEAST with strict clock with

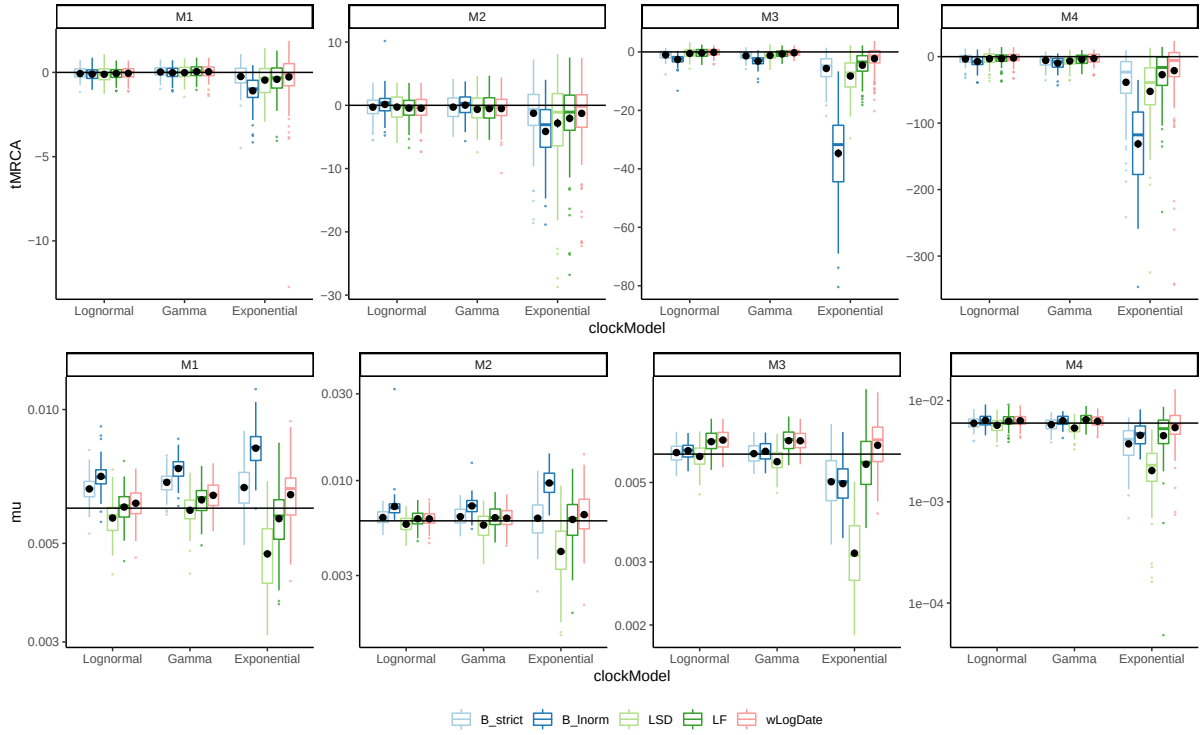

Figure 3: The inferred (top) tMRCA and (bottom) expected mutation rate on different tree models and clock models. Distributions are over 100 replicates. The solid horizontal lines indicate the true mutation rate and tMRCA. Each black is the average of the inferred values for each method under each model condition. We remove 6 outlier data points (2 LF, 1 LSD, 2 BEAST-LogNormal, 1 BEAST-Strict) with exceptional incorrect tMRCA (< -350) in the M4/Exponential model. Mutation rates are in log scale.

a mean rank 2; mean normalized RMSE of wLogDate, LF, BEAST-strict, BEAST-LogNormal, and LSD are 0.072, 0.074, 0.077, 0.087, and 0.116, respectively, showing that wLogDate is the best method overall.

For all methods, errors are an order of magnitude smaller for the LogNormal and Gamma models of rate variations compared to the Exponential model. In terms of trees, M4, which is meant to simulate inter-host evolution, presents the most challenging case for all methods. Interestingly, wLogDate has the best accuracy under all parameters of M4 tree and also all parameters of M3 (thus, both inter-host conditions). On M1, all methods have very low error and perform similarly (Fig. 2).

Among other methods, patterns of accuracy are consistent with the literature. Despite its conceptual similarity to wLogDate, LSD has the worst accuracy. On M1 and M2, LSD is competitive with other methods; however, on M3 and especially M4, it has a much higher error, in particular with the Exponential model of rate variation. With the LogNormal clock model, BEAST-LogNormal is better than BEAST-strict *only* for M4 *but not* for M1-M3; in fact, BEAST-LogNormal has the highest error for the M2 condition. Given the match between the model and the method, this result is surprising. However, as expected, the only cases where BEAST-LogNormal has high accuracy are those with the LogNormal model of rate variation; BEAST-LogNormal is one of the two worst methods elsewhere. Thus, BEAST-LogNormal is sensitive to model misspeficication. In contrast, BEAST-strict is much less sensitive to the model of rate variation and ranks among the top three in most cases. In particular, BEAST-strict is always the best method for intra-host ladder-like trees M1 and M2.

Accuracy of tMRCA followed similar patterns (Fig. 3). Again, the Exponential rate variation model is the most difficult case for all methods, resulting in biased results and highly variable error rates for most methods. In all conditions of M3 and M4, wLogDate has the best accuracy

		BEAST_strict	BEAST_lognorm	LF	LSD	wLogDate
M4	Lognormal	6.99	9.50	6.66	7.38	**6.11** (9% ↓)
	Gamma	7.83	10.48	7.02	8.48	**6.28** (12% ↓)
	Exponential	43.5	140.9	116.2	62.2	**32.5** (34% ↓)
M3	Lognormal	1.37	2.60	1.21	1.39	**1.03** (17% ↓)
	Gamma	1.60	3.14	1.23	1.67	**0.97** (27% ↓)
	Exponential	5.76	34.67	4.87	8.35	**2.94** (66% ↓)
M2	Lognormal	**1.40**	1.41	1.50	1.63	1.47 (5% ↑)
	Gamma	1.54	**1.44**	1.75	1.92	1.66 (15% ↑)
	Exponential	**3.39**	4.59	4.28	5.27	3.72 (10% ↑)
M1	Lognormal	**0.28**	**0.28**	0.30	0.37	0.30 (7% ↑)
	Gamma	**0.27**	0.29	0.32	0.35	0.30 (11% ↑)
	Exponential	**0.60**	1.11	0.79	0.82	0.69 (15% ↑)
Average		6.21	17.54	12.17	8.13	**4.83**

Table 2: Mean absolute error of the inferred tMRCA of BEAST_strict, BEAST_lognorm, LF, LSD, and wLogDate. For wLogDate, parenthetically, we compare it with the best (↑) or second best (↓) method for each condition. We show percent improvement by wLogDate, as measured by the increase in the error in the second best method (wLogDate or the alternative) divided by the error of the best method (wLogDate or the alternative).

and improves on the second best method by 9 – 66% (Table 2). For M1 and M2, BEAST-strict is often the best method. The mean tMRCA error of wLogDate across all conditions is 4.83 (years), which is substantially better than the second best method, BEAST-strict (6.21).

In terms of the mutation rate, wLogDate is the best method jointly with the two strict clock models BEAST-strict and LF (Table S1). Overall, even though LF and wLogDate tend to over-estimate mutation rates, both seem to have less biased results compared to other methods (Fig. 3). LSD and BEAST-LogNormal have the highest errors; depending on the condition, each can overestimate or underestimate the rate but LSD tends to underestimate while BEAST-LogNormal tends to overestimate. On M1, wLogDate and LF have a clear advantage over BEAST-strict, which tends to over-estimate the rate. On M2, the three methods have similar accuracy. For M3 and M4, BEAST-strict under-estimates the rate under the Exponential model of rate variation, and wLogDate and LF are closer to the true value. For all methods, M4, which has a much higher root to tip distance, presents the most challenging case.

3.2 Biological data

On the H1N1 dataset, the best available evidence has suggested a tMRCA between December 2008 and January 2009 [35–37]. wLogDate inferred the tMRCA to be 14 December 2008 (Fig. 4a), which is consistent with the literature. LF and LSD both infer a slightly earlier tMRCA (10 November 2008), followed by BEAST-strict and BEAST-lognorm (October 2008 and July 2008), and finally BEAST runs using the phyML tree (Feb. 2008 for strict and July 2007 for LogNormal). While the exact tMRCA is not known on this real data, the results demonstrate that wLogDate, on a real data, produces times that closer to the results of previous studies comparing to other methods.

On the HIV dataset, wLogDate inferred a tMRCA of 1958 with only a handful of lineages coalescing in the 1950s and most others coalescing in 1960s and early 1970s (Fig. S3). The recovered tMRCAs is within the range postulated in the literature for subtype B [38, 39] and the fact that randomly sampled HIV lineages across USA tend to coalesce deep in the tree is a known phenomenon. LF and LSD recovered the tMRCA of 1952 and 1953, respectively. Comparing to wLogDate, these two strict-clock methods postulate an earlier burst of subtype B (Fig. 4c). We were not able to run BEAST on this dataset.

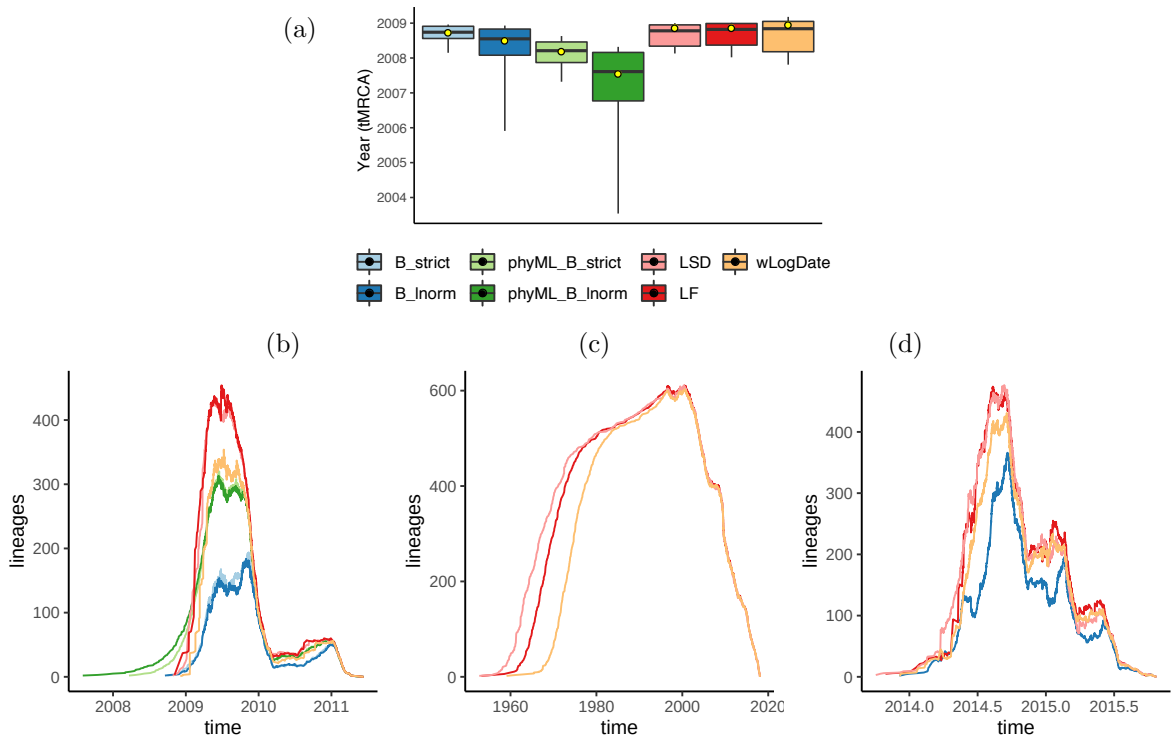

Figure 4: (a) Inferred tMRCA of the H1N1 dataset. Boxplots represent the median, maximum, minimum, 97.5% and 2.5% quantiles of the bootstrap estimates for LF, LSD, and wLogDate, and of the posterior distribution for BEAST. Yellow dot shows the inferred tMRCA of the best ML or MAP tree. BEAST was run with 4 different settings: B_strict and B_lnorm allow BEAST to infer both tree topology and branch lengths, with strict and LogNormal clock models; phyML_B_strict and phyML_B_lnorm fixed the topology to the rooted phyML tree given to BEAST. All other methods (LSD, LF, and wLogDate) were run on the rooted phyML trees. Results for LSD, LF, and BEAST are all obtained from To *et al.* [8]. (b) LTT plot for all methods on the H1N1 data. (c) LTT plot of fast methods on the HIV dataset. (d) LTT plot of BEAST, LSD, LF, and wLogDate on the Ebola dataset.

On the Ebola dataset, the BEAST-1000 trees obtained from Dudas *et. al* [32] inferred the tMRCA to be between 13 September 2013 and 26 January 2014 (95 per cent credible interval) and the BEAST-MCC inferred the tMRCA to be 5 December 2013 (results reported previously [12]). Here, wLogDate inferred a tMRCA on 7 December 2013, which is very close to the estimate by BEAST. Both LF and LSD inferred an earlier tMRCA: 29 October 2013 for LF and 2 October 2013 for LSD, but still within the 95 per cent credible interval of BEAST-1000. LTT plots showed a similar reconstruction by all methods for this dataset (Fig. 4d).

We also compare running time of the 3 dating methods on the three real biological datasets (Table S2). LSD was always the fastest, running in just seconds. LF is faster on the H1N1 and HIV data, while on Ebola data wLogDate is faster. Here we report the running time for wLogDate as a sequential run of 10 independent starting points. We note that wLogDate can easily be parallelized to gain speedup.

4 Discussion and future work

We introduced (w)LogDate, a new method for dating phylogenies based on a non-convex optimization problem. We showed that by log-transforming the rates before minimizing their variance, we obtain a method that performs much better than LSD, which is a similar method without log-transform. Our relatively simple method also outperformed other existing methods, including the Bayesian methods, which are much slower. The improvements were most pronounced in terms of the estimation of tMRCA and individual node times and less so for the mutation rate. Moreover, improvements are most visible under the hardest model conditions.

The log transformation results in a non-convex optimization problem that is harder than the convex problem solved by LSD. However, given the advances in numerical methods for solving non-convex optimization problems, insistence on convex problems seems unnecessary. Our results indicate that this non-convex problem can be solved efficiently in the varied settings we tested (we note that the problem is convex for rate multipliers between 0 and e). The main benefit of the log transformation is that it allow us to define a scoring function that assigns symmetrical penalties for increased or decreased rates compared to the mean rate (Fig. 1).

The accuracy of LogDate under varied conditions we tested is remarkable and is a testament to the power of non-parametric methods. LogDate does not depend on a particular model of rate evolution and is based on a simple optimization problem. We emphasize that the parametric models used in practice are employed for mathematical convenience and not because of a strong biological reason to believe that they capture real variations in rates. Thus, lack of reliance on a model, in our judgment, is a strength of the method.

The LogDate approach can be further improved in the future. The weighting scheme used in LogDate is chosen heuristically to deal with the Poisson process that governs deviations of estimated branch lengths from the true branch length. In future, the weighting schema should be studied more carefully, both in terms of theoretical properties and empirical performance. Moreover, LogDate, like many other methods, is based on an uncorrelated model: rates across branches do not have any relationships. Future work should consider adding correlation terms to the optimization problem. Finally, we described, implemented, and tested LogDate in the phylodynamic setting [4] where sampling times are available, as opposed to the paleontological setting with fossil calibrations. While the algorithms extends easily, treatment of fossil calibrations is a complex topic and adds other difficulties such as the uncertainty of times and placement of fossils on trees [40, 41]. Thus, applying LogDate in such settings requires further tweaks to the optimization formulation, including changing equality to inequality constraints.

We have left several questions for the future work. Most importantly, all the simulation models we used assumed that evolutionary rates, even when variable, are drawn from a fixed distribution at random. Thus, rates are stationary and do not change in distribution through time. Real evolution, in addition to changing rates, may have a direction, where, for example, rates may reduce with time. Moreover, it is reasonable to assume rates of adjacent branches are not independent, a condition we did not simulate. Future work should test LogDate under models of rate variation that include correlations and are time-sensitive. Moreover, to facilitate estimation of error, we used the true topology (albeit with estimated branch lengths). Future work should study the impact of the incorrect topology on LogDate.

Code availability. The LogDate code and all the data used here are available on https://github.com/uym2/LogDate.

References

[1] Frank Rutschmann. Molecular dating of phylogenetic trees: A brief review of current methods that estimate divergence times. *Diversity & Distributions*, 12(1):35–48, 1 2006.

[2] Flix Forest. Calibrating the Tree of Life: fossils, molecules and evolutionary timescales. *Annals of Botany*, 104(5):789–794, 10 2009.

[3] David M Hillis, C Moritz, and B K Mable. *Molecular Systematics*, volume 2nd. Sinauer Associates, 1996.

[4] Erik M. Volz, Katia Koelle, and Trevor Bedford. Viral Phylodynamics. *PLoS Computational Biology*, 9(3), 2013.

[5] Ullasa Kodandaramaiah. Tectonic calibrations in molecular dating. *Current Zoology*, 57(1):116–124, 2 2011.

[6] Michael J Sanderson. Estimating rate and time in molecular phylogenies: beyond the molecular clock? In *Molecular systematics of plants II*, pages 242–264. Springer, 1998.

[7] Emile Zuckerkandl. Molecular disease, evolution, and genetic heterogeneity. *Horizons in biochemistry*, pages 189–225, 1962.

[8] Thu-Hien To, Matthieu Jung, Samantha Lycett, and Olivier Gascuel. Fast Dating Using Least-Squares Criteria and Algorithms. *Systematic Biology*, 65(1):82–97, 09 2015.

[9] Charles H. Langley and Walter M. Fitch. An examination of the constancy of the rate of molecular evolution. *Journal of Molecular Evolution*, 3(3):161–177, Sep 1974.

[10] R Shankarappa, J B Margolick, S J Gange, A G Rodrigo, D Upchurch, H Farzadegan, P Gupta, C R Rinaldo, G H Learn, X He, X L Huang, and J I Mullins. Consistent viral evolutionary changes associated with the progression of human immunodeficiency virus type 1 infection. *Journal of virology*, 73(12):10489–502, 12 1999.

[11] Alexei J. Drummond, Simon Y. W Ho, Matthew J. Phillips, and Andrew Rambaut. Relaxed Phylogenetics and Dating with Confidence. *PLoS Biology*, 4(5):e88, 3 2006.

[12] E. M. Volz and S. D. W. Frost. Scalable relaxed clock phylogenetic dating. *Virus Evolution*, 3(2), 08 2017. vex025.

[13] Orjan Akerborg, Bengt Sennblad, and Jens Lagergren. Birth-death prior on phylogeny and speed dating. *BMC Evolutionary Biology*, 8(1):77, 2008.

[14] John P. Huelsenbeck, Bret Larget, and David Swofford. A compound poisson process for relaxing the molecular clock. *Genetics*, 154(4):1879–92, 4 2000.

[15] Jeffrey L. Thorne, Hirohisa Kishino, and Ian S. Painter. Estimating the rate of evolution of the rate of molecular evolution. *Molecular Biology and Evolution*, 15(12):1647–1657, 12 1998.

[16] Hirohisa Kishino, Jeffrey L. Thorne, and William J. Bruno. Performance of a Divergence Time Estimation Method under a Probabilistic Model of Rate Evolution. *Molecular Biology and Evolution*, 18(3):352–361, 3 2001.

[17] Michael J. Sanderson. Estimating Absolute Rates of Molecular Evolution and Divergence Times: A Penalized Likelihood Approach. *Molecular Biology and Evolution*, 19(1):101–109, 1 2002.

[18] Thomas Lepage, David Bryant, Herv Philippe, and Nicolas Lartillot. A General Comparison of Relaxed Molecular Clock Models. *Molecular Biology and Evolution*, 24(12):2669–2680, 6 2007.

[19] Alexei J. Drummond and Marc A. Suchard. Bayesian random local clocks, or one rate to rule them all. *BMC Biology*, 8(1):114, 12 2010.

[20] Koichiro Tamura, Fabia Ursula Battistuzzi, Paul Billing-Ross, Oscar Murillo, Alan Filipski, and Sudhir Kumar. Estimating divergence times in large molecular phylogenies. *Proceedings of the National Academy of Sciences*, 109(47):19333–19338, 11 2012.

[21] Sagi Snir, Yuri I. Wolf, and Eugene V. Koonin. Universal Pacemaker of Genome Evolution. *PLoS Computational Biology*, 8(11):e1002785, 11 2012.

[22] Michael J. Sanderson. A Nonparametric Approach to Estimating Divergence Times in the Absence of Rate Constancy. *Molecular Biology and Evolution*, 14(12):1218–1231, 12 1997.

[23] Mrio J.F. Pulqurio and Richard A. Nichols. Dates from the molecular clock: how wrong can we be? *Trends in Ecology & Evolution*, 22(4):180 – 184, 2007.

[24] Jeffrey H. Schwartz and Bruno Maresca. Do molecular clocks run at all? a critique of molecular systematics. *Biological Theory*, 1(4):357–371, Dec 2006.

[25] Simon Y.W. Ho. The changing face of the molecular evolutionary clock. *Trends in Ecology & Evolution*, 29(9):496 – 503, 2014.

[26] Alexei J Drummond and Andrew Rambaut. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC evolutionary biology*, 7:214, 2007.

[27] Michael J. Sanderson. r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. *Bioinformatics*, 19(2):301–302, 1 2003.

[28] Marucha Lalee, Jorge Nocedal, and Todd Plantenga. On the implementation of an algo-

rithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, 8(3):682–706, 1998.

[29] Andrew Rambaut and Nicholas C. Grass. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Bioinformatics*, 13(3):235–238, 06 1997.

[30] Stphane Guindon, Jean-Franois Dufayard, Vincent Lefort, Maria Anisimova, Wim Hordijk, and Olivier Gascuel. New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. *Systematic Biology*, 59(3):307–321, 5 2010.

[31] Lam Tung Nguyen, Heiko A. Schmidt, Arndt Von Haeseler, and Bui Quang Minh. IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32(1), 2015.

[32] Gytis Dudas, Luiz Max Carvalho, Trevor Bedford, Andrew J. Tatem, Guy Baele, Nuno R. Faria, Daniel J. Park, Jason T. Ladner, Armando Arias, Danny Asogun, Filip Bielejec, Sarah L. Caddy, Matthew Cotten, Jonathan D'Ambrozio, Simon Dellicour, Antonino Di Caro, Joseph W. Diclaro, Sophie Duraffour, Michael J. Elmore, Lawrence S. Fakoli, Ousmane Faye, Merle L. Gilbert, Sahr M. Gevao, Stephen Gire, Adrianne Gladden-Young, Andreas Gnirke, Augustine Goba, Donald S. Grant, Bart L. Haagmans, Julian A. Hiscox, Umaru Jah, Jeffrey R. Kugelman, Di Liu, Jia Lu, Christine M. Malboeuf, Suzanne Mate, David A. Matthews, Christian B. Matranga, Luke W. Meredith, James Qu, Joshua Quick, Suzan D. Pas, My V. T. Phan, Georgios Pollakis, Chantal B. Reusken, Mariano Sanchez-Lockhart, Stephen F. Schaffner, John S. Schieffelin, Rachel S. Sealfon, Etienne Simon-Loriere, Saskia L. Smits, Kilian Stoecker, Lucy Thorne, Ekaete Alice Tobin, Mohamed A. Vandi, Simon J. Watson, Kendra West, Shannon Whitmer, Michael R. Wiley, Sarah M. Winnicki, Shirlee Wohl, Roman Wölfel, Nathan L. Yozwiak, Kristian G. Andersen, Sylvia O. Blyden, Fatorma Bolay, Miles W. Carroll, Bernice Dahn, Boubacar Diallo, Pierre Formenty, Christophe Fraser, George F. Gao, Robert F. Garry, Ian Goodfellow, Stephan Günther, Christian T. Happi, Edward C. Holmes, Brima Kargbo, Sakoba Keïta, Paul Kellam, Marion P. G. Koopmans, Jens H. Kuhn, Nicholas J. Loman, N'Faly Magassouba, Dhamari Naidoo, Stuart T. Nichol, Tolbert Nyenswah, Gustavo Palacios, Oliver G. Pybus, Pardis C. Sabeti, Amadou Sall, Ute Ströher, Isatta Wurie, Marc A. Suchard, Philippe Lemey, and Andrew Rambaut. Virus genomes reveal factors that spread and sustained the ebola epidemic. *Nature*, 544(7650):309–315, 2017.

[33] Andreas Sand, Morten K Holt, Jens Johansen, Rolf Fagerberg, Gerth Stlting Brodal, Christian N S Pedersen, and Thomas Mailund. Algorithms for computing the triplet and quartet distances for binary and general trees. *Biology*, 2(4):1189–209, 2013.

[34] S. Nee, R. M. May, and P. H. Harvey. The reconstructed evolutionary process. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 344(1309):305–311, 5 1994.

[35] Philippe Lemey, Marc Suchard, and Andrew Rambaut. Reconstructing the initial global spread of a human influenza pandemic: A bayesian spatial-temporal model for the global spread of h1n1pdm. *PLoS currents*, 1:RRN1031–RRN1031, Sep 2009. 20029613[pmid].

[36] Andrew Rambaut and Edward Holmes. The early molecular epidemiology of the swine-origin a/h1n1 human influenza pandemic. *PLoS currents*, 1:RRN1003–RRN1003, Aug 2009. 20025195[pmid].

[37] J. Hedge, S. J. Lycett, and A. Rambaut. Real-time characterization of the molecular epidemiology of an influenza pandemic. *Biology Letters*, 9(5):20130331, 10 2013.

[38] M. T. P. Gilbert, A. Rambaut, G. Wlasiuk, T. J. Spira, A. E. Pitchenik, and M. Worobey. The emergence of HIV/AIDS in the Americas and beyond. *Proceedings of the National Academy of Sciences*, 104(47):18566–18570, 11 2007.

[39] Joel O. Wertheim, Mathieu Fourment, and Sergei L. Kosakovsky Pond. Inconsistencies in Estimating the Age of HIV-1 Subtypes Due to Heterotachy. *Molecular Biology and Evolution*, 29(2):451–456, 2 2012.

[40] Simon Y W Ho and Matthew J. Phillips. Accounting for Calibration Uncertainty in Phylogenetic Estimation of Evolutionary Divergence Times. *Systematic Biology*, 58(3):367–380, 7 2009.

[41] Tracy A Heath. A hierarchical Bayesian model for calibrating estimates of species divergence times. *Systematic biology*, 61(5):793–809, 10 2012.

S1 Supplementary appendix

Algorithm 1 Setup the linear constraints Ψ for tree T given a set of calibrated points. This algorithm assumes all leaves have sampling times. Can be extended to other cases in a natural way.

```
function SetupConstraints(T)
    Ψ ← {}
    for i in post-order traversal of T do
        if i is a leaf then
            nearest_timepoint(i) ← i
        else
            (i_1, i_2) ← Children(i)
            u ← nearest_timepoint(i_1)
            v ← nearest_timepoint(i_2)
            if i is calibrated then
                nearest_timepoint(i) ← i
                Ψ ← Ψ ∪ ψ(i, i_1)
                Ψ ← Ψ ∪ ψ(i, i_2)
            else
                nearest_timepoint(i) ← u if d(i, i_1) < d(i, i_2) else v
                Ψ ← Ψ ∪ ψ(i_1, i_2)
```

Claim 1. *The optimal* $\mathbf{x}^*$ *in Eq. 5 is the Maximum Likelihood (ML) estimate of the model of rate variation where* $\nu_i \sim LogNormal(\mu, \sigma^2)$ *and* $Mode(\nu_i) = 1$.

Proof. We have: $\nu_i \sim LogNormal(\mu, \sigma^2) \implies Mode(\nu_i) = e^{\mu-\sigma^2} \implies 1 = e^{\mu-\sigma^2} \implies \mu = \sigma^2$ Therefore, the log-likelihood function under the LogNormal model of rate variation can be written as follow:

$$\log L^{lG} = \sum_{i=1}^{2n-2} \log\left(\frac{1}{\nu_i \sigma\sqrt{2\pi}} \exp\left(-\frac{(\log \nu_i - \mu)^2}{2\sigma^2}\right)\right) \propto \sum_{i=1}^{2n-2}\left(-\log \nu_i - \frac{\log^2 \nu_i - 2\mu \log \nu_i}{2\mu}\right) \propto -\log^2 \nu_i \tag{S1}$$

Thus, the ML estimate can be found by minimizing Eq. 5. □

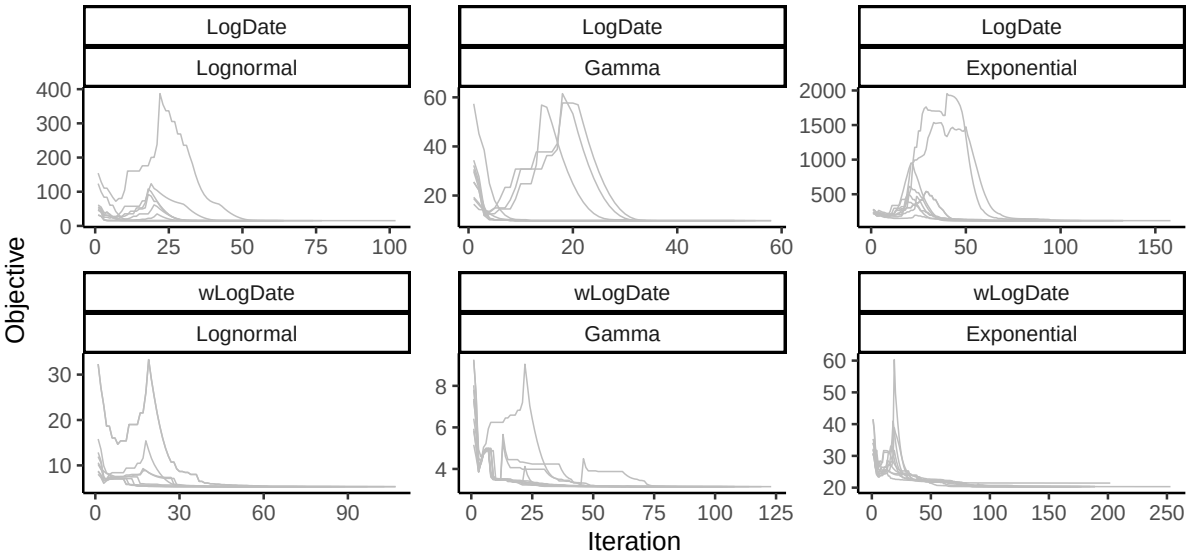

Figure S1: Objective value versus iteration of the LogDate and wLogDate runs on one arbitrarily selected simulated tree (M4, replicate 2). Each of the two methods were run using 10 random initial points generated using the strategy described in 2.3.2.

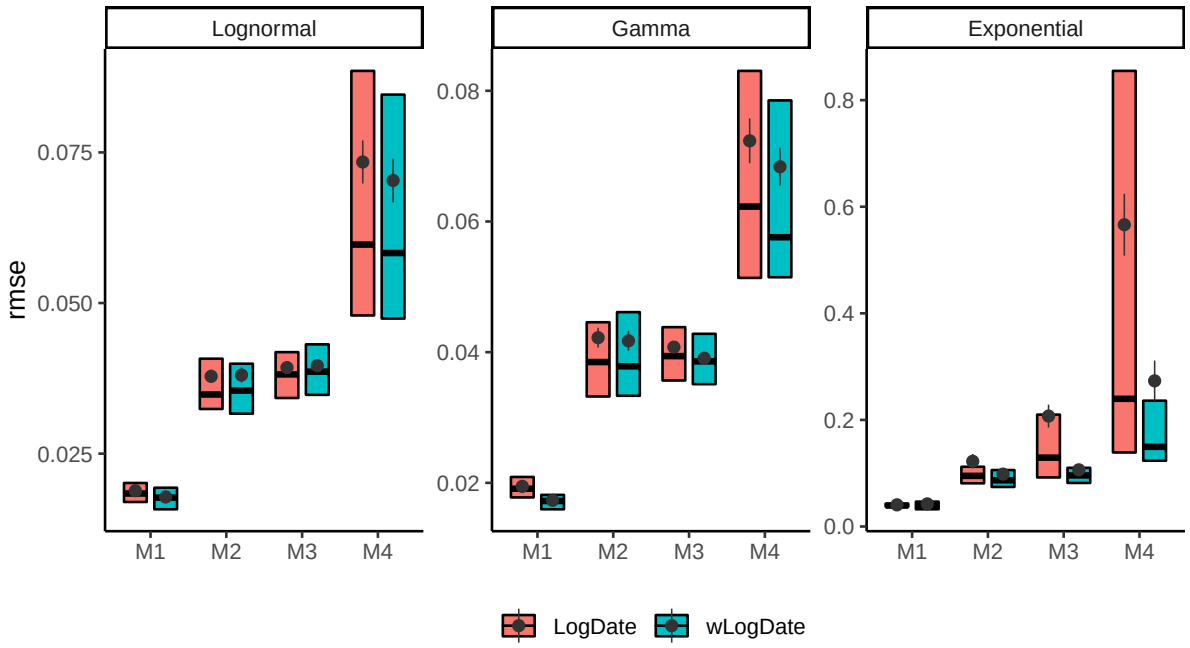

Figure S2: RMSE of the internal node ages inferred by LogDate and wLogDate (the weighted version).

		BEAST_strict	BEAST_lnorm	LF	LSD	wLogDate
	Lognormal	0.0007	0.0011	**0.0004**	0.0005	**0.0004**
M1	Gamma	0.0009	0.0014	0.0005	**0.0004**	0.0006
	Exponential	0.0009	0.0022	**0.0008**	0.0013	0.0009
	Lognormal	**0.0005**	0.0014	**0.0005**	**0.0005**	**0.0005**
M2	Gamma	**0.0006**	0.0013	0.0007	0.0007	0.0007
	Exponential	**0.0013**	0.0038	0.0015	0.0020	0.0015
	Lognormal	**0.0003**	**0.0003**	0.0006	0.0004	0.0006
M3	Gamma	**0.0003**	**0.0003**	0.0006	0.0004	0.0006
	Exponential	0.0010	0.0011	**0.0009**	0.0027	**0.0009**
	Lognormal	**0.0007**	0.0008	0.0008	**0.0007**	0.0008
M4	Gamma	**0.0006**	0.0007	0.0008	0.0008	0.0007
	Exponential	0.0020	**0.0016**	**0.0016**	0.0037	0.0017
Average		**0.0008**	0.0013	**0.0008**	0.0012	**0.0008**

Table S1: Mean absolute error of the inferred mutation rate of BEAST_strict, BEAST_lognorm, LF, LSD, and wLogDate.

Data	LSD	LF	wLogDate
H1N1 (n=892)	$<$ 1 sec	1 min	3 mins
HIV San Diego (n=904)	$<$ 1 sec	11 mins	24 mins
Ebola (n=1610)	2 secs	4 mins	3 mins

Table S2: Running time of LSD, LF, and wLogDate on both biological datasets.

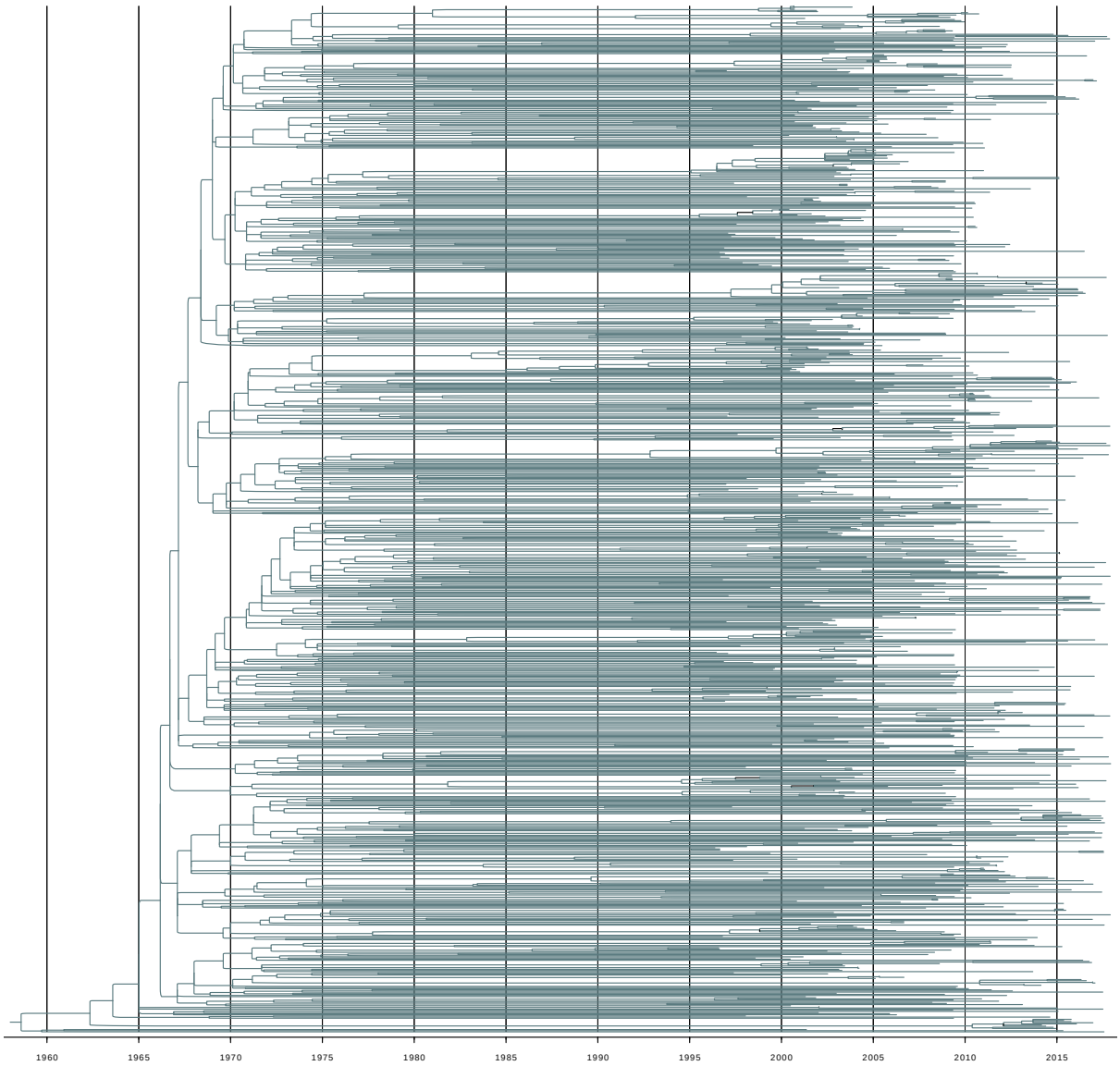

Figure S3: A TimeTree of San Diego HIV epidemic according to wLogDate.