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 inferred from sequence data often have branch lengths measured in the expected number of substitutions and therefore, do not have divergence times estimated. These trees give an incomplete view of evolutionary histories since many applications of phylogenies require time trees. Many methods have been developed to convert the inferred branch lengths from substitution unit to time unit 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 various model assumptions.

Keywords: time tree, divergence time estimation, phylogenetic dating, molecular dating, non-convex optimization.

1 Introduction

Phylogenetic inference from sequence data does not reveal divergence time (i.e. exact timing of evolutionary events) unless paired with external timing information. Because the level of divergence given by sequence data is a function of unknown time and mutation rates, the inferred phylogenies measure branch lengths in the unit of expected number of substitutions per site. Dating a phylogeny is the process of translating branch lengths from substitution unit to time unit. Such a process requires soft or hard constraints for the timing of *some* nodes of the tree, leaving the divergence times of the remaining nodes to be inferred. Timing information is often represented in the form of calibration points obtained by carbon-dating fossils [16] or sampling times for leaves of the tree in fast-evolving viruses and bacteria. While dating is difficult [26], it is crucial for understanding evolutionary processes [6, 11] and is a fundamental need in many clinical applications of phylogenetics and phylodynamics [39].

A large number of computational methods for dating phylogenies are available [e.g., see 26**?**]. A main point of differentiation between these methods is the clock model they assume [29]. Some methods rely on strict molecular clock [41] where rates are assumed to be close to fixed [e.g., 18, 33, 37]. Other methods allow rates to be drawn independently from a stationary distribution [1, 4, 38] or model the evolution of rates with time [14] and allow correlated rates across branches [e.g., 3, 15, 20, 30, 34–36].

Another distinction between methods is the use of parametric models [28]. Many dating methods use an explicit parametric statistical model and formulate dating as estimating parameters in a maximum likelihood (ML) or Bayesian inference framework [e.g., 4, 18, 37, 38]. Yet there are non-parametric methods that rely on optimization problems that do not compute likelihood under explicit statistical model [e.g., 30, 35], albeit with some underlying assumptions. When the assumed parametric model is close to the reality, we expect parametric methods to perform well. However, parametric methods can be sensitive to model deviations, a problem that my be avoided by non-parametric methods.

In this paper, we introduce LogDate, a new method of dating rooted phylogenies based on a non-parametric approach. We define mutation rates necessary to compute time unit branch

lengths as the product of a single global rate and a set of rate multipliers, one per branch. We seek to find the overall rate and all rate multipliers such that the log-transformed rate multipliers 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. While formulation of dating as an optimization problem is not new [18, 37], our contribution here is the realization that log-transforming the rate multipliers results in more accurate dates. Although the LogDate method is non-parametric, it corresponds to the ML estimate under specific parameters of a statistical model. In extensive simulation studies and three biological data sets, we show that a weighted version of LogDate, namely wLogDate, has higher accuracy in inferring node ages compared to alternative methods, including some that rely on time-consuming Bayesian inference.

2 Methods

2.1 Notations

For a rooted binary tree T with n leaves, we let $[0, \ldots, n-2]$ uniquely index internal nodes (root is assigned 0) and let $[n-1, \ldots, 2n-2]$ 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 between $par(i)$ and i as e_i.

We can measure each branch e_i of T in either time unit or substitution unit. 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. We then let b_i be the length of e_i measured in the unit of expected number of substitutions incurred along edge e_i. Let $\tau = [\tau_1, \ldots, \tau_{2n-2}]$ and $\mathbf{b} = [b_1, \ldots, b_{2n-1}]$. To convert from τ_i to b_i, let μ_i be the mutation rate along branch e_i, then $b_i = \mu_i \tau_i$.

For models of sequence evolution that depend on b_i only, μ_i and τ_i are not identifiable. Moreover, $\mathbf{b}$ is not observable directly. Instead, the tree generates sequences with length s, from which we can estimate $\hat{\mathbf{b}} = [\hat{b}_1, \ldots, \hat{b}_{2n-1}]$ using standard approaches (e.g., ML). Thus, the given data is $\hat{\mathbf{b}}$ instead of $\mathbf{b}$.

2.2 Dating as an optimization problem

Given the inferred branch lengths in substitution unit $\hat{\mathbf{b}}$ and the divergence time of some nodes, the goal of phylogenetic dating is to find τ. Let $\mathbf{f} = [f_1, \ldots, f_{2n-2}]$ be unknown variables where each $f_i = \frac{\tau_i}{\hat{b}_i}$. Our goal of finding τ is identical to finding $\mathbf{f}$. Assuming the mutation rates on the branches are distributed around a global rate μ, we define $\nu_i = f_i \mu = \frac{\mu \tau_i}{\hat{b}_i}$. We formulate dating as a constrained optimization problem on $2n-1$ variables $\mathbf{x} = [\nu_1, \ldots, \nu_{2n-2}, \mu]$. Hard calibration points and sampling times define linear constraints on $\mathbf{x}$ (detailed below). Because constraints do not fully determine τ, we need extra assumptions on $\mathbf{x}$, which can 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 Linear constraints Ψ from sampling times

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, $k(k-1)/2$ constraints must hold. However, 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 $t_1, \ldots, t_k$,

we can setup k constraints of the form:

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

For any pair (i, j), the linear constraint given in Eq. 1 can be derived by $C_j - C_i$. Also, we can remove t_0 from the set of constraints by subtracting C_1 from all other constraints $C_2, \ldots, C_k$. This gives us the final $k-1$ linear constraints on ν, which we denote as Ψ. We can easily build Ψ using Algorithm 1 (Supplementary material) in linear time.

2.2.2 Optimization Criteria

Recall that each f_i is influenced both by the distribution of μ_i and the random error in the $\hat{b}_i$ estimation of b_i. In traditional strict-clock models [41] such as those used by Least-Squares Dating (LSD) [37] and Langley-Fitch (LF) [18], a constant 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.

Langley and Fitch [18] (LF) modeled the number of substitutions *per sequence site* on a branch i by a Poisson distribution with mean $\lambda = \mu\tau_i$. The total *observed* number of substitutions can be modeled as $s\hat{b}_i \sim Poisson(s\mu\tau_i)$ ($s =$ the sequence length). The log likelihood function is:

$$\sum_{i=1}^{2n-2} \left(s\hat{b}_i \log(s\hat{b}_i) - \log((s\hat{b}_i)!) \right) + \sum_{i=1}^{2n-2} s\hat{b}_i \left(\log \nu_i - \nu_i \right).$$

Given s and $\hat{b}_i$, LF finds $\mathbf{x}$ that maximizes likelihood subject to the constraints:

$$\mathbf{x}^* = \arg\min_{\nu} \sum_{i=1}^{2n-2} \hat{b}_i \big(\nu_i - \log \nu_i\big) \quad \text{subject to } \Psi. \tag{2}$$

To *et al.* 37 assume $\hat{b}_i$ follows a Gaussian model $\hat{b}_i \sim Gaussian(\mu\tau_i, \sigma_i^2)$ and approximate the variance by $\frac{\hat{b}_i}{s}$ (the method includes smoothing strategies omitted here). Then, the negative log likelihood function is proportional to:

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

Thus, the ML estimate can be formulated as:

$$\mathbf{x}^* = \arg\min_{\nu} \sum_{i=1}^{2n-2} \hat{b}_i (1 - \nu_i)^2 \quad \text{subject to } \Psi. \tag{3}$$

Both LF and LSD have convex formulations. Langley and Fitch 18 proved that their negative log-likelihood function is convex and thus the local minimum is also the global minimum. Our constraint-based formulation of LF also can be easily proved convex by showing its Hessian matrix is positive definite. To *et al.* 37 pointed out their objective function is a weighted least-squares. Using our formulation, we also see that Eq. 3 together with the calibration constraints form a standard convex quadratic optimization problem which has a unique analytical solution.

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 [12, 32**?**], motivating relaxed clocks, typically by assuming that μ_is are drawn i.i.d. from some distribution [e.g., 1, 4, 38]. Most methods rely on presumed parametric distributions (typically, LogNormal, Exponential, or Gamma) and estimate parameters using ML [38], MAP [1], or MCMC [2, 4].

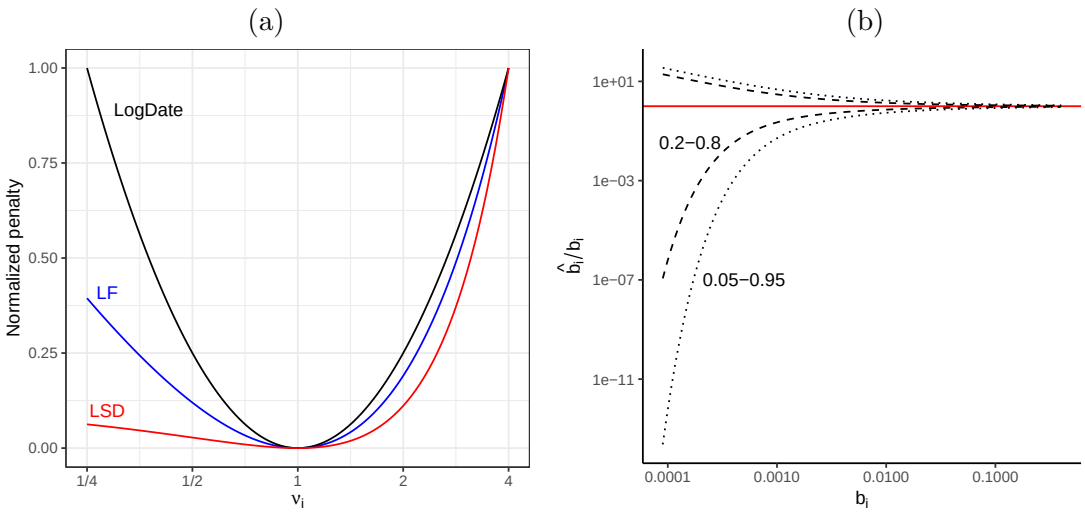

Figure 1: (a) The penalty associated to multiplying a single edge i with multiplier ν_i in LSD, LF, and LogDate approaches, as shown in Equations 2, 3, and 4. To allow comparison, we normalize the penalty to be zero at $\nu = 1$ and to be 1 at $\nu = 4$. (b). The confidence interval of the ratio between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

To develop a non-parametric method that does not assume a strict clock, we avoid explicit assumptions on the model that generates the rate multipliers. Instead, we follow a simple 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, we prefer solutions that minimize deviations from a strict clock while allowing deviations.

A natural way to formalize this principle is to minimize the variance of f_i. This can be achieved by finding μ and all ν_i such that ν_i is centered at 1 and $\sum_{i=1}^{2n-2}(\nu_i - 1)^2$ is minimized. Interestingly, the ML function used by LSD (Eq. 3) 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 log-transformation is the main insight of our method.

2.3.1 LogDate optimization function

Given $\hat{\mathbf{b}}$ and the set of calibration constraints described earlier, we seek to find $\mathbf{x} = [\nu_1, \ldots, \nu_{2n-2}, \mu]$ as:

$$\mathbf{x}^* = \arg\min_{\nu} \sum_{1}^{2n-1} \log^2(\nu_i) \quad \text{subject to } \Psi. \tag{4}$$

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. 4, the optimal solution would push μ to the mean rate.

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. The formal proof is given in Claim 1 (Supplementary material).

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 ($\frac{\hat{b}_i}{b_i}$). 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 $var(\hat{b}_i) = \frac{\mu\tau_i}{s} = \frac{b_i}{s}$. 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. 2 and 3) 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 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 the weights are meant to reflect the variance of be_i (logarithmic scale), we infer that weights should be monotonically increasing with $\hat{b}_i$ (Fig. 1b). We use $\sqrt{\hat{b}_i}$ as weights, a selection driven by simplicity and empirical performance.

The shortest branches require even more care. When the branch is very short, for a limited-size alignment, with high probability, we will see zero mutations. 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, Sanderson 31 collapses all branches with length $\hat{b}_i < 1/s$). To *et al.* 37 proposed adding a smoothing constant c/s to each $\hat{b}_i$ to estimate b_i variance, where c is a parameter that the user can tune. 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 $1-\alpha$ for $\alpha \in [0, 1]$. Assuming $s\hat{b} \sim Poisson(sb)$, it is easy to show that the maximum $\tilde{b}$ that satisfies $Pr(s\hat{b} = 0 | b = \tilde{b}) \geq 1 - \alpha$ is $\tilde{b} = \frac{-\log(1-\alpha)}{s}$. By default, we set $\alpha = 0.01$ but allow the user to adjust this parameter. Thus, we define the weighted LogDate (wLogDate) as follows:

$$\mathbf{x}^* = \arg\min_{\nu} \sum_{1}^{2n-1} \sqrt{\tilde{b} + \hat{b}_i} \log^2(\nu_i) \quad \text{subject to } \Psi. \tag{5}$$

2.3.2 Solving the optimization problem

LogDate problems (Eq. 4 and Eq. 5) 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 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` [17] method. To help the solver work efficiently, we incorporate three techniques that we next describe.

Computing Jacobian and Hessian matrices analytically helps speedup the search. By taking the partial derivative of each ν_i, we can compute the Jacobian, J, of Eq. 5. Also, since Eq. 5 is separable, its Hessian H is a $(2n-2) \times (2n-2)$ diagonal matrix. Simple derivations give us:

$$
\begin{aligned}
J =& \Big[2\sqrt{\tilde{b}+\hat{b}_1}\frac{\log \nu_1}{\nu_1}, \ldots, 2\sqrt{\tilde{b}+\hat{b}_{2n-2}}\frac{\log \nu_{2n-2}}{\nu_{2n-2}}\Big]^T \\
\text{and} \quad H_{ii} =& 2\sqrt{\tilde{b}+\hat{b}_i}\frac{1-\log \nu_i}{\nu_i^2} \ .
\end{aligned}
$$

Sparse matrix representation further saves space and computational time. The Hessian matrix is diagonal, allowing us to store only the diagonal elements. In addition, the constraints matrix defined by Ψ is highly sparse. If all sampling times are given at the leaves, the number of non-zero elements in our $(n-1) \times (2n-1)$ matrix is $O(n \log n)$ (Claim 3; Supplementary material). If the tree is either caterpillar or balanced, the number of non-zeroes reduced to $\Theta(n)$. Thus, we use sparse matrix representation implemented in the Scipy package. This significantly reduces the running time of LogDate.

Starting from multiple feasible initial points is necessary given that our optimization problem is non-convex. Providing initial points that are feasible (i.e. satisfied the calibration constraints) helps the SciPy solver work efficiently. We designed a heuristic strategy to find multiple initial points given sampling times $t_1, \ldots, t_n$ of all the leaves (as is common in phylodynamics).

We first describe the process to get a single initial point. We compute the root age t_0 and μ using root-to-tip regression (RTT) [33]. 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. Finally, we set μ to the mutation rate given by RTT at the root and for each branch i, compute ν_i as $\frac{\mu\tau_i}{\hat{b}_i}$. In a tree of n leaves, we have $2^{(n-1)} - 1$ ways to select the initial non-empty set S, giving us enough room for randomization.

2.4 Experiments

2.4.1 Datasets

Simulated data. To *et al.* 37 simulated a 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.* 37, 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 To *et al.* 37. 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 [24] under the same settings as To *et al.* 37.

H1N1 2009 pandemic. We re-analyze the H1N1 biological data provided by To *et al.* 37 which includes 892 H1N1pdm09 sequences collected worldwide between 13 March 2009 and 9 June 2011. We reuse the estimated PhyML [8] trees, 100 bootstrap replicates, and all the results of the dating methods other than LogDate that are provided by To *et al.* 37.

San Diego HIV. We study a dataset of 926 HIV-1 subtype B *pol* sequences obtained in San Diego between 1996 and 2018 as part of the PIRC study. We use IQTree [23] 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.* 5 using BEAST and re-analyzed by Volz and Frost 38 using IQTree to estimate the ML tree and *treedater* to infer node ages. We run LSD, LF, and wLogDate on the IQTree from Volz and Frost 38 and use the BEAST trees from Dudas *et al.* 5, 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 [27] between the IQTree and the BEAST-MCC tree.

2.4.2 Methods Compared

We compare wLogDate to three other methods: LSD, LF as implemented in r8s [31], and BEAST [2]. We used phyML [8] 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.

For the Bayesian method BEAST, we fixed the true rooted tree topology and only inferred node ages. Following To *et al.* 37, 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). We always set the length of the MCMC chain to 10^7 generations, burn-in to 10%, and sampling to every 10^4 generations (identical to To *et al.* 37).

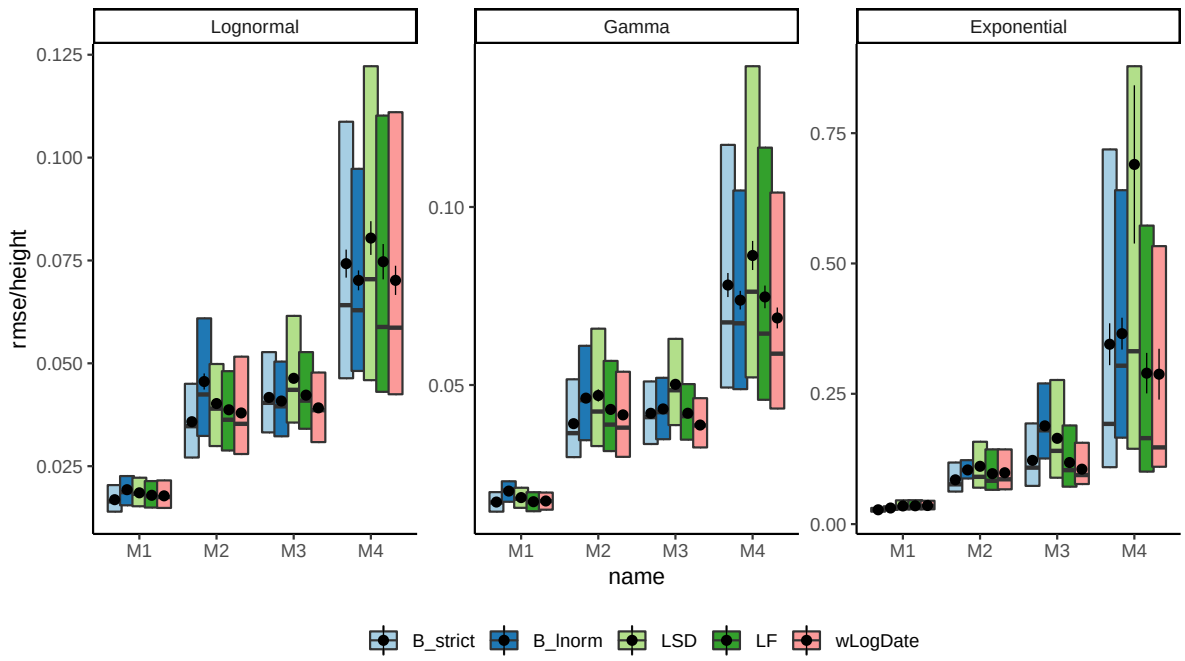

Figure 2: Distributions of RMSE normalized by the tree height for internal node ages inferred by all methods on model trees M1–M4, each with clock models Lognorm, Gamma, and Exponential. Boxes show median, 10% and 90% quantiles; dots and error bars show mean and standard error (100 replicates).

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, we examine the time to Most Recent Common Ancestor (tMRCA) of all nodes and the mutation rate. On real data, we show lineage-through-time (LTT) plots [22], which trace the number of lineages at any point in time, and compare tMRCA times to the literature.

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 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.

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

model	Clock model	B_lnorm	B_strict	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
Averaged 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.

simulates inter-host evolution and high the largest height, 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, results 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 M4, it has a much higher error, especially 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. This result is surprising given the correct model specification. Nevertheless, BEAST-LogNormal is competitive only under the LogNormal model of rate variation and is one of the two worst methods elsewhere. Thus, BEAST-LogNormal is sensitive to model misspecification. In contrast, BEAST-strict is 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 follows 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 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, the distinction between methods is less pronounced (Table S1). wLogDate is the best method jointly with the two strict clock models BEAST-strict and LF. Overall, even though LF and wLogDate tend to over-estimate mutation rates, both 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

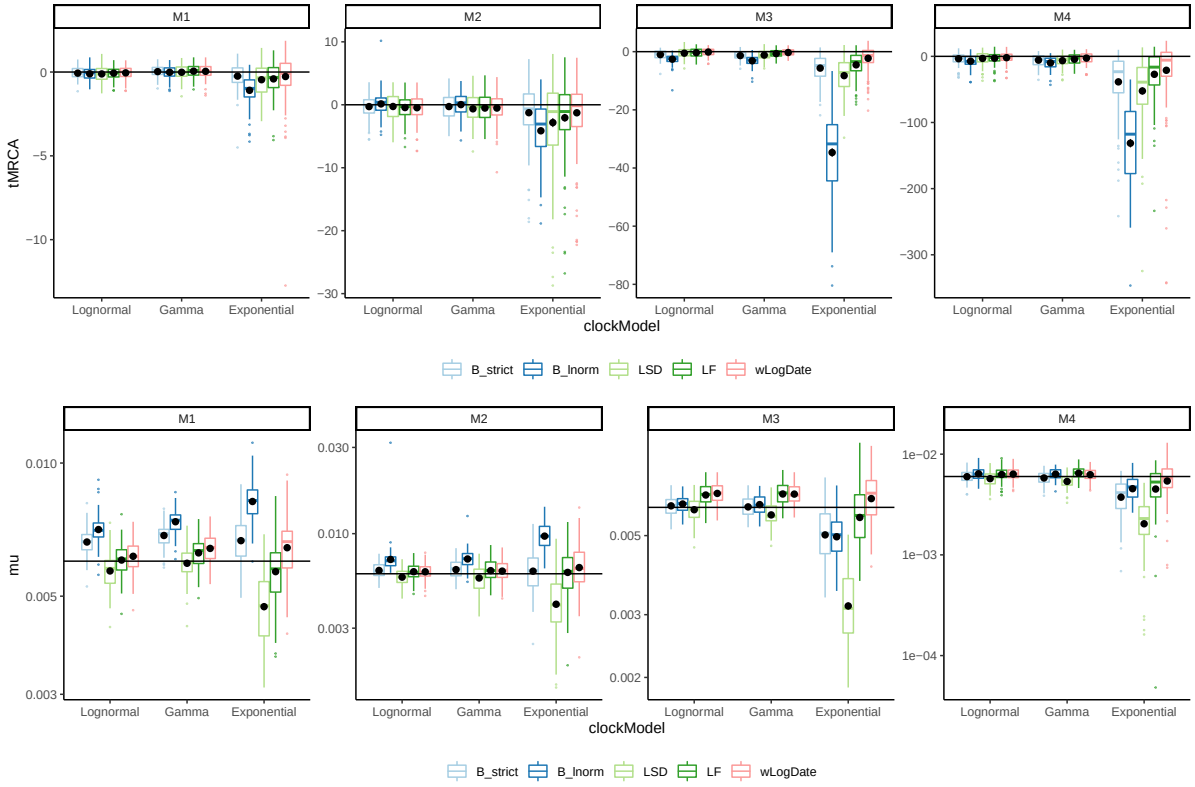

Figure 3: The inferred (top) tMRCA and (bottom) expected mutation rate on different tree models and clock models. Distributions are over 100 replicates. The solid horizontal lines indicate the true mutation rate and tMRCA. Each black is the average of the inferred values for each method under each model 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.

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 is 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 [10, 19, 25]. 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 match the presumed ground truth.

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 [7, 40] 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.

On the Ebola dataset, the BEAST-1000 trees obtained from Dudas *et al.* 5 inferred the tMRCA to be between 13 September 2013 and 26 January 2014 (95% credible interval) and the

Tree	Clock Model	B_strict	B_lnorm	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 of the second best method (wLogDate or the alternative) divided by the error of the best method.

BEAST-MCC inferred the tMRCA to be 5 December 2013 as reported by Volz and Frost 38. 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 times of dating methods on the three real biological datasets (Table S2). LSD was always the fastest, running in just seconds, compared to minutes for LF and wLogDate. LF is faster than wLogDate on the H1N1 and HIV data, while on Ebola data, wLogDate is faster. We report the running time for wLogDate as the sequential run of 10 independent starting points and note that wLogDate can easily be parallelized.

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 the log transformation. 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 ages 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, which is harder to solve than the convex problems solved by LSD and LF. However, we note that the problem is convex for rate multipliers between 0 and e. In addition, 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. 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 (Fig. 1a).

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

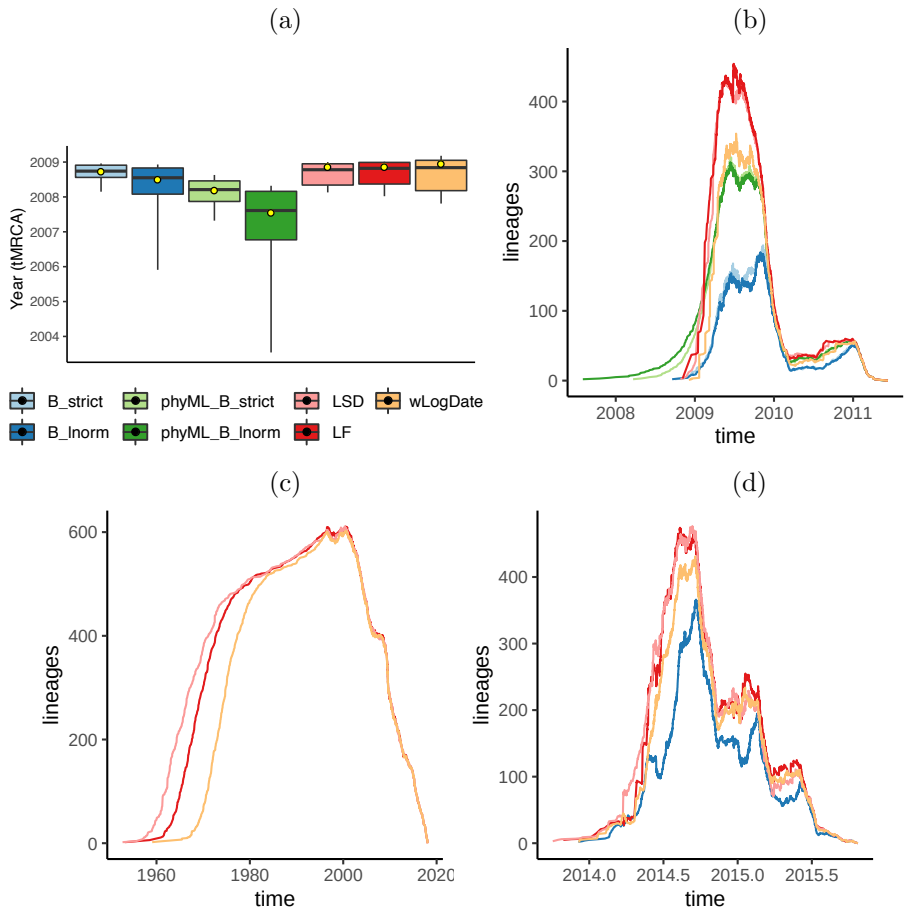

Figure 4: (a) Inferred tMRCA of the H1N1 dataset. Boxplots represent the median, maximum, minimum, 97.5% and 2.5% quantiles of the bootstrap estimates for LF, LSD, and wLogDate, and of the posterior distribution for BEAST. Yellow dot shows the inferred tMRCA of the best ML or MAP tree. BEAST was run 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.* 37. (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.

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, being free from a parametric model, in our judgment, is a strength of the method.

The LogDate approach can be further improved in several aspects. First, the current formulation of LogDate assumes a rooted phylogenetic tree, whereas most inferred trees are unrooted. Rooting phylogenies is a non-trivial problem and can also be done based on principles of minimizing rate variation [21]. Similar to LSD, LogDate can be generalized to unrooted trees by rooting the tree on each branch, solving the optimization problem for each root, and choosing the root that minimizes the (w)LogDate objective function. We leave the careful study of such an approach to the future work.

Beyond rooting, the future work can explore the possibility of building a specialized solver for LogDate to gain speedup. One approach could be exploiting the special structure of the search space defined by the tree, which is the strategy employed by LSD to solve the least-squares optimization in linear time. Divide-and-conquer may also prove effective.

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. More generally, 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 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 [9, 13]. Thus, applying LogDate in such settings requires further tweaks to the optimization formulation, including changing equality to inequality constraints.

In the studies of LogDate accuracy, we have left several questions for the future works. 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 with estimated branch lengths. Future work should study the impact of the incorrect topology on LogDate and other dating methods.

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

5 Acknowledgments

We thank Susan B. Little for providing the San Diego HIV sequence dataset used in this study.

This work was supported by the National Science Foundation (NSF) grant III-1845967 and a developmental grant from the University of California, San Diego Center for AIDS Research (P30 AI036214), supported by the National Institutes of Health.

References

[1] Akerborg, O., Sennblad, B., and Lagergren, J. 2008. Birth-death prior on phylogeny and speed dating. *BMC Evolutionary Biology*, 8(1): 77.

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

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

[4] Drummond, A. J., Ho, S. Y. W., Phillips, M. J., and Rambaut, A. 2006. Relaxed Phylogenetics and Dating with Confidence. *PLoS Biology*, 4(5): e88.

[5] Dudas, G., Carvalho, L. M., Bedford, T., Tatem, A. J., Baele, G., Faria, N. R., Park, D. J., Ladner, J. T., Arias, A., Asogun, D., Bielejec, F., Caddy, S. L., Cotten, M., D'Ambrozio, J., Dellicour, S., Di Caro, A., Diclaro, J. W., Duraffour, S., Elmore, M. J., Fakoli, L. S., Faye, O., Gilbert, M. L., Gevao, S. M., Gire, S., Gladden-Young, A., Gnirke, A., Goba, A., Grant, D. S., Haagmans, B. L., Hiscox, J. A., Jah, U., Kugelman, J. R., Liu, D., Lu, J., Malboeuf, C. M., Mate, S., Matthews, D. A., Matranga, C. B., Meredith, L. W., Qu, J., Quick, J., Pas, S. D., Phan, M. V. T., Pollakis, G., Reusken, C. B., Sanchez-Lockhart, M., Schaffner, S. F., Schieffelin, J. S., Sealfon, R. S., Simon-Loriere, E., Smits, S. L., Stoecker, K., Thorne, L., Tobin, E. A., Vandi, M. A., Watson, S. J., West, K., Whitmer, S., Wiley, M. R., Winnicki, S. M., Wohl, S., Wölfel, R., Yozwiak, N. L., Andersen, K. G., Blyden, S. O., Bolay, F., Carroll, M. W., Dahn, B., Diallo, B., Formenty, P., Fraser, C., Gao, G. F., Garry, R. F.,

Goodfellow, I., Günther, S., Happi, C. T., Holmes, E. C., Kargbo, B., Keïta, S., Kellam, P., Koopmans, M. P. G., Kuhn, J. H., Loman, N. J., Magassouba, N., Naidoo, D., Nichol, S. T., Nyenswah, T., Palacios, G., Pybus, O. G., Sabeti, P. C., Sall, A., Ströher, U., Wurie, I., Suchard, M. A., Lemey, P., and Rambaut, A. 2017. Virus genomes reveal factors that spread and sustained the ebola epidemic. *Nature*, 544(7650): 309–315.

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

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

[8] Guindon, S., Dufayard, J.-F., Lefort, V., Anisimova, M., Hordijk, W., and Gascuel, O. 2010. New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. *Systematic Biology*, 59(3): 307–321.

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

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

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

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

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

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

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

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

[17] Lalee, M., Nocedal, J., and Plantenga, T. 1998. On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, 8(3): 682–706.

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

[19] Lemey, P., Suchard, M., and Rambaut, A. 2009. 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. 20029613[pmid].

[20] Lepage, T., Bryant, D., Philippe, H., and Lartillot, N. 2007. A General Comparison of Relaxed Molecular Clock Models. *Molecular Biology and Evolution*, 24(12): 2669–2680.

[21] Mai, U., Sayyari, E., and Mirarab, S. 2017. Minimum variance rooting of phylogenetic trees and implications for species tree reconstruction. *PLOS ONE*, 12(8): e0182238.

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

[23] Nguyen, L. T., Schmidt, H. A., Von Haeseler, A., and Minh, B. Q. 2015. IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32(1).

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

[25] Rambaut, A. and Holmes, E. 2009. The early molecular epidemiology of the swine-origin

a/h1n1 human influenza pandemic. *PLoS currents*, 1: RRN1003–RRN1003. 20025195[pmid].

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

[27] Sand, A., Holt, M. K., Johansen, J., Fagerberg, R., Brodal, G. S., Pedersen, C. N. S., and Mailund, T. 2013. Algorithms for computing the triplet and quartet distances for binary and general trees. *Biology*, 2(4): 1189–209.

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

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

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

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

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

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

[34] Snir, S., Wolf, Y. I., and Koonin, E. V. 2012. Universal Pacemaker of Genome Evolution. *PLoS Computational Biology*, 8(11): e1002785.

[35] Tamura, K., Battistuzzi, F. U., Billing-Ross, P., Murillo, O., Filipski, A., and Kumar, S. 2012. Estimating divergence times in large molecular phylogenies. *Proceedings of the National Academy of Sciences*, 109(47): 19333–19338.

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

[37] To, T.-H., Jung, M., Lycett, S., and Gascuel, O. 2015. Fast Dating Using Least-Squares Criteria and Algorithms. *Systematic Biology*, 65(1): 82–97.

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

[39] Volz, E. M., Koelle, K., and Bedford, T. 2013. Viral Phylodynamics. *PLoS Computational Biology*, 9(3).

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

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

Supplementary material

Supplementary text

Algorithm 1 Setup the linear constraints Ψ for tree T given a set of calibration points.

```
function SetupConstraints(T)
    Ψ ← {}
    for w in post-order traversal of T do
        if w is a leaf then
            if  w is calibrated then
                nearest_timepoint(w) ← w
            else
                nearest_timepoint(w) ← ∅
        else
            (w1, w2) ← Children(w)
            u ← nearest_timepoint(w1)
            v ← nearest_timepoint(w2)
            if w is calibrated  then
                nearest_timepoint(w) ← w
                if u! = ∅ then
                    Ψ ← Ψ ∪ ψ(w, u)
                if v! = ∅ then
                    Ψ ← Ψ ∪ ψ(w, v)
            else
                if u == ∅ then
                    nearest_timepoint(w) ← v
                else
                    if v == ∅ then
                        nearest_timepoint(w) ← u
                    else
                        nearest_timepoint(w) ← u if (d(w, u) < d(w, v)) else v
                        Ψ ← Ψ ∪ ψ(u, v)
```

Claim 1. *The optimal* $\mathbf{x}^*$ *in Eq. 4 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 follows:

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

Thus, the ML estimate can be found by minimizing Eq. 4. □

Lemma 2. *The length of the shortest path from the root of a binary tree to its leaves is at most* $\log n$ *where* n *is the number of leaves in the tree.*

Proof. Consider a rooted binary tree $\mathcal{T}$ with n leaves; let r be the root and h be the length of the shortest path from r to the leaves of $\mathcal{T}$. We need to prove that $h \leq \log_2 n$.

Let $\mathcal{D}_i$ be the set of nodes in $\mathcal{T}$ with depth i, that is, $\mathcal{D}_i = \{w \in \mathcal{T} | d(r, w) = i\}$. We first prove that $|\mathcal{D}_i| = 2^i \ \forall i \leq h$ where $|\mathcal{D}_i|$ denotes the cardinality of $\mathcal{D}_i$. We prove this by induction. The base case $i = 0$ holds since the root r is the only node with depth 0. Suppose we have $|\mathcal{D}_k| = 2^k$ and $k < h$, we need to prove that if $k + 1 \leq h$ then $|\mathcal{D}_{k+1}| = 2^{k+1}$. Note that a node $v \in \mathcal{D}_{k+1}$ if and only if its parent $par(v) \in \mathcal{D}_k$. Because $\mathcal{T}$ is a binary tree, each node in $\mathcal{T}$ must either has no child (leaf node) or two children (internal node). Since $k < h$, there must be no leaf node in $\mathcal{D}_k$, otherwise, a leaf v in $\mathcal{D}_k$ has $d(r, v) = k < h$, which defines a root-to-leaf path that is shorter than h and contradicts the definition of h. Thus, each node in $\mathcal{D}_k$ has exactly 2 children, making $|\mathcal{D}_{k+1}| = 2 * |\mathcal{D}_k| = 2 * 2^k = 2^{k+1}$.

Now we have $|\mathcal{D}_h| = 2^h$. To prove that $h \leq \log_2 n$, note that $\mathcal{D}_h$ contains a mixture of leaves and internal nodes and each internal node in $\mathcal{D}_h$ must have more than one leaf below it. Therefore, the size of $\mathcal{D}_h$ is at most the size of the leaf set of $\mathcal{T}$; that is, $|\mathcal{D}_h| \leq n$. Thus, we have $2^h = |\mathcal{D}_h| \leq n \implies h \leq \log_2 n$. □

Claim 3. *If all the leaves have sampling times and there is no other calibration points given for internal nodes, the matrix corresponding to the constraints* Ψ *setup by Algorithm 1 has* $O(n\log(n))$ *non-zero elements, where* n *is the number of leaves in the input tree* $\mathcal{T}$.

Proof. Let $\mathcal{P}(w)$ denote the shortest path from a node w to its leaves and let $|\mathcal{P}(w)|$ denote the length of this path. Let $\mathcal{T}_w$ be the clade of $\mathcal{T}$ below w and let $|w|$ denote the size of this clade (i.e. the number of leaves below w). Applying lemma 2 on $\mathcal{T}_w$, we have $|\mathcal{P}(w)| \leq \log_2 |w| \leq \log_2 n$ for all $w \in \mathcal{T}$.

Note that if all leaves have sampling times, Algorithm 1 adds exactly one constraint for each internal node in the tree. For each node w with two children $c_l(w)$ and $c_r(w)$, the non-zero elements of the constraint added when node w is visited must locate on $\mathcal{P}(c_l(w))$, $\mathcal{P}(c_r(w))$, and the two branches $(w, c_l(w))$ and $(w, c_r(w))$. Let η_w denote the number of non-zero elements of the constraint defined by node w, then $\eta_w \leq |\mathcal{P}(c_l(w))| + |\mathcal{P}(c_r(w))| + 1 + 1 \leq 2\log_2 n + 2$. Thus, the total number of non-zeros in all constraints corresponding to the $n-1$ internal nodes is bounded above by $(n-1)(2\log_2 n + 2) \in O(n\log n)$. □

Supplementary figures and tables

Tree Model	Clock Model	BEAST_strict	BEAST_lnorm	LF	LSD	wLogDate
M1	Lognormal	0.0007	0.0011	**0.0004**	0.0005	**0.0004**
	Gamma	0.0009	0.0014	0.0005	**0.0004**	0.0006
	Exponential	0.0009	0.0022	**0.0008**	0.0013	0.0009
M2	Lognormal	**0.0005**	0.0014	**0.0005**	**0.0005**	**0.0005**
	Gamma	**0.0006**	0.0013	0.0007	0.0007	0.0007
	Exponential	**0.0013**	0.0038	0.0015	0.0020	0.0015
M3	Lognormal	**0.0003**	**0.0003**	0.0006	0.0004	0.0006
	Gamma	**0.0003**	**0.0003**	0.0006	0.0004	0.0006
	Exponential	0.0010	0.0011	**0.0009**	0.0027	**0.0009**
M4	Lognormal	**0.0007**	0.0008	0.0008	**0.0007**	0.0008
	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.

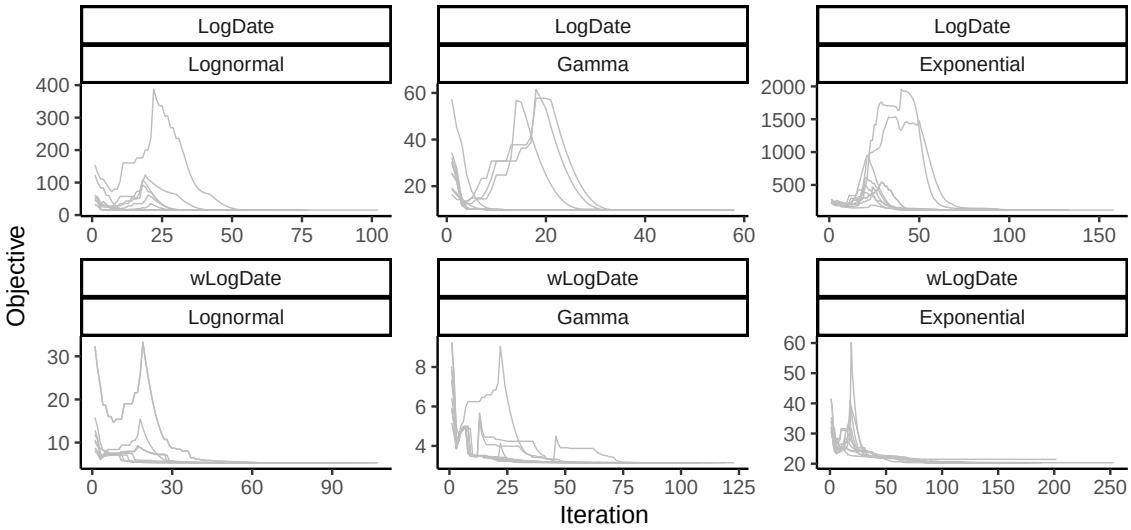

Figure S1: Objective value versus iteration of the LogDate and wLogDate runs on one arbitrarily selected simulated tree (M4, replicate 2). Each of the two methods were run using 10 random initial points generated using the strategy described in **??**.

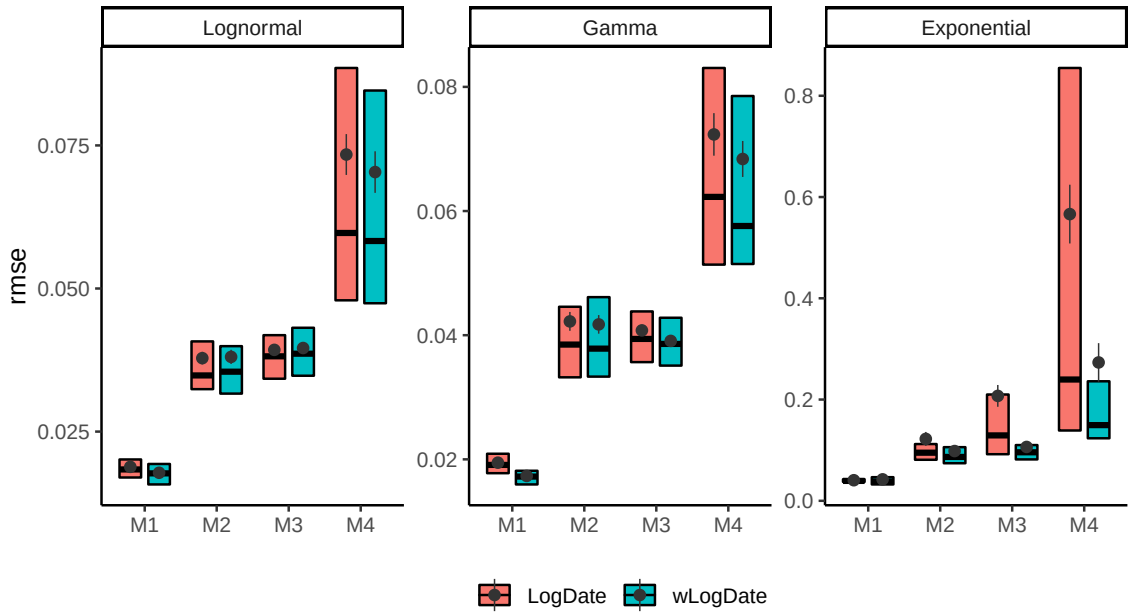

Figure S2: RMSE of the internal node ages inferred by LogDate and wLogDate (the weighted version).

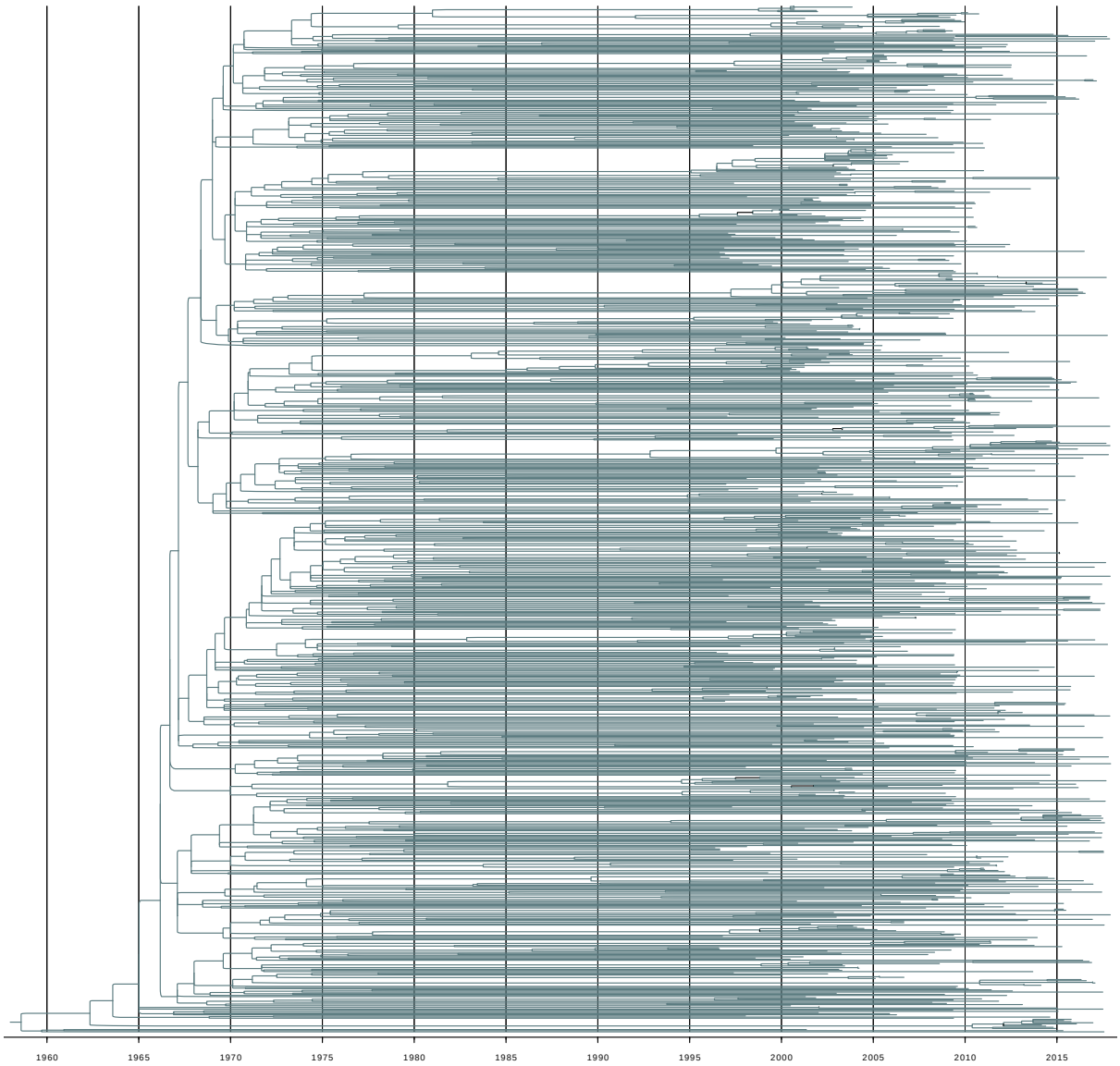

Figure S3: A TimeTree of San Diego HIV epidemic according to wLogDate.