Systematic Biology Advance Access published November 11, 2015

Syst. Biol. 00(0):1–16, 2015
© The Author(s) 2015. Published by Oxford University Press, on behalf of the Society of Systematic Biologists.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.
DOI:10.1093/sysbio/syv068

Fast Dating Using Least-Squares Criteria and Algorithms

THU-HIEN TO[1], MATTHIEU JUNG[1,2], SAMANTHA LYCETT[3], AND OLIVIER GASCUEL[1,*]

[1]*Institut de Biologie Computationnelle, LIRMM, UMR 5506 CNRS – Université de Montpellier, France;* [2]*IGBMC (Institut de Génétique et de Biologie Moléculaire et Cellulaire), INSERM, U596, CNRS, UMR7104, Université de Strasbourg, Illkirch, France;* [3]*Institute of Evolutionary Biology, University of Edinburgh, Ashworth Laboratories, Edinburgh, UK*

**Correspondence to be sent to: Institut de Biologie Computationnelle, LIRMM, UMR 5506 CNRS – Université de Montpellier, 161 rue Ada, 34392 Montpellier, France; Email: gascuel@lirmm.fr*

Received 10 September 2014; reviews returned 16 September 2015; accepted 16 September 2015
Associate Editor: Lars Jermiin

Abstract.—Phylogenies provide a useful way to understand the evolutionary history of genetic samples, and data sets with more than a thousand taxa are becoming increasingly common, notably with viruses (e.g., human immunodeficiency virus (HIV)). Dating ancestral events is one of the first, essential goals with such data. However, current sophisticated probabilistic approaches struggle to handle data sets of this size. Here, we present very fast dating algorithms, based on a Gaussian model closely related to the Langley–Fitch molecular-clock model. We show that this model is robust to uncorrelated violations of the molecular clock. Our algorithms apply to serial data, where the tips of the tree have been sampled through times. They estimate the substitution rate and the dates of all ancestral nodes. When the input tree is unrooted, they can provide an estimate for the root position, thus representing a new, practical alternative to the standard rooting methods (e.g., midpoint). Our algorithms exploit the tree (recursive) structure of the problem at hand, and the close relationships between least-squares and linear algebra. We distinguish between an unconstrained setting and the case where the temporal precedence constraint (i.e., an ancestral node must be older that its daughter nodes) is accounted for. With rooted trees, the former is solved using linear algebra in linear computing time (i.e., proportional to the number of taxa), while the resolution of the latter, constrained setting, is based on an active-set method that runs in nearly linear time. With unrooted trees the computing time becomes (nearly) quadratic (i.e., proportional to the square of the number of taxa). In all cases, very large input trees ($> 10,000$ taxa) can easily be processed and transformed into time-scaled trees. We compare these algorithms to standard methods (root-to-tip, r8s version of Langley–Fitch method, and BEAST). Using simulated data, we show that their estimation accuracy is similar to that of the most sophisticated methods, while their computing time is much faster. We apply these algorithms on a large data set comprising 1194 strains of Influenza virus from the pdm09 H1N1 Human pandemic. Again the results show that these algorithms provide a very fast alternative with results similar to those of other computer programs. These algorithms are implemented in the LSD software (least-squares dating), which can be downloaded from http://www.atgc-montpellier.fr/LSD/, along with all our data sets and detailed results. An Online Appendix, providing additional algorithm descriptions, tables, and figures can be found in the Supplementary Material available on Dryad at http://dx.doi.org/10.5061/dryad.968t3. [Active-set method; algorithms; computer simulations; dating; influenza (H1N1); least-squares; linear algebra; molecular clock; serial data; substitution rate estimation; temporal precedence constraints; viruses.]

INTRODUCTION

The explosion of genetic data and progress in phylogenetic reconstruction algorithms has resulted in increasing utility and popularity of phylogenetic analyses. Data sets with thousands of taxa are becoming more and more common, especially amongst virus evolution studies. Moreover, a number of studies have used molecular-dating techniques to tackle a wide range of biological questions, for example, in systematics for timing the tree of life (Hedges and Kumar 2009; Jetz et al. 2014), in epidemiology to trace back the phylodynamics and phylogeography of epidemics (Grenfell et al. 2004; Volz et al. 2013), and in functional genomics to decipher orthology/paralogy relationships within gene families and improve reconciliation inferences (Akerborg et al. 2009; Doyon et al. 2011; Rasmussen and Kellis 2012).

Currently, the most popular dating approaches are based on sophisticated probabilistic models, most often implemented in the Bayesian framework and able to account for complex priors (Thorne and Kishino 2002; Rannala and Yang 2007; Drummond and Rambaut 2007; Guindon et al. 2010). Maximum-likelihood methods have also been designed to deal with simpler models (Rambaut 2000). Corresponding computer programs take a sequence alignment and a set of known dates as input and return a time-scaled tree, with estimates of the substitution rate(s) and of the dates of all tree nodes. Some programs (e.g., PAML, Rannala and Yang 2007) perform calculations on a fixed, user-supplied tree, while others (e.g., BEAST, Drummond and Rambaut 2007; Drummond et al. 2012) infer the tree from the sequence alignment. These programs typically contain several submodels, which describe the substitution process (e.g., GTR, Γ distribution of rates across sites, etc.), the tree (e.g., coalescent, constant or varying population size, birth–death, etc.), priors on the parameter values and, most importantly regarding dating, the molecular clock. We distinguish the strict molecular clock (SMC) model, where the substitution rate is assumed to be constant across all tree branches, and uncorrelated and correlated relaxed-clock models. With uncorrelated models, the rate associated with each branch is drawn independently from a common underlying distribution; these models are commonly used with fast-evolving species over short time periods, typically with viruses for which there is no strong evidence of rate correlation among branches (Drummond et al. 2006). With correlated (also

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

called autocorrelated) models, the rate distribution for a particular branch depends on the rate value of the neighboring branches; the use of correlated models seems to be the preferred choice with large groups of slowly evolving species, for example mammals, where it has been demonstrated that some subgroups evolve faster than others (e.g., the rodents, Douzery et al. 2003). However, the advantages and limitations of this large variety of models is still a question of debate (Drummond et al. 2006; Lepage et al. 2007; Battistuzzi et al. 2010). All these models and methods have shown to be useful in a number of studies, but they are computationally intensive, making it virtually impossible to deal with the larger data sets available today, even when using sophisticated implementations and powerful computers (Ayres et al. 2012). Typically, days of computations are required to analyze a few hundred taxa, although faster approaches are available, using complex algorithmic approaches (Akerborg et al. 2008; Guindon et al. 2010) and multinormal approximations of the likelihood function (Thorne et al. 1998).

Here we are interested in dating very large phylogenies, typically with a thousand tips or more, a need that is becoming increasingly common, for example, in molecular epidemiology. We propose distance-based algorithms to estimate rates and dates, a mathematical and computational framework that has proven to produce fast and fairly accurate tools in phylogenetics (e.g., NJ, Saitou and Nei 1987). Several distance-based (as opposed to sequence-based, see above) dating methods have already been proposed. Most of these methods deal with time calibration points, where the dates of certain ancestral nodes in the tree are known, possibly with uncertainty (e.g., min–max values), and all of the tree tips are contemporaneous. These methods input a rooted tree with time calibration points, and return a time-scaled, ultrametric tree. PATHd8 (Britton et al. 2007) and the Tamura et al. (2012) method use smoothing and averaging techniques to accommodate for local rate variations. Xia and Yang's (2011) method assumes a SMC or two different local clocks, and achieves least-squares estimations under these assumptions. Sanderson's (1997, 2002) approach is based on a penalized-likelihood criterion to account for the autocorrelation of rates, combined with standard optimization techniques (see also TreePL, Smith and O'Meara 2012). Based on computer simulations, these fast methods were shown to be accurate by their authors, producing time-scaled trees similar to those obtained using sequence-based approaches.

The focus of the present study is on serial phylogenies, where the tips of the tree have been sampled through times. Such phylogenies are common with fast-evolving organisms (e.g., human immunodeficiency virus (HIV)), where a few years of evolution induce significant changes at the sequence level (Drummond et al. 2003a). Serial phylogenies are also used with ancient DNA (Lambert et al. 2002). Moreover, close relationships exist between the calibration-points and dated-tips approaches (Ronquist et al. 2012). Several methods have been proposed in this framework. One of the very first is root-to-tip regression (RTT) (Shankarappa et al. 1999; Drummond et al. 2003b): assuming a SMC, the root-to-tip distance in the input tree should be proportional to the corresponding elapsed time; then, a standard regression of the root-to-tip distance for every tip as a function of its date provides estimates of the substitution rate (regression slope) and root date (intercept with X-axis). This method is very fast and can be extended to unrooted trees by searching among all tree branches for the best root position, according to some numerical criterion (e.g., the sum of regression residues, to be minimized). However, this method does not provide estimates for the dates of internal nodes, and thus does not output time-scaled trees. The same holds for TREBLE (Yang et al. 2007), which is a triplet-based alternative to RTT that is also able to process unrooted trees. To obtain date estimates of the internal nodes, sUPGMA (Drummond and Rodrigo 2000) combines a regression method to estimate the substitution rate in a first step, corrects the non-contemporaneous tips into contemporaneous tips in a second step and then uses UPGMA (Sokal and Michener 1958) to compute the tree. Unlike the former approaches, Langley and Fitch's (LF; 1974) method uses an explicit model. The LF method assumes a SMC with a constant substitution rate, and models the number of substitutions along each branch of the tree by a Poisson distribution. The estimates of the global substitution rate and of the internal node dates are then obtained by maximizing the likelihood of the input, rooted tree. LF is implemented in r8s (Sanderson 2003).

In this article, we study a model analogous to LF's, but using a normal approximation that allows for a least-squares approach, and show that this model is robust to uncorrelated violations of the molecular clock. Using the tree (recursive) structure of the problem at hand, and the close relationships between least-squares and linear algebra, we propose very fast algorithms to estimate the substitution rate and the dates of all internal tree nodes. With rooted trees, the time complexity is nearly linear (i.e., proportional to the number of taxa), while with unrooted trees, it becomes nearly quadratic (i.e., proportional to the square of the number of taxa). In both cases, very large trees (>10,000 taxa) can easily be processed and transformed into time-scaled trees. The article is organized as follows: we first define the model and show its ability to handle uncorrelated rate variations among tree branches, as is commonly assumed with virus data. We then present our two main algorithms, distinguishing the unconstrained setting and the case where the temporal precedence constraints (i.e., an ancestral node must be older than its daughter nodes) are accounted for. Last, we compare these algorithms to standard approaches using simulated data and a large influenza data set. Our algorithms are implemented in the LSD program (least-squares dating), which can be downloaded (along with all data and results reported

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

here) from http://www.atgc-montpellier.fr/LSD/ (last accessed October 2015). An Online Appendix, providing additional algorithm descriptions, tables, and figures, can be found in the in the Supplementary Material available on Dryad at http://dx.doi.org/10.5061/dryad.968t3.

Models and Algorithms

Preliminaries and Notation

Our algorithms take as input a binary phylogenetic tree with branch lengths, inferred by any tree building program, and sampling dates associated with the taxa. As our algorithms are very fast, it is consistent to combine them with fast tree-building methods, for example distance-based methods (e.g., NJ, Saitou and Nei 1987, or FastME, Desper and Gascuel 2002, Lefort et al. 2015), but more accurate results are expected from trees obtained using maximum-likelihood (ML) methods (e.g., PhyML, Guindon and Gascuel 2003, Guindon et al. 2010). However, we shall see that results obtained with both approaches are close. The algorithms accept a rooted or unrooted tree, and for unrooted trees we propose a method to estimate the root position, though simulations show that the use of an outgroup is generally preferable. In the following, we first assume that the tree is rooted, and then summarize the rooting procedure, which is described in more details in the Online Appendix (available as Supplementary Material on Dryad at http://dx.doi.org/10.5061/dryad.968t3).

Given a set of n serially dated sequences, let R be the input rooted binary phylogenetic tree on these sequences with known branch lengths. Enumerate the internal nodes of R by $1,2,\ldots n-1$ and the leaves by n, $n+1,\ldots,2n-1$. Node 1 corresponds to the root. The date of node i is denoted by t_i. So $t_n, t_{n+1},\ldots,t_{2n-1}$ are known. Times are measured from the origin, that is, $t_i \geqslant t_j$ when i is more recent than j.

For every node i different from the root ($i=1$), let $a(i)$ be the parent node of i. For every internal node i, let $s_1(i)$ and $s_2(i)$ be the two direct descendants of i. Let b_i be the length of the branch $(i,a(i))$; b_i is an estimate of the number of substitutions per site that occurred along the branch from time $t_{a(i)}$ to t_i. With a SMC, the substitution rate (i.e., the expected number of substitutions per site per time unit) along the tree is constant and is denoted as ω. The goal of our algorithms is to estimate the substitution rate and the dates of all internal nodes, that is $(\omega, t_1,\ldots,t_{n-1})$.

Probabilistic Model and Objective Function

We use a Gaussian model, which is closely related to that proposed by Langley and Fitch (1974). Assuming a SMC, the expected branch length $E(b_i)$ is equal to ω times the time interval $(t_i - t_{a(i)})$. Due to sampling noise and estimation errors, the branch length estimate b_i (available in input tree R) can be expressed as:

$$b_i = \omega\left(t_i - t_{a(i)}\right) + \varepsilon_i, \tag{1}$$

where ε_i is the noise (error) term. Langley and Fitch's (1974) method assumes a Poisson model for ε_i, which is biologically meaningful (at least with low substitution rates and simple mutation processes). Here, we use a normal approximation for the distribution of the noise term ε_i (such an approximation is quite standard in computational statistics to accelerate the calculations, with a huge number of successful applications in many domains, and sound justifications related to the Law of Large Numbers). We thus assume:

$$\varepsilon_i = N\left(0, \sigma_i^2\right),$$

where $N(0,\sigma_i^2)$ denotes the normal distribution with mean 0 and variance σ_i^2. A limit of this model is that short branches may be negative according to Equation (1), but we impose positivity using temporal precedence constraints (see below). As evolution is independent from one branch to another, we consistently assume that the noise terms are mutually independent. The weighted least squares (WLS) criterion to be minimized (proportional to the log-likelihood assuming this model) is given by:

$$\phi\left(\omega, t_1,\ldots,t_{n-1}\right) = \sum_{i=2}^{2n-1} \frac{1}{\sigma_i^2}\left(b_i - \omega(t_i - t_{a(i)})\right)^2. \tag{2}$$

One difficulty with such a WLS criterion lies in the variance terms σ_i^2, which are unknown and depend on the (unknown) branch lengths and possibly on some model parameters (e.g., Γ distribution of site rates). Fitch and Margoliash's (1967) tree inference method use the square of the pairwise evolutionary distance estimate. We use here another standard approach (for discussion, see Gascuel 1997) derived from the Poisson nature of the substitution process, where

$$\sigma_i^2 = \frac{E(b_i)}{s} = \frac{\omega\left(t_i - t_{a(i)}\right)}{s} \text{ and } \widehat{\sigma_i^2} = \frac{b_i}{s}, \tag{3}$$

with s being the sequence length.

However, the limit of such variance estimates is that overconfidence is given on very short branches, while their short length may be due to sampling randomness or estimation errors. For example, with a null branch length estimate ($b_i = 0$), we have an infinite weight in Equation (2). This makes the method inapplicable, while the observation that $b_i = 0$ most likely is due to the limited amount of sites available. To avoid this problem, we use the following additive smoothing for the variance estimates:

$$\widehat{\sigma_i^2} = \frac{b_i + c/s}{s}, \tag{4}$$

where c is a constant. The higher c is, the closer we are to equal variances, that is, ordinary least squares (OLS). A value of $c=1$ corresponds to Laplace's Rule

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

of Succession, which is commonly used to estimate probabilities with limited numbers of observations (with short branches, b_i is very close to a frequency of observed differences, or p-distance, and $E(b_i)$ to the corresponding probability). Simulation experiments (not shown) indicate that $c=1$ is not large enough and that $c=10$ provides best average results; this is the default value in our computer program, but c can be chosen by the user.

This model accommodates some violations of the molecular clock. Assume a simple model (similar to Drummond et al. 2006; see also Thorne et al. 1998) where the rate ω_i attached to the branch $(i,a(i))$ follows a normal distribution $N(\omega,\xi^2)$. Moreover, assume a simple model for b_i where ε_i in Equation (1) does not depend on the branch specific rate ω_i, but on its expectation ω, that is

$$b_i=\omega_i\left(t_i-t_{a(i)}\right)+N\left(0,\frac{\omega(t_i-t_{a(i)})}{s}\right).$$

Then, it is easily seen that

$$b_i=\omega\left(t_i-t_{a(i)}\right)+N\left(0,\xi^2\left(t_i-t_{a(i)}\right)^2+\frac{\omega(t_i-t_{a(i)})}{s}\right).$$

In other words, b_i follows a normal distribution having a similar form as Equation (1), but the error term incorporates an additional factor (*i.e.*, $\xi^2(t_i-t_{a(i)})^2$), the relevance of which may be tested against the SMC. Moreover, the variance term is an increasing function of b_i, as in Equation (3), meaning that using our algorithms with uncorrelated violations of the molecular clock is still well founded.

To summarize, our model (Eq. (1)) is a normal approximation of the LF model and it naturally accommodates uncorrelated variation of rates across branches. This corresponds to the default option in several programs (e.g., BEAST), which have shown their accuracy and usefulness with numerous data sets (typically viruses, see section 'Introduction'). We certainly do not pretend that this model depicts all the complexity of sequence evolution, but it makes possible very efficient calculations with little loss in terms of estimation accuracy, as described later.

Outline of the Approach

The rate ω is positive, and we can fix in LSD the minimum value of the estimated rate to $\hat{\omega}\geqslant\omega_{min}>0$. Moreover, time is measured forward from the root to the tips of the tree, so it must satisfy the temporal precedence constraints $t_i\geqslant t_{a(i)}$ for every node i that is not the tree root ($i=1$). In other words, any daughter node ($i>1$) is more recent than its parent node ($a(i)$). This is an obvious requirement, analogous to the positivity of branch lengths in phylogenetic trees. However, not all dating methods comply with this requirement (e.g., see our example below with BEAST and the influenza data set), just as some phylogenetic algorithms (e.g., NJ) infer trees with negative branch lengths. The reasons for this are mostly computational. Imposing positivity constraints has a computational cost, as we shall see below in our dating context.

The estimates are obtained by minimizing the objective function ϕ defined in Equation (2). By using $\beta_i=\omega t_i$ for every node $i=1,\ldots,n-1$ (*i.e.* i is an internal node) and $w_i=1/\sigma_i^2$, the function ϕ in Equation (2) becomes:

$$\Psi\left(\omega,\beta_1,\ldots,\beta_{n-1}\right)=\sum_{i=2}^{n-1}w_i\left(b_i-\beta_i+\beta_{a(i)}\right)^2+\sum_{i=n}^{2n-1}w_i\left(b_i-\omega t_i+\beta_{a(i)}\right)^2. \quad (2b)$$

This function is a convex quadratic form (O'Meara 2000) and has a unique minimum (see Proof in the Online Appendix). Therefore, Equation (2) also has a unique minimum. However, to improve numerical precision our algorithms use Equation (2) and not Equation (2b), as in Equation (2b) we have to divide the variables β_i by ω (another variable) to obtain the t_i, which are the true variables of interest.

We propose two different algorithms. One takes into account the temporal precedence constraints, while the other does not. For each algorithm implemented in our computer program LSD, we have two versions: weighted, where each term in Equation (2) is associated with a weight denoted $w_i=1/\widehat{\sigma_i^2}$ (cf. Eq. (4)), and unweighted (all w_i are equal and set to 1). We present the weighted versions in the following, as the unweighted versions are simply obtained by fixing the w_i to 1.

Linear Dating (LD) Algorithm, Without Constraints

Let $B'=(b'_2,\ldots,b'_{2n-1})$, where:

$$\begin{cases}b'_i=b_i, & \text{for } i=2,\ldots,n-1,\\ b'_i=\omega t_i-b_i, & \text{for } i=n,\ldots,2n-1.\end{cases}$$

Then, Equation (1) becomes:

$$\begin{cases}b'_i=\omega\left(t_i-t_{a(i)}\right)+\varepsilon_i, & \text{for } i=2,\ldots,n-1,\\ b'_i=\omega t_{a(i)}+\varepsilon_i, & \text{for } i=n,\ldots,2n-1.\end{cases}$$

These equations can be rewritten, using matrix notation, as $B'=\omega AT+E$, where $T=(t_1,\ldots,t_{n-1})$, E is the error (noise) vector, and A is a $(2n-2)\times(n-1)$ matrix, which depends on the topology of R, such that for any $i=1,\ldots,2n-2$ and $j=1,\ldots,n-1$, we have:

$$A_{ij}=\begin{cases}1, & \text{if } (i+1<n \text{ and } j=i+1) \text{ or } (i+1\geqslant n \text{ and } j=a(i+1)),\\ -1, & \text{if } i+1<n \text{ and } j=a(i+1),\\ 0, & \text{otherwise.}\end{cases}$$

The objective function (Eq. (2)) is then written as $\phi=(B'-\omega AT)^T W(B'-\omega AT)$, where $W=(w_2,\ldots,w_{2n-1})$ is the diagonal matrix of inversed variances. By the

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

pseudo-inverse method, the estimates of ω and T that minimize ϕ, satisfy $\omega(A^T WA)T = A^T WB'$. The latter equation is equivalent to the following system of equations:

$$t_1 = \frac{1}{w_{s_1(1)} + w_{s_2(1)}} \left[w_{s_1(1)} \left(t_{s_1(1)} - \frac{b_{s_1(1)}}{\omega} \right) + w_{s_2(1)} \left(t_{s_2(1)} - \frac{b_{s_2(1)}}{\omega} \right) \right], \quad (5.1)$$

$$t_i = \frac{1}{w_{s_1(i)} + w_{s_2(i)} + w_i} \left[w_{s_1(i)} \left(t_{s_1(i)} - \frac{b_{s_1(i)}}{\omega} \right) + w_{s_2(i)} \left(t_{s_2(i)} - \frac{b_{s_2(i)}}{\omega} \right) + w_i \left(t_{a(i)} + \frac{b_i}{\omega} \right) \right], \quad \text{for } i = 2, \ldots, n-1. \quad (5.i)$$

This system of Equations (5) can also be obtained by taking the first-order derivatives of ϕ with respect to each variable $t_1, \ldots, t_{n-1}$. Based on Equation (1), $t_i = t_{s_1(i)} - b_{s_1(i)}/\omega + \varepsilon_{s_1(i)}/\omega$. Consequently, Equations (5) mean that the estimate of t_i is equal to the weighted average of its estimates with respect to all i's neighbors (2 for tree root ($i=1$) in Eq. (5.1), and 3 for other internal nodes ($i = 2, \ldots, n-1$) in Eqs. (5.i)). The resolution of Equations (5) can be achieved in linear time (i.e., $O(n)$, where n is the number of tree tips), while solving such a system with generic tools requires cubic time (i.e., $O(n^3)$). The technical details of the LD algorithm are given in the Online Appendix. The main idea is to simplify progressively this system (Eq. (5)) by recursive replacements using specific tree traversals. After the first, bottom-up set of replacements, we have

$$t_i = x_i t_{a(i)} + y_i + \frac{z_i}{\omega}, \text{ for } i = 2, \ldots, n-1, \text{ where } x_i, y_i, z_i \text{ are constants.} \quad (6.i)$$

After the second, top-down set of replacements, we obtain

$$t_i = u_i + \frac{v_i}{\omega}, \text{ for } i = 1, \ldots, n-1, \text{ where } u_i \text{ and } v_i \text{ are constants.} \quad (7.i)$$

By using Equations (7) into Equation (2), ϕ becomes a quadratic function of one variable ω. Then, it is easy to compute the unique $\hat{\omega}$ value that minimizes this function. If $\hat{\omega} < \omega_{min}$, then we set $\hat{\omega} = \omega_{min}$ (optimality is shown in the Online Appendix). Last, ω in Equations (7) is replaced by $\hat{\omega}$ to obtain all date estimates $\hat{t_i}$.

This algorithm can be extended to non-binary trees. However, nothing guarantees that the date estimates satisfy the temporal precedence constraints. This is why we designed the QPD (quadratic programming dating) algorithm, which we describe now.

QPD Algorithm

QPD is based on an active-set method, which is commonly used to solve optimization problems with linear constraints (Nocedal and Wright 2006). Let $x = (\omega, t_1, \ldots, t_{n-1})$; the function to minimize is $\phi(x)$ defined by Equation (2), subject to the constraints $t_i - t_{a(i)} \geqslant 0$, for $i = 2, \ldots, 2n-1$. For the sake of simplicity, we do not include the $(\omega \geqslant \omega_{min} > 0)$ constraint, as it is already accounted for in the LD algorithm, which is part of QPD. x is a "feasible" point if and only if it satisfies all the constraints. A constraint i is "active" at x if and only if $t_i = t_{a(i)}$. The active-set method applied to our problem can be summarized as follows (see the Online Appendix for details): starting from a feasible point x with C being the set of active constraints, we compute the minimal solution of Equation (2) with respect to C, that is, the minimal solution such that $t_i = t_{a(i)}$, for every $i \in C$. We thus have to calculate the stationary point (x^*, λ^*) of the Lagrange function:

$$\Gamma(x, \lambda) = \phi(x) - \sum_{i \in C} \lambda_i \left(t_i - t_{a(i)} \right). \quad (8)$$

We then check if: (i) some constraints are violated in x^*, and (ii) all constraints in C are useful. C is updated accordingly, by relaxing the "most useless" constraint and adding the "most violated" one. The algorithm stops when all constraints in C are useful and no more constraints are violated (Karush–Kuhn–Tucker (KKT) conditions, Boyd and Vandenberghe 2004). With strictly convex quadratic functions, this method is ensured to converge to the unique global minimum (Nocedal and Wright 2006). Although Equation (2) does not comply with these requirements, a proof of QPD convergence to the unique minimum is provided in the Online Appendix.

The active-set method is especially efficient here, because we can find the stationary point of the Lagrange function (Eq. (8)) in linear time. Indeed, x^* is computed by a modified version of the LD algorithm, which applies to a new tree obtained from the input tree R by collapsing the branches corresponding to the active set C. Then, λ^* can also be calculated in linear time (Online Appendix).

The time complexity of QPD is $O(f \times n)$, where f is the number of iterations needed to reach the optimal solution, and n is the number of taxa. f depends on the data and the chosen starting point. We use here the LD algorithm, initializing C with the violated constraints $(t_i < t_{a(i)})$ in the LD solution, which are combined to obtain a feasible point. In our experiments (described below), QPD performs 3 iterations on average with simulated trees of 110 taxa, and 69 iterations with an H1N1 influenza data set of 891 taxa. Although, it is difficult to extrapolate from these experiments, it seems that in practice f is much smaller than n, and thus the computing time of QPD appears to be nearly linear.

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

Estimating the Root Position for Unrooted Trees

Given an unrooted tree, we estimate the root position by searching for the point in the tree that minimizes the objective function (Eq. (2)) when the tree is rooted at this point. A similar approach is used in RTT-based Path-O-Gen software (Rambaut 2007). In essence, this is the point that makes the tree the most molecular clock-like. Let R be an unrooted tree with the internal nodes enumerated from 2 to $n-1$, and the external nodes from n to $2n-1$. Let r be a point on a branch $[r_1, r_2]$ of length b. Let t_r be the date of r assuming r is the tree root, and μ a variable in $[0,1]$ such that the length of branch (r_1, r) is equal to μb; then, the objective function (Eq. (2)) of the tree rooted at r becomes:

$$\begin{aligned}\varphi(t_r, t_2 \ldots, t_{n-1}, \omega, \mu) &= \big(\mu b - \omega(t_{r_1} - t_r)\big)^2 + \big((1-\mu)b - \omega(t_{r_2} - t_r)\big)^2 \\ &\quad + \sum_{i \neq r_1, r_2} \big(b_i - \omega(t_i - t_{a(i)})\big)^2. \end{aligned} \quad (9)$$

Note that we do not use weights (variances) in the objective function, since weights depend on their associated branch lengths, which are unknown for the two branches containing the assumed root. Optimizing this function without and with constraints can be done by slightly modifying the LD and QPD algorithms, without changing their time complexities. The technical details are given in the Online Appendix. For each branch, we calculate the root position (μ) which minimizes Equation (9), and then take the minimum point among all branches. Therefore, the time complexity is n times that of the LD and QPD algorithms. Since LD is linear, the corresponding rooting algorithm is quadratic. For QPD, to avoid exploring all branches, which could be time consuming with large trees, we pre-estimate the position of the root using LD, and then we use QPD to perform a greedy search for the local minimum around that position. This rooting method is also applicable when all tips are contemporaneous, thus representing a new alternative to the standard rooting methods (midpoint, minimum-variance, etc.).

Results with Simulated Data

Data Simulation

We implemented a tree generator based on a simple birth–death model with periodic sampling times, mimicking typical intrahost studies with yearly sampling, or (interhost) epidemic surveillance through time. We first assumed a SMC, and then a lognormal relaxed molecular clock (RMC). Let us start with SMC. At time $t=0$, there is one single individual ($n=1$), which is iteratively subdivided. At each step, one of the n individuals is randomly selected and divided into two individuals, resulting in $n+1$ individuals. The elapsed time between the previous division event and the new one is equal to $1/n$ (i.e., like the standard Yule tree, where the expected time is equal to $1/n$). This process is continued until we have 1000 individuals. Then we proceed with sampling and death: the evolution of a number of individuals (e.g., 750) is stopped, most of them (e.g., 725) are removed from the tree (or "culled"), while the remaining ones (e.g., 25) are retained to be the sampled individuals of the first sampling time. The process continues with the nonculled and nonsampled individuals (250 in our example), which are further divided using the same Yule-type rule until we again have 1000 individuals to be sampled, culled, or conserved for the next step. The whole process is continued until we attain the desired number of sampling times. The final set of sampled individuals is exactly the taxon set (or leaves) of the final tree. This tree is then rescaled so that the time between the first and the last sampling time is 20 years, with the root date being zero. An advantage of this scheme is that the time elapsed from one sampling time to the next one is constant, thus emulating the sampling of DNA sequences from an evolving population on a regular basis, as opposed to standard birth–death tree generators (Stadler 2010). Moreover, with birth–death trees the divergence times vary among replicates, while here we use fixed divergence times for easy estimation of method accuracy and presentation of the results.

We generated two kinds of trees, intended to simulate interhost and intrahost HIV evolution (Volz et al. 2013), by using two death rates (ratio of individuals removed at each sampling time): 750/1000 was used for interhost trees, and 995/1000 for intrahost (typically ladderized) trees. For each, we used 3 sampling times (separated by 10 years) with 25 selected individuals at each time, and 11 sampling times (separated by 2 years) with 10 selected individuals at each time. See Figure 1 for examples of trees. Additionally, we added one outgroup to simulate the search for the root position using the standard outgroup-based approach. The length of the branch from the ingroup root to the outgroup was three times the length from the ingroup root to the nearest ingroup leaf. Last, to simulate sequence evolution, we used the substitution rate ω to obtain the length of each branch $b_i = \omega(t_i - t_{a(i)})$, corresponding to the expected number of substitutions per site along that branch; ω was equal to 0.006 substitutions per site and per year, which is similar to the substitution rate of the HIV *env* gene (Bello et al. 2008). With each combination of these parameters, 100 trees were randomly generated. Hence, there are in total 4×100 SMC trees, denoted as (death rate/sampling scheme): $750/3 \times 25$, $750/11 \times 10$, $995/3 \times 25$, and $995/11 \times 10$.

To simulate trees with RMC, we used the uncorrelated lognormal model, which is one of the most widely used in BEAST (Drummond et al. 2006). For this purpose, we reused the previous trees, but multiplied every branch length by a random variable following a lognormal distribution with mean 1 and standard deviation 0.4. This value is between the estimates we obtained for *pol* and *env* HIV genes (unpublished results). We thus obtained 4×100 RMC trees.

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

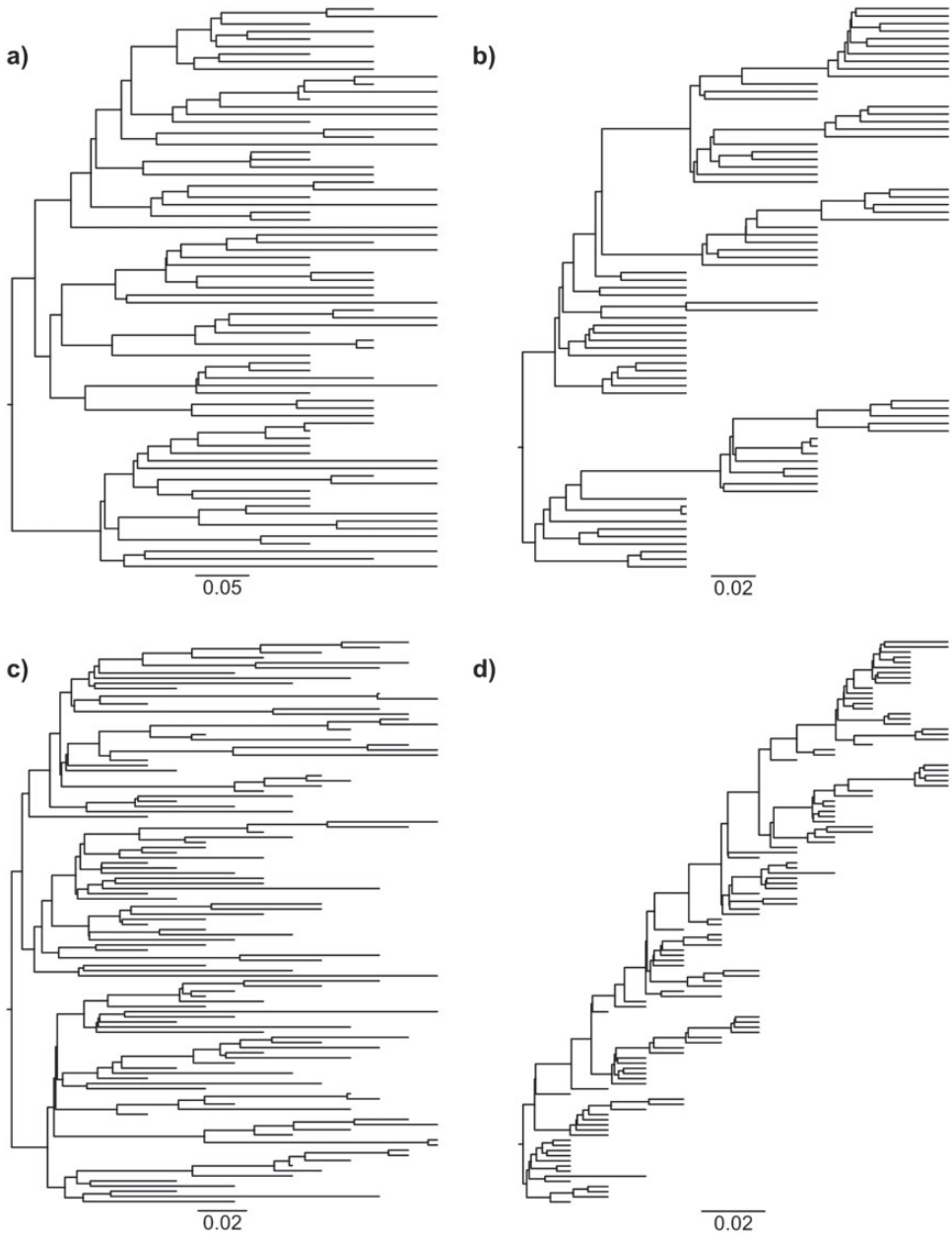


FIGURE 1. Examples of simulated trees. Four examples of trees extracted from our simulated data sets. Trees a) and c) are intended to simulate inter-host evolution of HIV (one tip per host; 750/1000 of the strains are removed at each sampling date). Trees b) and d) are intended to simulate intra-host evolution of HIV, with its typical "ladder shape" (all tips from a single host; 995/1000 of the strains are removed at each sampling date). Trees a) and b) have each 3 sampling dates with 25 sampled strains each. Trees c) and d) have each 11 sampling dates with 10 sampled strains each. See text and Volz et al. (2013) for explanations.

DNA sequences of length 1000 were evolved along these trees using Seq-Gen (Rambaut and Grassly 1997), version 1.3.2x. We used the F84 model with a Γ distribution with shape parameter 1.0 and 8 rate categories, a transition/transversion rate ratio of 2.5, and nucleotide frequencies of (A, C, G, T) = (0.35, 0.20, 0.20, 0.25). These parameter values are similar to estimates already observed with the *env* region of HIV (Posada and Crandall 2001).

To assess the accuracy of the distance-based dating methods, we inferred trees from these alignments. First, we used the correct tree topology but re-estimated the branch lengths using PhyML+F84+Γ8; the aim was to measure the impact of topological errors that are unavoidable in real studies; moreover, we used these trees to assess the performance of the various methods to estimate all tree node dates, instead of the root date only. Second, we used DNADIST+F84+Γ

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

(PHYLIP, Felsenstein 1989, v3.69) to estimate pairwise evolutionary distance matrices, and then FastME with SPR option to estimate trees (negative branch lengths were set to zero); the distance estimation parameters in DNADIST were the same as those used to generate the data with Seq-Gen; the aim was to check the accuracy of a fast distance approach, being easily bootstrapped or able to tackle very large data sets. Last, we used PhyML+F84+Γ8 with SPR option to infer both the tree topology and branch lengths, that is, a standard ML method with high accuracy, but slower than DNADIST+FastME. All these trees were used in two ways: (i) the outgroup was used to produce rooted trees, from which the outgroup was deleted; (ii) we simply removed the outgroup to obtain unrooted trees. All of our data sets (model trees, alignments, distance matrices, inferred trees, etc.) are available at http://www.atgc-montpellier.fr/LSD/.

Methods for Comparison

These simulated data were used to assess the performance of our two methods (LD and QPD) and of three other methods: RTT, Langley-Fitch (LF*), and BEAST (BSMC with a SMC model and BRMC with a relaxed clock):

- For LD and QPD, if the tree is rooted, the program uses the given tree; the methods are then denoted as LD* and QPD*, and we use the "variance" option (WLS). Otherwise, the root position is estimated and the methods are simply denoted as LD and QPD.

- For RTT, we re-implemented the linear regression method, which takes both rooted and unrooted trees as input. Given unrooted trees, it estimates the position of the root by minimizing the sum of squared residues. Given rooted trees, the method is a standard regression and is denoted as RTT*. For dozens of data sets, we checked that our implementation gives the same result as Path-O-Gen v1.3 (Rambaut 2007). Unlike other methods used here, RTT does not estimate the dates of internal nodes but only the root date and the substitution rate.

- For LF*, we used the program implemented in the *r8s* package v1.8 (Sanderson 2003); the likelihood function was optimized thanks to Powel's algorithm (TN algorithm was much faster, but returned inconsistent results with ∼20% of our data sets); this program has no ability to search for the root position and takes only rooted trees as input, hence the notation LF*.

- For BSMC and BRMC, BEAST version 1.7 was used with HKY+Γ8 (closely related to F84+Γ8 used to simulate the data) and coalescent with constant population size tree prior. We used a SMC with an uninformative prior (*clock rate* had a uniform distribution between 0 and 1). The length of the MCMC chain was 5×10^6 generations, with a burn-in of 10% and a sampling every 5×10^3 generation. For the relaxed-clock data, we also used a lognormal relaxed-clock model (i.e., the model used to generate the data); the prior of the *ucld.mean* parameter had a uniform distribution between 0 and 1, and the prior of *ucld.stdev* had an exponential distribution with parameter 1/3 (default value). The MCMC chain length was increased to 20×10^6 generations, with a burn-in of 10% and a sampling every 20×10^3 generations. These parameter values are standard and default options were used in all of our analyses. We increased the burn-in up to 25%, but did not observe significant changes. Additional runs with several alternative priors were also performed (uniform prior in a much more narrow interval [0, 0.05] for *clock.rate* and *ucld.mean* parameters; uniform prior on the inverse of these parameters; birth–death tree prior), but without improvement, and the same held with alternative program options (Drummond A., Yanez R., personal communication). Moreover, other runs of BEAST were carried out to assess the accuracy of internal node date estimations. We then used the true rooted tree topology (otherwise date comparisons are meaningless), and forced it to be constant in BEAST, so that only the branch lengths were re-estimated, just as with PhyML (see above). The length of the MCMC chain was set to 10×10^6 generations, with a burn-in of 10% and a sampling every 10×10^3 generations. In all of our analyses, we used *meanRate* estimator for rate estimations with BRMC, since it was more accurate than *ucld.mean*, and *clock.rate* with BSMC; *treeModel.rootHeight* was used to estimate the root date with both BSMC and BRMC. BEAST xml and log files with the 800 simulated data sets are available at http://www.atgc-montpellier.fr/LSD/.

Comparison Criteria

With simulated data, the true value of the parameters (substitution rate, root and node dates) are known. We used standard quadratic error measures to compare the true and estimated values and assess the accuracy of the methods being compared. An advantage of these measures is that they can be decomposed into variance and bias terms, thus indicating whether the estimation method shows some tendency to over- or underestimate the true parameter value, and whether the main source of errors is, or is not, the variance of the estimates.

- For the substitution rate, let ω be the true value, $\hat{\omega}_i$ the value estimated by a given method with the i^{th} data set among m (= 100 in our experiments), and

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

$\bar{\omega}$ the average of the m estimates. The accuracy of that method in estimating the substitution rate is measured by the relative error:

$$\frac{1}{\omega}\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\omega-\hat{\omega}_i\right)^2},$$

and the relative bias is defined by:

$$\frac{1}{\omega}\left(\bar{\omega}-\omega\right).$$

- Similar measures are used for the root date, with relative error defined by:

$$\frac{1}{t_c}\sqrt{\frac{1}{m}\sum_{i=1}^{m}\hat{t}_i^2},$$

and relative bias:

$$\frac{\bar{t}}{t_c},$$

where $\hat{t}_i$ is the estimated root date with the i^{th} data set, $\bar{t}$ is the average root date estimate, and t_c is the contemporary time (which is the same for all trees within each tree model); moreover, remember that the true root date is zero. These relative error terms can be interpreted as percentages; for example, a bias of -0.1 means that the true value is underestimated by 10%, in average. A basic result in estimation theory is that the square of the bias plus the variance of the estimates is equal to the mean square error. It follows that our relative bias is less than the relative error and that their difference corresponds to the relative, standard deviation of the estimates. We calculated the confidence intervals of these error measures using the bootstrap method; for each data set of 100 trees, we re-sampled 10,000 times with replacement the set of the 100 estimated values and computed the corresponding error; then, the 2.5% and 97.5% quantiles were picked up to form 95% confidence intervals.

- For the dates of internal nodes, we used the absolute error (measured in years and thus easily interpreted) defined by:

$$\sqrt{\frac{1}{m(n-1)}\sum_{i=1}^{m}\sum_{k=1}^{n-1}\left(\hat{t}_{ik}-t_{ik}\right)^2},$$

where $i=1,\ldots,m$ represents one of the m trees, and $k=1,\ldots,n-1$ is one of the internal nodes (including the root, where $k=1$), t_{ik} is the date of the node k in the tree i and $\hat{t}_{ik}$ is its estimated value. Again we used the bootstrap to build confidence intervals.

Results

The detailed results of all tested methods using above criteria are available from our web site http://www.atgc-montpellier.fr/LSD/ and in the Online Appendix.

Distance-based dating methods have negligible computing times with these data (~0.1 seconds or less, even with unrooted trees where the root position has to be searched among all edges), except LF*, which is still fast but requires a few seconds with rooted trees. In contrast, BEAST requires a few hours with a SMC, and a dozen hours with a relaxed clock. For a fair comparison, we also have to account for tree building, as BEAST infers both the tree and the dates. However, PhyML is much faster, requiring 8 min for the largest 110-taxon trees. The computing time difference between distance-based approaches and BEAST is thus very large (see Online Appendix Supplementary Table S1 for details), but does not correspond to gains in estimation accuracy, as discussed below.

With SMC data (Fig. 2a,c,e), the relative errors are low (~5%) and most methods have similar, high accuracy. RTT and RTT* are a bit less accurate than the others for both root date and rate estimations, most likely due to their overly simple model. BEAST is also behind the others regarding rate estimation, with a substantial positive bias (up to ~10% with 995/11x10 trees, Online Appendix Supplementary Table S2), but performs well with date estimation, both for the root (Fig. 2c) and all internal nodes (Fig. 2e). As a general tendency (Fig. 2a,c, e.g., LD vs. LD*, and QPD vs. QPD*), molecular clock-based rooting produces similar results to outgroup-based rooting for both the root date and the rate, as expected since trees were generated with SMC. Surprisingly, the accuracy of rate and root date estimations are not significantly affected by topological errors: although the FastME and PhyML trees contain a substantial amount of erroneous branches, we see very little difference in accuracy between the results obtained with the true and inferred topologies. Moreover, there is almost no difference between the results obtained with FastME (topological error ~15%, Online Appendix Supplementary Table S4) and PhyML (topological error ~10%). This suggests the use of (much faster) FastME rather than PhyML, when the aim is not to obtain a fully correct tree topology but to quickly estimate rates and dates, or to perform bootstrap analyses. The topological accuracy of BEAST and PhyML are quite similar (Supplementary Table S4), with BEAST providing a slight advantage, meaning that the high error of BEAST in rate estimation is not due to topological errors, but to the positive bias already indicated above. BEAST results with the fixed, true topology confirm this finding (Online Appendix Supplementary Fig. S5).

With RMC data (Fig. 2b,d,f), the relative errors of all methods are much higher (from ~10% to ~20%) than with SMC data (~5%). Again, the topological errors have little impact on the accuracy of rate and date estimations, and cannot explain the differences

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

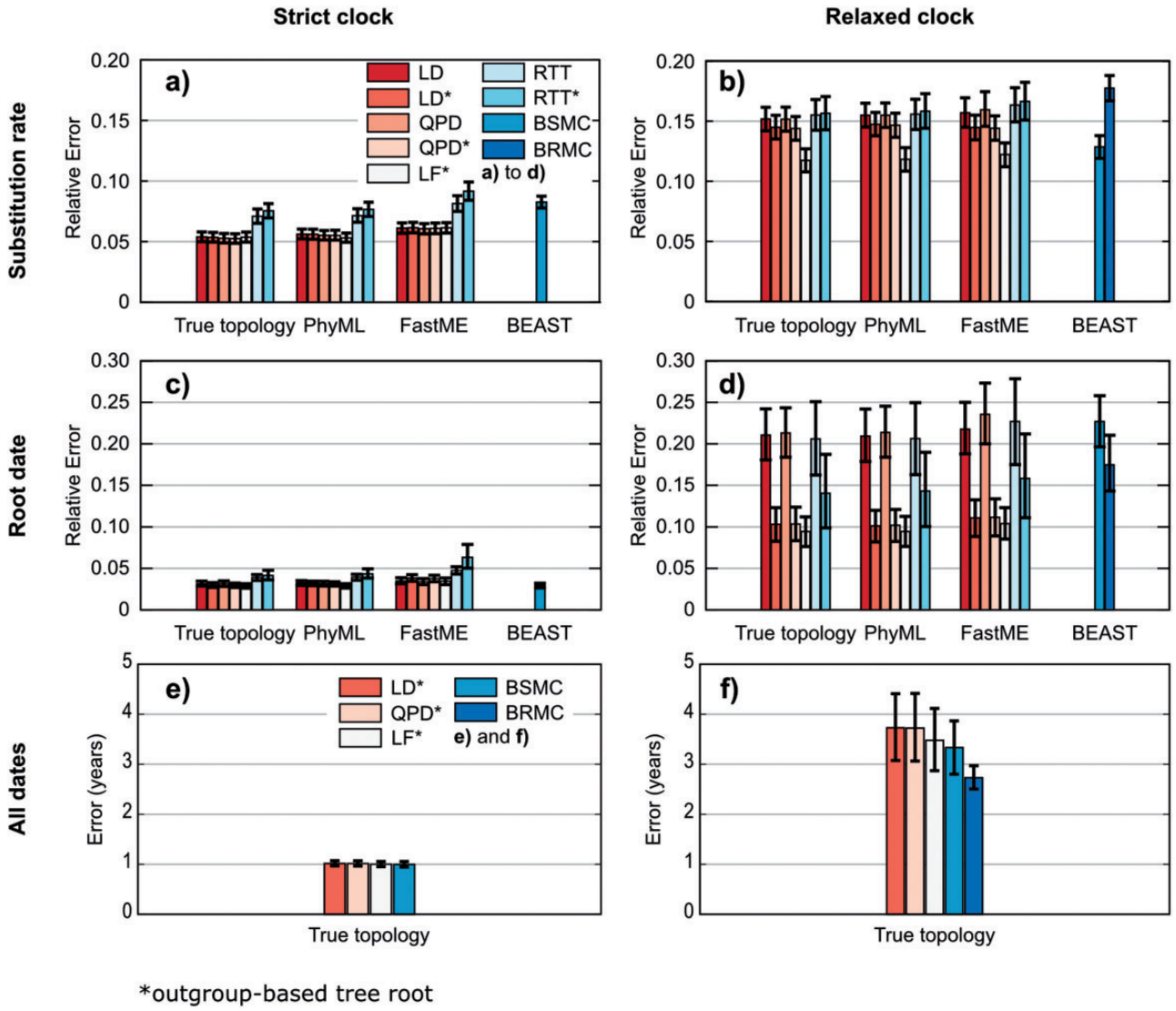


FIGURE 2. Summary results with simulated data. Panels a), c) and e) contain summary results of the trees with a SMC, panels b), d) and f) those with a lognormal, RMC. Panels a) and b) show the relative error of the substitution rate estimates, panels c) and d) show the relative error of the root date estimates, panels e) and f) show the average error (in years) of the data estimates of all tree nodes. See text for the definitions of these measures. From left to right (see legends) tested methods are: linear dating with tree root estimation (LD); linear dating with outgroup-based tree rooting (LD*); quadratic programing dating with tree root estimation (QPD); quadratic programing dating with outgroup-based tree rooting (QPD*); Langley-Fitch that uses rooted trees only (LF*); root-to-type regression with tree root estimation (RTT); root-to-type regression with outgroup-based tree rooting (RTT*); BEAST with strict molecular clock (BSMC); BEAST with lognormal, relaxed molecular clock (BRMC).

among the various methods, especially with BEAST the topological accuracy of which is still slightly better than PhyML's (Supplementary Table S4). Again, FastME and PhyML trees produce rate and date estimates showing similar accuracy. As expected the main factor is root positioning, which has a high impact on root date estimations. If the root is misplaced, the tree cannot be dated precisely. Among the methods directly inferring the root position (i.e., without outgroup), LD, QPD, and RTT show similar accuracy (poor regarding root date), while BEAST results differ depending on the clock model. With BSMC the rate is well estimated but the date is not any better than with direct distance-based approaches; with BRMC the rate is poorly estimated due to a high positive bias (> 10%), but the root date is fairly well estimated. Results with the fixed, true topology confirm these findings: BEAST rate estimations are not improved (Supplementary Fig. S5), but BEAST with the RMC model is the most accurate method to estimate internal node dates (Fig. 2f), which is to be expected since the data were generated using the very same model. Moreover, the global average results (Fig. 2) hide that BEAST does well with model trees with low death rate (750/1000, Online Appendix Supplementary Figs. S1–S2), but not so with high death rate (995/1000, Online Appendix Supplementary Figs. S3–S4). Among

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

the methods using outgroup-based rooted trees: LF* is best to estimate the rate and slightly (but not significantly) better than LD* and QPD* in estimating root and internal node dates; RTT* is worse both to estimate the rate and the root date.

Up until now, we mostly discussed average results over the four types of model trees (Fig. 2). As expected, the accuracy of the various methods differs depending on the model tree (Online Appendix Supplementary Figs. S1–S4). The accuracy of the estimates is better with the larger sample of 110 dated sequences, than with 75 sequences, and the impact is especially sensible with date estimations since we have 11 sampling times (every 2 years) instead of 3 (every 10 years). Moreover, the ladder-shaped trees (995/1000 death rate) are easier than the more star-like trees (750/1000), an outcome already observed with real data, for example from human seasonal influenza (ladder-shaped) versus HIV (star-like) epidemics (Grenfell et al. 2004). However, the global properties and the ranking of the various methods remain similar compared to average analysis (except with BEAST, see above).

Most results in these simulations were expected. Among distance-based methods, LF* has the most realistic (Poisson) model and obtains the best results; LD* and QPD* use a simplified (normal) version of the LF* model, and their results are not as good as those of LF*, although the difference is not significant in most cases; RTT* is the worst distance method, as expected since its model is too simple and does not account for the fact that the root-to-tip paths are highly correlated (Drummond et al. 2003b). The main surprise comes from the results of BEAST, expected to be the best due to its sophisticated model, being identical or very close to the data model, but in fact the results on the data sets used here do not suggest this. However, results in Figure 2 have to be interpreted with care: first, BEAST in our experiments does not use an outgroup as the best distance approaches do, and thus should be compared to the direct methods (LD, QPD, and RTT); second, BEAST shows a substantial bias with rate estimation that remains to be explained, but performs well (Fig. 2d) to very well (Fig. 2f) with dating.

Let us conclude these simulations with practical guidelines. Tree rooting is a difficult task; thus, if possible, use an outgroup and compare the results with the direct ones, obtained by assuming some (relaxed) clock model. When having a well-supported and consistent root position, use LF* or QPD*, selecting the latter with large data sets and bootstrap studies. ML trees are preferable to minimize topological errors, but (fast) distance-based trees provide nearly identical rate and date estimates. LD and QPD (resp. LD* and QPD*) have nearly identical accuracy in these simulations. However, LD and LD* violate a substantial number of temporal constraints (~4% by more than 1 month with 110-taxon trees and RMC), and the advantage of QPD and QPD* will become even more apparent with real (imperfect) influenza data.

Application to an Influenza Data Set

To illustrate the results of our algorithms on large data sets, we used a set comprising 1194 strains of influenza A virus subtype H1N1pdm09, which caused the first human influenza pandemic of the 21st century. The first two cases were reported in children from southern California on 21 April 2009. Soon after, other cases were reported, and by 11 June 2009, 27,000 cases of infection had been observed from 74 countries, including 141 deaths. On that date, the World Health Organization (WHO) declared a pandemic, and the end of the pandemic was declared in August 2010 (for details, see Christman et al. 2011).

Molecular epidemiology studies on this virus were performed at an early stage of the epidemic, using 242 strains collected between 30 March and 12 July 2009 (Lemey et al. 2009; Rambaut and Holmes 2009). These studies indicated that this virus has a high evolutionary rate of 4.96×10^{-3} [4.10×10^{-3}; 5.87×10^{-3}] substitutions per site and per year (for concatenated hemagglutinin (HA) and neuraminidase (NA) genes), and the estimated date for the tMRCA was 27 January 2009 [29 December 2008; 22 February 2009]. This MRCA date was confirmed by Hedge et al. (2013) using 328 whole virus genomes sampled in North America before April 2010. To our knowledge, no other molecular dating study has been published on a more comprehensive set of strains sampled over a longer time period.

The (1194) strains used here were collected worldwide between 13 March 2009 and 9 June 2011 (see Online Appendix Supplementary Table S5 for further details). The A/Swine/Hong Kong/1110/2006 (subtype H1N2) strain was used as outgroup to root the phylogenetic tree. The HA gene sequences were aligned by codon using MUSCLE in MEGA 5.0 and checked manually, resulting in an alignment of 1194 (+1 outgroup) sequences and 1701 sites. As many sequences were identical but collected at different time points, we retained for each set of identical sequences only one exemplar with a sampling date equal to the average of the dates of the corresponding strains. We thus obtained 891 (+1 outgroup) different sequences, each with a unique sampling date, from which a phylogenetic tree was computed. Note that grouping identical sequences does not impact phylogeny inference (identical sequences are separated by branches of length zero) but accelerates the computations and is consistent with our dating model which has difficulty in dealing with branches of length zero but different dates at both extremities (see Eqs. (1)–(4), and notably the variance term). However, this simplification was not used with BEAST, which handles such data due to its coalescent, population genetics model.

To run our dating algorithms, we first have to infer a phylogenetic tree. Two methods were used, as in our simulation study, with different speed/accuracy tradeoff: (i) a fast distance-based method, namely FastME with the SPR option and distances estimated by DNADIST under F84+Γ (the Γ parameter was set

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

to 1.0, as in other experiments); (ii) a more accurate but slower ML method, namely PhyML with SPR option and GTR+I+Γ4. For both methods we analyzed the ingroup sequences only and considered both the outgroup-based rooted tree, and the unrooted tree obtained by root removal. To compute confidence intervals we used the bootstrap method with 100 replicates generated with SEQBOOT from the PHYLIP package. To improve computational efficiency, the tree topology was kept constant and equal to the topology inferred using the original data set; only the branch lengths were re-estimated from the bootstrap samples.

We compared the same methods as in the previous sections, using the same options. LD and QPD were run with the "variance" (WLS) option both in the rooted and unrooted settings; they are denoted as LD* and QPD* with rooted trees. Langley-Fitch (LF*) from r8s was run with the rooted tree only, as it has no means to infer the tree root. RTT (our implementation, equivalent to Path-O-Gen v1.3, Rambaut 2007) was performed with both the unrooted (RTT) and the rooted (RTT*) trees. BEAST was run from the complete, ingroup alignment using GTR+I+Γ4, a coalescent constant population size tree prior and two molecular clock models: strict (SMC) and relaxed lognormal (RMC), with normal clock priors (mean $= 4 \times 10^{-3}$ substitutions per site per year (s/s/y), standard deviation $= 2 \times 10^{-3}$ s/s/y). Two independent MCMC chains were used per model, with a minimum of 250 million generations each, sampling every 10,000 generations. The first 25 million generations in each run were discarded as burn-in, and the Highest Posterior Density statistics for each parameter were calculated over a posterior sample of 1000 states using Tracer 1.5. Moreover, as we observed a strong discrepancy between BEAST and the other methods regarding substitution rate estimations (see below), we also launched BEAST with the cleaned data set where identical sequences were grouped (891 taxa), and using the PhyML rooted tree topology which was kept constant all along the computations, solely sampling the branch lengths and model parameters. Such use of BEAST seems to be rather uncommon, but corresponds to the way a number of other dating programs proceed, for example, PAML (Rannala and Yang 2007).

All methods except BEAST were run on our server (Intel(R) Xeon(R) X5650 @ 2.67GHz, single core, no parallelization), while BEAST was run on a Dell Precision T7500 workstation (Intel(R) Xeon(R) X5687 @ 3.6GHz CPUs, one core per model) using the Beagle library with the SSE, Double Precision, and Dynamic Rescaling options (Ayres et al. 2012). The computing times (Table 1) to obtain point estimates for the substitution rate and all node dates with distance-based approaches are very fast: at most 1 second with rooted trees for LD*, QPD*, and RTT*, and ∼1 min for LF* which is the slowest distance-based method; with unrooted trees, the methods inferring the tree root (LD, QPD, and RTT) are inevitably slower as they have to search all tree branches, but are still fast requiring less than 1 min. To obtain bootstrap intervals the computing times are multiplied by 100 as we have 100 replicates, varying from a few seconds (RTT* and LD*) to ∼1 h (LF*), with QPD* requiring ∼2 min. This shows the advantage brought by our algorithms, since both LF* and QPD* use closely related models and show similar accuracy (Fig. 2). However, the time to build trees has to be accounted for, especially when bootstraps are used. DNADIST+FastME is remarkably fast, requiring ∼1 h to infer the original and 100 bootstrap trees, while PhyML is much slower, requiring ∼4 d for the same task. To get a good posterior sample of time resolved Bayesian phylogenetic trees with the 1194 sequences requires running BEAST for a minimum of 20 d, using at least 250 million MCMC generations at approximately 2 h per million generations. For the 891 cleaned sequence set, using a fixed rooted, PhyML topology in BEAST, only 50 million MCMC generations are needed, taking 5 d at approximately 2.5 h per million generations.

TABLE 1. Computing time for the H1N1pdm09 Flu data set

	Original sample	100 bootstrap samples
Phylogeny inference		
Distance-based (DNADIST+FastME)	127	3434
ML (PhyML)	∼60 h	∼35 h
Dates and rate estimation		
LD	31	2791
LD*	<1	15
QPD	38	3329
QPD*	1	120
Root-to-tip	12	2765
Root-to-tip*	<1	2
Langley–Fitch*	54	4177
BEAST (BSMC and BRMC)	∼20 d	–
BEAST* (BSMC* and BRMC*)	∼5 d	–

Note: Time is expressed in seconds, except otherwise specified. With bootstrap samples, only the branch lengths were reoptimized; the tree topology was kept constant and equal to the topology inferred using the original alignment. BEAST was run to infer all model parameters, including the tree topology and tree root, while with BEAST* we used the PhyML rooted tree topology which was kept constant along the computations. The asterrisk (*) denotes methods using out-group based rooted trees.

We see little difference (Fig. 3) between the results obtained with FastME and PhyML trees, especially for the tMRCA where point and interval estimates are nearly the same for every distance-based estimation method (except root-to-tip). This strongly suggests using FastME when the focus is on rates and dates, at least for large data sets, as it is several orders of magnitude faster than PhyML. Moreover, both tree building and dating are then consistently based on similar distance-based approaches.

Regarding rate estimation (Fig. 3a), all distance-based methods provide similar results, except root-to-tip regression with faster rate estimates and much larger confidence intervals. QPD also shows relatively large intervals, likely due to the fact that it has

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

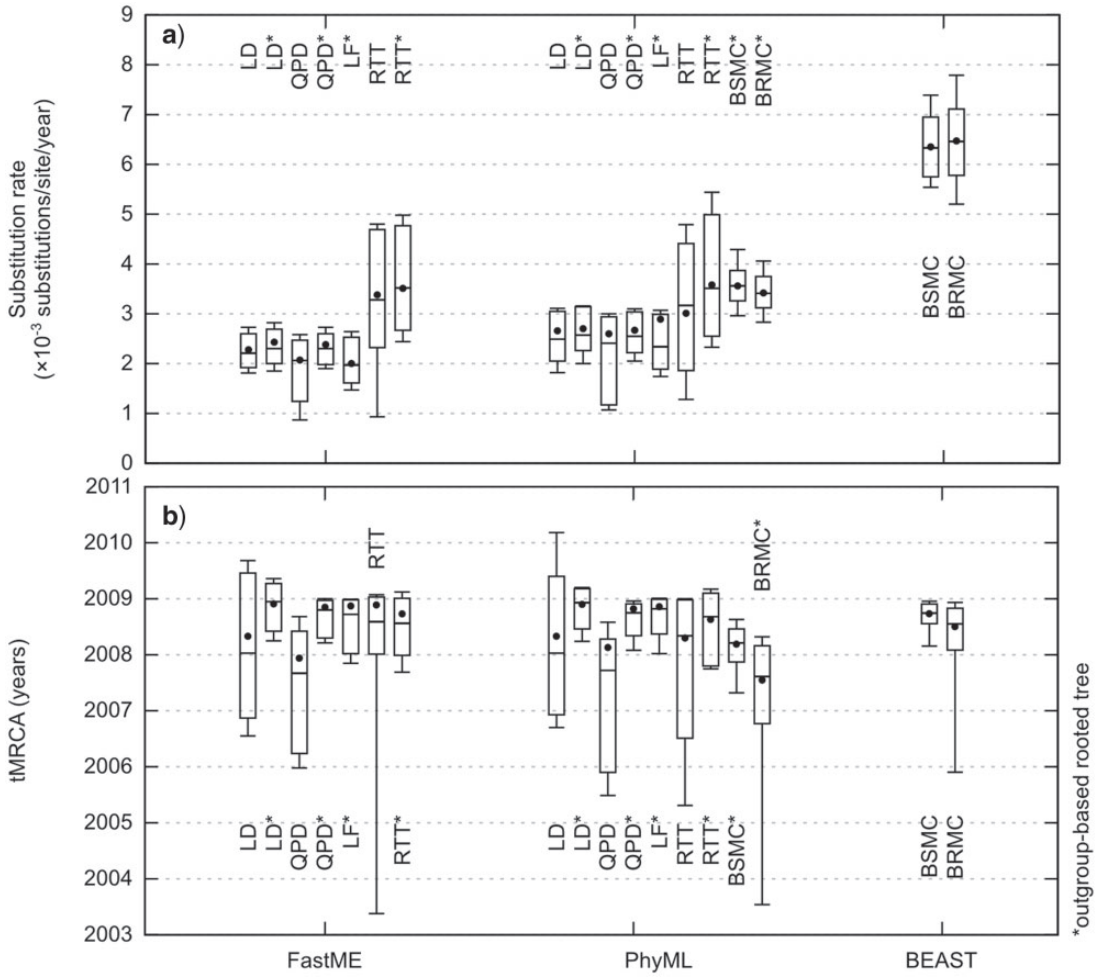


FIGURE 3. Estimations of the substitution rate (panel (a)) and tMRCA (panel (b)) with the H1N1pdm09 Flu data set. Distance-based estimation methods are run with both DNADIST+FastME and PhyML trees. LD, QPD, and RTT (root-to-tip) are run from unrooted trees and search for the best tree root position. LD*, QPD*, LF*, and RTT* are run using outgroup-based rooted trees. BEAST is run from the ingroup sequence alignment, with both a strict molecular clock (BSMC) and a lognormal relaxed clock (BRMC); BSMC and BRMC infer the tree topology and root position, while BSMC* and BRMC* use the fixed, rooted tree topology inferred by PhyML. The box plots represent the median, maximum, minimum, 97.5% and 2.5% quantiles of the bootstrap estimates with distance-based methods, and of the posterior distribution with BEAST. The distance-based point estimates and BEAST posterior means are represented by a dot.

to infer the tree root and is thus subject to more variability and possible rooting errors. However, the main fact here is that distance-based and BEAST rate estimates (obtained from the complete data set, while optimizing the tree topology) widely differ ($\sim 3.0 \times 10^{-3}$ and $\sim 6.5 \times 10^{-3}$, respectively, with non-overlapping confidence and credibility intervals). With simulated data we found that BEAST with the specified priors and options may overestimate the substitution rate (Online Appendix Supplementary Tables S2 and S3). We also observed similar discrepancies between both approaches on other biological data sets (results not shown). However, the gap here was so large that we ran BEAST with the cleaned data set and the fixed PhyML rooted tree topology that was used with other approaches. Then, BEAST rate estimates (BSMC* and BRMC* in Fig. 3a) became much closer to the others, being still somewhat faster ($\sim 3.5 \times 10^{-3}$ instead of $\sim 3.0 \times 10^{-3}$) but with mostly overlapping intervals. BEAST (combined with TREEANNOTATOR) infers a tree where ~2% of the temporal precedence constraints are violated with the complete data set, while with the cleaned data set and the fixed PhyML rooted tree, all constraints are satisfied. The reasons for these findings are still unclear. One explanation could be that with such a large data set (> 1000 sequences) BEAST has difficulty in converging on a reasonable rooted tree topology, notably because it does not use any outgroup to root the tree. Such calculations in a Bayesian setting could simply be too heavy, thus supporting the use of simpler PAML-like approaches for estimating dates and rates from fixed rooted tree topologies.

Paradoxically, no such gap is observed for the date of the MRCA (Fig. 3b): the best distance-based methods, namely QPD* and LF*, find nearly the same point estimates and confidence intervals as BEAST used in

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

a standard free-topology way, at least with a strict molecular clock (BSMC), that is, end of 2008. This date is compatible with Rambaut and Holmes (2009), Lemey et al. (2009) and Hedges et al. (2013) studies, but slightly older, as expected due to our larger data set incorporating more ancient strains. When using a fixed rooted tree topology, BEAST tMRCA becomes clearly older by 1 year or so, especially with a relaxed clock (BRMC*). However the discrepancy with distance methods involves only the MRCA and a few basal nodes (2 nodes with difference >6 months between QPD* and BSMC*, and 22 with BRMC*), while for most of the nodes the dates are highly similar (Pearson correlation coefficient with all node dates: QPD*/BSMC* ≈ 0.95, QPD*/BRMC* ≈ 0.91). The main difference among distance-based methods is between those using an outgroup to root the tree (LD*, QPD*, LF*, and RTT*) and the others (LD, QPD, and RTT) which infer the root position from the ingroup sequences only. The latter show more variability, larger confidence intervals, and tend to produce older date estimates, around the beginning of 2008 (these intervals and dates, however, are still statistically compatible with those of other methods). Again this larger variability is likely explained by the difficulty of tree rooting. Another factor for LD, and to some extent LD*, is the absence of temporal constraints: we see that their confidence intervals include a few root date estimates that are more recent (mid-2009) than our earliest strains (13 March 2009). This is clearly impossible and shows the advantage of incorporating temporal constraints, as in QPD and QPD*. With this data set, the solution of LD has $\sim$7% of branches such that the descendant node is older by 1 month or more than its parent ($\sim$1.5% when the time difference is larger than 2 months, and $\sim$0.5% with 3 months).

To summarize, while the best distance-based methods (QPD* and LF*, used with FastME) are considerably faster than BEAST (especially QPD*, with negligible computing times), their dating results are quite similar. Regarding substitution rate estimation, we observe a large discrepancy between distance-based methods and BEAST, when used in the usual way estimating all parameters, including the tree topology and its root. However, with the fixed rooted tree topology, BEAST estimates of the substitution rate become similar to those of distance-based approaches.

Discussion and Conclusion

We have described very fast algorithms to estimate rates and dates from serial data. These algorithms are based on a Gaussian noise, least-squares model, simplifying the Langley and Fitch's (1974) Poisson model implemented in the r8s package (Sanderson 2003). We showed that this model should be robust to uncorrelated violations of the molecular clock, and our simulation results confirm this theoretical prediction. LD uses a pure linear algebra approach, while QPD accounts for temporal precedence constraints, which appears to be important with real data. Given an input tree with dated tips, our algorithms provide the user with estimates of the substitution rate, the root date and the dates of all internal tree nodes, a task that is not achieved by RTT (also based on a simple, least-squares approach, but not able to date internal nodes). Our algorithms can be used to root the input tree when no outgroup is available, a feature that is not available in the r8s implementation of LF, and would be time consuming in the Poisson setting. Consequently, LD and QPD are also new fast, practical methods for tree rooting, which represent an alternative to the standard midpoint and minimum-variance approaches.

Computer simulations show that the accuracy of our algorithms is better than RTT's, and just slightly behind LF's with rooted trees. Compared to BEAST, our algorithms (combined with standard tree building methods) have a similar or better accuracy in estimating the substitution rate, while regarding dates the results depend very much on the presence of an outgroup and the way BEAST is used, estimating all parameters including the tree topology and its root, or using a fixed rooted tree topology. Globally, we did not observe any obvious limitation of our algorithms compared to BEAST, with simulated as well as real data sets. Moreover, our results clearly show the importance of having an accurate root position, a difficult goal when no outgroup is available and with relaxed (realistic) molecular clock.

Our algorithms require (quasi)linear computing times with rooted trees, as a function of n, the number of leaves. With unrooted trees, the computing time is (nearly) quadratic in n. This is obtained with complex algorithms, exploiting the closeness between least-squares and linear algebra; we also exploit the tree structure which makes it possible to design fast recursive procedures. This speed is important for current applications of phylogenetics. In Mourad et al. (2015), we analyzed a tree containing $\sim$24,000 dated HIV strains; running QPD* required $\sim$30 min on a standard desktop, while LF from r8s did not return any result after 2 weeks of computation. LSD has also been used by the members of the PANGEA_HIV consortium to study the phylodynamics of HIV epidemics in Africa using very large data sets (Fraser C., Ratmann O., personal communication; http://www.pangea-hiv.org/Projects/#phylodynamic, last accessed October 2015).

Our approach could be developed in several directions. First, we currently use a bootstrap approach to obtain confidence intervals, which is possible due to the speed of the algorithms, but still slow. Much faster approaches could be designed, for example, using the second derivative of the log-likelihood (least-squares) function. Second, we have described here the application of these algorithms to serial phylogenies with dated tips; easy adaptations should make it possible to use the very same approach to deal with phylogenies with time calibration points, attaching dated tips to ancestral nodes and using intervals (constraints) to account for ancestral date uncertainty. Last, an important direction is to

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

implement fast methods that are able to cope with more complex, correlated molecular clock models, typically combining the least-squares framework with penalized criteria, similar to Sanderson (2002), or using some of our algorithmic solutions to deal with multi-normal approximations of the likelihood function (Thorne et al. 1998).

Supplementary Material

Data available from the Dryad Digital Repository: http://dx.doi.org/10.5061/dryad.968t3, and from http://www.atgc-montpellier.fr/LSD/.

Funding

This work was supported by ANR project PhyloSpace, Labex NUMEV, VIROGENESIS EU-H2020 project (grant number 634650; O. G.), Wellcome Trust (grant number 092807; S. J. L.).

Acknowledgements

Thanks to Tanja Stadler, Jessica Hedge, Gabriel Leventhal, Raphaël Mourad, Alexei Drummond, and Romina Yanez for help with BEAST and testing LSD on large data sets, and to David Bryant, Mike Sanderson, Lars Jermin, and Frank Anderson for their comments that greatly helped us to improve the first version of this paper.

References

Akerborg O., Sennblad B., Lagergren J. 2008. Birth-death prior on phylogeny and speed dating. BMC Evol. Biol. 8:77.

Akerborg O., Sennblad B., Arvestad L., Lagergren J. 2009. Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. PNAS 106(14):5714–5719.

Ayres D.L., Darling A., Zwickl D.J., Beerli P., Holder M.T., Lewis P.O., Huelsenbeck J.P., Ronquist F., Swofford D.L., Cummings M.P., Rambaut A., Suchard M.A. 2012. BEAGLE: an application programming interface and high-performance computing library for statistical phylogenetics. Syst. Biol. 61(1):170–173.

Battistuzzi F.U., Filipski.A, Hedges S.B., Kumar S. 2010. Performance of relaxed-clock methods in estimating evolutionary divergence times and their credibility intervals. Mol. Biol. Evol. 27(6):1289–1300.

Bello G., Passaes C.P., Guimarães M.L., Lorete R.S., Matos Almeida S.E., Medeiros R.M., Alencastro P.R., Morgado M.G. 2008. Origin and evolutionary history of HIV-1 subtype C in Brazil. AIDS 22: 1993–2000.

Bertsekas D.P. 1999. Nonlinear programming. Cambridge, MA: Athena Scientific,

Boyd S., Vandenberghe L. 2004. Convex optimization, Chapter 5:244. Cambridge: Cambridge University Press.

Britton T., Anderson C.L., Jacquet D., Lundqvist S., Bremer K. 2007. Estimating divergence times in large phylogenetic trees. Syst. Biol. 56(5):741–752.

Christman M.C., Kedwaii A., Xu J., Donis R.O., Lu G. 2011. Pandemic (H1N1) 2009 virus revisited: an evolutionary retrospective. Infect. Genet. Evol. 11:803–811.

Desper R., Gascuel O. 2002. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. J. Comput. Biol. 19(5):687–705.

Douzery E.J., Delsuc F., Stanhope M.J., Huchon D. 2003. Local molecular clocks in three nuclear genes: divergence times for rodents and other mammals and incompatibility among fossil calibrations. J. Mol. Evol. 57(1 Suppl):S201–S213.

Doyon J.P., Ranwez V., Daubin V., Berry V. 2011. Models, algorithms and programs for phylogeny reconciliation. Brief. Bioinform. 12(5): 392–400.

Drummond A., Rodrigo A. 2000. Reconstruction genealogies of serial samples under the assumption of a molecular clock using serial-sample UPGMA. Mol. Biol. Evol., 17(12):1807–1815.

Drummond A.J., Pybus O.G., Rambaut A., Forsberg R., Rodrigo A.G. 2003a. Measurably evolving populations. Trends Ecol. Evol. 18: 481–488.

Drummond A., Pybus O.G., Rambaut A. 2003b. Inference of viral evolutionary rates from molecular sequences. Adv. Parasitol. 54: 331–358.

Drummond A.J., Ho S.Y.W., Phillips M.J., Rambaut A. 2006. Relaxed phylogenetics and dating with confidence. PLoS Biol. 4:e88.

Drummond A.J., Rambaut A. 2007. BEAST: Bayesian evolutionary analysis by sampling trees. BMC Evol. Biol. 7:214.

Drummond A.J., Suchard M.A., Xie D., Rambaut A. 2012. Bayesian phylogenetics with BEAUti and the BEAST 1.7. Mol. Biol. Evol. 29(8):1969–1973.

Felsenstein, J. 1989. PHYLIP - phylogeny inference package (Version 3.2). Cladistics 5:164–166.

Fitch W.M., Margoliash E. 1967. Construction of phylogenetic trees. Science 155:279–284.

Gascuel O. 1997. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. Mol. Biol. Evol. 14(7): 685–695.

Grenfell B.T., Pybus O.G., Gog J.R., Wood J.L., Daly J.M., Mumford J.A., Holmes E.C. 2004. Unifying the epidemiological and evolutionary dynamics of pathogens. Science 303:327–332.

Guindon S. 2010. Bayesian estimation of divergence times from large sequence alignments. Mol. Biol. Evol. 27:1768–1781.

Guindon S., Gascuel O. 2003. A simple, fast and accurate algorithm to infer large phylogenies by maximum-likelihood. Syst. Biol. 52(5):696–704.

Guindon S., Dufayard J.-F., Lefort V., Anisimova M., Hordijk W., Gascuel O. 2010. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. Syst. Biol. 59:307–321.

Hedge J., Lycett S.J., Rambaut A. 2013. Real-time characterization of the molecular epidemiology of an influenza pandemic. Biol. Lett. 9:5 20130331.

Hedges S.B., Kumar S., editors. 2009. The timetree of life. New York: Oxford University Press.

Jetz W., Thomas G.H., Joy J.B., Redding D.W., Hartmann K., Mooers A.O. 2014. Global distribution and conservation of evolutionary distinctness in birds. Curr. Biol. 24(9):919–930.

Lambert D.M., Ritchie P.A., Millar C.D., Holland B., Drummond A.J., Baroni C. 2002. Rates of evolution in ancient DNA from Adelie penguins. Science 295:2270–2273.

Langley C.H., Fitch W. 1974. An examination of the constancy of the rate of molecular evolution. J. Mol. Evol. 3:161–177.

Lefort V., Desper R., Gascuel O. 2015 (forthcoming). FastME 2.0: a comprehensive, accurate, and fast distance-based phylogeny inference program. Mol. Biol. Evol. 32:2798–2800.

Lemey P., Suchard M., 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 Curr. 1:RRN1031.

Lepage T., Bryant D., Philippe H. Lartillot N. 2007. A general comparison of relaxed molecular clock models. Mol. Biol. Evol. 24:2669–2680.

Mourad R., Chevennet F., Dunn D.T. Fearnhill E., Delpech V., Asboe D., Gascuel O., Hue S. 2015. A phylotype-based analysis highlights the role of drug-naive HIV-positive individuals in the transmission of antiretroviral resistance in the UK. AIDS 500:15-00106.

Murty K.G. 1988. Linear complementarity, linear and nonlinear programming. Sigma Series in Applied Mathematics, Vol. 3. Berlin: Heldermann Verlag. p. 447–448

Nocedal J., Wright S.J. 2006. Numerical optimization. 2nd ed., Chapter 16. Springer-Verlag, Berlin, New York. p. 455–468

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

O'Meara, O.T. 2000. Introduction to quadratic forms. Berlin and New York: Springer-Verlag.
Posada D., Crandall K.A. 2001. Evaluation of methods for detecting recombination from DNA sequences: Computer simulations. PNAS 98(24):13757–13762.
Rambaut A. 2000. Estimating the rate of molecular evolution: incorporating non-contemporaneous sequences into maximum likelihood phylogenies. Bioinformatics 16(4):395–399.
Rambaut A. 2007. Path-O-Gen web server: http://tree.bio.ed.ac.uk/software/pathogen/
Rambaut A., Grassly N.C. 1997. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. 13:235–238.
Rambaut A., Holmes E. 2009. The early molecular epidemiology of the swine-origin A/H1N1 human influenza pandemic. PLoS Curr. 1:RRN1003.
Rannala B.,Yang Z. 2007. Inferring speciation times under an episodic molecular clock. Syst. Biol. 56(3):453–466.
Rasmussen M.D., Kellis M. 2012. Unified modeling of gene duplication, loss, and coalescence using a locus tree. Genome Res. 22(4):755–765.
Ronquist F., Klopfstein S., Vilhelmsen L., Schulmeister S., Murray D.L., Rasnitsyn A.P. 2012. A total-evidence approach to dating with fossils, applied to the early radiation of the hymenoptera. Syst. Biol. 61(6):973–999.
Saitou N., Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4(4):406–425.
Sanderson M.J. 1997. A nonparametric approach to estimating divergence times in the absence of rate constancy. Mol. Biol. Evol. 14:1218–1231.
Sanderson M.J. 2002. Estimating absolute rates of molecular evolution and divergence times: a penalized likelihood approach. Mol. Biol. Evol. 19:101–109.
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.
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., Mullins J.I. 1999. Consistent viral evolutionary changes associated with the progression of human immunodeficiency virus type 1 infection. J. Virol. 73:10489–10502.
Smith S. A., O'Meara B.C. 2012. treePL: divergence time estimation using penalized likelihood for large phylogenies. Bioinformatics 28:2689–2690.
Sokal R., Michener C. 1958. A statistical method for evaluating systematic relationships. Kans. Univ. Sci. Bull. 38:1409–1438.
Stadler T. 2010. Sampling-through-time in birth-death trees. J. Theor. Biol. 267:396–404.
Tarjan R. E. 1983. Data structures and network algorithms. Philadelphia: SIAM.
Tamura K., Battistuzzi F.U., Billing-Ross P., Murillo O., Filipski A., Kumar S. 2012. Estimating divergence times in large molecular phylogenies. PNAS 109:19333–19338.
Thorne J.L., Kishino H., Painter S. 1998. Estimating the rate of evolution of the rate of molecular evolution. Mol. Biol. Evol. 15(12): 1647–1657.
Thorne J.L., Kishino H. 2002. Divergence time and evolutionary rate estimation with multilocus data. Syst. Biol. 51(5):689–702.
Volz E.M., Koelle K., Bedford T. 2013. Viral phylodynamics. PLoS Comput. Biol. 9:e1002947.
Xia X., Yang Q. 2011. A distance-based least-square method for dating speciation events. Mol. Phylogenet. Evol. 59:342–353.
Yang Z., O'Brien J.D., Zheng X., Zhu H.Q., She Z.S. 2007. Tree and rate estimation by local evaluation of heterochronous nucleotide data. Bioinformatics 23(2):169–176.

Downloaded from http://sysbio.oxfordjournals.org/ at IRMA Strasbourg on November 13, 2015

ONLINE APPENDIX

Fast dating using least-squares criteria and algorithms

Thu Hien TO[1], Matthieu JUNG[1,3], Samantha LYCETT[2] and Olivier GASCUEL[1,*]

[1]Institut de Biologie Computationnelle, LIRMM, UMR 5506 CNRS – Université Montpellier 2, Montpellier, France
[2]University of Edinburgh, Institute of Evolutionary Biology, Ashworth Laboratories, Edinburgh, United Kingdom
[3]Current address: IGBMC (Institut de Génétique et de Biologie Moléculaire et Cellulaire), INSERM, U596, CNRS, UMR7104, Université de Strasbourg, Illkirch, France

*Corresponding author: gascuel@lirmm.fr

- Unicity of the minimum in both constrained and unconstrained settings p. 2
- LD algorithm p. 3
- QPD algorithm p. 5
- Lagrange linear-time algorithm to find the stationary point of Equation (8) p. 7
- Convergence of the QPD algorithm p. 11
- Root estimation p. 13
- Supp. Figure S1. Summary results with the 750/3×25 simulated data set p. 15
- Supp. Figure S2. Summary results with the 750/11×10 simulated data set p. 16
- Supp. Figure S3. Summary results with the 995/3×25 simulated data set p. 17
- Supp. Figure S4. Summary results with the 995/11×10 simulated data set p. 18
- Supp. Figure S5. Error in rate estimation when using the true phylogeny p. 19
- Supp. Table S1: Computing times with simulated data p. 20
- Supp. Table S2: Simulation results with a strict molecular clock p. 21
- Supp. Table S3: Simulation results with a lognormal, relaxed molecular clock p. 22
- Supp. Table S4: Topological errors with simulated data sets p. 23
- Supp. Table S5. Origins and dates of influenza A H1N1pdm09 strains p. 24

ϕ (EQ. 2) HAS A UNIQUE MINIMUM IN BOTH CONSTRAINED AND UNCONSTRAINED SETTINGS

Ψ in Equation (2b) is a quadratic form (O'Meara 2000) and there exists a matrix Q, a vector v and a positive constant k such that $\Psi(x) = \frac{1}{2}x^T Q x + v^T x + k$, where $x = (\omega, \beta_1, \dots, \beta_{n-1})$. The quadratic term

$$\frac{1}{2}x^T Q x = \sum_{i=2}^{n-1} w_i \left(\beta_i - \beta_{a(i)}\right)^2 + \sum_{i=n}^{2n-1} w_i \left(\omega t_i - \beta_{a(i)}\right)^2$$

is always positive for any x. So Q is positive definite, and thus Ψ is strictly convex and has a unique global minimum. It is easily seen that the ranges of ϕ (in Eq. 2) and Ψ are the same, and each vector $(\omega, t_1, \dots, t_{n-1})$ corresponds to one and only one vector $(\omega, \beta_1, \dots, \beta_{n-1})$. So, ϕ also has a unique global minimum.

The function Ψ remains strictly convex on the restricted domain of constraints. Indeed, by definition of strict convexity: $\big(\Psi(X) + \Psi(X')\big)/2 > \Psi((X + X')/2)$ for every two vectors $X = (\omega, \beta_1, \dots, \beta_{n-1})$ and $X' = (\omega', \beta'_1, \dots, \beta'_{n-1})$; moreover, if X and X' satisfy the constraints, then $(X + X')/2$ also satisfies the constraints. So Ψ is strictly convex and has a unique minimum on the domain of constraints. Therefore, ϕ also has a unique global minimum on the domain of constraints.

LD ALGORITHM

LD Algorithm

Input: a rooted tree with: branch lengths $b_2, \ldots, b_{2n-1}$; the dates of the leaves $t_n, \ldots, t_{2n-1}$; and ω_{min}

Output: $\widehat{\omega} \geq \omega_{min}$ and $\hat{t}_1, \ldots, \hat{t}_{n-1}$ that minimize Eq. (2)

Calculate the system of Equations (5)

$pos \leftarrow$ post-order of the internal nodes

For i from $pos[1]$ to $pos[n-1]$ and $i \neq 1$

> **If** $s_1(i)$ and $s_2(i)$ are leaves **then** the dates $t_{s_1(t)}$ and $t_{s_2(t)}$ are known and Eq. $(5.i)$ is turned into:
>
> $$t_i = x_i t_{a(i)} + y_i + \frac{z_i}{\omega} \quad (6.\text{i})$$
>
> **Else if** $s_1(i)$ (resp. $s_2(i)$) is not a leaf, it has been already processed and we replace Eq. $(6.s_1(i))$ (resp. Eq. $(6.s_2(i))$ into Eq. $(5.i)$ to obtain Eq. (6.i)

$pre \leftarrow$ pre-order of the internal nodes

For i from $pre[1]$ to $pre[n-1]$

> **If** $i = 1$ **then** replace Eq. $(6.s_1(1))$ and Eq. $(6.s_2(1))$ into Eq. (5.1) to obtain $t_1 = u_1 + \frac{v_1}{\omega}$ (Eq. (7.1))
>
> **Else** replace $t_i = u_i + \frac{v_i}{\omega}$ (7.i) into Eq. $(6.s_1(i))$ and into Eq. $(6.s_2(i))$, to obtain Eq. $(7.s_1(i))$ and Eq. $(7.s_2(i))$

Replace $n-1$ equations Eq. (7.i) into Eq. (2) to compute $\widehat{\omega}$ that minimizes Eq. (2).

If $\widehat{\omega} < \omega_{min}$ then $\widehat{\omega} = \omega_{min}$.

Replace $\widehat{\omega}$ into Eq. (7.i) to compute $\hat{t}_1, \ldots, \hat{t}_{n-1}$

This algorithm takes as input a binary rooted tree with branch lengths and the sampling date associated to each tip. The outputs are the substitution rate and the dates of all internal nodes (including the root) that minimize Equation (2). All calculations are performed in linear time. LD

starts by calculating the system of Equations (5), and then resolves this system with recursive replacements using post-order and pre-order traversals (Tarjan 1983) of the input, rooted tree R.

By using a post-order (from the leaves up to the root) in the first loop, in each iteration i, if s is a descendant of i, then either s is a leaf or the iteration s has already been achieved. If s is a leaf, then t_s is known. Otherwise, x_s, y_s and z_s of Equation (6.s) have been computed in iteration s. In both cases, by replacing the value of t_s or the Equation (6.s) into Equation (5.i), we deduce x_i, y_i and z_i of Equation (6.i).

Next, by using a pre-order (from the root down to the leaves) in the second loop, in each iteration i, either i is the root or the iteration of $a(i)$, the ancestor of i, has already been achieved. If $i = 1$ (the root), by replacing Equations $(6.s_1(1))$ and $(6.s_2(1))$ into Equation (5.1), we infer two constants u_1 and v_1 such that $t_1 = u_1 + v_1/\omega$, as required by Equation (7.1). Otherwise, $u_{a(i)}$ and $v_{a(i)}$ have been computed in $a(i)$ iteration, so by replacing Equation $(7.a(i))$ into Equation $(6.i)$, we deduce u_i and v_i of Equation $(7.i)$.

Afterwards, the $(n-1)$ Equations (7) for every i are replaced into Equation (2) to obtain $\widehat{\omega}$. If $\widehat{\omega} < \omega_{min}$, then ω_{min} is the optimal rate under $(\widehat{\omega} \geq \omega_{min} > 0)$ constraint. This property follows from the strict convexity of the equation obtained from Equation (2) after replacement of the $(n-1)$ Equations (7). We thus assign $\widehat{\omega} = \omega_{min}$ in this case.

Lastly, $\widehat{\omega}$ is used to obtain all time estimates $\widehat{t_i}$ using Equations (7).

QPD ALGORITHM

QPD Algorithm

Input: a rooted tree with: branch lengths $b_2, \dots, b_{2n-1}$; the dates of the leaves $t_n, \dots, t_{2n-1}$; and ω_{min}

Output: $\hat{\omega} \geq \omega_{min}$ and $\hat{t}_1, \dots, \hat{t}_{n-1}$ that minimizes ϕ (Eq. 2) and satisfy the temporal constraints

$x = (\omega, t_1, \dots, t_{n-1}) \leftarrow$ the solution of LD; $C = \emptyset$;

$pos \leftarrow$ a post-order of internal nodes of the input tree;

For $i = pos[1]$ to $pos[n-1]$

> **If** $t_{s_1(i)} < t_{s_2(i)}$ **then** *; bottom-up initialization of the active set C*
>
> **If** $t_i > t_{s_1(i)}$ **then** $t_i \leftarrow t_{s_1(i)}$; add $s_1(i)$ into C; *; x is made feasible*
>
> **Else If** $t_i > t_{s_2(i)}$ **then** $t_i \leftarrow t_{s_2(i)}$; add $s_2(i)$ into C.

For $k = 0$ *to* $MaxNumberIterations$

> Use Lagrange Algorithm to compute the stationary point (x^*, λ^*) of Eq. (8) with respects to C
>
> **If** (x^* is feasible) **then**
>
> **If** ($\lambda_h^* \geq 0$ for every $h \in C$) **then** return solution ; *KKT conditions, x* is feasible and optimal*
>
> **Else** let λ_h^* be the most negative component of λ^*; remove h from C ; *h is useless*
>
> **Else**
>
> Let $V = \{i \in 2, \dots, 2n-1 | t_i^* < t_{a(i)}^*\}$; *V is the set of violated constraints in x**
>
> **If** there exists $i \in V$ s.t. $t_i = t_{a(i)}$ **then** add i into C *; x remains unchanged*
>
> **Else** Let $\Delta_i = (t_i^* - t_{a(i)}^*)/(t_i - t_{a(i)})$ for all i in V
>
> Let p be the index such that Δ_p is minimum over all Δ_i terms, and $\alpha = 1/(1 - \Delta_p)$
>
> Add p into C *; p is the "first violated" constraint*
>
> Set $x \longleftarrow x + \alpha(x^* - x)$ *; x is feasible and as close as possible to x**

Return *x.* *; x is feasible and nearly optimal*

This (active set, Nocedal and Wright 2006) algorithm takes as input a binary rooted tree with branch lengths and the sampling dates associated to the tips. The outputs are the substitution rate and the dates of all internal nodes (including the root) which minimize Equation (2), such that $t_i \geq t_{a(i)}$, for every i, and $\omega \geq \omega_{min}$. QPD runs in $O(f \times n)$, where f is the number of iterations needed to reach the optimal feasible solution (x*, when KKT conditions are fulfilled), and n is the number of taxa. We observed $f \ll n$ in all of our experiments (e.g. 4-5 with 110-taxon simulated trees, and 69 with the 891-taxon influenza data set), and the maximum number of iterations (set to 1,000 in our implementation) was never reached, by far.

The starting point x is obtained by LD, and the active set C is computed from the constraints being violated in x, using a post-order tree traversal. In each iteration, we compute the optimal solution of Equation (2) such that $t_i = t_{a(i)}$ for all $i \in C$. This is equivalent to searching for the stationary point (x^*, λ^*) of the Lagrange Equation (8). To achieve this task we use a slightly modified version of LD where the branches corresponding to violated constraints are collapsed; see Lagrange Algorithm below. If the KKT conditions are satisfied (x^* is feasible and λ^* does not contain any negative elements), then the algorithm stops and returns x^*. Otherwise the active set C is updated as follows. If x^* is feasible and λ^* contains negative elements, then the constraint corresponding to the most negative value is relaxed. Otherwise, we search for a violated constraint to add into C and a new feasible point x being closer to the optimum. To this purpose, x is moved as close as possible to x^* following the direction $d = x^* - x$. In other words, we have to find the largest number α in [0,1] such that $x + \alpha d$ is feasible. Note that if a constraint i is valid in x^*, then it is also valid in $x + \alpha d$ for every α in [0,1]. Thus, we only need to consider the constraints i that are violated in x^*. Let $V = \{i \in 2, \ldots 2n-1 | t_i^* < t_{a(i)}^*\}$, then $t_i + \alpha(t_i^* - t_i) \geq t_{a(i)} + \alpha(t_{a(i)}^* - t_{a(i)})$ must hold for every $i \in V$. If there exists $i \in V$ such that $t_i = t_{a(i)}$, then $\alpha = 0$ and i is the new constraint added into C. Otherwise, $t_i > t_{a(i)}$ for all $i \in V$ (because x is feabsible), thus let $\Delta_i = (t_i^* - t_{a(i)}^*)/(t_i - t_{a(i)})$, the feasibility condition becomes $\alpha \leq 1/(1 - \Delta_i)$ for every $i \in V$. Let p be the index such that Δ_p is minimum over all Δ_i terms with $i \in V$. Then, $\alpha = 1/(1 - \Delta_p)$, x becomes $x + \alpha d$, and p is added into the active set. We repeat these calculations until the KKT conditions are satisfied (then x* is returned) or the given maximum number of iterations is reached (then x is returned). We prove below that this algorithm converges to the optimal solution when the number of iterations is large enough.

LAGRANGE LINEAR-TIME ALGORITHM TO FIND THE STATIONARY POINT OF EQUATION (8)

Lagrange algorithm

Input: The Lagrange function (8) and the active set C

Output: The stationary point (x^*, λ^*) of (8)

Calculate the system of equations (10,11,12)

For each internal node i such that i is not in C, and either $s_1(i)$ or $s_1(i)$ is in C

> Calculate the set C_i
>
> **If** C_i contains only internal nodes, then collapse all branches in C_i
>
> **Else** assign the dates of all nodes in C_i by the date of the leaf contained in C_i

Use the generalized version of LD algorithm to compute x^* on the collapsed tree

For each C_i

> **For** each j in C_i with respect to the order defined by rules 1,2,3,4
>
> Calculate λ_j^* by using x^* and the already computed elements of λ^* in Equation (10.j)

Return x^* and λ^*

The stationary point (x^*, λ^*) of the Lagrange Equation (8) is computed by solving the system of equations in which the partial derivatives of Γ (Eq. (8)) are null (the weight terms w_i are ignored in the formulae for simplicity):

$$\frac{\partial \Gamma\left(x,\lambda\right)}{\partial t_1} = 2\omega\left(b_{s_1(1)} - \omega t_{s_1(1)} + \omega t_1\right) + 2\omega\left(b_{s_2(1)} - \omega t_{s_2(1)} + \omega t_1\right) + \lambda_{s_1(1)} + \lambda_{s_2(1)} \qquad \textbf{(10.1)}$$
$$= 0,$$

$$\frac{\partial\Gamma(x,\lambda)}{\partial t_i} = -2\omega\left(b_i - \omega t_i + \omega t_{a(i)}\right) + 2\omega\left(b_{s_1(i)} - \omega t_{s_1(i)} + \omega t_i\right) + 2\omega\left(b_{s_2(i)} - \omega t_{s_2(i)} + \omega t_i\right) - \lambda_i + \lambda_{s_1(i)} + \lambda_{s_2(i)} = 0, \text{ for } i = 2,\ldots,n-1, \quad \textbf{(10.i)}$$

$$\frac{\partial\Gamma(x,\lambda)}{\partial\omega} = -2\times\sum_{i=2}^{2n-1}\left(t_i - t_{a(i)}\right)\left(b_i - \omega t_i + \omega t_{a(i)}\right) = 0. \quad \textbf{(11)}$$

This system of equations is solved under the conditions that:

$$t_i = t_{a(i)}, \forall i \in C, \text{ and } \lambda_i = 0, \forall i \notin C. \quad \textbf{(12)}$$

The "Lagrange" algorithm (summarized above) solves Equations (10, 11, 12) in linear time. Through the first loop, the active constraints of C are imposed on the tree. Then we use a generalized version of LD on the collapsed tree to compute x^*. Lastly, we compute λ^* by using x^* and recursive replacements following the specific order of C_i in Equations (10). Let us detail these calculations:

Computing x^*: the key remark is that if i is in C, then the equation $\partial\Gamma/\partial t_i$ contains $-\lambda_i$, and the equation $\partial\Gamma/\partial t_{a(i)}$ contains $+\lambda_i$. So λ_i is eliminated by taking the sum of these two equations, and we can eliminate all Lagrange multipliers in all Equations (10.i) by taking the sum of appropriate sets of equations. Each set corresponds to a maximal set of nodes whose dates are equal due to active constraints. Let call i the "top" of such a set if i is not in C but either $s_1(i)$ or $s_2(i)$ are in C. We then define for each top i a set of active constraints C_i by the following rules: (1) every child of i that is in C is in C_i; (2) if $j \in C$ and $a(j) \in C_i$, then $j \in C_i$. It is easy to see that these sets are pairwise disjoint. Moreover, each C_i contains at most one leaf all along QPD iterations, as we shall prove recursively. In the initial active set C each internal node has at most one child in C, so each C_i contains at most one leaf. Moreover, if C_i contains a leaf k, then $t_k \leq t_{k'}$ for every leaf $k' \neq k$ below i. Suppose that k' has its parent in C_i; so in the next iteration, if k

is still in C_i then k' is not added into C_i, because the constraint k' is not violated ($t_{k'} \geq t_{a(k')} = t_k$). If k is removed from C_i, then whether k' is added into C_i or not, C_i contains at most one leaf. If the parent of k' is not in C_i, then if k' is added into C in the next iteration, it must be contained in a C_j where $j \neq i$. Therefore, each C_i contains at most one leaf in every iteration of QPD. If C_i contains a leaf, then the dates of all nodes in C_i are equal to the date of this leaf, which is known. So it is not necessary to compute the dates of these nodes in this case.

Consider now the case where C_i only contains internal nodes. The sum of the equations $\partial\Gamma/\partial t_j$ for all $j \in C_i$ is $-2\omega\left(b_i - \omega t_i + \omega t_{a(i)}\right) + 2\omega \sum_{s\in S(i)}(b_s - \omega t_s + \omega t_i) = 0$, where $S(i)$ consists of the children of the lowest nodes of C_i. This equation does not contain any Lagrange multiplier, and has the same form as Equations (5) but on a tree where all branches in C_i are collapsed. Therefore, by using on the new tree a generalized version of LD able to process unresolved trees, we can compute in linear time, for every internal node i such that $i \notin C$, the constants u_i and v_i such that $t_i = u_i + v_i/\omega$. The later equations and the value of all known t_i are then replaced into (11) to calculate ω^*, and then t_i^*.

Computing λ^*: we can also compute λ^* in linear time by replacing x^* into Equations (10.i) following a specific order which ensures that each time there is one and only one unknown λ_i in the equation at hand, so that we can easily deduce its value. This order is defined on the C_i sets, distinguishing two cases:

(1) If C_i contains internal nodes only, then one simply uses a bottom-up order: for every $j_1, j_2 \in C_i$, if j_1 is below j_2, then $j_1 < j_2$ (rule 1).

(2) If C_i contains a leaf k, then as we proved above, k is the only leaf in C_i. Denote by C_i^1 the set of nodes in C_i that are not above or equal to the leaf k, and C_i^2 the set of nodes in C_i that are above or equal to k. The order of the elements in C_i respects the following rules: the nodes in C_i^1 follow a bottom-up order (rule 2); the nodes in C_i^2 follow a top-down order (rule 3); the nodes in C_i^1 are smaller than the nodes in C_i^2 (rule 4).

Then, λ_j is computed following the order defined above: for every $j \in C_i$, either j is the smallest node in C_i (so the first one in C_i to be computed) or all Lagrange multipliers of nodes smaller than j in C_i have already been calculated. If C_i only contains internal nodes (case (1)), then $\forall j \in C$, j always has two direct descendant $s_1(j)$, $s_2(j)$. Moreover, $\lambda_{s_1(j)}$ (resp. $\lambda_{s_2(j)}$) is either 0 (not in the active set) or has already been calculated (because $s_1(j)$, $s_2(j)$ are smaller than j by rule 1). In other words, λ_j is the only unknown variable in Equation (10.j) and is easily computed. Let us now deal with case (2) and let k be the only leaf in C_i. If j is not above or equal to k, then $\lambda_{s_1(j)}$ (resp. $\lambda_{s_2(j)}$) is either 0 or has already been calculated (because $s_1(j)$, $s_2(j)$ are smaller than j by rule 2); therefore, λ_j is determined by Equation (10.j). If j is above or equal to k, then let j' be the sibling node of j. So j' is not above or equal to k. Hence, $\lambda_{a(j)}$ (resp. $\lambda_{j'}$) is either 0 or has already been calculated (they are smaller than j by the rule 3 (resp. rule 4)). Therefore, λ_j is the only unknown variable in Equation (10.j) and thus is easily computed.

CONVERGENCE OF THE QPD ALGORITHM

Let us consider the problem of finding the minimal solution of Ψ (Eq. 2b), subject to the constraints $\omega \geq \omega_{min}$, $\beta_i \geq \beta_{a(i)}$ for $i = 2, \ldots, n-1$, and $\omega t_i \geq \beta_{a(i)}$ for $i = n, \ldots, 2n-1$. Ψ is convex quadratic, so the active set method is ensured to converge to its unique global minimum on the constraint domain (Nocedal and Wright 2006). By choosing the starting feasible point and the initial active set of Ψ similarly to the one of ϕ in QPD algorithm (*i.e.*, the minimal solution of Ψ without constraints, and then deriving *C* from the set of violated constraints, see above), we prove here the equivalence of the active sets and feasible points of Ψ and ϕ in every iteration of both algorithms. This implies the convergence of the QPD algorithm, which is the active set method applied to ϕ (Eq. 2).

Let $x = (\omega_x, t_1, \ldots, t_{n-1})$ be the variable vector of ϕ (Eq. 2), and $y = (\omega_y, \beta_1, \ldots, \beta_{n-1})$ be the variable vector of Ψ (Eq. 2b). We say that x and y are equivalent if and only if $\omega_x = \omega_y$ and $\beta_i = \omega_y t_i$ for every i from 1 to $n-1$.

First, it is easy to see that $y = (\omega, \beta_1, \ldots, \beta_{n-1})$ is the minimal solution of Ψ (Eq. 2b), without constraints, if and only if $x = (\omega, \beta_1/\omega, \ldots, \beta_{n-1}/\omega)$ is the minimal solution of ϕ (Eq. 2), without constraints. In other words, the minimal solution of ϕ and Ψ, without constraints, are equivalent, and is easy to define for Ψ a starting point equivalent to ϕ's and an initial active set exactly the same as ϕ's (see QPD algorithm above).

Second, given an active set C, it is easy to see that $y = (\omega, \beta_1, \ldots, \beta_{n-1})$ is the minimal solution of Ψ (Eq. 2b) subjects to C (*i.e.*, $\forall i \in W, \beta_i = \beta_{a(i)}$ if $i < n$, and $\omega t_i = \beta_{a(i)}$ if $i \geq n$) if and only if $x = (\omega, \beta_1/\omega, \ldots, \beta_{n-1}/\omega)$ is the minimal solution of ϕ (Eq. 2) subjects to C (*i.e.*, $\forall i \in C, t_i = t_{a(i)}$). Hence, if (x^*, λ_x^*) (resp. (y^*, λ_y^*)) is the stationary point of the Lagrange function of ϕ (Eq. 8) (resp. Ψ, Eq. (8b)) with respects to C, then x^* and y^* are equivalent. We will prove that $\lambda_x^* = \omega^* \lambda_y^*$. The Lagrange function of Ψ with respects to C is:

$$\mathcal{L}(y, \lambda) = \Psi(y) - \sum_{i \in C, i<n} \lambda_i (\beta_i - \beta_{a(i)}) - \sum_{i \in C, i \geq n} \lambda_i (\omega t_i - \beta_{a(i)}). \quad \textbf{(8b)}$$

By taking the partial derivatives of $\mathcal{L}$ in Equation (8b), we have a system of equations similar to Equations (10, 11, 12) but without ω factor before each term as in Equations (10). By replacing x^* and y^* (which are equivalent) into these equations, we deduce that $\lambda_x^* = \omega^* \lambda_y^*$.

Finally, we prove that if the two starting active sets C_x, C_y are the same, then they remain the same after each update. Suppose that they are the same at the k^{th} iteration, then by using the claim above, we have that x^*, y^* are equivalent and $\lambda_x^* = \omega^* \lambda_y^*$. So the KKT condition on (y^*, λ_y^*) is satisfied if and only if it is satisfied on (x^*, λ_x^*). The indices of the most negative components of λ_x^* and λ_y^* are the same, so the constraints to relax are the same. It remains to prove that the two new active constraints added into C_x, C_y are also the same. Since x^* and y^* are equivalent, then the set of constraints that are violated in x^* are the same as in y^*. In other words, let $V_{x^*} = \{i = 2, \dots, 2n-1 | t_i^* < t_{a(i)}^*\}$, and $V_{y^*} = \{i = 2, \dots, n-1 | \beta_i^* < \beta_{a(i)}^*\} \cup \{i = n, \dots, 2n-1 | \omega^* t_i < \beta_{a(i)}^*\}$, then $V_{x^*} = V_{y^*}$. If there exists $i \in V_{x^*}$ such that $t_i^* = t_{a(i)}^*$, then $\beta_i^* = \beta_{a(i)}^*$ if i<n or $\omega^* t_i = \beta_{a(i)}^*$ if $i \geq$n, and i is the new constraint to be added into C_x and C_y. Otherwise, let $\Delta_i = (t_i^* - t_{a(i)}^*)/(t_i - t_{a(i)})$ and $\nabla_i = (\omega^*/\omega)\Delta_i$ for all $i \in V_{x^*}$. As explained in in QPD description (above), the constraint p to be added into C_x corresponds to the Δ_p index that is minimum over all Δ_i with $i \in V_{x^*}$. Similarly, we can prove that the constraint q to be added into C_y corresponds to the ∇_q index that is minimum of all ∇_i with $i \in V_{y^*}$. Because ω^* and ω are positive, and $V_{x^*} = V_{y^*}$, then we always have $p = q$. In other words, we add the same constraint into the active sets C_x, C_y. Hence, the two active sets are the same at the $(k+1)^{th}$ iteration, and they are the same until the end, when the KKT condition is satisfied for both. In conclusion, the QPD algorithm is converged.

ROOT ESTIMATION

Without constraints

Optimizing Equation (9) without constraints can also be solved in linear time by slightly modifying the algorithm LD. Indeed, by calculating the first order derivatives of φ with respect to each variable, we have a system of equations:

$$\begin{cases} \mu = \dfrac{\omega}{2b}\left(t_{r_1} - t_{r_2}\right) + \dfrac{1}{2}, \\ t_r = \dfrac{1}{2}\left(t_{r_1} + t_{r_2} - \dfrac{1}{\omega}b\right), \\ t_i = \dfrac{1}{3}\left[\left(t_{s_1(i)} - \dfrac{1}{\omega}b_{s_1(i)}\right) + \left(t_{s_2(i)} - \dfrac{1}{\omega}b_{s_2(i)}\right) + \left(t_r + \dfrac{1}{\omega}\mu b\right)\right], & i = r_1 \\ t_i = \dfrac{1}{3}\left[\left(t_{s_1(i)} - \dfrac{1}{\omega}b_{s_1(i)}\right) + \left(t_{s_2(i)} - \dfrac{1}{\omega}b_{s_2(i)}\right) + \left(t_r + \dfrac{1}{\omega}(1-\mu)b\right)\right], & i = r_2 \\ t_i = \dfrac{1}{3}\left[\left(t_{s_1(i)} - \dfrac{1}{\omega}b_{s_1(i)}\right) + \left(t_{s_2(i)} - \dfrac{1}{\omega}b_{s_2(i)}\right) + \left(t_{a(i)} + \dfrac{1}{\omega}b_i\right)\right], & i \neq r_1, r_2 \\ \omega = \dfrac{\mu b\left(t_{r_1} - t_r\right) + (1-\mu)b\left(t_{r_2} - t_r\right) + \sum_{i \neq r_1, r_2} b_i\left(t_i - t_{a(i)}\right)}{\left(t_{r_1} - t_r\right)^2 + \left(t_{r_2} - t_r\right)^2 + \sum_{i \neq r_1, r_2}\left(t_i - t_{a(i)}\right)^2}, \end{cases}$$

which simplifies into:

$$\begin{cases} \mu = \dfrac{\omega}{2b}\left(t_{r_1} - t_{r_2}\right) + \dfrac{1}{2}, \\ t_r = \dfrac{1}{2}\left(t_{r_1} + t_{r_2} - \dfrac{1}{\omega}b\right), \\ t_i = \dfrac{1}{2}\left[\left(t_{s_1(i)} - \dfrac{1}{\omega}b_{s_1(i)}\right) + \left(t_{s_2(i)} - \dfrac{1}{\omega}b_{s_2(i)}\right)\right], & i = r_1, r_2 \\ t_i = \dfrac{1}{3}\left[\left(t_{s_1(i)} - \dfrac{1}{\omega}b_{s_1(i)}\right) + \left(t_{s_2(i)} - \dfrac{1}{\omega}b_{s_2(i)}\right) + \left(t_{a(i)} + \dfrac{1}{\omega}b_i\right)\right], & i \neq r_1, r_2 \\ \omega = \dfrac{\mu b\left(t_{r_1} - t_r\right) + (1-\mu)b\left(t_{r_2} - t_r\right) + \sum_{i \neq r_1, r_2} b_i\left(t_i - t_{a(i)}\right)}{\left(t_{r_1} - t_r\right)^2 + \left(t_{r_2} - t_r\right)^2 + \sum_{i \neq r_1, r_2}\left(t_i - t_{a(i)}\right)^2}. \end{cases}$$

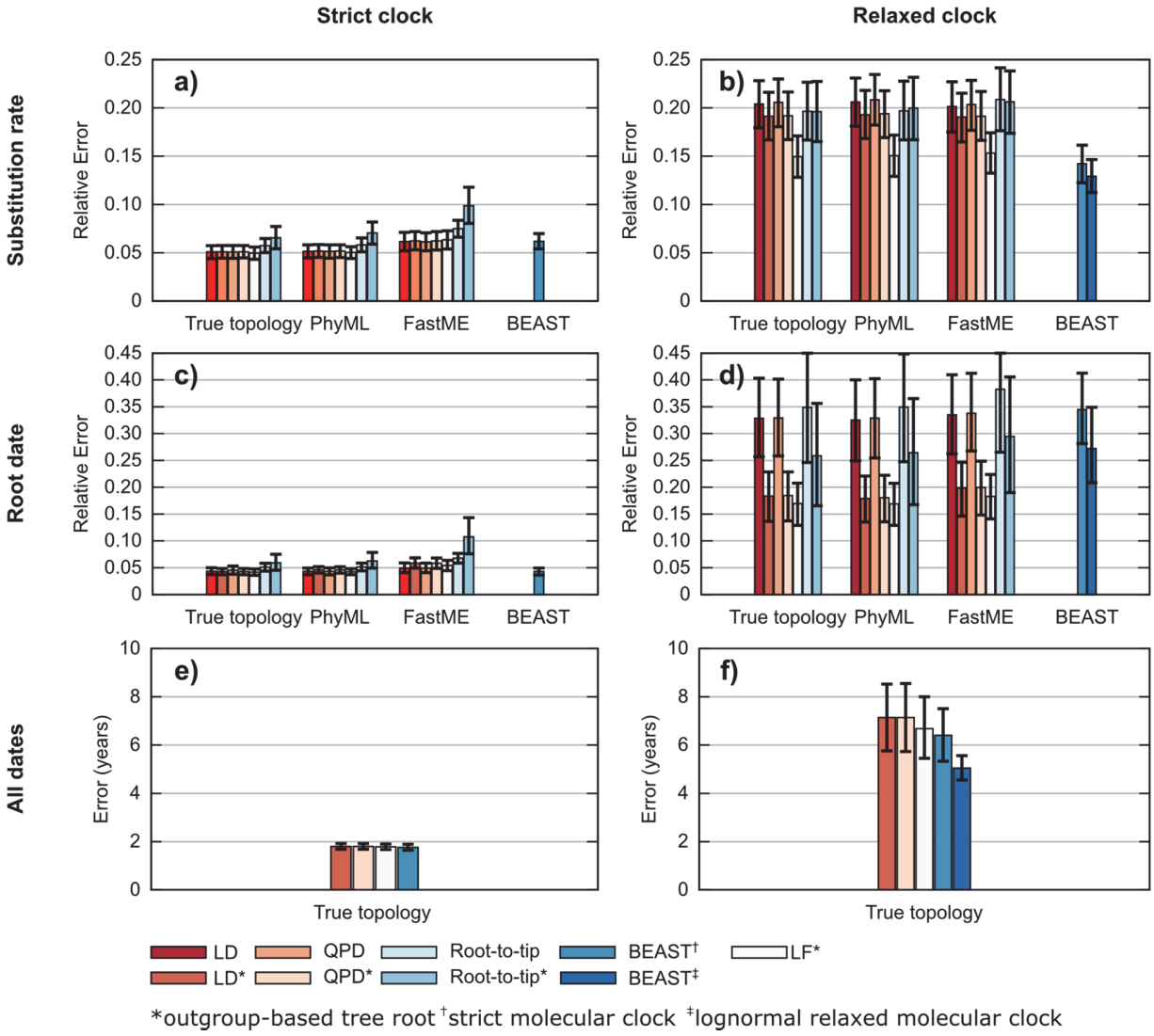
Just as with Equation (2), the above system has a unique solution and can be resolved in linear time using the same approach as LD (*i.e.,* simplify progressively by recursive replacements upon post-order and then pre-order, see above). Hence, the function φ has only one critical point. By considering its second order derivatives, it is easy to see that this point corresponds to the minimum solution. To solve this problem when μ is bounded between 0 and 1, we first calculate

the minimum solution in the general case. If the minimum value of μ is not in [0, 1], then we compute the minimum solution of Equation (9) in the cases where $\mu = 0$ and $\mu = 1$, and then take the one which makes φ smaller. By running this procedure on all branches, the time complexity is quadratic in the number of tips.

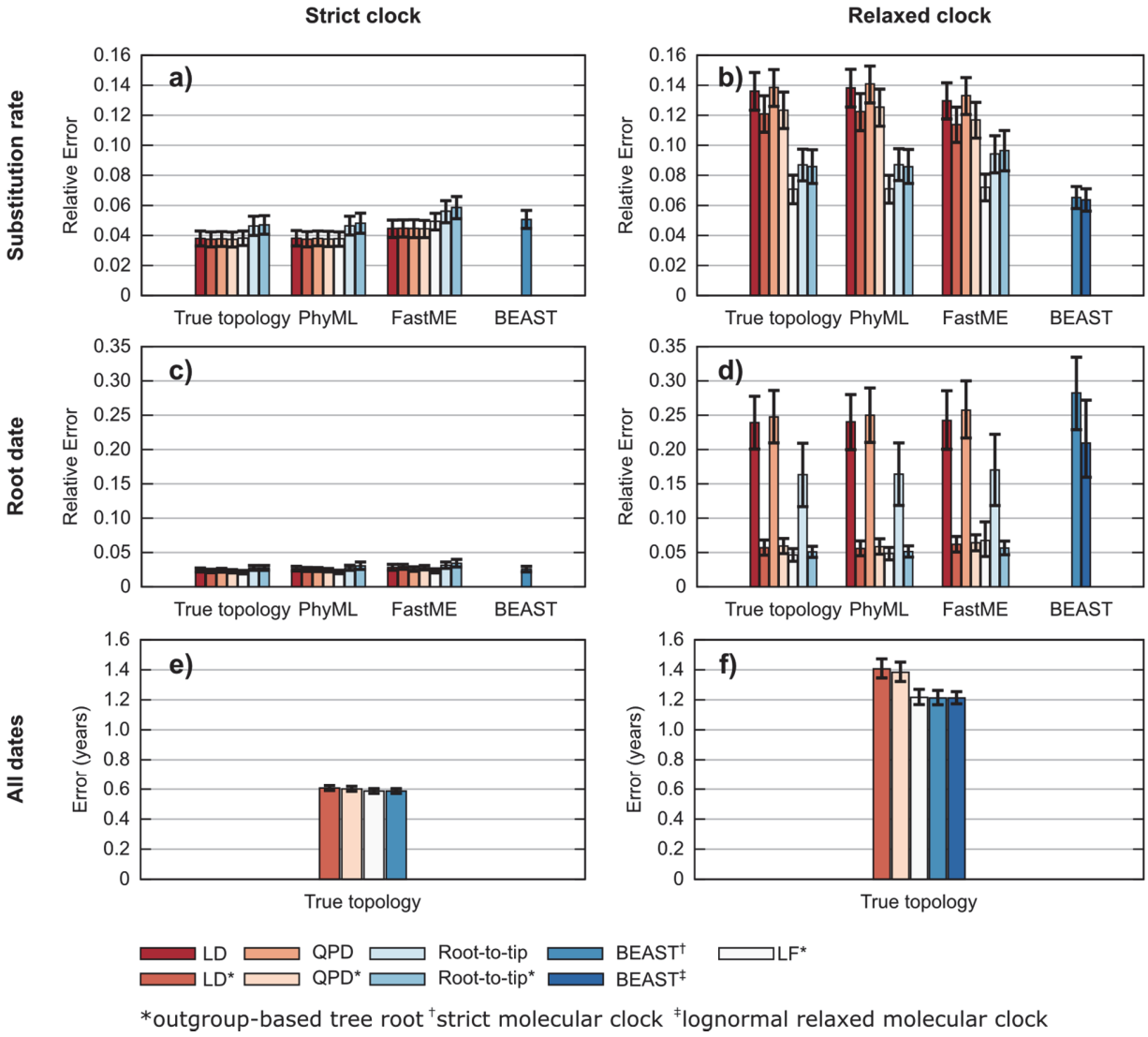
With constraints

We use again the active set method (see QPD algorithm) to optimize φ in Equation (9), subject to the constraints $\omega \geq \omega_{min}$, $t_i \geq t_{a(i)}$, and $0 \leq \mu \leq 1$. In this case, $x = (\omega, t_1, \ldots, t_{n-1}, \mu)$. Let $\Im$ be the Lagrange function of Equation (9). To find the stationary point of $\Im$, we use an algorithm similar to the Lagrange algorithm (see above). The Lagrange multipliers are computed in the same way, using that λ_μ can be computed directly from equation $\partial\Im/\partial\mu = 0$.

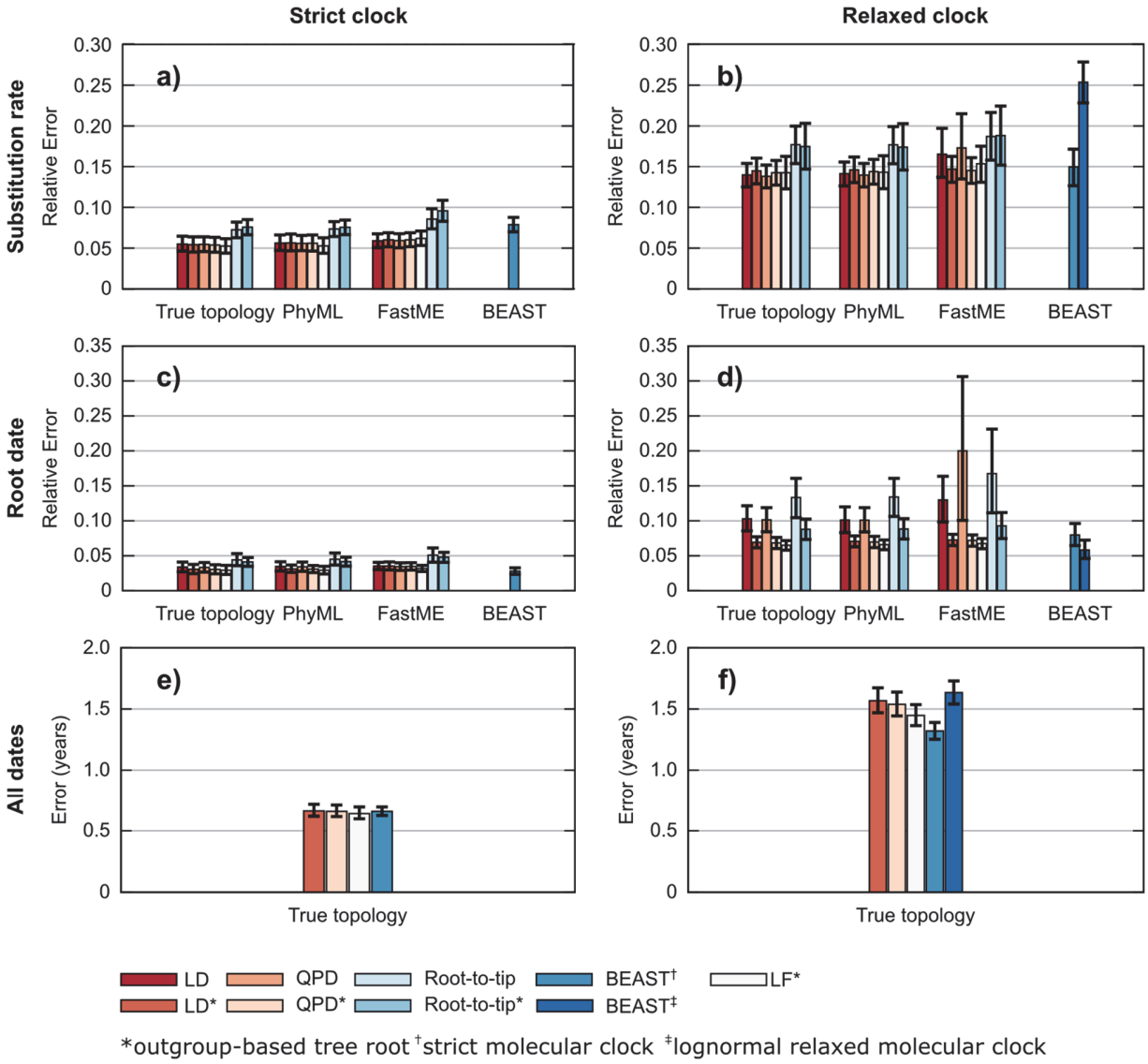
To avoid exploring all branches, which could require heavy computing times with large data sets, we pre-estimate the root position without temporal constraints. Then, we use the method with constraints to search for the local minimum around that position. Let o be the branch that contains the pre-estimated root position, and E be a set that initially contains the branch o only. Denote by φ_i the optimum value of φ in Equation (9), when the root is situated on the branch i. Let r be the branch that contains the estimated root (r is initialized with o), and $\varphi^* = \varphi_r$ the corresponding objective function value. Consider a branch i in E; add into E all neighbors j of i such that $\varphi_j<\varphi_i$, and then remove i from E. If the smallest value of these neighboring branches is smaller than φ^*, then update φ^* with this value and r with the corresponding branch. Continue exploring the neighbors of all branches in E and stop when E is empty. It is easy to check that the explored branches form the largest subtree in which the value of φ at o is the greatest, and that this value decreases from o to every terminal branch of that subtree. Moreover, r corresponds to the terminal branch having the smallest value. In practice, starting from the pre-estimated root without constraints, only a few branches need to be explored to reach to the optimal one with constraints. For example, with our simulated data of 110 taxa, we usually have to consider 4-5 branches in average, and in most cases the pre-estimated root is on the same branch as that of the final root. However, both fast and complete options are available in LSD, as we observed a few rare cases where the fast option performed poorly.



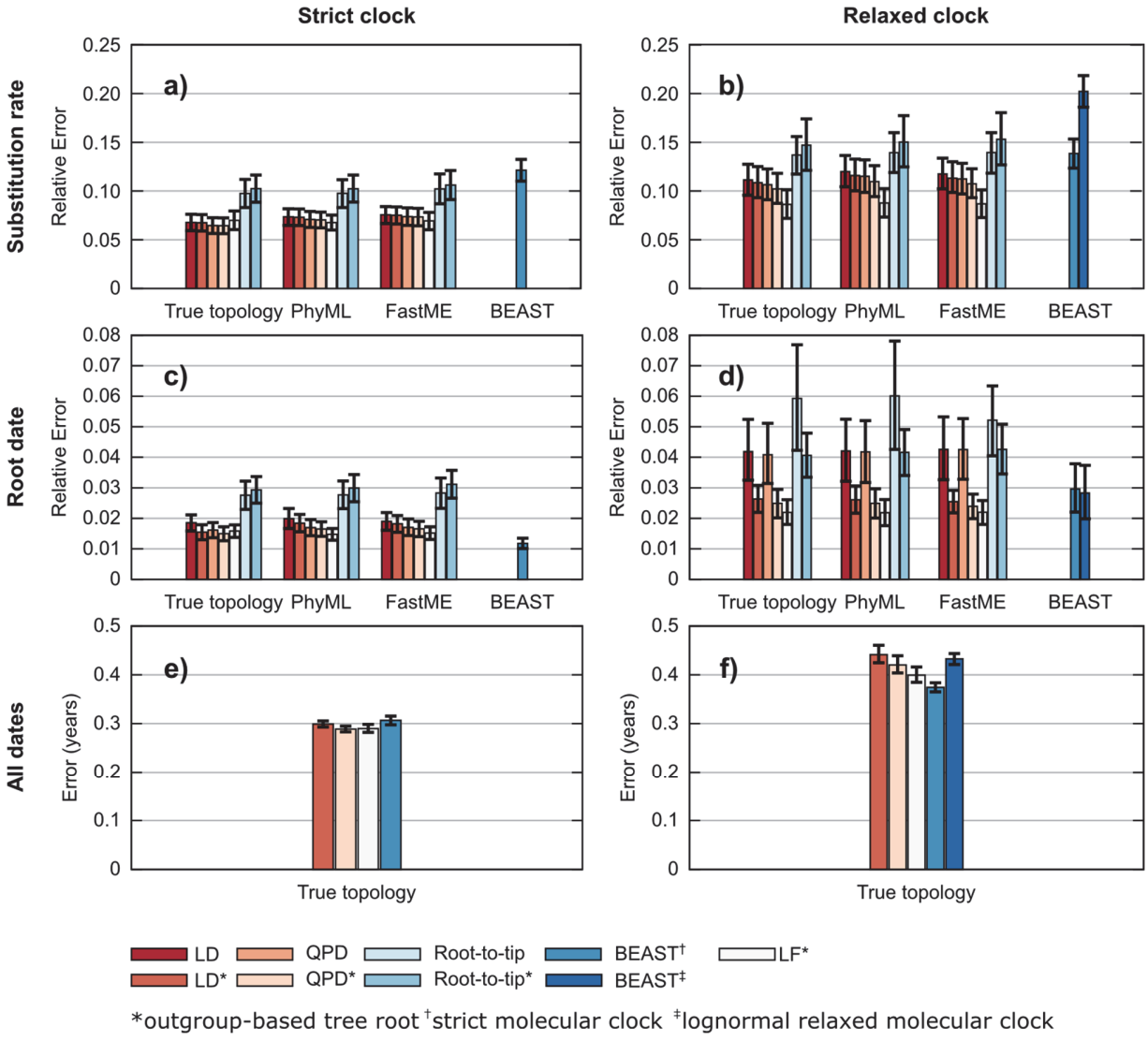
Supplementary Figure S1. Summary results with the 750/3×25 simulated data set. See legend of Figure 2.



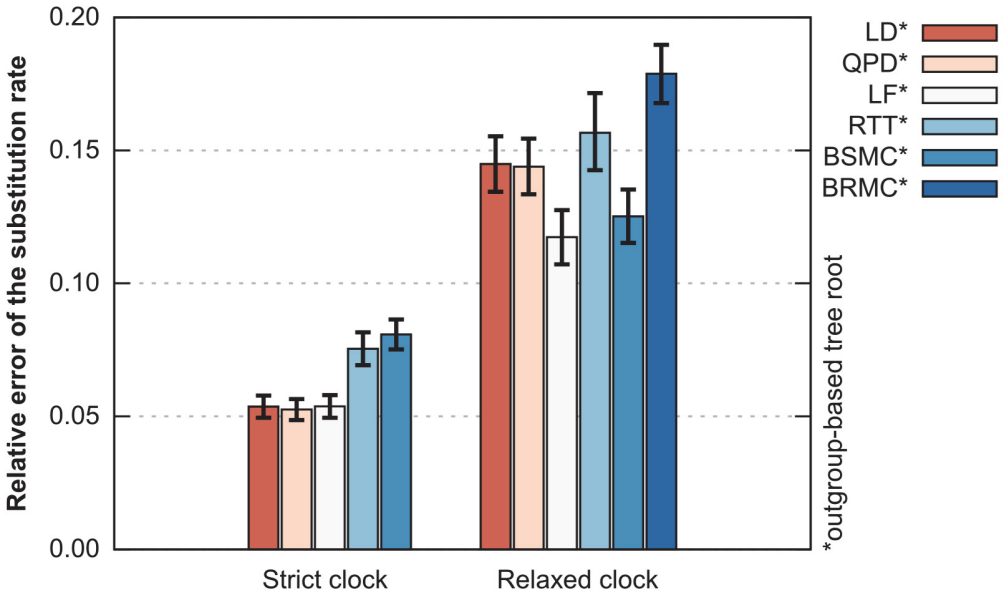
Supplementary Figure S2. Summary results with the 750/11×10 simulated data set. See legend of Figure 2.



Supplementary Figure S3. Summary results with the 995/3×25 simulated data set. See legend of Figure 2.



Supplementary Figure S4. Summary results with the 995/11×10 simulated data set. See legend of Figure 2.



Supplementary Figure S5. Error in rate estimation when using the true topology. The true, rooted topology is used by all methods. With distance methods (LD*, QPD*, LF*, RTT*) the branch lengths are estimated by PhyML. BEAST (BSMC*: strict clock, BRMC*: lognormal, relaxed clock) sample the branch lengths (and the dates and the substitution rate) from sequence data. We measure here the relative error in estimating the substitution rate. We see that the high estimation error of BEAST is not due to topological errors, and that the ranking of methods is basically preserved, compared to the (realistic case) where an approximate (FastME, PhyML or BEAST) topology is used to estimate de substitution rate (and other date parameters).

	750/11 ×10	995/3 ×25
Phylogeny inference		
DNAdist+FastME	5	2
PhyML	8mn	4mn
Dates and rate estimation		
LD	0.1	<0.1
LD*	<0.1	<0.1
QPD	0.2	<0.1
QPD*	<0.1	<0.1
Root-to-tip	<0.1	<0.1
Root-to-tip*	<0.1	<0.1
LF*	3.5	1.5
BEAST with a strict molecular clock	4h	3h
BEAST with a relaxed molecular clock	17h	12h

*outgroup-based rooted tree

Table S1. Computing time with simulated data sets. The time is expressed in seconds, except otherwise specified. Two types of trees are used: 750/11×10 with a death ratio of 750/1,000 and 110 leaves; 995/3×25 with a death ratio of 995/1,000 and 75 leaves. All calculations have been performed on our server: Intel(R) Xeon(R) X5650 @ 2.67GHz, single core, no parallelization.

		LD			LD*			QPD			QPD*			Root-to-tip			Root-to-tip*			Langley-Fitch*			BSMC†
		True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME	
Substitution rate	750/3×25	0.0509	0.0514	0.0614	0.0510	0.0518	0.0623	0.0508	0.0513	0.0613	0.0510	0.0518	0.0623	0.0574	0.0581	0.0749	0.0655	0.0705	0.0985	0.0496	0.0501	0.0635	0.0618
		-0.0080	-0.0097	-0.0007	-0.0082	-0.0099	0.0005	-0.0080	-0.0098	-0.0008	-0.0082	-0.0100	0.0005	-0.0050	-0.0066	0.0025	-0.0023	-0.0027	0.0054	-0.0007	-0.0023	0.0095	0.0355
	750/11×10	0.0380	0.0382	0.0447	0.0374	0.0376	0.0447	0.0376	0.0381	0.0447	0.0373	0.0378	0.0446	0.0463	0.0464	0.0561	0.0471	0.0480	0.0587	0.0382	0.0377	0.0492	0.0507
		-0.0037	-0.0053	0.0064	-0.0027	-0.0041	0.0081	-0.0036	-0.0053	0.0065	-0.0029	-0.0046	0.0076	0.0064	0.0048	0.0200	0.0070	0.0043	0.0215	0.0095	0.0080	0.0223	0.0327
	995/3×25	0.0550	0.0563	0.0592	0.0543	0.0565	0.0602	0.0546	0.0558	0.0590	0.0540	0.0559	0.0603	0.0726	0.0734	0.0857	0.0758	0.0757	0.0960	0.0524	0.0527	0.0621	0.0788
		-0.0094	-0.0139	-0.0002	-0.0096	-0.0142	-0.0010	-0.0085	-0.0129	-0.0000	-0.0088	-0.0133	-0.0006	0.0001	-0.0036	0.0139	0.0016	-0.0008	0.0238	0.0038	0.0014	0.0191	0.0606
	995/11×10	0.0678	0.0736	0.0757	0.0676	0.0732	0.0753	0.0646	0.0706	0.0736	0.0645	0.0704	0.0735	0.0975	0.0977	0.1020	0.1024	0.1023	0.1062	0.0699	0.0677	0.0694	0.1214
		-0.0263	-0.0373	-0.0354	-0.0261	-0.0368	-0.0346	-0.0206	-0.0313	-0.0315	-0.0201	-0.0307	-0.0305	-0.0024	-0.0059	0.0012	0.0026	-0.0005	0.0089	-0.0114	-0.0137	-0.0044	0.1071
	Summary	**0.0540**	**0.0563**	**0.0612**	**0.0537**	**0.0562**	**0.0616**	**0.0528**	**0.0552**	**0.0605**	**0.0526**	**0.0552**	**0.0610**	**0.0711**	**0.0716**	**0.0814**	**0.0754**	**0.0766**	**0.0917**	**0.0537**	**0.0531**	**0.0615**	**0.0826**
		-0.0119	**-0.0165**	**-0.0075**	**-0.0117**	**-0.0163**	**-0.0067**	**-0.0102**	**-0.0148**	**-0.0065**	**-0.0100**	**-0.0146**	**-0.0058**	**-0.0002**	**-0.0028**	**0.0094**	**0.0022**	**0.0001**	**0.0149**	**0.0003**	**-0.0017**	**0.0116**	**0.0590**
tMRCA	750/3×25	0.0436	0.0432	0.0494	0.0428	0.0461	0.0586	0.0432	0.0433	0.0494	0.0427	0.0458	0.0583	0.0510	0.0512	0.0678	0.0590	0.0626	0.1075	0.0416	0.0424	0.0541	0.0430
		0.0005	0.0004	0.0009	0.0022	0.0056	0.0135	0.0002	0.0001	0.0009	0.0021	0.0053	0.0133	-0.0031	-0.0034	-0.0128	-0.0073	-0.0077	-0.0379	-0.0007	0.0003	-0.0081	-0.0032
	750/11×10	0.0246	0.0265	0.0283	0.0232	0.0252	0.0292	0.0244	0.0255	0.0258	0.0227	0.0242	0.0279	0.0276	0.0277	0.0314	0.0275	0.0305	0.0343	0.0214	0.0216	0.0231	0.0258
		0.0042	0.0059	0.0058	0.0079	0.0110	0.0124	0.0033	0.0046	0.0039	0.0068	0.0096	0.0110	0.0012	0.0014	-0.0019	0.0025	0.0015	-0.0085	0.0061	0.0064	0.0009	-0.0157
	995/3×25	0.0339	0.0347	0.0354	0.0306	0.0312	0.0354	0.0333	0.0342	0.0349	0.0304	0.0309	0.0348	0.0442	0.0451	0.0508	0.0413	0.0417	0.0479	0.0295	0.0293	0.0320	0.0280
		-0.0061	-0.0069	-0.0051	-0.0029	-0.0027	0.0022	-0.0062	-0.0068	-0.0061	-0.0030	-0.0028	0.0017	-0.0085	-0.0096	-0.0119	-0.0049	-0.0058	-0.0092	-0.0034	-0.0038	-0.0051	-0.0003
	995/11×10	0.0185	0.0199	0.0190	0.0154	0.0184	0.0182	0.0161	0.0170	0.0172	0.0149	0.0165	0.0166	0.0276	0.0277	0.0283	0.0293	0.0299	0.0311	0.0158	0.0147	0.0152	0.0118
		-0.0005	0.0000	-0.0011	0.0012	0.0037	0.0046	-0.0036	-0.0034	-0.0037	0.0005	0.0021	0.0029	-0.0032	-0.0035	-0.0042	-0.0014	-0.0026	-0.0046	0.0003	0.0005	-0.0009	-0.0005
	Summary	**0.0316**	**0.0323**	**0.0348**	**0.0298**	**0.0319**	**0.0383**	**0.0310**	**0.0315**	**0.0340**	**0.0295**	**0.0312**	**0.0376**	**0.0390**	**0.0393**	**0.0474**	**0.0412**	**0.0432**	**0.0633**	**0.0288**	**0.0289**	**0.0343**	**0.0293**
		-0.0005	**-0.0001**	**0.0001**	**0.0021**	**0.0044**	**0.0082**	**-0.0016**	**-0.0014**	**-0.0013**	**0.0016**	**0.0035**	**0.0072**	**-0.0034**	**-0.0038**	**-0.0077**	**-0.0028**	**-0.0036**	**-0.0150**	**0.0006**	**0.0008**	**-0.0033**	**-0.0049**

*outgroup-based rooted tree

†Beast with strict molecular clock model and coalescent constant size tree prior

Supplementary Table S2. Detailed estimation results with data simulated with a strict molecular clock (SMC). This table displays the errors of the various methods in estimating the root date (tMRCA) and the substitution rate, for the four type of trees. Two values are given in each cell: the relative error on top, and the relative bias on bottom. See paper for the definitions of these measures and the details and options of the tested methods. See also legend of Figure 2. The formula used to compute the summary value of the relative error is:

$$\sqrt{\frac{1}{400}\left[\sum_{i=1..100}\frac{(x-x_A^i)^2}{{x_A}^2}+\sum_{i=1..100}\frac{(x-x_B^i)^2}{{x_B}^2}+\sum_{i=1..100}\frac{(x-x_C^i)^2}{{x_C}^2}+\sum_{i=1..100}\frac{(x-x_D^i)^2}{{x_D}^2}\right]}$$

For the substitution rate: $x = x_A = x_B = x_C = x_D = 0.006$ is the true substitution rate; x_X^i is the estimated value for the ith data set (among 100) with tree type X. For the tMRCA: the four tree types are denoted as A = 750|3×25, B = 750|11×10, C = 995|3×25, and D = 995|11×10; $x = 0$ is the true root date; x_X^i is the estimated root date with the ith data set (among 100) with tree type X; x_X is the date (in years) of the most recent tips with tree type X, that is, $x_A = 66.833416$, $x_B = 29.36668$, $x_C = 32.466935$, $x_D = 22.493387$. For the global relative bias we use the standard arithmetic mean of the four relative biases of the four data sets.

		LD			LD*			QPD			QPD*			Root-to-tip			Root-to-tip*			Langley-Fitch*			BSMC†	BRMC‡
		True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME	True topology	PhyML	FastME		
Substitution rate	750/3×25	0.2042	0.2062	0.2016	0.1913	0.1931	0.1907	0.2058	0.2083	0.2036	0.1919	0.1942	0.1915	0.1965	0.1969	0.2086	0.1963	0.1999	0.2063	0.1495	0.1506	0.1533	0.1421	0.1289
		-0.1512	-0.1526	-0.1459	-0.1363	-0.1380	-0.1297	-0.1502	-0.1516	-0.1454	-0.1377	-0.1400	-0.1318	-0.0392	-0.0414	-0.0358	-0.0124	-0.0149	-0.0048	-0.0389	-0.0412	-0.0287	-0.0600	0.0324
	750/11×10	0.1362	0.1383	0.1297	0.1211	0.1225	0.1140	0.1387	0.1409	0.1331	0.1234	0.1255	0.1170	0.0871	0.0871	0.0942	0.0859	0.0859	0.0966	0.0708	0.0711	0.0720	0.0653	0.0635
		-0.1144	-0.1168	-0.1074	-0.0971	-0.0993	-0.0895	-0.1175	-0.1195	-0.1103	-0.1004	-0.1030	-0.0933	-0.0021	-0.0045	0.0110	0.0039	0.0021	0.0204	-0.0044	-0.0067	0.0066	-0.0130	0.0072
	995/3×25	0.1400	0.1414	0.1653	0.1449	0.1462	0.1471	0.1381	0.1397	0.1730	0.1426	0.1439	0.1453	0.1770	0.1770	0.1870	0.1748	0.1739	0.1882	0.1427	0.1432	0.1535	0.1497	0.2537
		-0.0691	-0.0716	-0.0695	-0.0558	-0.0585	-0.0470	-0.0648	-0.0679	-0.0677	-0.0518	-0.0548	-0.0438	-0.0227	-0.0262	-0.0154	0.0115	0.0085	0.0306	0.0266	0.0243	0.0419	0.0837	0.2208
	995/11×10	0.1116	0.1201	0.1177	0.1086	0.1159	0.1133	0.1065	0.1152	0.1125	0.1022	0.1097	0.1076	0.1370	0.1394	0.1394	0.1470	0.1502	0.1532	0.0861	0.0877	0.0870	0.1387	0.2023
		-0.0715	-0.0850	-0.0821	-0.0670	-0.0797	-0.0755	-0.0618	-0.0758	-0.0730	-0.0561	-0.0690	-0.0661	-0.0208	-0.0229	-0.0077	-0.0039	-0.0054	0.0083	-0.0105	-0.0141	-0.0028	0.1146	0.1854
	Summary	**0.1519**	**0.1550**	**0.1570**	**0.1449**	**0.1476**	**0.1448**	**0.1517**	**0.1550**	**0.1595**	**0.1439**	**0.1468**	**0.1441**	**0.1551**	**0.1558**	**0.1634**	**0.1566**	**0.1582**	**0.1664**	**0.1174**	**0.1183**	**0.1223**	**0.1285**	**0.1774**
		-0.1015	**-0.1065**	**-0.1012**	**-0.0890**	**-0.0939**	**-0.0854**	**-0.0986**	**-0.1037**	**-0.0991**	**-0.0865**	**-0.0917**	**-0.0837**	**-0.0212**	**-0.0238**	**-0.0120**	**-0.0002**	**-0.0024**	**0.0136**	**-0.0068**	**-0.0094**	**0.0042**	**0.0313**	**0.1115**
tMRCA	750/3×25	0.3287	0.3250	0.3351	0.1835	0.1790	0.1985	0.3292	0.3292	0.3379	0.1844	0.1804	0.1996	0.3487	0.3493	0.3825	0.2587	0.2643	0.2946	0.1696	0.1688	0.1828	0.3448	0.2722
		-0.1881	-0.1856	-0.1971	-0.0723	-0.0678	-0.0750	-0.1985	-0.1991	-0.2089	-0.0749	-0.0715	-0.0787	-0.1337	-0.1344	-0.1539	-0.0560	-0.0588	-0.0828	-0.0638	-0.0637	-0.0744	-0.2266	-0.1814
	750/11×10	0.2392	0.2403	0.2425	0.0569	0.0558	0.0620	0.2475	0.2500	0.2574	0.0594	0.0585	0.0641	0.1634	0.1640	0.1702	0.0508	0.0514	0.0566	0.0465	0.0485	0.0675	0.2825	0.2095
		-0.1642	-0.1738	-0.1726	-0.0187	-0.0169	-0.0207	-0.1920	-0.1955	-0.1984	-0.0256	-0.0245	-0.0280	-0.0822	-0.0824	-0.0851	-0.0032	-0.0040	-0.0129	-0.0147	-0.0156	-0.0257	-0.2012	-0.1635
	995/3×25	0.1030	0.1011	0.1299	0.0693	0.0706	0.0726	0.1016	0.1007	0.2000	0.0684	0.0697	0.0720	0.1334	0.1341	0.1674	0.0879	0.0883	0.0929	0.0651	0.0659	0.0676	0.0799	0.0583
		-0.0456	-0.0449	-0.0522	-0.0031	-0.0022	-0.0009	-0.0482	-0.0486	-0.0705	-0.0035	-0.0027	-0.0016	-0.0617	-0.0630	-0.0731	-0.0080	-0.0095	-0.0120	-0.0008	-0.0016	-0.0025	-0.0199	-0.0018
	995/11×10	0.0419	0.0420	0.0426	0.0264	0.0261	0.0255	0.0408	0.0417	0.0425	0.0249	0.0249	0.0239	0.0593	0.0601	0.0522	0.0406	0.0416	0.0426	0.0220	0.0218	0.0220	0.0296	0.0283
		-0.0173	-0.0187	-0.0192	0.0018	0.0031	0.0040	-0.0224	-0.0227	-0.0235	-0.0005	0.0002	0.0014	-0.0253	-0.0254	-0.0214	-0.0024	-0.0037	-0.0036	0.0006	0.0002	-0.0010	-0.0105	-0.0123
	Summary	**0.2107**	**0.2094**	**0.2178**	**0.1030**	**0.1010**	**0.1109**	**0.2131**	**0.2137**	**0.2357**	**0.1035**	**0.1018**	**0.1115**	**0.2059**	**0.2064**	**0.2269**	**0.1404**	**0.1432**	**0.1584**	**0.0944**	**0.0944**	**0.1037**	**0.2269**	**0.1748**
		-0.1038	**-0.1058**	**-0.1103**	**-0.0231**	**-0.0210**	**-0.0232**	**-0.1153**	**-0.1165**	**-0.1253**	**-0.0261**	**-0.0246**	**-0.0267**	**-0.0757**	**-0.0763**	**-0.0834**	**-0.0174**	**-0.0190**	**-0.0278**	**-0.0197**	**-0.0202**	**-0.0259**	**-0.1145**	**-0.0897**

*outgroup-based rooted tree

†Beast with strict molecular clock and coalescent constant size tree prior

‡Beast with lognormal relaxed molecular clock and coalescent constant size tree prior

Supplementary Table S3. Detailed estimation results with data simulated with a lognormal relaxed molecular clock (RMC). See legend of Figure 2 and Table S2.

	Strict molecular clock (SMC)			Relaxed molecular clock (RMC)			
	PhyML	**FastME**	**BSMC**	**PhyML**	**FastME**	**BSMC**	**BRMC**
750/3×25	0.06	0.12	0.06	0.08	0.13	0.12	0.08
750/11×10	0.08	0.14	0.07	0.10	0.16	0.12	0.10
995/3×25	0.08	0.12	0.08	0.09	0.14	0.10	0.09
995/11×10	0.17	0.19	0.16	0.19	0.20	0.18	0.18
Summary	**0.10**	**0.14**	**0.09**	**0.11**	**0.16**	**0.14**	**0.11**

Supplementary Table S4. Topological errors with simulated data sets. This table displays the Robinson and Foulds topological distance between the true and inferred trees, for the various tree building methods and tree models. This distance is equal to the number of bipartitions (branches) that are in one tree but not in the other, divided by the total number of bipartitions; it is equal to 0.0 when both trees are identical, and 1.0 when they do not share any bipartition in common. Results are averaged over the 100 data sets corresponding to each tree model. BSMC: BEAST with strict molecular clock model; BRMC: BEAST with lognormal relaxed molecular clock. As expected likelihood-based methods (PhyML and BEAST) have a better topological accuracy than (distance-based) FastME. BEAST (incorporating a tree model constraining branch lengths) is slightly more accurate than PhyML.

Country	2009	2010	2011	Total	Country	2009	2010	2011	Total
Argentina	17			17	Korea Republic of	2			2
Australia	30	7	6	43	Kuwait	1			1
Austria	5	1		6	Kyrgyzstan	4			4
Bangladesh	2			2	Lao Peoples Democratic Republic	15			15
Belgium	11			11	Luxembourg	1			1
Bhutan	1			1	Macao		2		2
Bolivia	1			1	Malaysia	5			5
Brazil	1			1	Mali	1			1
Brunei			1	1	Mexico	43			43
Canada	126	3		129	Mongolia	7			7
Chile	4			4	Myanmar	4			4
China	66	18	19	103	Nepal	1			1
Colombia	2			2	Netherlands	21	2		23
Costa Rica	3			3	New Zealand	1	2		3
Cote dIvoire	1			1	Nicaragua	26			26
Czech Republic	2		1	3	Norway	7			7
Denmark	18			18	Papua New Guinea		1		1
Dominican Republic	9			9	Peru	1			1
Ecuador	1			1	Philippines	1			1
Egypt	1			1	Poland	3			3
El Salvador	7			7	Puerto Rico	2			2
Estonia	2	2		4	Russian Federation	27	5	9	41
Ethiopia	2	1		3	Saudi Arabia	1			1
Finland	3			3	Seychelles	1			1
France	4			4	Singapore	31	4		35
Germany	21	4		25	Spain	18			18
Greece	13	14		27	Suriname		1		1
Guam	19	1		20	Taiwan Province of China	25	4	10	39
Haiti	1			1	Thailand	7	3	1	11
Hong Kong	5			5	Turkey	1			1
India	1			1	Turkmenistan	1			1
Iraq	2			2	Uganda	1			1
Ireland	6			6	United Kingdom	54			54
Italy	5			5	United States	219	43	16	278
Japan	84			84	Uruguay	1			1
Jordan	3			3					
Kenya	1			1	**Total**	**1,013**	**118**	**63**	**1,194**

Supplementary Table S5. Origins and dates of influenza A H1N1pdm09 strains.