# Smarter dating and faster proposals: revisiting the phylogenetic relaxed clock model

Jordan Douglas[1,2*], Rong Zhang[1,2], Alexei J. Drummond[1,2,3], Remco Bouckaert[1,2]

**1** Centre for Computational Evolution, University of Auckland, Auckland, New Zealand
**2** School of Computer Science, University of Auckland, Auckland, New Zealand
**3** School of Biological Sciences, University of Auckland, Auckland, New Zealand

* jordan.douglas@auckland.ac.nz

## Abstract

## Author summary

## Introduction

The molecular clock hypothesis states that the evolutionary rates of biological sequences are approximately constant throughout time [1]. This assumption forms the basis of phylogenetics, under which evolutionary trees and divergence dates of life forms are inferred from biological sequences, such as nucleotides and amino acids [2,3]. In Bayesian phylogenetics, these trees and their associated parameters are estimated as probabilitiy distributions [4–6]. Statistical inference is typically performed by sampling under the Metropolis-Hastings Markov chain Monte Carlo (MCMC) algorithm [7,8]. Bayesian phylogenetic clock models have been implemented in many platforms including BEAST [9], BEAST2 [10], MrBayes [11], and RevBayes [12].

The simplest phylogenetic clock model – the strick clock – makes the convenient assumption that the evolutionary rate is constant across all lineages [4,5,13]. However, molecular substitution rates are known to vary over time, over population sizes, over evolutionary pressures, and over DNA replicative machineries [14–16]. Moreover, any given dataset may be clock-like (where the substitution rate has a small variance across lineages) or non clock-like (a large variance). In the latter case, a strict clock is probably not suitable.

This led to the development of relaxed (uncorrelated) clock models, under which each branch in the phylogenetic tree has its own substitution rate [3]. Branch rates can be drawn from a range of probability distributions including log-normal, exponential, gamma, and inverse-gamma distributions [3,17,18]. This class of models is widely used, and has aided insight into many recent biological problems, including the 2016 Zika virus outbreak [19] and the COVID-19 pandemic [20].

Finally, although not the focus of this article, the class of correlated clock models assumes some form of auto-correlation between rates over time. The correlation itself can invoke a range of stochastic models, including compound Poisson [21] and CIR processes [17], or it can exist as a series of local relaxed clocks [22]. However, due to the correlated and discrete nature of such models, the time until MCMC convergence may be cumbersome, particularly for larger datasets [22].

With the increasing availabilty of biological sequence data, the development of efficient Bayesian phylogenetic methods is more important than ever. The performance

of MCMC is dependent not only on computational runtime but also the efficacy of an MCMC setup to achieve its convergence. A critical task therein lies the further advancement of MCMC operators. Recent developments in this area include the advancement of guided tree proposals [23–25], coupled MCMC [26, 27], adaptive multivariate transition kernels [28], and other explorative proposal kernels [29, 30]. In the case of clock models, smart tree proposals can account for correlations between substitution rates and divergence times [31]. The rate parameterisation itself can also affect the ability to "mix" during MCMC [3, 18, 31].

While a range of advanced operators and other MCMC optimisation methods have arisen over the years, there has yet to be a widescale performance benchmarking of such methods as applied to the relaxed clock model. In this article, we systematically evaluate how the relaxed clock model can benefit from i) different substitution rate parameterisations, ii) the use of Bactrian proposal kernels [29], iii) tree operators which account for correlations between substitution rates and times, and iv) adaptive multivariate operators [28]. The discussed methods are implemented in and compared using BEAST2 [10].

# Models

## Preliminaries

Let $\mathcal{T}$ be a binary rooted time tree with $N$ taxa. Let $L$ be the number of sites within the multiple sequence alignment $D$, and let $L_{\text{eff}}$ be the effective number of sites in the alignment (i.e. the number of unique patterns across all sites in the alignment). The posterior density of a phylogenetic model is described by

$$p(\mathcal{T}, \vec{\mathcal{R}}\,, \sigma, \theta|D) \propto p(D|\mathcal{T}, r(\vec{\mathcal{R}}\,), \theta)\; p(\mathcal{T}|\theta)\; p(\vec{\mathcal{R}}\,|\sigma)\; p(\sigma)\; p(\theta), \tag{1}$$

for substitution rate standard deviation $\sigma$ and other model parameters $\theta$. $\vec{\mathcal{R}}$ is a vector of abstracted substitution rates, which is transformed into real rates by $r(\vec{\mathcal{R}}\,)$. Three methods of representing rates as $\vec{\mathcal{R}}$ are presented in **Substitution rate parameterisations**.

Let $t_i$ be the height (time) of node $i$. Every internal and leaf node $i$ in $\mathcal{T}$ is associated with a parental branch length (the height difference between $i$ and its parent) $\tau_i$ and a parental branch substitution rate $r_i = r(\mathcal{R}_i)$. There are a total of $2N - 2$ internal and leaf nodes. Under the relaxed clock model, each $\mathcal{R}_i$ is independently distributed under the prior.

The posterior distribution is sampled by MCMC. The probability of accepting proposed state $x'$ from state $x$ is calculated using the Metropolis-Hastings ratio [7, 8]:

$$\alpha(x'|x) = \min\left(1, \frac{p(x'|D)}{p(x|D)} \frac{q(x|x')}{q(x'|x)}\right). \tag{2}$$

$q(a|b)$ is the transition kernel: the probability of proposing state $b$ from state $a$. The ratio between the two $\frac{q(x|x')}{q(x'|x)}$ is known as the Hastings ratio [8].

## Substitution rate parameterisations

In Bayesian inference, the way parameters are represented in the model can affect the mixing ability of the model and the meaning of the model itself [32]. Three methods for parameterising substitution rates are described below, and are later evaluated in

**Results**. Each parameterisation technique is associated with i) an abstraction of the rates $\vec{\mathcal{R}}$, ii) some function for transforming this parameter into real rates $r(\vec{\mathcal{R}})$, and iii) a prior density function of the abstraction $p(\vec{\mathcal{R}}|\sigma)$. The three methods are summarised in **Fig 1**.

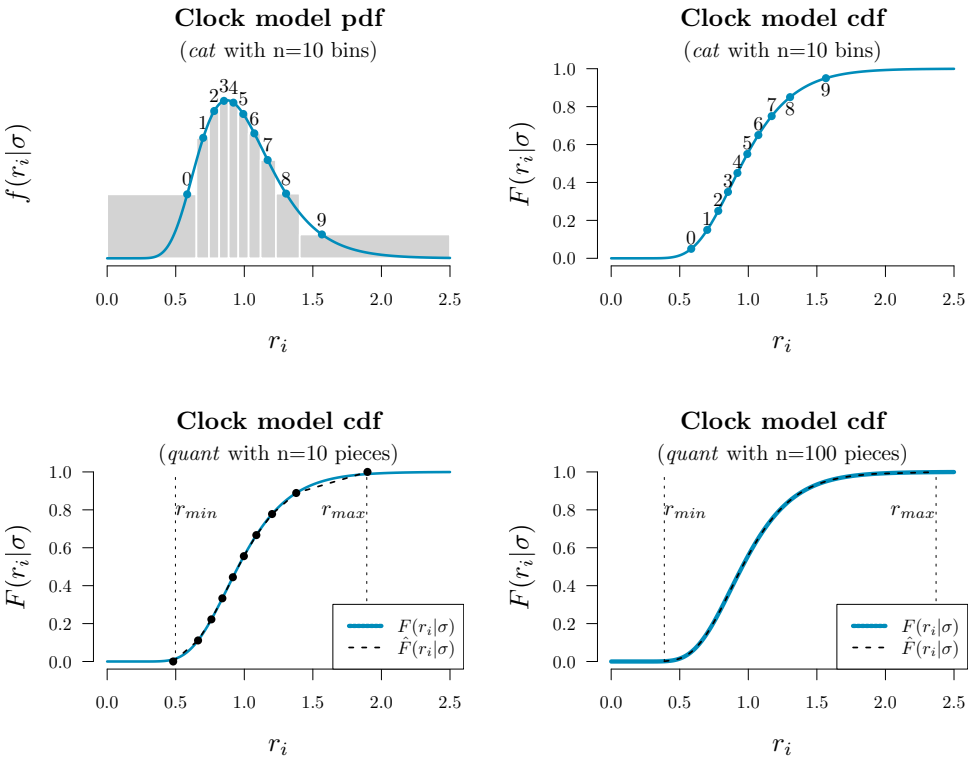


**Fig 1. Methods of rate parameterisation.** The *cat* and *quant* approximations are plotted on top of the true underlying rate prior distribution (*real*). In this example, rates are drawn from a LogNormal($\mu = -0.045, \sigma = 0.3$) distribution. The probability density function (pdf) and cumulative density function (cdf) of this distribution are shown.

### 1. Real rates

The natural (and unabstracted) parameterisation of a substitution rate is a real number $\mathcal{R}_i \in \mathbb{R}, \mathcal{R}_i > 0$ which is equal to the rate itself. Thus, under the *real* parameterisation:

$$r(\vec{\mathcal{R}}) = \vec{\mathcal{R}}\,. \tag{3}$$

Under the prior distribution $p(\vec{\mathcal{R}}\,|\sigma)$, rates are log-normally distributed with a mean of 1:

$$p(\mathcal{R}_i|\sigma) = \frac{1}{\mathcal{R}_i\sigma\sqrt{2\pi}} \exp\big(-\frac{(\ln \mathcal{R}_i - \mu)^2}{2\sigma^2}\big) \tag{4}$$

where $\mu = -0.5\sigma^2$ is set such that the expected value of the log-normal distribution is 1. In this article we only consider log-normal clock priors, however the methods discussed are general.

Zhang and Drummond 2020 present a series of tree operators which propose internal/root node heights, and then recompute the rates of incident branches such that their genetic distances ($r_i \times \tau_i$) remain constant after the proposal [31]. These operators account for the correlation which exists between branch rates and branch times. By keeping the genetic distance of each branch constant, the likelihood is unaltered by the proposal.

### 2. Categories

The category parameterisation (*cat*) is an abstraction of the *real* parameterisation. Each branch is assigned an integer from 0 to $n-1$:

$$\vec{\mathcal{R}} \in \{0, 1, \ldots, n-1\}^{2N-2}. \tag{5}$$

These integers correspond to $n$ rate categories. The domain of $\vec{\mathcal{R}}$ is uniformly distributed:

$$p(\mathcal{R}_i|\sigma) = p(\mathcal{R}_i) = \frac{1}{n}. \tag{6}$$

Let $f(x|\sigma)$ be the probability density function (pdf) and let $F(x|\sigma) = \int_0^x f(t|\sigma)\,dt$ be the cumulative distribution function (cdf) of the prior distribution used by the underlying *real* clock model. Then, in the *cat* parameterisation, $f(x|\sigma)$ is discretised into $n$ bins and the elements of $\vec{\mathcal{R}}$ each point to one of these bins. Each bin contains uniform probability density $1/n$. The rate of each bin is equal to the median value within the bin

$$r(\mathcal{R}_i) = F^{-1}\big(\frac{\mathcal{R}_i + 0.5}{n}\big), \tag{7}$$

where $F^{-1}$ is the inverse cumulative distribution function (i-cdf).

The key advantage of the *cat* parameterisation is the removal of a term from the posterior density (Equation 1), or more accurately the replacement of a non-trivial

$p(\vec{\mathcal{R}}\,|\sigma)$ term with that of a uniform prior. Thus, one fewer term needs to be estimated per rate.

This method has been widely used in BEAST and BEAST2 analyses [3]. However, the recently developed constant distance operators – which are incompatible with the *cat* parameterisation – yield an increase in mixing rate under *real* by up to an order of magnitude over that of *cat* [31].

### 3. Quantiles

Finally, rates can be parameterised as real numbers $0 < \mathcal{R}_i < 1$ which describe the rate's quantile with respect to some underlying clock model distribution. Under the *quant* parameterisation, each element in $\vec{\mathcal{R}}$ is uniformly distributed.

$$\vec{\mathcal{R}} \in \mathbb{R}^{2N-2}, 0 < \mathcal{R}_i < 1 \tag{8}$$

$$p(\mathcal{R}_i|\sigma) = p(\mathcal{R}_i) = 1 \tag{9}$$

Transforming these quantiles into rates invokes the i-cdf of the underlying *real* clock model distribution. Thus, while this approach has clear similarities with *cat*, the domain of rates here is continuous (as opposed to being confined to a discrete number of bins) and is therefore compatible with constant distance operators [31].

A potential disadvantage of the *quant* method would be the computational requirements of continuously evaluating the i-cdf. As the number of quantiles grows with the taxon count $N$, this drawback would be at its worst on large trees. Hence, rather than evaluating the exact i-cdf $F^{-1}$, an approximation $\hat{F}^{-1}$ will be used instead:

$$r(\mathcal{R}_i) = \hat{F}^{-1}(\mathcal{R}_i). \tag{10}$$

In this article we have extended *quant* through a linear piecewise approximation of the i-cdf. As the piecewise approximation is linear, evaluating the derivatives $\frac{\partial}{\partial \mathcal{R}_i}\hat{F}^{-1}(\mathcal{R}_i) = D\hat{F}^{-1}(\mathcal{R}_i)$ and $\frac{\partial}{\partial r_i}\hat{F}(r_i) = D\hat{F}(r_i)$ – which are required for computing the Hastings ratios – is trivial. The approximation is comprised of $n$ pieces (where $n$ is fixed) and upper and lower rate boundaries $r_{\min}$ and $r_{\max}$. The approximation is displayed in **Fig 1** and further detailed in **S1 Appendix**.

Zhang and Drummond 2020 introduced several tree operators for the *real* parameterisation – including `Constant Distance`, `Simple Distance`, and `Small Pulley`. In this project, we extended these three operators so that they are compatible with the *quant* parameterisation. These are presented in **S1 Appendix**.

## Adaptive operator sampling

Throughout the MCMC chain, `AdaptiveOperatorSampler` goes through three phases. In the first phase (burn-in), `AdaptiveOperatorSampler` samples operators uniformly at random. In the second phase (learn-in), the operator continues to sample operators uniformly at random however it starts learning several terms related to the operators it samples from and the numeric parameters they operate on. In the third phase, `AdaptiveOperatorSampler` makes each proposal first by generating a random number. With probability 0.01, each $\omega_i$ is sampled uniformly at random. With probability 0.99, operator $\omega_i$ is sampled with probabiltiy:

$$p(\omega_i) \propto \frac{1}{n_i \mathbb{T}_i} \sum_{p \in \text{params}} \frac{1}{\sigma_p^2} \Big[ \sum_{x \in \text{accepts}_i} (x_p - x_p')^2 \Big]. \tag{11}$$

The terms which are learned during the second and third phases are: the number of proposals $n_i$ made by each operator $\omega_i$ and its the average computational runtime $\mathbb{T}_i$, the sample variance $\sigma_p^2$ of each parameter $p$, and the sum-of-squares $\sum_{x \in \text{accepts}_i} (x_p - x'_p)^2$, where $x_p$ and $x'_p$ are the values of $p$ before and after each acceptance of a proposal made by $\omega_i$.

Under Equation 11, operators which effect larger changes on the parameters of interest, in shorter runtime, are sampled with greater probabilities. Division of the sum-of-squares term by the parameter variance $\sigma_p^2$ enables comparison between multiple parameters.

## Bactrian proposal kernel

The step size of a proposal kernal $q(x'|x)$ should be such that the proposed state $x'$ is sufficiently far from the current state $x$ to explore vast areas of parameter space, but not so large that the proposal is rejected too often [33]. Yang et al. have challenged the widely used uniform proposal kernel in place of the Bactrian kernel [29, 30]. The Bactrian($m$) distribution is defined as the sum of two Normal distributions:

$$\Sigma \sim \text{Bactrian}(m) \equiv \frac{1}{2}\text{Normal}(-m, 1-m^2) + \frac{1}{2}\text{Normal}(m, 1-m^2) \qquad (12)$$

where $0 \leq m < 1$ describes the modality of the Bactrian distribution. When $m = 0$, the Bactrian distribution is equivalent to a Normal$(0, 1)$ distribution. As $m \to 1$, the distribution becomes increasingly bimodal (**Fig. 2**). Yang et al. 2013 [29] suggest that Bactrian($m = 0.95$) yields a proposal kernel superior to the uniform kernel, by placing minimal probability on steps which are too small or too large.

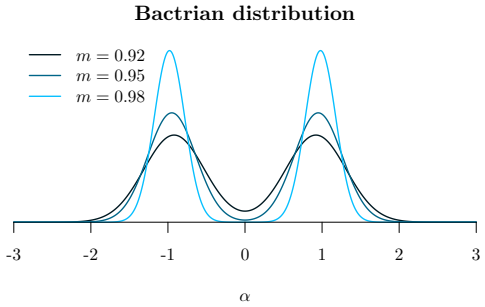


**Fig 2. The Bactrian proposal kernel.** Y-axis corresponds to probability density $f(\Sigma|m)$.

In this article we compare the performance of uniform and Bactrian proposal kernels in the clock model. Two Bactrian distributions are compared($m = 0.95$ and $m = 0.98$). The clock model operators which these proposal kernels apply to are described in **Table 1**.

| | Operator | Proposal | Parameters |
|---|---|---|---|
| 1 | `Random walk` operator | $x' \leftarrow x + s\Sigma$ | $\sigma, r, q$ |
| 2 | `Scale` operator | $x' \leftarrow x \times e^{s\Sigma}$ | $\sigma, r$ |
| 3 | `Interval` operator | $y \leftarrow \frac{u-x}{x-l} \times e^{s\Sigma}$<br>$x' \leftarrow \frac{u+l*y}{y+1}$ | $q \ \ (l=0, u=1)$ |
| 4 | `Constant distance` operator | $x' \leftarrow x + s\Sigma$ | $t$ |

**Table 1.** Summary of proposal kernels $q(x'|x)$ of clock model operators. `Random walk`, `Scale`, and `Interval` are standard BEAST2 operators. In each operator, $\Sigma$ is drawn from either a Bactrian($m$) or Uniform distribution (distributions are normalised so that they have a mean of 0 and a variance of 1). The scale size $s$ is tunable. The proposal kernel may apply to node heights $t$, clock standard deviation $\sigma$, clock rates $r$ (*real* only), and clock rate quantiles $q$ (*quant* only). The Scale operator acts on parameters with non-negative domains. The Interval operator proposes values which respect the domain of the parameter i.e. $l < x, x' < u$.

## Narrow Exchange Rate

The `Narrow Exchange` operator [34], widely used in BEAST [9, 35] and BEAST2 [10], is similar to NNI, and works as follows (**Fig. 3**):

*Step 1*. Sample an internal/root node $E$ from tree $\mathcal{T}$, where $E$ has grandchildren.

*Step 2*. Identify the child of $E$ with the greater height. Denote this child as $D$ and its sibling as $C$ (i.e. $t_D > t_C$).

*Step 3*. Randomly identify the two children of $D$ as $A$ and $B$.

*Step 4*. Relocate the $B - D$ branch onto the $C - E$ branch, so that $B$ and $C$ become siblings and their parent is $D$. All node heights are unchanged.

Lakner et al. 2008 [36] found that tree operators which perterb topology (such as NNI and SPR) consistently perform better than those which also change branch lengths (such as LOCAL [37] and Continuous Change [38]). If `Narrow Exchange` was adapted to the relaxed clock model by ensuring that genetic distances remain constant after the proposal, its performance may be improved even further. This may in turn permit proposing a new node height $t_D$ and therefore changing branch (time) lengths.

Here, we present the `Narrow Exchange Rate` (NER) operator. Let $r_A$, $r_B$, $r_C$, and $r_D$ be the clock rates of nodes $A$, $B$, $C$, and $D$, respectively. In addition to the modest topological change applied by `Narrow Exchange`, NER also proposes new clock rates ${r_A}'$, ${r_B}'$, ${r_C}'$, and ${r_D}'$. While NER does not alter $t_D$ (i.e. ${t_D}' \leftarrow t_D$), we also consider NERw - a special case of the NER operator which embarks $t_D$ on a random walk:

$$ {t_D}' \leftarrow t_D + s\Sigma \tag{13} $$

for random walk step size $s\Sigma$ where $s$ is a tunable scaler parameter and $\Sigma$ is drawn from a uniform or **Bactrian proposal kernel**. NER (and NERw) are compatible with both the *real* and *quant* parameterisations. Analagous to the `Constant Distance` operator, new rates are proposed such that genetic distances between nodes $A$, $B$, $C$, and $E$ are maintained. Thus, there are $\binom{4}{2} = 6$ pairwise distance constraints.

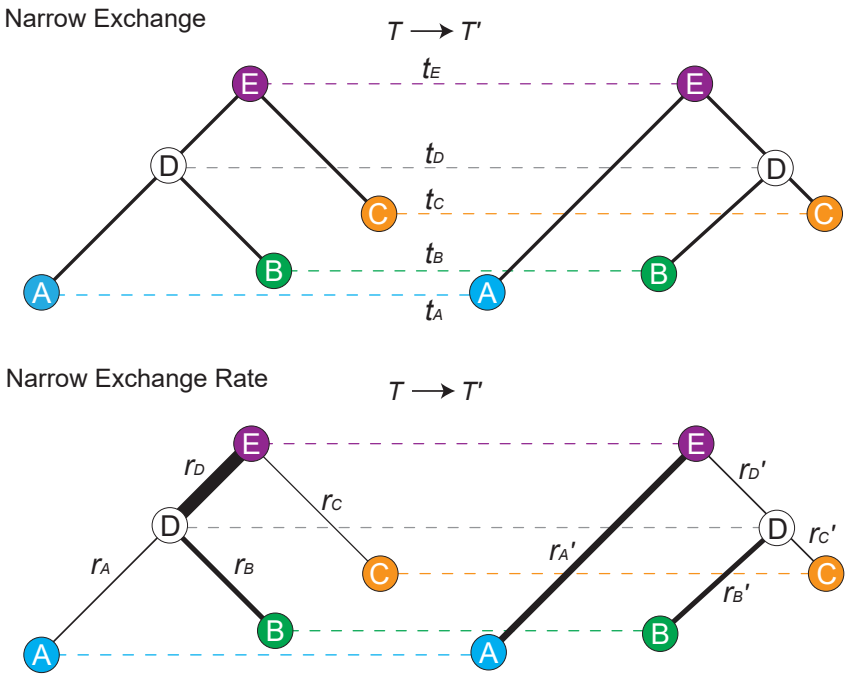


**Fig 3. Depiction of `Narrow Exchange` and `Narrow Exchange Rate` operators.** Proposals are denoted by $\mathcal{T} \to \mathcal{T}'$. The vertical axis corresponds to node height $t$. In the bottom figure, branch rates $r$ are indicated by line thickness. In this example, the $\mathcal{D}_{AE}$ and $\mathcal{D}_{CE}$ constraints are satisfied.

$$\begin{aligned}
\mathcal{D}_{AB}: \quad & r_A(t_D - t_A) + r_B(t_D - t_B) = \\
& r_A'(t_E - t_A) + r_D'(t_E - t_D') + r_B'(t_D' - t_B) && (14)\\
\mathcal{D}_{AC}: \quad & r_A(t_D - t_A) + r_D(t_E - t_D) + r_C(t_E - t_C) = \\
& r_A'(t_E - t_A) + r_D'(t_E - t_D') + r_C'(t_D' - t_C) && (15)\\
\mathcal{D}_{AE}: \quad & r_A(t_D - t_A) + r_D(t_E - t_D) = \\
& r_A'(t_E - t_A) && (16)\\
\mathcal{D}_{BC}: \quad & r_B(t_D - t_B) + r_D(t_E - t_D) + r_C(t_E - t_D) = \\
& r_B'(t_D' - t_B) + r_C'(t_D' - t_C) && (17)\\
\mathcal{D}_{BE}: \quad & r_B(t_D - t_B) + r_D(t_E - t_D) = \\
& r_B'(t_D' - t_B) + r_D'(t_E - t_D') && (18)\\
\mathcal{D}_{CE}: \quad & r_C(t_E - t_C) = \\
& r_C'(t_D' - t_C) + r_D'(t_E - t_D') && (19)
\end{aligned}$$

Further constraints are imposed by the model itself:

$$r_i > 0 \text{ and } r_i' > 0 \text{ for } i \in \{A, B, C, D\} \quad (20)$$

$$\max\{t_B, t_C\} < t_D' < t_E. \quad (21)$$

Unfortunately, it is not possible to solve all six $\mathcal{D}_{ij}$ constraints without permitting non-positive rates or illegal trees. Therefore rather than conserving all six pairwise

distances, NER conserves a *subset* of distances. It is not immediately clear which distances should be conserved.

### Automated generation of operators and constraint satisfaction

The total space of NER operators is comprised of all possible subsets of distance constraints (i.e. $\{\}, \{\mathcal{D}_{AB}\}, \{\mathcal{D}_{AC}\}, \ldots, \{\mathcal{D}_{AB}, \mathcal{D}_{AC}, \mathcal{D}_{AE}, \mathcal{D}_{BC}, \mathcal{D}_{BE}, \mathcal{D}_{CE}\}$) which are solvable. The simplest NER – the null operator denoted by NER$\{\}$ – does not satisfy any distance constraints. This is equivalent to `Narrow Exchange`.

As it is unclear which NER variants would perform the best, we developed an automated pipeline for generating and testing these operators.

**1. Solution finding.** Using standard analytical linear-system solving libraries in MATLAB, the $2^6 = 64$ subsets of distance constraints are solved. 54 out of the 64 subsets were found to be solvable, and the unsolvables were discarded.

**2. Solving Jacobian determinants.** The determinant of the Jacobian matrix $J$ is required for computng the Hastings ratio of the proposal. $J$ is defined as

$$J = \begin{bmatrix} \frac{\partial r_A{}'}{\partial r_A} & \frac{\partial r_A{}'}{\partial r_B} & \frac{\partial r_A{}'}{\partial r_C} & \frac{\partial r_A{}'}{\partial r_D} \\ \frac{\partial r_B{}'}{\partial r_A} & \frac{\partial r_B{}'}{\partial r_B} & \frac{\partial r_B{}'}{\partial r_C} & \frac{\partial r_B{}'}{\partial r_D} \\ \frac{\partial r_C{}'}{\partial r_A} & \frac{\partial r_C{}'}{\partial r_B} & \frac{\partial r_C{}'}{\partial r_C} & \frac{\partial r_C{}'}{\partial r_D} \\ \frac{\partial r_D{}'}{\partial r_A} & \frac{\partial r_D{}'}{\partial r_B} & \frac{\partial r_D{}'}{\partial r_C} & \frac{\partial r_D{}'}{\partial r_D} \end{bmatrix}. \tag{22}$$

Computing the determinant $|J|$ invokes standard analytical differentiation and linear algebra libraries of MATLAB. 6 of the 54 solvable operators were found to have $|J| = 0$, corresponding to irreversible proposals, and were discarded.

**3. Automated generation of BEAST2 operators.** Java class files are generated using string processing. Each class corresponds to a single operator, extends the class of a meta-NER-operator, and is comprised of the solutions found in **1** and the Jacobian determinant found in **2**. $|J|$ is further augmented if the *quant* parameterisation is employed (**S1 Appendix**).

The 48 operators generated by this pipeline are evaluated and compared in **Results**. Each operator is considered with and without a random walk on $t_D$ and thus there are 96 total settings.

## A guided adaptive leaf rate operator

A *guided* operator incorporates knowledge about neighbouring states, while an *adaptive* operator undergoes a training process to improve its efficiency over time [39]. In previous work, parsimony scores and conditional clade probabilities of neighbouring trees have been employed by guided tree operators [23–25] and the latter has also been explored as the basis of adaptive tree operators [24, 25]. The (adaptive) mirror kernel [30] learns a target distribution which acts as a 'mirror image' of the current point $x$. The adaptable variance multivariate normal (AVMVN) kernel [28, 35] learns correlations between parameters during MCMC. Baele et al. 2017 observed a large increase ($\approx 5 - 10\times$) in sampling efficiency from using the AVMVN kernel on clock rates and substitution model parameters across partitions [28].

In this article we consider application of the AVMVN kernel to the branch rates of leaf nodes. This operator is not readily applicable to internal node branch rates due to their dependencies on tree topology.

### AVMVN kernel

The AVMVN kernel assumes its parameters live in $x \in \mathbb{R}^N$ and that these parameters follow a multivariate normal distribution with covariance matrix $\Sigma_N$. Hence, the kernel operates on the logarithmic or logistic transformation of the $N$ leaf branch rates, depending on the rate parameterisation:

$$x_i = \begin{cases} \log r_i \text{ for } \textit{real} \\ \log \frac{q_i}{1-q_i} \text{ for } \textit{quant} \end{cases} \tag{23}$$

where $r_i$ is a real rate and $q_i$ is a rate quantile. The AVMVN probability density is defined by

$$\mathcal{AVMVN}(x) = \mathcal{MVN}\big(x, (1-\beta)\frac{\Sigma_N}{N} + \beta\frac{\mathbb{I}_N}{N}\big), \tag{24}$$

where $\mathcal{MVN}$ is the multivariate normal probability density. $\beta$ $(= 0.05)$ is a constant which determines the fraction of the proposal determined by the identity matrix $\mathbb{I}_N$, as opposed to the covariance matrix $\Sigma_D$ which is trained during MCMC.

The AVMVN proposal kernel is computed as

$$x' \leftarrow x + \sum_{i=1}^{N}\sum_{j=i}^{N} c_{i,j} \times s\Sigma \tag{25}$$

$$\text{where } c = \text{cholesky}\left((1-\beta)\frac{\Sigma_N}{N} + \beta\frac{\mathbb{I}_N}{N}\right). \tag{26}$$

The cholesky$(Y)$ decomposition returns a lower diagonal matrix $L$, with positive real diagonal entries, such that $Y = LL'$ [40, 41]. $s$ is a tunable step size parameter and $\Sigma$ is a random variable drawn from a proposal kernel (uniform or Bactrian for instance). Our BEAST2 implementation of the AVMVN kernel is adapted from that of BEAST [35].

In Results, we evaluate this operator for its ability to estimate leaf rates. As the size of the covariance matrix $\Sigma_N$ grows with the number of taxa $N$, AVMVN is hypothesised to work well on small trees but become less efficient with larger taxon sets.

# Results

## Assessment criteria and datasets

To avoid a cross-product explosion, the five targets for clock model improvement are evaluated sequentially in the following order: **Adaptive operator sampling** **Substitution rate parameterisations**, **Bactrian proposal kernel**, **Narrow Exchange Rate**, and **A guided adaptive leaf rate operator**.

The setting which is considered to be the best in each step is then incorporated into the following step. This protocol and its outcomes are summarised in **Fig. 4**.

Methodologies are assessed according to the following criteria.

**1. Validation**. This is assessed by measuring the coverage of all estimated parameters in a well-calibrated simulation study, using 100 simulated datasets (with

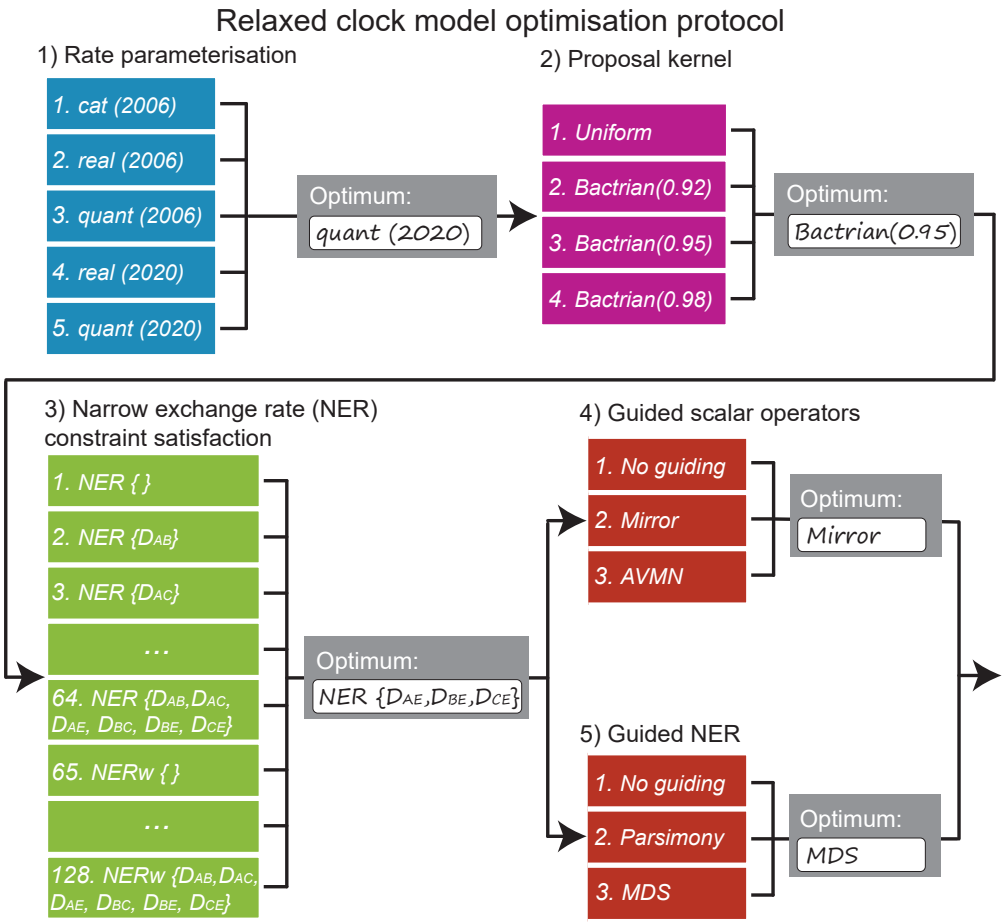


**Fig 4. Protocol for optimising methodology settings.** The three areas (detailed in **Models**) are optimised sequentially, where the best setting from each step is used when optimising the following step.

$N = 100$ taxa and $L = 5000$ nucleotide alignments). These are presented in **S2 Appendix**.

**2. Mixing of parameters**. Key parameters are evaluated for the number of effective samples generated per hour (ESS/hr). This calculation is performed by BEAST2 [10].

Methodologies are benchmarked using one simulated and nine empirical datasets – the latter were compiled [42] and partitioned [43] by Lanfear et al. as 'benchmark alignments' (**Table 2**). To facilitate convergence on complex datasets, each posterior distribution is sampled using coupled MCMC ($MC^3$) with 4 chains [27]. Each methodology is benchmarked across 5 replicates on each dataset, using an the Intel Xeon Gold 6138 CPU (2.00 GHz).

## Evaluating constant distance operators

## Comparison of rate parameterisations

We compared the three rate parameterisations described in **Substitution rate parameterisations**. All three settings use the standard BEAST2 clock model operators from Drummond et al. 2006 [3]. *real* and *quant* additionally use the

| | $N$ | $P$ | $L$ (kb) | $L_{\text{eff}}$ (kb) | **Description** |
|---|---|---|---|---|---|
| 1 | 38 | 8 | 9.1 | 6.45 | Seed plants (Ran 2018 [44]) |
| 2 | 44 | 7 | 5.9 | 1.8 | Squirrel Fishes (Dornburg 2012 [45]) |
| 3 | 44 | 3 | 1.9 | 0.8 | Bark beetles (Cognato 2001 [46]) |
| 4 | 51 | 6 | 5.4 | 1.8 | Southern beeches (Sauquet 2011 [47]) |
| 5 | 61 | 8 | 6.9 | 4.3 | Bony fishes (Broughton 2013 [48]) |
| 6 | 70 | 3 | 2.2 | 0.9 | Caterpillars (Kawahara 2013 [49]) |
| 7 | 78 | 8 | 3.4 | 3.1 | Animals (Cannon 2016 [50]) |
| 8 | 80 | 1 | 10.0 | 4.2 | *Simulated data* |
| 9 | 94 | 4 | 2.2 | 1 | Bees (Rightmyer 2013 [51]) |
| 10 | 106 | 1 | 0.8 | 0.5 | Songbirds (Moyle 2016 [52]) |

**Table 2.** Datasets used during benchmarking, sorted in increasing order of taxa count $N$. Number of partitions $P$, total alignment length $L$, and number of patterns $L_{\text{eff}}$ are also specified.

constant-distance tree operators [31]. To determine whether the difference in performance between *real/quant* versus *cat* is because of the constant-distance tree operators or the parameterisation itself, we also included benchmarked two additional settings: *real 2006* and *quant 2006*, which do not used the constant-distance operators. These five settings are validated in **S2 Appendix**.

## Comparison of Bactrian and uniform proposal kernels on the clock model

## Comparison of NER variants

The **Narrow Exchange Rate** (NER) operators are evaluated. This protocol selects the best among 48 NER (no random walk) and 48 NERw (Bactrian(0.95) random walk) operators, and has two phases. First, the best of the 96 is selected by comparing operator acceptance rates on simulated data. Second, the selected operator is benchmarked with respect to convergence time and sampling rate on real data (**Table 2**). The analyses in this section invoke the *quant* parameterisation and Bactrian(0.95) proposal kernels on clock model parameters.

### Initial screening by acceptance rate

We selected the best operator variant by performing MCMC on 300 simulated datasets, where each MCMC employed all 96 NER/NERw variants. Simulated datasets have $N = 30$ taxa and an alignment with $L \sim \text{Uniform}(10^2, 10^4)$ sites. The acceptance rate of each operator is compared to that of the null operator NER{} (i.e. `Narrow Exchange`).

**Fig. 6** shows that NER variants which satisfy the genetic distances between nodes $B$ and $A$ (i.e. $\mathcal{D}_{AB}$) or between $B$ and $C$ (i.e. $\mathcal{D}_{BC}$) usually perform worse than the standard `Narrow Exchange` operator, where $B$ is the node being interchanged from the $A$ branch to the $C$ branch (**Fig. 3**). This is an intuitive result. If the posterior distribution is relatively flat, and the data presents high uncertainty in the positioning of $B$, with respect to $A$ and $C$, then the topological rearrangement performed by `Narrow Exchange` will be favoured. However, this uncertainty in the *topology* is likely coupled with uncertainty in the *distance* between $B$ and $A$ or between $B$ and $C$. Thus,

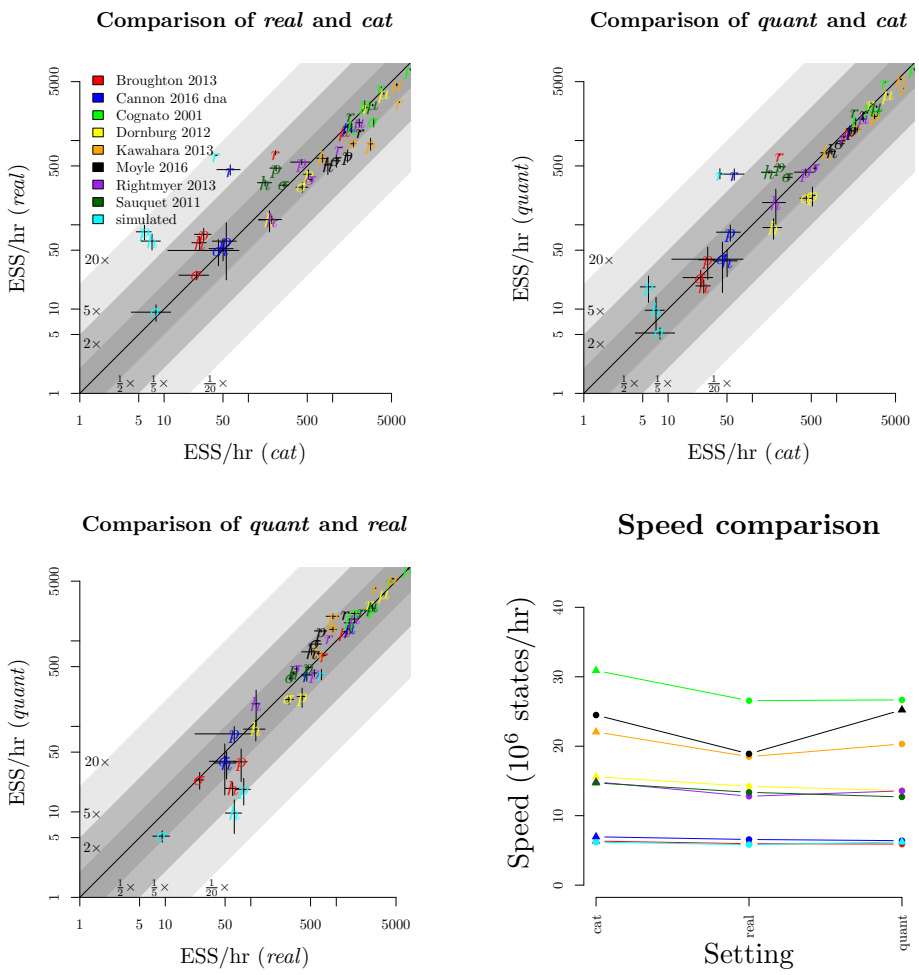


**Fig 5. Rate parameterisation performance evaluation.** Comparison of ESS/hr (averaged across five independent MC3 analyses) with respect to relevant terms – $P$: posterior density; $L$: likelihood, $p$: prior density, $r$: clock rate ESS averaged across all leaves, $\hat{r}$: branch rate mean, $v$: branch rate variance, $\sigma$: clock standard deviation, $\kappa$: HKY model transition-transversion ratio, $\lambda$: Yule model birth rate. $h$: tree height. Datasets are displayed in **Table 2**.

in this case, respecting the $\mathcal{D}_{AB}$ and $\mathcal{D}_{BC}$ constraints (by proposing branch rates) makes too many unnecessary changes to the state and the operator performs worse.

**Fig. 6** also reveals a cluster of NER variants which – under the conditions of the simulation – performed better than the null operator NER{} around 25% of the time and performed worse around 10% of the time. One such operator is NER$\{\mathcal{D}_{AE}, \mathcal{D}_{BE}, \mathcal{D}_{CE}\}$. This variant conserves the genetic distance between all child nodes $A$, $B$, and $C$, and the grandparent node $E$. This is performed by proposing rates for $r_A$, $r_B$, and $r_C$ while obeying the distance constraints imposed by the operator. Exploring this operator further, we can see that NER$\{\mathcal{D}_{AE}, \mathcal{D}_{BE}, \mathcal{D}_{CE}\}$ is at its best when there is a large variance in branch rate i.e. when clock standard deviation $\sigma$ is high ($\sigma \gtrsim 0.5$ for $N = 30$), corresponding to data which is not clock-like. On the other hand, NER{} is much preferred when the operator's acceptance rate is high ($\gtrsim 0.15$) – corresponding with datasets with a small number of site patterns ($L_{eff} \lesssim 500$ for

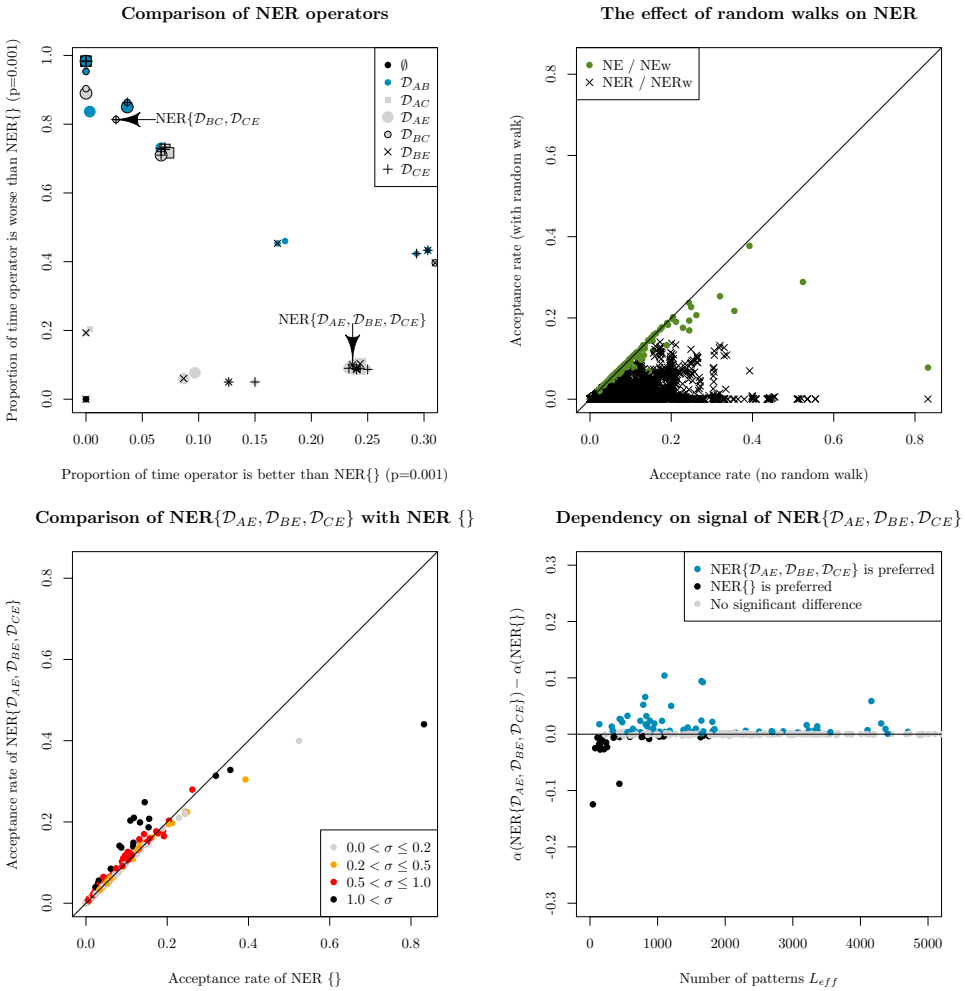


**Fig 6. Screening of NER and NERw variants by acceptance rate.** Top left: comparison of NER variants with the null operator NER{} (i.e. `Narrow Exchange`). Each of the 48 operators are represented by a single point, uniquely encoded by the point stylings. The number of times each operator is proposed and accepted is compared with that of NER{}, and one-sided z-tests are performed to assess the statistical significance between the two acceptance rates ($p = 0.001$). This process is repeated for each of 300 simulated datasets. The axes of each plot are the proportion of these 300 simulations for which there is evidence that the operator is better than NER{} (x-axis) or worse than NER{} (y-axis). Top right: comparison of NER and NERw acceptance rates. Each point is one NER/NERw variant from a single simulation. Bottom: relationship between the acceptance rates $\alpha$ of NER$\{\mathcal{D}_{AE}, \mathcal{D}_{BE}, \mathcal{D}_{CE}\}$ and NER{} with the clock model standard deviation $\sigma$ and the number of patterns $L_{eff}$. Each point is a single simulation.

$N = 30$) and thus poor signal. Overall, NER$\{\mathcal{D}_{AE}, \mathcal{D}_{BE}, \mathcal{D}_{CE}\}$ outperforms the standard `Narrow Exchange` operator when the data is not clock-like and contains enough signal.

Finally, **Fig. 6** shows that by applying a (Bactrian) random walk to $t_D$ – the height of internal node $D$ – the acceptance rate of NER plummets dramatically. This effect is most dominant for the NER variants which satisfy distance constraints (i.e. the

operators which are not NER{}). This result is unfortunate however not unexpected, and is consistent with Lakner et al. 2008 [36], who observed that tree operators perform best when they change either topology, or branch lengths, but not both.

Although there are several operators tying for first place, we selected the NER$\{\mathcal{D}_{AE}, \mathcal{D}_{BE}, \mathcal{D}_{CE}\}$ operator to proceed to the next round of optimisation.

#### Benchmarking convergence time

### Evaluating the AVMVN leaf rate operator

### Clock model averaging

## Discussion

## Conclusion

## Supporting information

**S1 Appendix. Rate quantiles.** The linear piecewise approximation used in the *quant* parameterisation is described. `Constant distance` tree operators [31] are extended to the *quant* parameterisation.

**S2 Appendix. Well-calibated simulation studies.** Methodologies are validated using well-calibated simulation studies.

## References


1. Zuckerkandl E. Molecular disease, evolution, and genetic heterogeneity. Horizons in biochemistry. 1962; p. 189–225.
2. Douzery EJ, Delsuc F, Stanhope MJ, Huchon D. Local molecular clocks in three nuclear genes: divergence times for rodents and other mammals and incompatibility among fossil calibrations. Journal of Molecular Evolution. 2003;57(1):S201–S213.
3. Drummond AJ, Ho SY, Phillips MJ, Rambaut A. Relaxed phylogenetics and dating with confidence. PLoS biology. 2006;4(5):e88.
4. Kuhner MK, Yamato J, Felsenstein J. Estimating effective population size and mutation rate from sequence data using Metropolis-Hastings sampling. Genetics. 1995;140(4):1421–1430.
5. Larget B, Simon DL. Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. Molecular biology and evolution. 1999;16(6):750–759.
6. Mau B, Newton MA, Larget B. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. Biometrics. 1999;55(1):1–12.
7. Metropolis N. Equation of state calculations by fast computing machines. J Chem Phys. 1953;21:1087–1092.
8. Hastings W. Monte-Carlo sampling methods using Markov chains and their applications. Biometrika. 1970;57:97–109.
9. Drummond AJ, Suchard MA, Xie D, Rambaut A. Bayesian phylogenetics with BEAUti and the BEAST 1.7. Molecular biology and evolution. 2012;29(8):1969–1973.
10. Bouckaert R, Vaughan TG, Barido-Sottani J, Duchêne S, Fourment M, Gavryushkina A, et al. BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis. PLoS computational biology. 2019;15(4):e1006650.
11. Ronquist F, Teslenko M, Van Der Mark P, Ayres DL, Darling A, Höhna S, et al. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. Systematic biology. 2012;61(3):539–542.
12. Höhna S, Landis MJ, Heath TA, Boussau B, Lartillot N, Moore BR, et al. RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. Systematic biology. 2016;65(4):726–736.
13. Zuckerkandl E, Pauling L. Evolutionary divergence and convergence in proteins. In: Evolving genes and proteins. Elsevier; 1965. p. 97–166.
14. Gillespie JH. The causes of molecular evolution. vol. 2. Oxford University Press On Demand; 1994.
15. Woolfit M. Effective population size and the rate and pattern of nucleotide substitutions. Biology letters. 2009;5(3):417–420.
16. Loh E, Salk JJ, Loeb LA. Optimization of DNA polymerase mutation rates during bacterial evolution. Proceedings of the National Academy of Sciences. 2010;107(3):1154–1159.

17. Lepage T, Bryant D, Philippe H, Lartillot N. A general comparison of relaxed molecular clock models. Molecular biology and evolution. 2007;24(12):2669–2680.

18. Li WLS, Drummond AJ. Model averaging and Bayes factor calculation of relaxed molecular clocks in Bayesian phylogenetics. Molecular biology and evolution. 2012;29(2):751–761.

19. Faria NR, Quick J, Claro I, Theze J, de Jesus JG, Giovanetti M, et al. Establishment and cryptic transmission of Zika virus in Brazil and the Americas. Nature. 2017;546(7658):406–410.

20. Giovanetti M, Benvenuto D, Angeletti S, Ciccozzi M. The first two cases of 2019-nCoV in Italy: Where they come from? Journal of medical virology. 2020;92(5):518–521.

21. Huelsenbeck JP, Larget B, Swofford D. A compound Poisson process for relaxing the molecular clock. Genetics. 2000;154(4):1879–1892.

22. Drummond AJ, Suchard MA. Bayesian random local clocks, or one rate to rule them all. BMC biology. 2010;8(1):1–12.

23. Zhang C, Huelsenbeck JP, Ronquist F. Using parsimony-guided tree proposals to accelerate convergence in Bayesian phylogenetic inference. Systematic Biology. 2020;.

24. Meyer X. Adaptive Tree Proposals for Bayesian Phylogenetic Inference. BioRxiv. 2019; p. 783597.

25. Höhna S, Drummond AJ. Guided tree topology proposals for Bayesian phylogenetic inference. Systematic biology. 2012;61(1):1–11.

26. Altekar G, Dwarkadas S, Huelsenbeck JP, Ronquist F. Parallel metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. Bioinformatics. 2004;20(3):407–415.

27. Müller NF, Bouckaert R. Coupled MCMC in Beast 2. bioRxiv. 2019;.

28. Baele G, Lemey P, Rambaut A, Suchard MA. Adaptive MCMC in Bayesian phylogenetics: an application to analyzing partitioned data in BEAST. Bioinformatics. 2017;33(12):1798–1805.

29. Yang Z, Rodríguez CE. Searching for efficient Markov chain Monte Carlo proposal kernels. Proceedings of the National Academy of Sciences. 2013;110(48):19307–19312.

30. Thawornwattana Y, Dalquen D, Yang Z, et al. Designing simple and efficient Markov chain Monte Carlo proposal kernels. Bayesian Analysis. 2018;13(4):1037–1063.

31. Zhang R, Drummond A. Improving the performance of Bayesian phylogenetic inference under relaxed clock models. BMC Evolutionary Biology. 2020;20:1–28.

32. Gelman A. Parameterization and Bayesian modeling. Journal of the American Statistical Association. 2004;99(466):537–545.

33. Roberts GO, Gelman A, Gilks WR, et al. Weak convergence and optimal scaling of random walk Metropolis algorithms. The annals of applied probability. 1997;7(1):110–120.

34. Drummond AJ, Nicholls GK, Rodrigo AG, Solomon W. Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. Genetics. 2002;161(3):1307–1320.

35. Suchard MA, Lemey P, Baele G, Ayres DL, Drummond AJ, Rambaut A. Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. Virus evolution. 2018;4(1):vey016.

36. Lakner C, Van Der Mark P, Huelsenbeck JP, Larget B, Ronquist F. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. Systematic biology. 2008;57(1):86–103.

37. Simon D, Larget B. Bayesian analysis in molecular biology and evolution (BAMBE) http://www.mathcs.duq.edu/larget/bambe.html. Pittsburgh, Pennsylvania. 1998;.

38. Jow H, Hudelot C, Rattray M, Higgs P. Bayesian phylogenetics using an RNA substitution model applied to early mammalian evolution. Molecular Biology and Evolution. 2002;19(9):1591–1601.

39. Roberts GO, Rosenthal JS. Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. Journal of applied probability. 2007;44(2):458–475.

40. Lindstrom MJ, Bates DM. Newton—Raphson and EM algorithms for linear mixed-effects models for repeated-measures data. Journal of the American Statistical Association. 1988;83(404):1014–1022.

41. Pourahmadi M. Cholesky decompositions and estimation of a covariance matrix: orthogonality of variance–correlation parameters. Biometrika. 2007;94(4):1006–1013.

42. Lanfear R. BenchmarkAlignments https://github.com/roblanf/BenchmarkAlignments. GitHub. 2019;.

43. Lanfear R, Frandsen PB, Wright AM, Senfeld T, Calcott B. PartitionFinder 2: new methods for selecting partitioned models of evolution for molecular and morphological phylogenetic analyses. Molecular biology and evolution. 2016;34(3):772–773.

44. Ran JH, Shen TT, Wang MM, Wang XQ. Phylogenomics resolves the deep phylogeny of seed plants and indicates partial convergent or homoplastic evolution between Gnetales and angiosperms. Proceedings of the Royal Society B: Biological Sciences. 2018;285(1881):20181012. doi:10.1098/rspb.2018.1012.

45. Dornburg A, Moore JA, Webster R, Warren DL, Brandley MC, Iglesias TL, et al. Molecular phylogenetics of squirrelfishes and soldierfishes (Teleostei: Beryciformes: Holocentridae): Reconciling more than 100 years of taxonomic confusion. Molecular Phylogenetics and Evolution. 2012;65(2):727–738. doi:10.1016/j.ympev.2012.07.020.

46. Cognato AI, Vogler AP. Exploring Data Interaction and Nucleotide Alignment in a Multiple Gene Analysis of Ips (Coleoptera: Scolytinae). Systematic Biology. 2001;50(6):758–780. doi:10.1080/106351501753462803.

47. Sauquet H, Ho SYW, Gandolfo MA, Jordan GJ, Wilf P, Cantrill DJ, et al. Testing the Impact of Calibration on Molecular Divergence Times Using a Fossil-Rich Group: The Case of Nothofagus (Fagales). Systematic Biology. 2011;61(2):289–313. doi:10.1093/sysbio/syr116.

48. Broughton RE, Betancur-R R, Li C, Arratia G, Ortí G. Multi-locus phylogenetic analysis reveals the pattern and tempo of bony fish evolution. PLoS Currents. 2013;doi:10.1371/currents.tol.2ca8041495ffafd0c92756e75247483e.

49. Kawahara AY, Rubinoff D. Convergent evolution of morphology and habitat use in the explosive Hawaiian fancy case caterpillar radiation. Journal of Evolutionary Biology. 2013;26(8):1763–1773. doi:10.1111/jeb.12176.

50. Cannon JT, Vellutini BC, Smith J, Ronquist F, Jondelius U, Hejnol A. Xenacoelomorpha is the sister group to Nephrozoa. Nature. 2016;530(7588):89–93. doi:10.1038/nature16520.

51. RIGHTMYER MG, GRISWOLD T, BRADY SG. Phylogeny and systematics of the bee genus Osmia ( H ymenoptera: M egachilidae) with emphasis on N orth A merican Melanosmia : subgenera, synonymies and nesting biology revisited. Systematic Entomology. 2013;38(3):561–576. doi:10.1111/syen.12013.

52. Moyle RG, Oliveros CH, Andersen MJ, Hosner PA, Benz BW, Manthey JD, et al. Tectonic collision and uplift of Wallacea triggered the global songbird radiation. Nature Communications. 2016;7(1). doi:10.1038/ncomms12709.