Improving the performance of Bayesian phylogenetic inference under relaxed clock models

Rong Zhang¹* and Alexei Drummond¹,²

Abstract

Background: Bayesian MCMC has become a common approach for phylogenetic inference. But the growing size of molecular sequence data sets has created a pressing need to improve the computational efficiency of Bayesian phylogenetic inference algorithms.

Results: This paper develops a new algorithm to improve the efficiency of Bayesian phylogenetic inference for models that include a per-branch rate parameter. In a Markov chain Monte Carlo algorithm, the presented proposal kernel changes evolutionary rates and divergence times at the same time, under the constraint that the implied genetic distances remain constant. Specifically, the proposal operates on the divergence time of an internal node and the three adjacent branch rates. For the root of a phylogenetic tree, there are three strategies discussed, named Simple Distance, Small Pulley and Big Pulley. Note that Big Pulley is able to change the tree topology, which enables the operator to sample all the possible rooted trees consistent with the implied unrooted tree. To validate its effectiveness, a series of experiments have been performed by implementing the proposed operator in the BEAST2 software.

Conclusions: The results demonstrate that the proposed operator is able to improve the performance by giving better estimates for a given chain length and by using less running time for a given level of accuracy. Measured by effective samples per hour, use of the proposed operator results in overall mixing more efficient than the current operators in BEAST2. Especially for large data sets, the improvement is up to half an order of magnitude.

Keywords: Bayesian MCMC, Bayesian phylogenetics, Proposal kernel, Genetic distances, Divergence times, Evolutionary rates

Background

Bayesian phylogenetics puts an emphasis on estimating a probability distribution over parameters of interest, including the phylogenetic tree topology and divergence times, given the data. The Metropolis-Hastings Markov chain Monte Carlo (MCMC) [1, 2] algorithm has been the primary computational tool used in Bayesian phylogenetics for sampling from the posterior distribution. This paper is aimed at improving the performance of the relaxed clock model in Bayesian phylogenetic analysis.

*Correspondence: rzha419@aucklanduni.ac.nz

¹School of Computer Science, University of Auckland, Auckland, New Zealand

Full list of author information is available at the end of the article

© The Author(s). 2020 Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.
Bayesian phylogenetic inference via MCMC is computationally intensive for large data sets. Two approaches to improve efficiency are (i) by making faster likelihood calculations, and (ii) by incorporating more effective proposal kernels. Calculating the phylogenetic likelihood is computationally expensive. Hence, researchers have tried many ways to tackle the computation burden in the likelihood calculations, such as detection of repeating sites [12], approximate methods (e.g. [13, 14]) and the use of parallelisation strategies (e.g. BEAGLE [15]).

However the overall efficiency of the sampling process also depends strongly on the construction of the proposal mechanism. An effective proposal mechanism is proficient at exploring the posterior distribution, and can do so with fewer steps in the MCMC chain. Therefore fewer likelihood calculations are required, since each step in the chain that changes the tree or substitution parameters requires a likelihood calculation.

A major limitation in Bayesian MCMC analysis of phylogeny lies in the efficiency with which operators sample the tree space [16, 17]. Fast and reliable estimation is dependent on a good mixture of operators, since the posterior distribution often exhibits correlations between the tree and other random variables.

In this paper, we present a novel operator that works alongside standard operators by proposing moves within a subspace of constant genetic distances. Namely, the proposed operator changes both divergence times of nodes and neighbouring branch rates so that the implied genetic distances are not changed. For time-reversible substitution models the phylogenetic likelihood will also be unchanged under this operation. The proposed operator has been implemented and tested in BEAST2 [18].

Preliminaries

Bayesian MCMC

Let \( D, g \) and \( \Phi \) denote the data, phylogenetic time-tree and a set of evolutionary parameters respectively. The time-tree \( g = \{E, t\} \) consists of a directed edge graph, \( E \), defining a rooted tree topology on a set of labelled taxa and a set of associated divergence times \( t \) (for details see e.g. [19]). The posterior probability density can be calculated using Eq. 1. It consists of prior distributions for the tree and the parameters, a phylogenetic likelihood that conveys information from data, and the posterior distribution to be inferred. These are denoted in the form of probability densities by \( p(g), p(\Phi), p(D|g, \Phi), p(g, \Phi|D) \) respectively. From a Bayesian perspective, the phylogenetic trees and the parameters are random variables described by a posterior probability distribution given the observed data \( D \).

\[
p(g, \Phi|D) = \frac{p(D|g, \Phi)p(g)p(\Phi)}{p(D)} \tag{1}
\]

However, due to the state space being high dimensional and the marginal likelihood being infeasible to calculate, MCMC is adopted to sample the posterior distribution. Specifically, MCMC algorithms construct a Markov chain whose stationary distribution is the posterior distribution \( p(g, \Phi|D) \), in such a way that the computation of the marginal likelihood \( p(D) \) is avoided.

Tree proposals

We use the term “operator” to describe an algorithm that can be used to draw a new state \( \theta' \) given an existing state \( \theta = (g, \Phi) \) from a specific proposal kernel \( q(\theta'|\theta) \) and also return the Hastings-Green ratio for the proposed state transition [2, 20].

Standard naïve operators such as the random walk operator propose the new state \( \theta' \) by adding a random variate to a component of the current state \( \theta \) [21]. Similarly, scale operators multiply a subset of the current state by a random scale factor [22]. They are suitable for working on a single random variable, or a single component of the model, for example the population size parameter of the coalescent tree prior. Standard operators for the tree topology and divergence times include the subtree slide operator, Wilson-balding and narrow exchange operators [19, 23].

In this paper, the novelty of the proposed operators lies in maintaining the genetic distance \( d \) while changing the rate \( r \) and divergence time \( t \). The reason is that the likelihood along one branch is constant if its distance is fixed, i.e. \( d = r \times t \), noting that the likelihood is calculated based on transition probability matrix for each branch of \( e^{Qt_i} \), where \( d_i \) is the branch length in units of substitutions per site for branch \( i \). In this way, the joint distribution on rates and divergence times can be explored without proposing states that would adversely affect the phylogenetic likelihood.

Uncorrelated relaxed clock model

Molecular clocks model how molecular sequences evolve along branches in the phylogenetic tree, so that a time tree can be reconciled with the genetic distances between sequences. In this paper, uncorrelated relaxed clock models are adopted, where the rates are drawn independently and identically from a given prior distribution, such as the log-normal distribution [7]. As a result, the rates can vary markedly between parent and child branches.

Referring to the Bayesian framework in Eq. (1), the joint inference of evolutionary rates \( r \) and the time tree \( g \) can be obtained by the conditional distribution in Eq. 2:

\[
p(g, r, \Phi|D) = \frac{p(D|g, r, \Phi)p(r)p(g)p(\Phi)}{p(D)}, \tag{2}
\]

where \( p(r) \) is the prior for rates specified in uncorrelated relaxed clock model. In the constructed Markov chain, the
operator proposes a new state \( \theta' = (r', g', \Phi') \), from the original state \( \theta = (r, g, \Phi) \).

While the proposed operator is introduced based on uncorrelated clock models, it could equally be applied to any other relaxed clock that applies a rate parameter to each branch, such as autocorrelated clock models [6].

Results
To validate the correctness and determine the efficiency, we conducted a series of experiments by implementing the Constant Distance operator in BEAST2 [18].

First, we perform a well-calibrated simulation study, which tests our operator alongside existing operators. Correctness was further confirmed by sampling trees from the prior distribution i.e. without data (Refer to Appendix 2 section for more details). By comparing effective sample sizes (ESS) [24] and running times, it is demonstrated that the performance is improved when including our proposed operator. Finally, the posterior correlation of rates and node times are discussed.

Well-calibrated simulation study
A well-calibrated simulation study is a powerful tool for evaluating and validating the implementation of a Bayesian model [25].

Figure 1 shows the Bayesian model used in this study, which includes the evolutionary model and the prior distributions of parameters. As is shown in the figure, the sequence alignment is simulated by a phylogenetic continuous-time Markov chain in BEAST2. It contains a substitution rate matrix given by the HKY85 [26] model and a substitution tree determined by an uncorrelated relaxed clock model and Yule model. More specifically, base frequencies \( \pi \) follow a Dirichlet distribution and the transition-transversion ratio \( \kappa \) follows a log-normal prior distribution. The distribution of node times is described in a Yule tree \( \psi \) with hyperparameter birth rate \( \lambda \) following a log-normal distribution. The rates \( r_i \) follow a log-normal distribution with mean of 1 and standard deviation \( s_1 \) following a hyperprior distribution.

First, we sampled parameters and trees from the full model 100 times. The random parameters included: standard deviation of rates across branches \( s_1 \), birth rate \( \lambda \), base frequencies \( \pi \) and transition-transversion bias \( \kappa \). Second, we simulated nucleotide alignments using the simulated parameters. In total, 100 data sets were simulated, each with 120 taxa. Third, we used BEAST2 with the Constant Distance operator to infer the tree and parameters from each of the 100 simulated data sets in turn. Finally, the posterior estimates of the parameters were compared with the real values that were used to simulate the corresponding sequence alignment. The comparisons are shown in Fig. 2.

These results show that the true values of the parameters are within the 95% highest posterior density (HPD) interval approximately 95% of the time (Table 1). This well-calibrated simulation study formed part of the validation of our implementation of the Constant Distance operator.

Performance comparison
To evaluate the performance of Constant Distance operator in a Bayesian phylogenetic analysis, we explored the time required to adequately sample the posterior distribution. This was achieved by examining (i) the total time
Fig. 1 The models and prior distributions to simulate the sequence data. The sequence alignment (SA) is simulated through a phylogenetic continuous-time Markov Chain (PhyloCTMC) that consists of a substitution model (HKY) and an uncorrelated relaxed clock model (UCRelaxedClockModel). The random variables in HKY model construct the mutation rate matrix ($Q$), including base frequencies ($\pi = \{\pi_A, \pi_C, \pi_G, \pi_T\}$) and kappa ($\kappa$). The time trees ($\Psi$) and branch rates ($r_i$ for each branch $i$ in $\Psi$) construct the substitution tree (ST). The branch rates have a LogNormal prior with fixed mean 1 and certain standard deviation (denoted by $s_1$). And the time trees have a Yule model prior with birth rate ($\lambda$) having a LogNormal prior. The other prior distributions include a Dirichlet distributions on $\pi$, a LogNormal distribution on $\kappa$, and a LogNormal distribution on $s_1$. For notations in LogNormal distributions, the uppercase letters represent the parameters in real space, and the lowercase letters represent the parameters in log space. In all the simulations, the number of taxa is fixed at 120 ($n = 120$).

We performed the analysis on two sets of simulated sequence alignment (See Appendix 3.2 for more details). The simulated data sets both have 20 taxa but different sequence lengths, i.e. one data set containing 500 sites, the other containing 1,000 sites. Moreover, we used four real data sets to further evaluate the performance of Constant Distance operator, including a primate data set [27] and three other datasets (Anolis [28], RSV2 [29, 30] and HIV-1 [31]) in BEAST2 [32].

The ESS and running time are summarised in Fig. 3 and Table 2. To be more specific, we measure the efficiency by ESS per hour, which is calculated by the ESS of parameters in one simulation divided by the running time in hours. Then we compare the efficiency of two configurations by calculating the ratios of ESS per hour for simulations in the two configurations. If the ratio is larger than 1, then ESS per hour of Cons configuration is larger than that of Category configuration. As is shown in Fig. 3, the efficiency varies in different data sets and also depends on what model is used in the analysis. For Anolis data set (29 taxa and 1456 sites), Category configuration performs better than Cons configuration, since most ratios of the parameters are slightly below the red line (which means smaller than 1). Moreover, for RSV2 (129 taxa and 629 sites) and HIV-1 data sets (117 taxa and...
some ratios of the parameters, “posterior” and “prior” in particular, are above the red line (larger than 1), which indicates that Cons configuration provides larger ESS per hour. Although there are several parameters sampled by Cons configuration having smaller ESS per hour, it should be noticed that the ratio is calculated by choosing random simulations in the two configurations (See Appendix 3.3 for more details). Additionally, it is worth noting that the efficiency is improved more obviously in simulated data set having 1000 sites, compared with the data sets having 500 sites. This indicates that the proposed operators behave better when sequence length is long.

More specifically, in Primates data set (87 taxa and 19220 sites), the longer molecular sequence provides more accurate genetic distances, which leads to peaked likelihood distributions. In this circumstance, the proposed operators sample rates and node times that fit the constant genetic distances more efficiently.

Table 2 lists the average running time of 20 simulations for each data set. It can be seen that Cons configuration finished simulations with a little bit more time in most cases. This is because the continuous rates have to be adjusted for a new clock standard deviation (See Appendix Section 4 for more details). Moreover, Table 2 also shows the parameter that has the smallest ESS in Category configuration, and is compare with the corresponding ESS in Cons configuration. Although the improvement in ESS is not obvious for both simulate data sets, it is noticed that ESS of the parameters are much larger in Cons configuration for all the real data sets. After calculating the ESS per hour, we conclude that Cons configuration improved the

663 sites), some ratios of the parameters, “posterior” and “prior” in particular, are above the red line (larger than 1), which indicates that Cons configuration provides larger ESS per hour. Although there are several parameters sampled by Cons configuration having smaller ESS per hour, it should be noticed that the ratio is calculated by choosing random simulations in the two configurations (See Appendix 3.3 for more details). Additionally, it is worth noting that the efficiency is improved more obviously in simulated data set having 1000 sites, compared with the data sets having 500 sites. This indicates that the proposed operators behave better when sequence length is long.

More specifically, in Primates data set (87 taxa and 19220 sites), the longer molecular sequence provides more accurate genetic distances, which leads to peaked likelihood distributions. In this circumstance, the proposed operators sample rates and node times that fit the constant genetic distances more efficiently.

Table 2 lists the average running time of 20 simulations for each data set. It can be seen that Cons configuration finished simulations with a little bit more time in most cases. This is because the continuous rates have to be adjusted for a new clock standard deviation (See Appendix Section 4 for more details). Moreover, Table 2 also shows the parameter that has the smallest ESS in Category configuration, and is compare with the corresponding ESS in Cons configuration. Although the improvement in ESS is not obvious for both simulate data sets, it is noticed that ESS of the parameters are much larger in Cons configuration for all the real data sets. After calculating the ESS per hour, we conclude that Cons configuration improved the

663 sites), some ratios of the parameters, “posterior” and “prior” in particular, are above the red line (larger than 1), which indicates that Cons configuration provides larger ESS per hour. Although there are several parameters sampled by Cons configuration having smaller ESS per hour, it should be noticed that the ratio is calculated by choosing random simulations in the two configurations (See Appendix 3.3 for more details). Additionally, it is worth noting that the efficiency is improved more obviously in simulated data set having 1000 sites, compared with the data sets having 500 sites. This indicates that the proposed operators behave better when sequence length is long.

More specifically, in Primates data set (87 taxa and 19220 sites), the longer molecular sequence provides more accurate genetic distances, which leads to peaked likelihood distributions. In this circumstance, the proposed operators sample rates and node times that fit the constant genetic distances more efficiently.

Table 2 lists the average running time of 20 simulations for each data set. It can be seen that Cons configuration finished simulations with a little bit more time in most cases. This is because the continuous rates have to be adjusted for a new clock standard deviation (See Appendix Section 4 for more details). Moreover, Table 2 also shows the parameter that has the smallest ESS in Category configuration, and is compare with the corresponding ESS in Cons configuration. Although the improvement in ESS is not obvious for both simulate data sets, it is noticed that ESS of the parameters are much larger in Cons configuration for all the real data sets. After calculating the ESS per hour, we conclude that Cons configuration improved the

663 sites), some ratios of the parameters, “posterior” and “prior” in particular, are above the red line (larger than 1), which indicates that Cons configuration provides larger ESS per hour. Although there are several parameters sampled by Cons configuration having smaller ESS per hour, it should be noticed that the ratio is calculated by choosing random simulations in the two configurations (See Appendix 3.3 for more details). Additionally, it is worth noting that the efficiency is improved more obviously in simulated data set having 1000 sites, compared with the data sets having 500 sites. This indicates that the proposed operators behave better when sequence length is long.

More specifically, in Primates data set (87 taxa and 19220 sites), the longer molecular sequence provides more accurate genetic distances, which leads to peaked likelihood distributions. In this circumstance, the proposed operators sample rates and node times that fit the constant genetic distances more efficiently.

Table 2 lists the average running time of 20 simulations for each data set. It can be seen that Cons configuration finished simulations with a little bit more time in most cases. This is because the continuous rates have to be adjusted for a new clock standard deviation (See Appendix Section 4 for more details). Moreover, Table 2 also shows the parameter that has the smallest ESS in Category configuration, and is compare with the corresponding ESS in Cons configuration. Although the improvement in ESS is not obvious for both simulate data sets, it is noticed that ESS of the parameters are much larger in Cons configuration for all the real data sets. After calculating the ESS per hour, we conclude that Cons configuration improved the
efficiency of the worst estimated parameter in Category configuration by a factor of 1.55 to 8.53.

**Correlation analysis of rates and branch lengths**

In this section, we conduct a pairwise comparison between rates and branch lengths in units of time. We used a data set of ratite mitochondrial genomes [33]. This data set includes 7 species of ratites and an alignment of 10767 sites. After analysing the ratites data set in BEAST2 using the Constant Distance operator, we calculated the Pearson coefficient between the rates and the times across branches to investigate the posterior correlation of these parameters.

The results are summarised in Fig. 4. Figure 4a presents the topology of the maximum clade credibility tree. We utilised the programme TreeStat2 [34] to obtain the filtered trees that have the same topology as the maximum clade credibility tree from the sampled trees in MCMC chain. This means the trees that have different shared common ancestors of each taxon from the reference tree are filtered out.

Afterwards, Fig. 4b shows the pairwise comparison of the 12 branch rates and 12 branch lengths (in time) on these filtered trees. As can be seen from the diagonal, the rate on one branch is negatively correlated with the length of that branch, which indicates that an older divergence time will lead to a smaller rate. This is because the primary signal in the data is genetic distance, so that there will be a range of rates and divergence times that are consistent with the genetic distances, but the products of these quantities will vary less than the individual parameters. The consequence is that there will tend to be a negative rela-
Table 2  Summary of ESS and running time

| Data                | Configuration | Average running time (hour) | Parameter | ESS   |
|---------------------|---------------|-----------------------------|-----------|-------|
| Andal            | Category      | 0.3788                      | frequency.A | 698.95|
|                  | Cons          | 0.4212                      |           | 750.53|
| RSV2              | Category      | 2.0509                      | prior     | 1231.32|
|                  | Cons          | 2.6742                      |           | 3409.88|
| HIV-1             | Category      | 2.6040                      | prior     | 387.83 |
|                  | Cons          | 3.2680                      |           | 753.48 |
| Primates          | Category      | 31.4059                     | rate.mean | 71.79 |
|                  | Cons          | 21.6584                     |           | 422.24 |
| Simulated data    | Category      | 0.0728                      | frequency.G | 819.39|
| with 500 sites    | Cons          | 0.0834                      |           | 837.50 |
| Simulated data    | Category      | 0.4403                      | frequency.G | 2760.41|
| with 1000 sites   | Cons          | 0.4863                      |           | 2961.67|

It is precisely this form of correlation structure in the posterior that our operator anticipates, and these correlations are the reason that our operator performs better than naive alternatives.

Sampling a fixed unrooted tree

A limiting case for the relaxed molecular clock model (and one exploited in some of our validation tests) occurs for long sequences, when the branch lengths of the unrooted

Fig. 4 Correlation analysis in the ratites tree. / represents the length of a branch, that is the time difference between a parent node and a child node, where \( l_1 = l_2 = l_3 = l_4 = l_5 = l_6 = l_7 = l_8 = l_9 = l_{10} = l_{11} = l_{12} = T - t_0 \). The rates and branch lengths are converted into log space and then Pearson’s coefficients are computed, which range from -1 to 1. Blue indicates positive correlations and red indicates negative correlations. The darker the colour, the stronger the correlation.
tree, in units of expected substitutions per site, become known without error. With full length genomes now available, although inferring trees from genomes involves complexities and assumptions such as a good partition scheme [35], this limiting case might be approached in some data sets. As a simple test in this paper, this gives rise to an alternative approach to analysis, where divergence times, a root position and the branch rates are random variables, and the data are a set of branch lengths in units of substitution on a known unrooted tree topology.

Previous work done by Reis and Yang [14] also tried to approximate the likelihood of such an unrooted tree in Bayesian phylogenetic inference. Similar researches in [6, 13] show that these methods can account for rate changes in a relaxed clock model, but the genetic distances are not fixed, for example Stéphane Guindon used a Gibbs sampling algorithm [13]. Outside of the Bayesian MCMC formalism, least-squares criteria [36] and maximum likelihood [37, 38], can also be applied to estimate substitution rates and divergence times in unrooted trees.

In this section, we investigated this approach on a fixed substitution tree reconstructed from whole mitochondrial genomes from a set of ratite species [33]. Since no uncertainty is admitted in the genetic distances and the proposed operator doesn’t change the genetic distances, the phylogenetic likelihood is no longer needed and the unrooted tree becomes the data, rather than a multiple sequence alignment.

First of all, we used the ratites data set to construct an unrooted tree with PhyML 3.0 [39, 40]. Figure 5a shows the unrooted tree with the genetic distances on the branches which are fixed in the subsequent relaxed clock analysis in BEAST2.

As an initial starting point, the root is assigned using the midpoint method. After that, according to the genetic distances among seven taxa and the position of the root, consistent divergence times are specified and assigned to each ancestral node, so that a valid rooted time tree is obtained. Once divergence times are determined, rates on the branches are also calculated.

![Fig. 5](image.png) Illustration of sampling a fixed unrooted tree. In subfigure (a), the unrooted tree is obtained from the ratites data set [33] by a maximum likelihood method [39] and the labeled numbers represent genetic distances. The three unique tree topologies in (b) (c) and (d) are obtained from the sampled trees by using program TreeTraceAnalysis [41]. The branch rates and node times are summarised by using program TreeAnnotator [42]. The labeled numbers represent the posterior mean of rates on the corresponding branches. The colour of branches from green to red indicates the rates increasing from small to large, and the blue bars represent the 95% HPD of the corresponding node times.
so that the products match the unrooted substitution tree.

Then we used Constant Distance operator to sample a Markov chain initiated by this starting tree. The resulting posterior distribution is shown in Fig. 5b-d. As can be seen, despite that there is some uncertainty in the root position, the most probable tree in Fig. 5b is consistent with previous analyses of this data (see Figure 2 in Ref. [33]). For large data sets of long sequences, the proposed operators may prove useful to provide faster divergence time estimates based on the assumption of known unrooted topology and branch lengths in units of expected substitutions per site.

Discussion

We have demonstrated that the presented operator is valid and able to improve the efficiency of phylogenetic MCMC for relaxed clock models. The overall performance of a Bayesian phylogenetic analysis will be affected by the proportion of MCMC steps that this operator is chosen to make the proposal. In the BEAST2 software, this can be changed by modifying the relative weights operators in the operator schedule. The ideal proportion is non-trivial to determine for an arbitrary data set. In this study, we assigned equal weights on operations to all internal nodes (including the root). How to assign weights to achieve better performance is not studied in this paper, and users may assign different weights in practice. Hence, an optimal method of assigning weights still needs further investigation.

The key idea of the presented operator (to maintain the genetic distances) shows a novel direction for more efficient proposals in Bayesian phylogenetic MCMC. For example, the operations on the internal nodes, in the current study, involve one random internal node, one node time and three branch rates. If two or more nodes are selected, then more associated rates and node times can be sampled in one proposal, which may achieve even better efficiency. Another possible approach is to make small changes to the genetic distances as well. To minimise the number of changes to genetic distances, a two-dimensional random draw will be used to change four parameters (one divergence time and one rate changed directly, the other two rates derived so as to minimise changes to genetic distances). What’s more, it should be pointed out that Small Pulley and Big Pulley can only be applied to reversible continuous-time Markov chain models where unrooted trees can be used in inference, because these operators require the underlying unrooted tree to be unchanged. Future work could elaborate a larger class of these operators require the underlying unrooted tree to be unchanged. Future work could elaborate a larger class of these operators.

Conclusions

As data sets have increased in size, the need for computational efficiency of Bayesian phylogenetic analyses has also increased. In this paper, we have discussed a new tree proposal that substantially increases the efficiency of Bayesian phylogenetic inference under a popular class of relaxed molecular clock models.

We demonstrate the correctness of this algorithm with a series of tests including a well-calibrated simulation study. Based on both simulated and real data sets, the proposed operator is more efficient than current algorithms implemented in BEAST2 for datasets with long sequences. This is a desirable property because efficiency is most important for larger datasets. The proposed operator is available for use as a package of BEAST2.

Methods

In this section, we define the Constant Distance Operator. Figure 6 illustrates the flow chart of the proposed operators. In a phylogenetic tree, the node to operate on is denoted by $X$ and the Constant Distance Operator works differently on internal nodes and the root node. The details of the operations are introduced step by step in the following subsections.

Operations on internal nodes

Figure 7 represents the tree (or subtree) with the node $X$ that is randomly selected from among the internal nodes. Let $g$ be the tree in the current state. The following steps propose a new divergence time in $g'$ and three rates in $r'$.

Step 1 Identify the parent node and two child nodes of $X$, denoted by $P$, $L$ and $R$ respectively.

Step 2 Denote the nodes times of $X$, $P$, $L$ and $R$ by $t_x$, $t_p$, $t_l$, $t_r$ respectively. Denote the rates on the branches above the nodes by $r_x$, $r_l$ and $r_R$ respectively.

Step 3 Propose a new node time for $X$ by $t_{x'} \leftarrow t_x + a$, where $a$ follows a Uniform distribution with a symmetric window size $w$, i.e. $a \sim \text{Uniform}[-w, +w]$, for some window size $w$. Make sure that the proposed time is valid, i.e.
max\{t_L, t_R\} < t'_X < t_P holds. Otherwise, we reject the proposal.

Step 4 Propose new rates by using Eq. 3.

\[ r'_X = \frac{t_P - t'_X}{t_P - t_X} \times \frac{t_X - t_L}{t_X - t'_L} \]
\[ r'_R = \frac{t_R - t'_X}{t_R - t_X} \times \frac{t_X - t_L}{t_X - t'_L} \]
\[ r'_L = \frac{t'_R - t'_X}{t'_R - t_X} \times \frac{t_X - t_L}{t_X - t'_L} \]

(3)

Step 5 Return the Green ratio \( \alpha_{IN} \) (Refer to Calculating the Green Ratio in the following subsection).

Operations on the root
We present three strategies for proposing the new rates and a new divergence time for the special case when \( X \) is the root node. i) The Simple Distance operator only
proposes a new root time. ii) Small Pulley adjusts the distances of branches on both sides of the root. iii) Big Pulley proposes a new tree topology by rearranging the root, without perturbing the unrooted tree. As is illustrated in Fig. 8a, all the operations on the root, including Big Pulley that changes the tree topology, do not change the underlying unrooted tree. For instance, no matter where the root \( X \) is (either on branch \( EF \) or \( AE \)), the operators maintain the distances \( d_{AB}, d_{AC}, d_{AD}, d_{BC}, d_{BD}, d_{CD} \) and preserve the unrooted tree at the same time.

**Simple distance**

Figure 8b, c and d show the trees that are rooted at the node \( X \). The original tree \( g \) in the current state is shown in Fig. 8b. Similar to the operations on internal nodes, we will use the following steps to propose a new root time in \( g' \) and two rates in \( r' \), as is illustrated in Fig. 8c. At the same time, the genetic distances of two branches linked to the root, i.e. \( d_L \) and \( d_R \), are kept constant.

**Step 1** Identify the child nodes of the root \( X \), denoted by \( L \) and \( R \). Their corresponding node times and branch rates are \( t_X, t_L, t_R \) and \( r_L, r_R \).

**Step 2** Propose a new node time for the root \( X \) by \( t_X' \leftarrow t_X + a \), where \( a \sim \text{Uniform}[-w, +w] \). Make sure that \( t_X' > \max\{t_L, t_R\} \) holds. Otherwise, we reject the proposal.

**Step 3** Propose new rates for branches on both sides of the root by using Eq. 4.

\[
\begin{align*}
    r'_L &= \frac{r_L \times (t_X - t_L)}{t_X' - t_L} \\
    r'_R &= \frac{r_R \times (t_X - t_R)}{t_X' - t_R}
\end{align*}
\]  

(4)

**Step 4** Return the Green ratio \( \alpha_{SD} \).

**Small pulley**

In contrast to Simple Distance, Small Pulley changes genetic distances of branches on both sides of the root. As is illustrated in Fig. 8d, two new rates in \( r' \) are proposed based on those in the original tree \( g \). In order to maintain the total genetic distance \( d_L + d_R \) of the two branches linked to the root, after \( d_L' \) is proposed, \( d_R \) will be adjusted simultaneously. In other words, Small Pulley keeps \( D = d_L + d_R \) constant. The detailed process includes the following 4 steps.

**Step 1** Identify the child nodes of the root \( X \), denoted by \( L \) and \( R \). Their corresponding node times and branch rates are \( t_X, t_L, t_R \) and \( r_L, r_R \). The implied genetic distances of the two branches linked to the root can be calculated by:

\[
\begin{align*}
    d_L &= r_L \times (t_X - t_L) \\
    d_R &= r_R \times (t_X - t_R)
\end{align*}
\]  

(5)
Step 2 Propose a new genetic distance for \( d_L \) by adding a random number that follows a Uniform distribution, i.e. \( d_L' \leftarrow d_L + b \), where \( b \sim \text{Uniform}[{-v}, {+v}] \), for some window size \( v \). Make sure that \( 0 < d_L' < D \) holds. Otherwise, we reject the proposal.

Step 3 Propose new rates for branches on each side of the root:

\[
\begin{align*}
    r_L' &= \frac{d_L'}{t_X - t_L} \\
    r_R' &= \frac{D - d_L'}{t_X - t_R}
\end{align*}
\]

(6)

Step 4 Return the Green ratio \( \alpha_{SP} \).

**Big Pulley**

Big Pulley resamples the rates and times while maintaining the implied unrooted tree in units of genetic distance. So the genetic distances between the taxa are held constant, but the location of the root in the time tree is readjusted.

Before describing the detailed steps, we introduce a method \( \text{Exchange} \) that proposes a new root position. In Fig. 9, let (i) \( X \) denote the root of tree \( g \), (ii) \( C \) and \( N \) denote the two child nodes of \( X \), (iii) \( S \) and \( M \) denote the two child nodes of \( C \). The \( \text{Exchange}(M, N) \) method involves the following steps:

- Swap the two nodes by pruning and regrafting, i.e. cutting \( M \) (\( N \)) at its original position and attaching it to the original position of \( N \) (\( M \)).
- Propose \( d_C' \leftarrow d_C + b \), where \( b \sim \text{Uniform}[{-v}, {+v}] \). Make sure that \( 0 < d_C' < D \) holds, where \( D = d_C + d_N \). Otherwise, we reject the proposal.
- The distances on the other three branches, i.e. \( d_S, d_M \) and \( d_N \), will be adjusted:

\[
d_S' = d_S \quad d_M' = d_M - d_C' \quad d_N' = d_N + d_C
\]

(7)

As can be seen from the above descriptions, the method \( \text{Exchange}(M, N) \) swaps two nodes and adjusts distances \( (d_S, d_M, d_N \) and \( d_C) \) on the four branches so as to maintain the implied genetic distances among three taxa \( S, M \) and \( N \).

Additionally, operations in Big Pulley vary depending on the shape of phylogenetic tree. In Fig. 10, a symmetric tree is shown on the left, in which both the child nodes of the root have two child nodes, i.e. \( L \) having children \( H_1, H_2 \) and \( R \) having children \( H_3, H_4 \). But in the asymmetric tree on the right, only one of the child nodes of the root has two child nodes below it, i.e. \( O \) having children \( G_1, G_2 \). But the other child node \( Y \) doesn't have any child node, which is a heterochronous tip. The corresponding operations are detailed in the following two parts.

**Symmetric tree** For the symmetric tree in Fig. 10, the operations are illustrated in Fig. 11, after which one of the four possible trees (1 2 3 4) will be proposed. The detailed process is described in Algorithm 1.

For example, suppose we are going to propose tree (1). After the new node times for the root \( X \) and \( L \) are proposed, we apply the method by \( \text{Exchange}(H_1, R) \), so that four distances are adjusted, as follows:

\[
d_{H_1'} = d_{H_1} - d_L' \quad d_{H_2'} = d_{H_2} + d_L' = d_L + b d_R'
\]

(8)

Finally, in this example the new rates would be updated by:

\[
\begin{align*}
    r_{H_1'} &= \frac{d_{H_1'}}{t_X' - t_{H_1}} \quad r_{H_2'} &= \frac{d_{H_2'}}{t_X' - t_{H_2}} \\
    r_L' &= \frac{d_L'}{t_X' - t_L} \quad r_R' &= \frac{d_R'}{t_X' - t_R}
\end{align*}
\]

(9)

**Asymmetric tree** For an asymmetric tree such as in Fig. 10 we would operate as illustrated in Fig. 12, in which
there are three possible trees (5 6 7). The operations are detailed in Algorithm 2.

To give an example, assume we are going to propose tree 5. Firstly, $t_X'$ and $t_O'$ are proposed in Step 3 and Step 4. Then, in Step 4, the method Exchange ($G_1$, $Y$) is applied, after which the four distances are adjusted as follows:

$$d_{G_1}' = d_{G_1} - d_O' d_{G_2}' = d_{G_2} d_O' = d_O + b d_Y' = d_Y + d_O$$

(10)

And the four rates are updated as follows:

$$r_{G_1}' = \frac{d_{G_1}'}{t_X' - t_{G_1}} r_{G_2}' = \frac{d_{G_2}'}{t_O' - t_{G_2}} r_O' = \frac{d_O'}{t_X' - t_O'} r_Y' = \frac{d_Y'}{t_Y - t_Y'}$$

(11)

**Calculating the green ratio**

MCMC operators must use reversible proposal distributions to satisfy the detailed balance requirements of the MCMC algorithm (Refer to Appendix section 1 for more details). Therefore, all of our operators involve a final step of calculating the Green ratio for the proposal.

According to the third and fourth steps in the operations for internal nodes, three rates on the branches linked to the selected internal node are proposed by one random number $a$ that is used to change the node time. There are four parameters involved in this proposal, comprised of a 3-dimensional rate space and a 1-dimensional time space. The proposed operator utilises one random number in time space and makes changes in both time and rate space, which leads to a dimension-matching problem. To solve this dimension-matching problem, as is mentioned in Green’s paper [20], it is necessary to construct the Jacobian matrix. In Eq. (12), $J_1$ deals with the parametric spaces before the proposal in vector $\mathbf{IN} = [t_X, r_X, r_L, r_R]$ and after the proposal in vector $\mathbf{OUT} = [t_X', r_X', r_L', r_R']$.

$$J_1 = \left[ \begin{array}{cccc} \frac{\partial f_1}{\partial t_X} & \frac{\partial f_1}{\partial r_X} & \frac{\partial f_1}{\partial r_L} & \frac{\partial f_1}{\partial r_R} \\ \frac{\partial f_2}{\partial t_X} & \frac{\partial f_2}{\partial r_X} & \frac{\partial f_2}{\partial r_L} & \frac{\partial f_2}{\partial r_R} \\ \frac{\partial f_3}{\partial t_X} & \frac{\partial f_3}{\partial r_X} & \frac{\partial f_3}{\partial r_L} & \frac{\partial f_3}{\partial r_R} \\ \frac{\partial f_4}{\partial t_X} & \frac{\partial f_4}{\partial r_X} & \frac{\partial f_4}{\partial r_L} & \frac{\partial f_4}{\partial r_R} \end{array} \right]$$

(12)

where the functions $f_1$, $f_2$, $f_3$ and $f_4$ represent how the operator makes a proposal. After substituting Eq. (3) in Eq. (12), the Green ratio for the internal nodes can be derived:

$$\alpha_{IN} = \frac{p(-a)}{p(a)} |J_1| = \frac{t_p - t_X}{t_p - t_{X'}} \times \frac{t_X - t_L}{t_X' - t_L} \times \frac{t_X - t_R}{t_X' - t_R}$$

(13)

where the proposal density $p(-a)$ is equal to $p(a)$ since the random number $a$ is drawn from Uniform distribution.

Likewise, the Green ratio for Simple Distance, Small Pulley and Big Pulley can be obtained:

$$\alpha_{SD} = \frac{t_X - t_L}{t_{X'} - t_L} \times \frac{t_X - t_R}{t_{X'} - t_R}$$

(14)

$$\alpha_{SP} = 1$$

(15)

$$\alpha_{BP} = \mu \times \frac{t_X' - t_C}{t_{X'} - t_{C'}} \times \frac{t_C - t_S}{t_C' - t_S} \times \frac{t_C - t_{N1}}{t_C' - t_{N1}} \times \frac{t_X - t_{N2}}{t_{C'} - t_{N2}}$$

(16)
where \( \mu = \frac{p(g',g)}{p(g,g')} \) is defined as the proposal ratio of topology change and is obtained by Algorithm 3. More details of how to calculate the determinant of the Jacobian matrix are explained in Appendix 1 section.

**Appendix**

1. the green ratio

When developing an operator for MCMC, the proposal function must be reversible. In other words, the probability that the operator propose a new state from the current state is required to be equal to the probability that the proposed state goes back to current state. To be specific, let \( \pi(x) \) be the target probability distribution and \( p(x,x') \) be the transition kernel in the continuous Markov chain. The reversibility condition requires that

\[
\pi(x) p(x,x') = \pi(x') p(x',x).
\]

And an operator provides a proposal \( q(x,x') \) with some probability \( \alpha(x,x') \) that the proposal is accepted. Thus, the reversibility condition is rewritten as

\[
\pi(x) q(x,x') \alpha(x,x') = \pi(x') q(x',x) \alpha(x',x).
\]
Algorithm 2 Proposal for asymmetric trees in Big pulley

[Step 1: Identify the child node of the root X that has two child nodes below, which is denoted by O. The other child node of the root, which does not have any child nodes and means a heterochronous tip, is denoted by Y. The node times of the root X, Y, O and its child nodes are denoted by \( t_X, t_Y, t_O, t_{G1} \) and \( t_{G2} \) respectively.]

Let X be the root of the tree.

Let O be the child of X that has children, and let Y be the child of X that does not have children.

[Step 2: Propose a new node time for the root X.]

\[ a \sim \text{Uniform}[-w, +w] \]

\[ t_X' \leftarrow t_X + a \]

[Step 3: Propose a new node time for the node O.]

\[ a_3 \sim \text{Uniform}[-w, +w] \]

\[ t_O' \leftarrow t_O + a_3 \]

if \( t_O' < t_Y \) or \( t_O' > t_X' \) then

Reject the proposal.

end if

[Step 4: Adjust the distances according to the tree corresponding topologies.]

if \( t_O' > \text{max}(t_{G1}, t_{G2}) \) or \( t_{G1} = t_{G2} \) then

if \( \alpha_3 \sim \text{Uniform}(0, 1) < 0.5 \) then

Apply Exchange \((G1, Y)\) and propose tree 5.

else

Apply Exchange \((G2, Y)\) and propose tree 6.

end if

else if \( \min(t_{G1}, t_{G2}) < t_O' < \text{max}(t_{G1}, t_{G2}) \) then

Exchange the older child of O and Y. (For the asymmetric tree in Fig. 5, we apply Exchange \((G1, Y)\) and propose tree 7).

else if \( t_O' < \min(t_{G1}, t_{G2}) \) then

Reject the proposal.

end if

[Step 4: Update the rates on the corresponding branches.]

[Step 5: Return the Green ratio \( \alpha_{BP} \).]

Algorithm 3 Calculation of \( \mu \) for Big pulley

Original tree is symmetric:

if the node that has been exchanged with L or R has child nodes then

\[ \alpha = \beta = 0.25 \]

else if \( t_R > t_L \) then

\[ \alpha = 1, \beta = 0.5 \]

else if \( t_R < t_L \) then

\[ \alpha = 0.5, \beta = 1 \]

else if \( t_R = t_L \) then

\[ \alpha = \beta = 1 \]

end if

if Proposed tree belongs to 1 or 2 then

Return \( \mu = \frac{\alpha}{0.5^2} \)

end if

if Proposed tree belongs to 3 or 4 then

Return \( \mu = \frac{\beta}{0.5^2} \)

end if

if Proposed tree belongs to 5 then

Return \( \mu = \frac{\gamma}{0.25} \)

end if

if Proposed tree belongs to 6 then

Return \( \mu = \frac{\gamma}{0.25} \)

end if

if Proposed tree belongs to 7 then

Return \( \mu = \frac{\gamma}{0.25} \)

end if

where \( A \in \varphi_1 \) and \( B \in \varphi_2 \) are two Borel sets. \( g(x, x') \) denotes the probability that the operator proposes a new state \( x' \) given the current state \( x \).

\[ \alpha_H(x, x') = \min \left\{ 1, \frac{g(x')p(x', x)}{g(x)p(x, x')} \right\}, \tag{18} \]

where \( p(x', dx')/p(x, dx) \) is known as the Hastings ratio.

However, for operators that do not have a symmetric measure, it is necessary to include the Jacobian matrix \( J \) in order to deal with the dimension matching problem, as is discussed in Green’s paper [20]. In this case, Eq. (18) is extended, as is shown in Eq. (19).

\[ \alpha_G(x, x') = \min \left\{ 1, \frac{g(x')p(x', x)}{g(x)p(x, x')} |J| \right\}, \tag{19} \]

where \( J = \nabla h(x, x') \) represents a vector differential matrix of deterministic function \( h \). \( \alpha = \frac{p(x, x)}{p(x', x')} |J| \) is defined as the Green ratio, and \( J \) ensures that the proposal have a symmetric measure on each subspace in state \( x \) and \( x' \).
1.1 calculating the green ratio for operations on internal nodes
The Constant Distance Operator firstly proposes a new time for the randomly selected internal node (Eq. (20a)), and then proposes three rates by the original distances and new node times (Eqs. (20b)–(20d)).

\[ f_1 : t_X' = t_X + a \]  
(20a)

\[ f_2 : r_X' = \frac{r_X \times (t_P - t_X)}{t_P - t_X'} \]  
(20b)

\[ f_3 : r_L' = \frac{r_L \times (t_X - t_L)}{t_X' - t_L} \]  
(20c)

\[ f_4 : r_R' = \frac{r_R \times (t_X - t_R)}{t_X' - t_R} \]  
(20d)

Substituting Eq. (20) in the Jacobian matrix \( J_1 \) (Eq. (12)), we get Eq. (21), so that the determinant of \( J_1 \) can be obtained by Eq. (22).

\[
J_1 = \begin{bmatrix}
t_X - t_X' & 0 & 0 & 0 \\
t_P - t_X & 0 & 0 & 0 \\
t_X' - t_L & 0 & 0 & 0 \\
t_X' - t_R & 0 & 0 & 0 \\
\end{bmatrix}
\]  
(21)

\[
|J_1| = 1 \times \begin{vmatrix}
t_P - t_X & 0 & 0 & 0 \\
t_X' - t_L & 0 & 0 & 0 \\
t_X' - t_R & 0 & 0 & 0 \\
\end{vmatrix}
\]  
(22)

1.2 calculating the green ratio for simple distance
Simple Distance proposes two rates by using Eqs. (23b) and (23c), according the new root time in Eq. (23a). So the Jacobian matrix can be obtained as shown in Eq. (24).

\[ t_X' = t_X + a \]  
(23a)

\[ r_L' = \frac{r_L \times (t_X - t_L)}{t_X' - t_L} \]  
(23b)

\[ r_R' = \frac{r_R \times (t_X - t_R)}{t_X' - t_R} \]  
(23c)

\[
J_2 = \begin{bmatrix}
\frac{\partial t_X'}{\partial t_X} & \frac{\partial t_X'}{\partial t_X'} & \frac{\partial t_X'}{\partial t_X'} & \frac{\partial t_X'}{\partial t_X'} \\
\frac{\partial r_L'}{\partial t_X} & \frac{\partial r_L'}{\partial t_X'} & \frac{\partial r_L'}{\partial t_X'} & \frac{\partial r_L'}{\partial t_X'} \\
\frac{\partial r_R'}{\partial t_X} & \frac{\partial r_R'}{\partial t_X'} & \frac{\partial r_R'}{\partial t_X'} & \frac{\partial r_R'}{\partial t_X'} \\
\frac{\partial r_X'}{\partial t_X} & \frac{\partial r_X'}{\partial t_X'} & \frac{\partial r_X'}{\partial t_X'} & \frac{\partial r_X'}{\partial t_X'} \\
\end{bmatrix}
\]  
(24)

So the determinant of \( J_2 \) is calculated by Eq. (25)

\[
|J_2| = \frac{t_X - t_L}{t_X' - t_L} \times \frac{t_X - t_R}{t_X' - t_R} \]  
(25)

1.3 calculating the green ratio for big pulley
Two new node times are proposed in Big Pulley. One is the root time (Eq. (28a)), the other is the node time of the child node of the root. It can be either children of the root, i.e. son and dau. So \( t_C' \) is used to denote the node time proposed, as is seen in Eq. (28b). In addition, the distances are adjusted by the method \( Exchange(M,N) \), dependent on which nodes are chosen. As a result, the four rates are proposed, as is shown in Eq. (28c)–Eq. (28f).

\[ t_X' = t_X + a \]  
(28a)

\[ t_C' = t_C + a_{1,2,3} \]  
(28b)

\[ r_C' = \frac{r_C \times (t_X - t_C) + b}{t_X' - t_C'} \]  
(28c)

\[ r_S' = \frac{r_S \times (t_C - t_S)}{t_C' - t_S} \]  
(28d)

\[ r_M' = \frac{r_M \times (t_C - t_M) - [r_C \times (t_X - t_C) + b]}{t_X' - t_M} \]  
(28e)

\[ r_N' = \frac{r_N \times (t_X - t_C) + r_N \times (t_X - t_N)}{t_C' - t_N} \]  
(28f)

where \( a_{1,2,3} \) is the random number to propose a new node time for the child node of the root. Depending on which child node is selected, the notation is different, i.e. \( a_1, a_2, a_3 \). Here, to make it a general case, \( a_x \) is used.
Therefore, the Jacobian matrix \( \mathbf{J}_4 \) for the six parameters in Eq. (28) is obtained by Eq. (29). And the determinant of \( \mathbf{J}_4 \) is calculated shown in Eq. (30).

\[
\mathbf{J}_4 = \begin{bmatrix}
\frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} \\
\frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} \\
\frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} \\
\frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} \\
\frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} \\
\frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c} & \frac{\partial \alpha}{\partial c}
\end{bmatrix}
\]

\[|\mathbf{J}_4| = \frac{t_{x'} - t_C}{t_{x'} - t_{C'}} \times \frac{t_C - t_S}{t_{C'} - t_S} \times \frac{t_C - t_M}{t_{C'} - t_M} \times \frac{t_X - t_N}{t_{C'} - t_N} \times \frac{t_{x'} - t_{C'}}{t_{C'} - t_{C'}} (30)\]

Last but not least, due to the change of tree topology in Exchange \((M, N)\), the probability of the proposed tree going back to the original tree \(p(g'g)\), as well as the probability of making the proposal \(p(g'|g)\), should be considered. As the ratio of \(p(g'|g)/p(g|g)\) is defined as \(\mu\), the calculation of \(\mu\) is detailed in the following algorithm.

2. sampling from the prior

In this section, we aim to validate the correctness of the proposed operators. To be more specific, we firstly run the simulations by sampling from prior distributions in BEAST2. Since the prior distributions are deterministic, we can analytically calculate the theoretical joint-distributions of sampled parameters in MCMC chains. By comparing the sampled distributions with the analytical results, we demonstrate whether the proposed operators are able to sample parameters correctly.

In Fig. 13, a tree with three taxa \(A\), \(B\) and \(C\) (plus one internal node \(D\), and root \(E\)) is used as a small example in the experiments in this section. In the figure, \(g_1\) is set as the initial tree. Firstly, a LogNormal distribution is used as the rate prior in the uncorrelated relaxed clock model, given by Eq. (31).

\[
r = [r_A \quad r_B \quad r_C \quad r_D] \sim \text{LogNormal}(m = -3, s = 0.25)
\]

In addition, a Coalescent model [47] with constant population size \((N = 0.3)\) is used to describe the tree prior. Hence, for the tree in Fig. 13, the probability of node times is calculated by Eq. (32).

\[
p(t = \{t_E, t_D\}) = \frac{1}{N} \times e^{-\frac{t_E}{2}} \times \frac{1}{N} \times e^{-\frac{t_D}{2}}
\]

After the priors are specified, the distribution to sample can be exactly known, since the samples are drawn from the prior distributions. In other words, as the rates are functions of its genetic distance and times, the joint distribution to sample can be represented by Eq. (33).

\[
p(r, t) = p(t_E, t_D) \times p(r_D) \times p(r_A) \times p(r_B) \times p(r_C)
\]

\[
= p(t_E, t_D) \times p\left(\frac{d_D}{t_E - t_D}\right) \times p\left(\frac{d_A}{t_D - t_A}\right)
\]

\[
\times p\left(\frac{d_B}{t_D - t_B}\right) \times p\left(\frac{d_C}{t_E - t_C}\right),
\]

where \(p(.)\) is the probability of certain rate values in the LogNormal distribution. Therefore, the whole probability can be obtained by conducting numerical integration on Eq. (33), which shows the probability distribution over all the possible values of parameters.

2.1 test the operator on internal nodes

The genetic distances, node times and rates for \(g_1\) in Fig. 13 are given in Table 3. To test roundly, two scenarios are designed. In each scenario, the genetic distances are fixed, the node time \(t_D\) starts from the initial value and will be changed by the proposed operator during the sampling process. Essentially, the proposed operator makes node \(D\)
move between node $A$ and $E$. Besides, to make sure that the result is robust, two different MCMC chain lengths are performed in each scenario, i.e. 10 million and 20 million.

The mean, mean error and the standard deviation of the MCMC samples are summarised in Table 4. Besides, according to Eq. (33), the actual joint distribution is obtained by using Eq. (34), and is used to evaluate the results, which is also included in Table 4. Moreover, the histograms of MCMC samples that indicate the sampled distributions, as well as the curves of the numerical integration of Eq. (34), are shown in Fig. 14. From Table 4 and Fig. 14, it can be seen that the red curves well fit the black histograms, and the mean values and standard deviations are consistent, which makes it safe to conclude that the proposed operator samples the internal node correctly.

\[ p(r, t) = \int_{t=0}^{t_E} p(t_E, t_D) \times p \left( \frac{d_A}{t_D} \right) \times p \left( \frac{d_B}{t_D} \right) \times p \left( \frac{d_D}{t_E - t_D} \right) \times p \left( \frac{d_C}{t_E} \right) \, dd_D \quad (34) \]

2.2.1 Using Simple Distance The root time $t_E$ is sampled by Simple Distance, which ranges from 1 to positive infinity theoretically. Namely, all the genetic distances and the node time $t_D$ are fixed. Similar to Eq. (34), the joint distribution of $t_E$ and rates to sample can be obtained by Eq. (35).

\[ p(r, t) = \int_{t=1}^{+\infty} p(t_E, t_D) \times p \left( \frac{d_A}{t_D} \right) \times p \left( \frac{d_B}{t_D} \right) \times p \left( \frac{d_D}{t_E - t_D} \right) \times p \left( \frac{d_C}{t_E} \right) \, dd_E \quad (35) \]

The results are given in Table 6 and Fig. 15a. As can be seen, the mean and the standard deviation of MCMC samples and numerical integration are close to each other, which confirms that the two distribution are the same. Thus, Simple Distance samples the root time and two branch rates correctly.

2.2.2 Using Small Pulley Although both $d_A$ and $d_i$ are changed during the sampling process when using Small Pulley, the sum of $d_D$ and $d_C$ are kept 0.67 in this test, as the initial setting shown in Table 5. To make it simple, only $d_D$ is compared.

Then, based on Eq. (33), the exact distribution of $d_i$ can be obtained by Eq. (36), which is compared with the sampled distribution in Table 6 and Fig. 15b. Even though there exist some errors, the sampled parameters can be considered to follow the same distribution. So the Small Pulley is also able to provide correct samples.

\[ p(r, t) = \int_{d_D=1}^{0.67} p(t_E, d_D) \times p \left( \frac{d_A}{t_D} \right) \times p \left( \frac{d_B}{t_D} \right) \times p \left( \frac{d_D}{t_E - t_D} \right) \times p \left( \frac{d_C}{t_E} \right) \, dd_D \quad (36) \]

2.2.3 Using Big Pulley For $g_1$ in Fig. 13, a new tree, together with the root time $t_E$ and node time of its older child $t_D$, as well as a genetic distance $d_i$, is proposed by Big Pulley. In this case, the initial tree $g_1$ will either go to $g_2$ or $g_3$, as is shown in Fig. 13. So the samples are repeatedly drawn from the 3 trees. Besides, according to the initial settings in Table 5, the genetic distances remain unchanged during the process, i.e. $d_{AB} = 1$, $d_{AC} = 1$ and $d_{BC} = 1$ hold. Hence, the distribution we are about to achieve can be calculated by Eq. (37).

### Table 3: Initial settings for testing operations on internal nodes

| Scenario | Genetic distances (fixed) | $t_0$ | $t_E$ | Initial rates |
|----------|---------------------------|-------|-------|---------------|
|          | $d_i$ | $d_x$ | $d_r$ | $d_i$ | $t_i$ | $t_r$ | $r_i$ |
| Scenario 1 | 0.1  | 0.2  | 0.4  | 0.27 | 1    | 10   | 0.1  | 0.2  | 0.04 | 0.03 |
| Scenario 2 | 0.4  | 0.8  | 2.4  | 1.6  | 0.4  | 0.8  | 1    | 2    | 3    | 4    |

### Table 4: Results of sampling the internal node

| Chain Length | Sample from MCMC | Integral curve | Plot |
|--------------|------------------|----------------|------|
| Scenario 1   |                  |                |      |
| 10000000     | 3.2727 8.3e-3    | 3.2669 1.3e-06 | Fig.14a |
| 20000000     | 3.271 6.1e-3     | 0.5616         | Fig.14b |
| Scenario 2   |                  |                |      |
| 10000000     | 0.4677 3.9e-04   | 0.4667 3.5e-05 | Fig.14c |
| 20000000     | 0.4672 2.8e-04   | 0.0262         | Fig.14d |
Fig. 14 Sampled parameters in tests of internal nodes. The horizontal axis represents the node time of D in Fig. 13. The two scenarios sample two trees with different distances specified in Table ??

### Table 5 Initial settings for operations on the root

| Strategy       | Genetic distances | 0.1 | 0.2 | 0.4 | 0.27 | Initial rates |
|----------------|-------------------|-----|-----|-----|------|---------------|
|                |                   | $d_1$ | $d_2$ | $d_3$ | $d_4$ | $r_1$ | $r_2$ | $r_3$ | $r_4$ |
| Simple Distance|                   | 0.1   | 0.2  | 0.4  | 0.27 | 1     | 10    | 0.1   | 0.2  | 0.04  | 0.03  |
| Small Pulley   |                   | 0.1   | 0.2  | 0.67 | 1    | 10    | 0.1   | 0.2  | 0.04  | 0.03  |
| Big Pulley     |                   | 0.5   | 0.5  | 0.5  | 5    | 10    | 0.1   | 0.1  | 0.03  | 0.04  |

### Table 6 Results of sampling the root

| Strategy        | Variable | Sample from MCMC | Integral curve | Plot   |
|-----------------|----------|------------------|----------------|--------|
|                 |          | Mean | St.dev | Mean | St.dev |             |        |
| Simple Distance | $t_E$    | 7.8081 | 1.2884 | 7.8187 | 1.2992 | Fig. 15a |
| Small Pulley    | $d_i$    | 0.3480 | 0.0492 | 0.3476 | 0.0494 | Fig. 15b |
| Big Pulley      | $d_i$    | 0.1016 | 0.0766 | 0.0960 | 0.0760 | Fig. 15c |
|                 | $t_E$    | 3.3017 | 0.6908 | 3.3095 | 0.6912 | Fig. 15d |
Fig. 15 Sampled parameters in test of the root. For the trees in Fig. 13, Simple Distance samples the root time $t_E$ only, Small Pulley samples the distance $d_D$ only, and Big Pulley samples $t_E, t_D, d_D$. To make it simple, $t_E$ and $d_D$ are compared.

\[
p(r, t) = \int_{t_E=0}^{+\infty} \int_{t_D=0}^{t_E} \int_{d_D=0}^{0.5} p(t_E, t_D) \times p\left(\frac{0.5}{t_D}\right) \times p\left(\frac{d_D}{t_E - t_D}\right) \times p\left(\frac{0.5 - d_D}{t_E}\right) dd_D dt_D dt_E
\]  

(37)

The statistical measurements, i.e. mean and standard deviation, are compared in Table 6. The histograms of samples and numerical curves of $d_D$ and $t_E$ are pictured in Fig. 15c and d. It is shown that the two distributions are consistent within the acceptable error range. Therefore, Big Pulley can also give the right combinations of rates and node times, under the condition that the genetic distances among taxa are constant.

3. performance analysis of operators

This section provides the details of the results presented in Performance comparison section.

3.1 Operator weights

The weights on operators for the simulations when comparing efficiency are listed in Table 7. Although how to assign weights to achieve better performance is not studied in this paper, we maintain the percentage of weights on three operator class in Category and Cons configurations. But we modified some weights on the operators inside the same class, and we assigned different weights for different data sets.

3.2 Simulated data sets

We simulated two sets of sequence alignment on the same tree with 20 taxa that is shown in Fig. 16. We used HKY model as substitution model with $\kappa = 2.4751$, and the base frequencies are $\pi = (0.2193, 0.2268, 0.3007, 0.2531)$. In the uncorrelated relaxed clock model, the standard deviation of the branch rates (Uclstdstdev) is 0.1803. The models and prior distributions are the same as is described in Fig. 1.

3.3 Efficiency measured by ESS per hour

Since we compare the efficiency based on ESS per hour using two configurations, i.e. Category and Cons, the ratio of ESS per hour is calculated by a random simulation in the two configurations, as is shown in Fig. 3. Then Table 2 lists the average running time and ESS of particular parameters in the simulations using different data sets. Here, we present the detailed running time and ESS of the simulations, which can be seen in Figs. 17, 18, 19, 20,
Table 7 Operator weights in MCMC chains

| Operator class       | Name                                      | Simulated data | Anolis | RSV2 | HIV-1 | Primates |
|----------------------|-------------------------------------------|----------------|--------|------|-------|----------|
|                      |   | Cons | Category | Cons | Category | Cons | Category | Cons | Category | Cons | Category |
| rates times          | Constant Distance Operator               | 0.2170         | -      | 0.2248 | -      | 0.2228 | -      | 0.2402 | -      |
|                      | Rate Normal Operators                     | 0.1302         | 0.2604 | 0.1349 | 0.2698 | 0.1337 | 0.2674 | 0.1337 | 0.2674 | 0.1441 | 0.2882 |
|                      | Stdev Scale Operator                      | 0.0260         | 0.0260 | 0.0270 | 0.0270 | 0.0267 | 0.0267 | 0.0267 | 0.0267 | 0.0288 | 0.0288 |
|                      | Mean Scale Operator                       | -              | -      | -     | -      | 0.0089 | 0.0089 | 0.0089 | 0.0089 | -      | -      |
|                      | Tree UpperDown Operator                   | -              | -      | -     | -      | 0.0267 | 0.0267 | 0.0267 | 0.0267 | -      | -      |
|                      | Internal Node Time Scale Operator         | 0.0174         | 0.0260 | 0.0280 | 0.0270 | 0.0178 | 0.0267 | 0.0178 | 0.0267 | 0.0192 | 0.0288 |
|                      | Root Age Scale Operator                   | 0.0174         | 0.0260 | 0.0280 | 0.0270 | 0.0178 | 0.0267 | 0.0178 | 0.0267 | 0.0192 | 0.0288 |
|                      | All Node Times Uniform Operator           | 0.1910         | 0.2604 | 0.1978 | 0.2698 | 0.1961 | 0.2674 | 0.1961 | 0.2674 | 0.2113 | 0.2882 |
|                      | Subtree Slide Operator                    | 0.1302         | 0.1302 | 0.0989 | 0.0989 | 0.1337 | 0.1337 | 0.1337 | 0.1337 | 0.4410 | 0.1441 |
|                      | Narrow Exchange Operator                  | 0.1302         | 0.1302 | 0.0989 | 0.0989 | 0.1337 | 0.1337 | 0.1337 | 0.1337 | 0.0480 | 0.0480 |
|                      | Wide Exchange Operator                    | 0.0434         | 0.0434 | 0.0270 | 0.0270 | 0.0267 | 0.0267 | 0.0267 | 0.0267 | 0.0480 | 0.0480 |
|                      | Wilson Balding Operator                   | 0.0434         | 0.0434 | 0.0270 | 0.0270 | 0.0267 | 0.0267 | 0.0267 | 0.0267 | 0.0480 | 0.0480 |
|                      | Birth Rate Scale Operator                 | 0.0434         | 0.0434 | 0.0629 | 0.0629 | -     | -      | -     | -      | 0.0288 | 0.0288 |
|                      | Death Rate Scale Operator                 | -              | -      | 0.0629 | 0.0629 | -     | -      | -     | -      | -      | -      |
|                      | Population Size Scale Operator            | -              | -      | -     | -      | 0.0267 | 0.0267 | 0.0267 | 0.0267 | -      | -      |
| Substitution model   | Kappa Scale Operator                      | 0.0087         | 0.0087 | 0.0009 | 0.0009 | 0.0009 | 0.0009 | 0.0009 | 0.0009 | 0.0192 | 0.0192 |
|                      | Frequencies Delta Exchange Operator       | 0.0017         | 0.0017 | 0.0009 | 0.0009 | 0.0009 | 0.0009 | 0.0009 | 0.0009 | 0.0010 | 0.0010 |

Note:
1: Random walk operator and Swap operator in Cons configuration, Random walk operator, Scale operator and Swap operator in Category configuration.
2: The operator introduced in Appendix section 4 is used in Cons configuration, a Scale operator is used in Category configuration.
-: The parameter is not sampled and no operator is assigned.
Fig. 16 The tree used to simulate sequence alignment. The taxa are denoted by t1 to t20. The divergence times are drawn near the node.

and 21. Overall, we conclude that the proposed operators are able to provide better performance, because the figures suggest that Cons configuration requires less running time and have larger ESS for most parameters in most simulations. Especially, for those poorly estimated parameters in Category configuration, the improvement is more obvious. For data sets such as primates and simulated data with 500 sites, the running time is slightly larger in Cons configuration, but the ESS are much larger, which makes it acceptable to reduce the MCMC chain length and get the same performance.

3.4 Efficiency measured by proposals The operators introduced in the paper utilise a random walk proposal for the new node time, which draws a random number from a uniform distribution and moves the node uniformly on the branch. However, others proposals, such as a Bactrian proposal [48] and a Beta proposal [49], assign

Fig. 17 Running time and ESS using Anolis data
a specific distribution on the new node time so that it is more probable to move to a certain height on the branch, either far away from or close to its original position. This section applied Random walk proposal (the operators in this paper), Bactrian proposal and Beta proposal to the three data sets, and the results are compared to those using Category configuration. The comparisons are shown in Figs. 22, 23, and 24. It is indicated that Beta proposal achieved worst performance in the three analysed data sets. The performance of the Constant Distance operator (Random walk) and Bactrian proposal achieved similar performance in RSV2 data set, while Bactrian proposal provided larger ESS per hour for most parameters in HIV-1 data set. Therefore, it still

![Diagram](image1)

**Fig. 18** Running time and ESS using RSV2 data

![Diagram](image2)

**Fig. 19** Running time and ESS using HIV-1 data
Fig. 20 Running time and ESS using primates data

Simulated data with 500 sites

Simulated data with 1000 sites

Fig. 21 Running time and ESS using simulated data
needs further investigation to demonstrate the effectiveness of different proposals when analysing various data sets. Our current implementation of the operators enables users to specify which proposal style will be used in Beast2 analysis.

4. uclstddevScaleOperator: a scale operator on standard deviation
It should be noted that the proposed ConstantDistance operator parameterises branch rates as continuous random variables, instead of discrete rate categories as is used...
in current BEAST2 settings. In uncorrelated relaxed clock model, branch rates are assumed to have a lognormal prior distribution, where the real mean is fixed to 1 and the standard deviation (denoted by Ucldstdev) is usually sampled with a hyper prior such as gamma($\alpha = 0.5396, \beta = 0.3819$). When a new Ucldstdev is proposed in one state during MCMC sampling by normal operators, the probability of all rates change as well under the new log normal distribution. Therefore, the authors implemented a separate operator working on Ucldstdev, which is able to solve this problem properly.

The first step is to propose a new Ucldstdev by a scale operation, which multiplies current Ucldstdev by a random factor, as is shown in Eq. (38).

$$Ucldstdev' = Ucldstdev \times \text{scale}$$

where \( \text{scale} = \text{Factor} + [\xi \times (\frac{1}{\text{Factor}} - \text{Factor})] \) and \( \xi \) is a random variable from a Uniform(0,1), Factor is a user-defined parameter to specify how bold the proposal is.

Secondly, all the branch rates are proposed based on the new \( Ucldstdev' \), given the probability of original \( Ucldstdev \), which is calculated using Eq. (39).

$$r_i' = \text{icdf}_{i ucldstdev'} \left[ \text{cdf}_{i ucldstdev}(r_i) \right]$$

where the notations \( \text{cdf}(\cdot) \) and \( \text{icdf}(\cdot) \) represent the cumulative and inverse cumulative density function of log normal distribution. Because of the calculation of \( \text{cdf}(\cdot) \) and \( \text{icdf}(\cdot) \) for each branch rates, the "Cons" configuration requires more running time than "category", as is discusses in "Performance comparison" section. However, it is acceptable as ConstantDistance operator gives larger ESS.

Finally, it is important to return the corrected hastings ratio, since the proposal is associated with one random variable, \( Ucldstdev \) and \((2n - 1)\) branch rates. As is shown in Eq. (40), the ratio includes the scale operation and rates changing under the same probability.

$$I_{Ucldstdev} = \frac{1}{\text{scale}} \prod_{i=1}^{2n-1} \frac{\partial \text{icdf}_{Ucldstdev} \left[ \text{cdf}_{Ucldstdev}(r_i) \right]}{\partial r_i}$$

In the comparison of ESS for the clock standard deviation (denoted by uclstd.dev in Fig. 3), we specified a normal scale operator in "Category" configuration. In "Cons" configuration, the UcldstdevScaleOperator is used to sample the clock standard deviation of continuous rates. To avoid the concern that the difference between "Category" and "Cons" is a result of how rates are parameterised (i.e. discrete or continuous), we set another configuration where continuous rates are sampled without using the ConstantDistance operator (denoted by "NoCons" configuration). The weights of the operators in "NoCons" are the same as those in "Category" which is detailed in Table 7. We ran the analysis using the three real data sets (Anolis, RSV2 and HIV-1) and the comparison of ESS per hour between "Category", "Cons" and "NoCons" is summarised in Fig. 25. The figure shows ESS per hour in \( \log_{10} \) space of uclstd.dev in 20 independent MCMC chains. As can be seen, "Cons" configuration gives similar performance, comparing with "Category". This indicates...
UcldstdevScaleOperator works properly on continuous rates. Moreover, ESS per hour is much larger in "Cons" than in "NoCons", where both continuous rates are sampled. Therefore, the proposed operators contribute to the improved performance. However, we noticed that the rate parameterisation does have some mixing issues in MCMC chains. In the future, we will further investigate how to parameterise branch rates to get better performance when using the proposed operators.

Abbreviations
MCMC: Markov chain Monte Carlo; ESS: effective sample sizes; HPD: highest posterior density; Category: Using the current operators in BEAST2 to sample discrete rate categories; Cons: Using the Constant Distance operator to sample continuous rates specified by an uncorrelated related clock model

Acknowledgements
Thanks are due to Jordan Douglas for assistance with the benchmarking experiments. The author(s) wish to acknowledge the use of New Zealand eScience Infrastructure (NeSI) high performance computing facilities, consulting support and/or training services as part of this research. New Zealand’s national facilities are provided by NeSI and funded jointly by NeSI’s collaborator institutions and through the Ministry of Business, Innovation & Employment’s Research Infrastructure programme. URL https://www.nesi.org.nz.

Authors’ contributions
RZ developed the operator and was a major contributor in writing the manuscript. AJD supervised the implementation of the operator and the writing process of the manuscript. Both authors read and approved the final manuscript.

Funding
The author(s) would like to acknowledge support from a Royal Society of New Zealand Marsden award (UO3161; 16-UOA-277). This work is also partially supported by scholarship from China Scholarship Council (No.201706990021).

Availability of data and materials
The source code of the proposed operator and the data sets analysed during the current study are available in the Github repository (https://github.com/Rong419/ConstantDistanceOperator.git).

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
1 School of Computer Science, University of Auckland, Auckland, New Zealand.
2 School of Biological Sciences, University of Auckland, Auckland, New Zealand.

Received: 16 April 2019 Accepted: 1 April 2020
Published online: 14 May 2020

References
1. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. Equation of state calculations by fast computing machines. J Chem Phys. 1953;21(6):1087–92.
2. Hastings WK. Monte carlo sampling methods using markov chains and their applications. Biometrika. 1970;57(1):97–109.
3. Zuckerkandl E, Pauling L. Evolutionary divergence and convergence in proteins. 196597–166. https://doi.org/10.1016/b978-1-4832-2734-4.50017-6.
4. Yang Z, Rannala B. Bayesian phylogenetic inference using dna sequences: a markov chain monte carlo method. Mol Biol Evol. 1997;14(7):717–24.
5. Rannala B, Yang Z. Bayes estimation of species divergence times and ancestral population sizes using dna sequences from multiple loci. Genetics. 2003;164(4):1645–56.
6. Thorne JL, Kishino H, Painter IS. Estimating the rate of evolution of the rate of molecular evolution. Mol Biol Evol. 1998;15(12):1647–57. https://doi.org/10.1093/oxfordjournals.molbev.a023892.
7. Drummond AJ, Ho SY, Phillips MJ, Rambaut A. Relaxed phylogenetics and dating with confidence. PLoS Biol. 2006;4(5):88.
8. Knapp M, Stockler K, Havell D, Deltsch F, Sebastiani F, Lockhart PJ. Relaxed molecular clock provides evidence for long-distance dispersal of nothofagus (southern beech). PLoS Biol. 2005;3(1):14.
9. Smith SA, Beaulieu JM, Donoghue MJ. An uncorrelated relaxed-clock analysis suggests an earlier origin for flowering plants. Proc Natl Acad Sci. 2010;107(13):5897–902. https://doi.org/10.1073/pnas.1001225107.
10. Ho SY, Phillips MJ, Drummond AJ, Cooper A. Accuracy of rate estimation using relaxed-clock models with a critical focus on the early metazoan radiation. Mol Biol Evol. 2005;22(5):1355–63.
11. Lepage T, Bryant D, Philippe H, Larillot N. A general comparison of relaxed molecular clock models. Mol Biol Evol. 2007;24(12):2669–80. https://doi.org/10.1093/molbev/msm234.
12. Kober K, Stamatatou A, Flouri T. Efficient detection of repeating sites to accelerate phylogenetic likelihood calculations. Syst Biol. 2017;66(2):205–17.

13. Guindon S. Bayesian estimation of divergence times from large sequence alignments. Mol Biol Evol. 2010;27(8):1768–81.

14. Res W, Yang Z. Approximate likelihood calculation on a phylogeny for Bayesian estimation of divergence times. Mol Biol Evol. 2011;28(7):2161–72.

15. Ayres DL, Darling A, Zwickl DJ, Belfer P, Holder MT, Lewis PO, Huelsenbeck JP, Ronquist F, Swoford DL, Cummings MP, et al. Beagle: an application programming interface and high-performance computing library for statistical phylogenetics. Syst Biol. 2011;61(1):170–3.

16. Lakner C, Van Der Mark P, Huelsenbeck JP, Larget B, Ronquist F. Efficiency of markov chain monte carlo tree proposals in Bayesian phylogenetics. Syst Biol. 2008;57(1):86–103.

17. Hohna S, Drummond AJ. Guided tree topology proposals for bayesian phylogenetic inference. Syst Biol. 2012;61(1):1–11. https://doi.org/10.1093/sysbio/sys074.

18. Bouckaert R, Heled J, Kühnert D, Vaughan T, Wu CH, Xie D, Suchard MA, Drummond AJ. Beast 2: a software platform for bayesian evolutionary analysis. PLoS Comput Biol. 2014;10(4):1003537.

19. Drummond A, Nicholls G, Rodrigo A, Solomon W. Estimating the relative rates of molecular evolution on a phylogenetic tree: a nonparametric approach. Mol Biol Evol. 1998;15(1):1–8.

20. Suchard MA. Stochastic models for horizontal gene transfer: taking a random walk through tree space. Genetics. 2005;170(1):419–31. https://doi.org/10.1534 genetics.103.025692.

21. Hisatoki T. Monte Carlo filter using the genetic algorithm operators. J Stat Comput Simul. 1997;59(1):1–13.

22. Hohna S, Defoin-Platel M, Drummond AJ. Clock-constrained tree proposal operators in bayesian phylogenetic inference. In: 2008 8th IEEE International Conference on Bioinformatics and BioEngineering. IEEE; 2008. https://doi.org/10.1109/ibibe.2008.4696663.

23. Ramíntej A, Drummond AJ, Xie D, Bao G, Suchard MA. Posterior Summarization in Bayesian Phylogenetics Using Tracer 1.7. Systematic Biology. 2018;67(5):901–904. https://doi.org/10.1093/sysbio/syy032.

24. Davyd AP. The well-calibrated bayesian. J Am Stat Assoc. 1982;77(379):605–10.

25. Hasegawa M, Kishino H, Yano T-a. Dating of the human-apes splitting by amolecular clock of mitochondrial dna. J Mol Evol. 1985;22(2):160–74.

26. Finstermeier K, Zinner D, Brameier M, Meyer M, Kreuz E, Hofreiter M, Roos C. A mitogenomic phylogeny of living primates. PLoS one. 2013;8(7):59504.

27. Jackman TR, Larson A, De Queiroz K, Lewis PO. Phylogenetic relationships and tempo of early diversification in anolis lizards. Syst Biol. 1999;48(2):254–85.

28. Zlateva KT, Lemey P, Vandamme A-M, Van Ranst M. Molecular evolution and circulation patterns of human respiratory syncytial virus subgroup a: positively selected sites in the attachment g glycoprotein. J Virol. 2004;78(9):4675–83.

29. Zlateva KT, Lemey P, Moes E, Vandamme A-M, Van Ranst M. Genetic variability and molecular evolution of the human respiratory syncytial virus subgroup b attachment g protein. J Virol. 2005;79(4):9157–67.

30. Shankarappa R, Margolick JB, Gange SJ, Rodrigo AG, Upchurch D, Farzadegan H, Gupta P, Rinaldo CR, Learn GH, et al. Consistent viral evolutionary changes associated with the progression of human immuno deficiency virus type 1 infection. J Virol. 1999;73(12):10489–502.

31. BEAST2 Data Sets. https://github.com/CompEvol/beast2/tree/master/examples/nexus. Accessed 13 Dec 2019.

32. Cooper A, Lalueza-Fox C, Anderson S, Rambaut A, Austin J, Ward R. Complete mitochondrial genome sequences of two extinct mosas clarify ratite evolution. Nature. 2001;409(6821):704.

33. TreeTraceAnalysis. https://github.com/CompEvol/beast2/blob/master/src/TreeTraceAnalysis.java. Accessed 13 Dec 2019.

34. Lefort F, Lefort V, Ansimova M, Hordijn W, Gascuel O. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML3.0. Syst Biol. 2010;59(3):307–21.

35. Beal ML. Variational Algorithms for Approximate Bayesian Inference. England: University of London; 2003, p. 281.

36. Zhang C, IV FAM. Variational bayesian phylogenetic inference. In: International Conference on Learning Representations. 2019. https://openreview.net/forum?id=SJmJkR9FX. Accessed 13 Mar 2019.

37. Sanderson MJ. r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. Bioinformatics. 2003;19(2):301–2.

38. PhyML3.0. New Algorithms, Methods and Utilities. http://www.atgc-montpellier.fr/phyml/. Accessed 13 Dec 2019.

39. Guindon S, Dufayard J-F, Lefort V, Anisimova M, Hordijn W, Gascuel O. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML3.0. Syst Biol. 2010;59(3):307–21.

Publisher’s Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.