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

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

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

* jordan.douglas@auckland.ac.nz

Abstract

Author summary

Introduction

Models

Preliminaries

Let $\mathcal{T}$ be a binary rooted time tree with N taxa (and $2N-2$ branches). Let L be the number of sites within the multiple sequence alignment D, and let L_{eff} be the *effective* number of sites in the alignment (ie. the number of site patterns). 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 rate standard deviation σ and other model parameters θ. $\vec{\mathcal{R}}$ 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 **Rate parameterisations**.

Under the *relaxed clock model*, each internal and leaf node is assigned a substitution rate $r_i = r(\mathcal{R}_i)$, which corresponds to its parent branch. There are a total of $|\vec{\mathcal{R}}| = 2N-2$ rates, which are independently distributed under the relaxed clock model prior [1].

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 [2]. 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**.

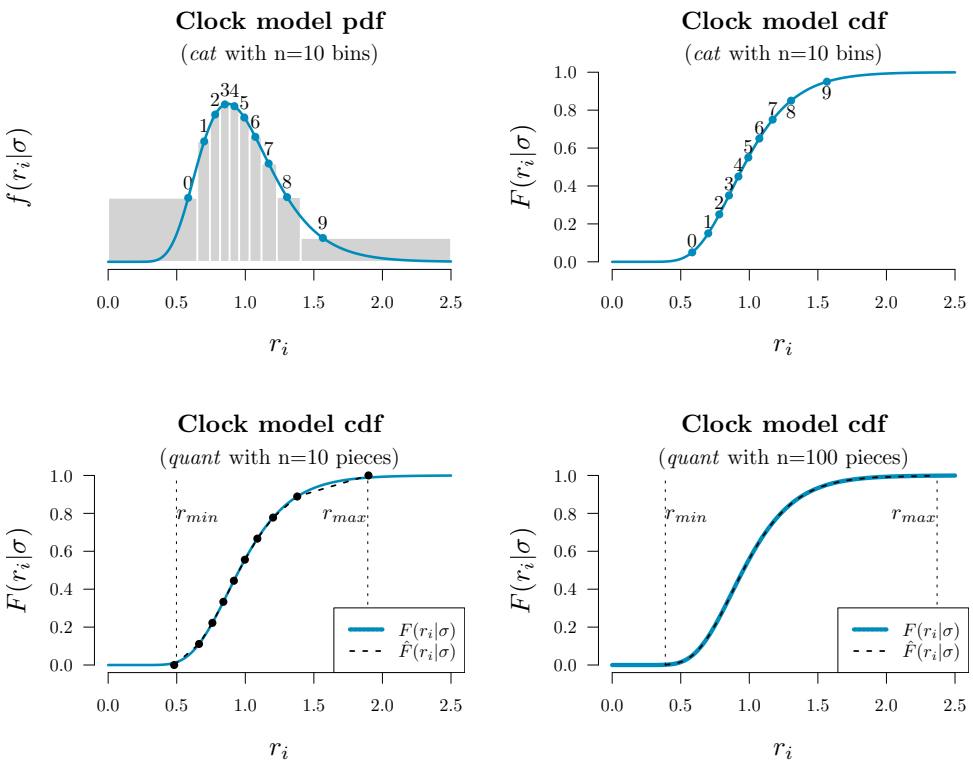

Fig 1. Methods of rate parameterisation. The *cat* and *quant* approximations are plotted on top of the true underlying rate prior distribution (*real*). In this example, rates are drawn from a LogNormal($\mu = -0.045, \sigma = 0.3$) distribution. The probability density function (pdf) and cumulative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

$$r(\vec{\mathcal{R}}) = \vec{\mathcal{R}}\,. \tag{2}$$

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

$$p(\mathcal{R}_i|\sigma) = \frac{1}{\mathcal{R}_i\sigma\sqrt{2\pi}} \exp\big(-\frac{(\ln \mathcal{R}_i - \mu)^2}{2\sigma^2}\big) \quad (\text{ LogNormal}(\mu, \sigma)\ ), \text{ or} \tag{3}$$

$$p(\mathcal{R}_i|\sigma) = p(\mathcal{R}_i) = e^{-\mathcal{R}_i} \quad (\text{ Exponential}(\lambda = 1)\ ) \tag{4}$$

where $\mu = -0.5\sigma^2$ is set such that the expected value of the log-normal distribution is 1.

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. By maintaining genetic distances the likelihood can also be maintained. These operators account for the

correlation which exists between branch rates and branch times – a correlation which is induced by the likelihood function.

2. Categories

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

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

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 was suggested in the original BEAST2 relaxed clock paper [1] and has been widely used. However, the constant distance operators since introduced by Zhang and Drummond 2020 – 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*. Moreover, the implementation of the *cat* model has come under criticism [3].

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 the class of operators described by Zhang and Drummond 2020.

A potential disadvantage of the *quant* method would be the computational requirements of continuously evaluating the i-cdf, especially for trees with large N. 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 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**.

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 [4]. Yang et al. have challenged the widely used uniform proposal kernel in place of the Bactrian kernel [5, 6]. The Bactrian(m) distribution is defined as the sum of two Normal distributions:

$$\Sigma \sim \text{Bactrian}(m) \equiv \frac{1}{2}\text{Normal}(-m, 1-m^2) + \frac{1}{2}\text{Normal}(m, 1-m^2) \tag{11}$$

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 [5] 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.

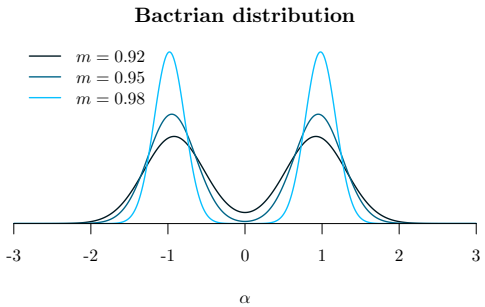

Fig 2. The Bactrian proposal kernel. Y-axis corresponds to probability density $f(\Sigma|m)$.

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**	$x' \leftarrow x + s\Sigma$	σ, r, q
2	**Scale operator**	$x' \leftarrow x \times e^{s\Sigma}$	σ, r
3	**Interval operator**	$y \leftarrow \frac{u-x}{x-l} \times e^{s\Sigma}$ $x' \leftarrow \frac{u+l*y}{y+1}$	$q \ \ (l=0, u=1)$
4	**Constant distance operators**	$x' \leftarrow x + s\Sigma$	t

Table 1. Summary of proposal kernels $q(x'|x)$ of clock model operators. In each operator, Σ 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 σ, 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 its domain ie. $l < x' < u$.

Narrow Exchange Rate

The **Narrow Exchange** operator [7], widely used in BEAST [8,9] 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 (ie. $t_D > t_C$).

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 [11] 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 [12] and Continuous Change [13]). 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 (ie. $t_D{}' \leftarrow t_D$), we also consider NERw - a special case of the NER operator which embarks t_D on a random walk:

$$t_D{}' \leftarrow t_D + s\Sigma \tag{12}$$

for random walk step size $s\Sigma$ where s is a tunable scaler parameter and Σ 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.

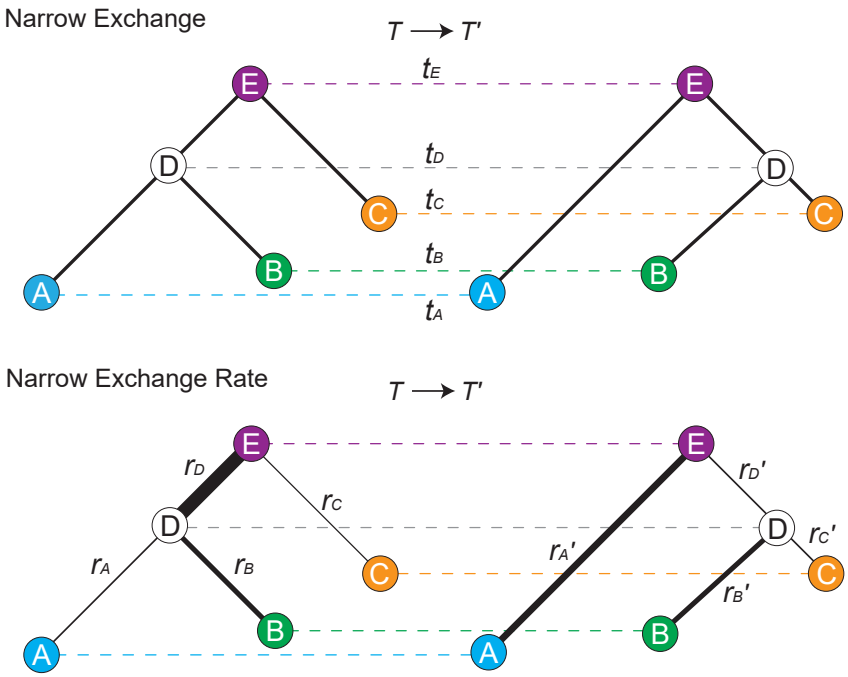

Fig 3. Depiction of Narrow Exchange and Narrow Exchange Rate operators. Proposals are denoted by $\mathcal{T} \to \mathcal{T}'$. The vertical axis corresponds to node height t. In the bottom figure, branch rates r are indicated by line thickness. In this example, the $\mathcal{D}_{AE}$ and $\mathcal{D}_{CE}$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

$$\begin{aligned}
\mathcal{D}_{AB}: \quad & r_A(t_D - t_A) + r_B(t_D - t_B) = \\
& r_A'(t_E - t_A) + r_D'(t_E - t_D') + r_B'(t_D' - t_B) && (13)\\
\mathcal{D}_{AC}: \quad & r_A(t_D - t_A) + r_D(t_E - t_D) + r_C(t_E - t_C) = \\
& r_A'(t_E - t_A) + r_D'(t_E - t_D') + r_C'(t_D' - t_C) && (14)\\
\mathcal{D}_{AE}: \quad & r_A(t_D - t_A) + r_D(t_E - t_D) = \\
& r_A'(t_E - t_A) && (15)\\
\mathcal{D}_{BC}: \quad & r_B(t_D - t_B) + r_D(t_E - t_D) + r_C(t_E - t_D) = \\
& r_B'(t_D' - t_B) + r_C'(t_D' - t_C) && (16)\\
\mathcal{D}_{BE}: \quad & r_B(t_D - t_B) + r_D(t_E - t_D) = \\
& r_B'(t_D' - t_B) + r_D'(t_E - t_D') && (17)\\
\mathcal{D}_{CE}: \quad & r_C(t_E - t_C) = \\
& r_C'(t_D' - t_C) + r_D'(t_E - t_D') && (18)
\end{aligned}$$

Further constraints are imposed by the model itself:

$$r_i > 0 \text{ and } r_i' > 0 \text{ for } i \in \{A, B, C, D\} \quad (19)$$

$$\max\{t_B, t_C\} < t_D' < t_E. \quad (20)$$

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 (ie. $\{\}, \{\mathcal{D}_{AB}\}, \{\mathcal{D}_{AC}\}, \ldots, \{\mathcal{D}_{AB}, \mathcal{D}_{AC}, \mathcal{D}_{AE}, \mathcal{D}_{BC}, \mathcal{D}_{BE}, \mathcal{D}_{CE}\}$) 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

$$J = \begin{bmatrix} \frac{\partial {r_A}'}{\partial r_A} & \frac{\partial {r_A}'}{\partial r_B} & \frac{\partial {r_A}'}{\partial r_C} & \frac{\partial {r_A}'}{\partial r_D} \\ \frac{\partial {r_B}'}{\partial r_A} & \frac{\partial {r_B}'}{\partial r_B} & \frac{\partial {r_B}'}{\partial r_C} & \frac{\partial {r_B}'}{\partial r_D} \\ \frac{\partial {r_C}'}{\partial r_A} & \frac{\partial {r_C}'}{\partial r_B} & \frac{\partial {r_C}'}{\partial r_C} & \frac{\partial {r_C}'}{\partial r_D} \\ \frac{\partial {r_D}'}{\partial r_A} & \frac{\partial {r_D}'}{\partial r_B} & \frac{\partial {r_D}'}{\partial r_C} & \frac{\partial {r_D}'}{\partial r_D} \end{bmatrix}. \tag{21}$$

Computing the determinant $|J|$ invokes standard analytical differentiation and linear algebra libraries of MATLAB. 6 of the 54 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.

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 [14]. In previous work, parsimony scores and conditional clade probabilities of neighbouring trees have been employed by guided tree operators [15–17] and the latter has also been explored as the basis of adaptive tree operators [15,17]. The (adaptive) mirror kernel [6] learns a target distribution which acts as a 'mirror image' of the current point x. The adaptable variance multivariate normal (AVMVN) kernel [9,18] 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 [18].

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 dependency on the topology of the tree.

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 Σ_N. Hence, the kernel operates on the logarithmic or logistic transformation of the N leaf branch rates, depending on the rate parameterisation:

$$x_i = \begin{cases} \log r_i \text{ for } \textit{real} \\ \log \frac{q_i}{1-q_i} \text{ for } \textit{quant} \end{cases} \tag{22}$$

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

$$\mathcal{AVMVN}(x) = (1-\beta)\mathcal{MVN}\big(x, \frac{\Sigma_N}{N}\big) + \beta\mathcal{MVN}\big(x, \frac{\mathbb{I}_N}{N}\big), \tag{23}$$

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

The AVMVN proposal kernel is computed as

$$x' \leftarrow x + \sum_{i=1}^{N}\sum_{j=i}^{N} c_{i,j} \times s\Sigma \tag{24}$$

$$\text{where } c = \texttt{cholesky}\{(1-\beta)\frac{\Sigma_N}{N} + \beta\frac{\mathbb{I}_N}{N}\}. \tag{25}$$

The $\texttt{cholesky}\{Y\}$ decomposition returns a lower diagonal matrix L, with positive real diagonal entries, such that $Y = LL'$ [19, 20]. s is a tunable step size parameter and Σ 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 [9].

In Results, we evaluate this operator for its ability to estimate leaf rates. As the size of the covariance matrix Σ_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.

Clock model averaging

In Bayesian model averaging (BMA), competing models are estimated in the posterior distribution alongside other parameters and are thus directly selected through usual Bayesian inference methods [21]. This is in contrast to other methods – such as path sampling [22] and stepping-stone sampling [23] – which operate outside of this central Bayesian inference process (in this case the MCMC algorithm). BMA has been applied to many domains [21] – including astrophysics [24], geographical information systems [25], and molecular biology [26] – and in phylogenetics has been applied to selection of substitution models [27–29], rate heterogeneity models [29], and clock models [30, 31].

Li and Drummond 2012 used Bayesian model averaging to select clock model prior distributions – LogNormal, Exponential, and Inverse Gaussian [30]. This was achieved

by a model indicator that points to the applicable clock model. However, due to its discrete nature, sampling the model indicator can be inefficient.

As a preliminary non-phylogenetic analysis, we explored the statistical power of distinguishing between log-normally and exponentially distributed data, and how this depends on sample size $2N-2$ (for N taxa) and standard deviation σ (for log-normally distributed data). Models are distinguished using the Bayesian information criterion (BIC):

$$BIC = \begin{cases} \log(2N-2) - 2\log\left(\hat{L}\ P(\hat{\sigma})\right) & \text{for LogNormal}(-0.5\sigma^2, \sigma) \\ -2\log\left(\hat{L}\right) & \text{for Exponential(1)} \end{cases} \quad (26)$$

where $\hat{L}$ is the maximim likelihood and $P(\hat{\sigma})$ is the prior density of σ. This analysis reveals that when there are a small number of rates, data simulated from a lognormal distribution is often best modelled with an exponential distribution, especially when σ is close to 1 (Fig. 4). However, for larger trees $N > 30$, clock model averaging should be a viable approach.

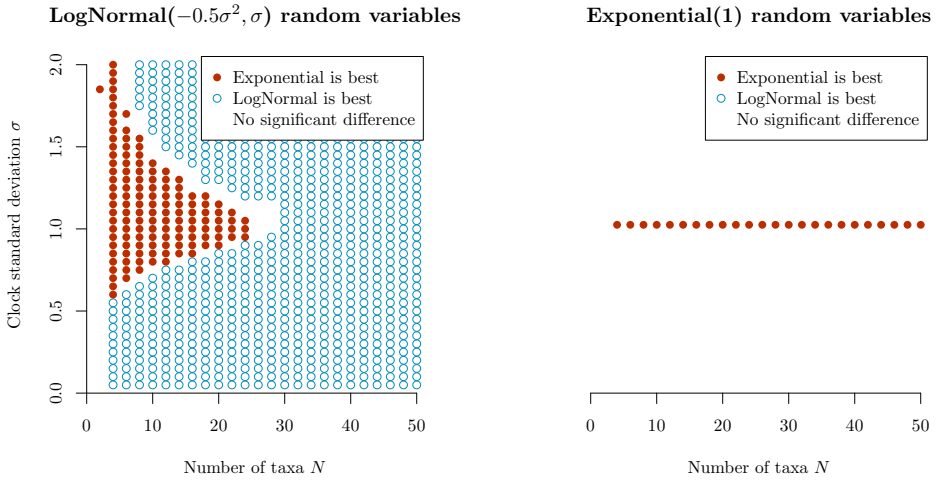

Fig 4. Ability to resolve LogNormal and Exponential distributed data. *Left:* at each grid point, $2N-2$ random variables are sampled from a LogNormal$(-0.5\sigma^2, \sigma)$ distribution. The point is assessed for its fit to i) a LogNormal$(-0.5\hat{\sigma}^2, \hat{\sigma})$ distributon, and ii) an Exponential(1) disribution using the Bayesian information criterion (BIC). This is repeated 1000 $\times$ at each grid point and the two sets of BIC scores are compared using a paired t-test ($p = 0.001$), which determines the colour of the point on the plot. *Right:* points are instead sampled from an Exponential(1) disribution. These simulations show that clock model averaging is most applicable for trees with more than 30 taxa.

Results

Assessment criteria and datasets

To avoid a cross-product explosion, the three targets for clock model improvement (**Rate parameterisations**, **Bactrian proposal kernel**, and **Narrow Exchange Rate**) are evaluated sequentially, in the order presented in this paper. The setting(s)

which are considered to be the best in each step are then incorporated into the following step. This protocol and its outcomes are summarised in **Fig. 5**.

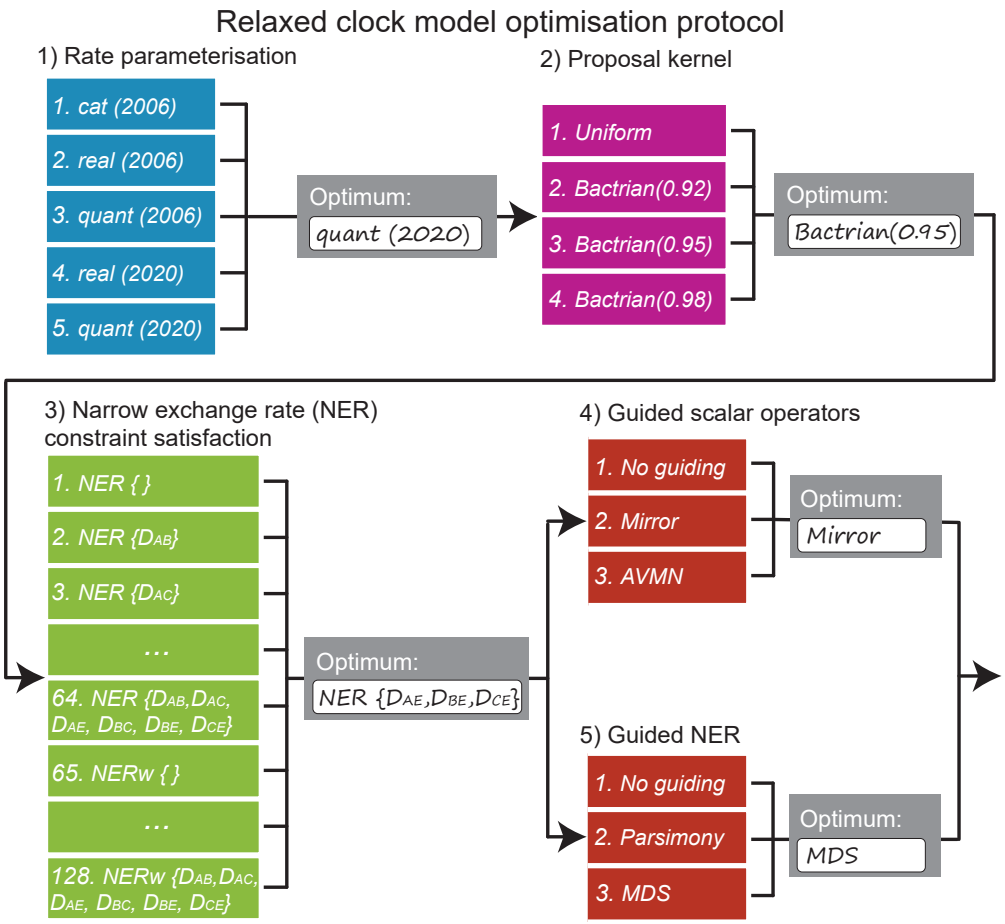

Fig 5. Protocol for optimising methodology settings. The three areas (detailed in **Models**) are optimised sequentially, where the best setting from each step is used when optimising the following step.

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

2. Time to convergence. Two independent MCMC chains are run and the time is measured until: a) the absolute difference in clade posterior probability between the two chains is less than 0.05 for all clades, b) the Rubin-Gelman statistic $\hat{R}$ [32] of every estimated parameter is less than 1.05, and c) the effective sample size [33] of every estimated parameter is greater than 200 in each chain.

3. Mixing of parameters. Key parameters are evaluated for the number of effective samples generated per hour (ESS/hr).

For the latter two criteria, methodologies are benchmarked using both simulated and empirical datasets – the latter were compiled [34] and partitioned [35] by Lanfear et al. as 'benchmark alignments' (**Table 2**). Each setting is run 10× per dataset and the average statistics across the $10 \times 2 = 20$ chains are reported. .

Methodologies are benchmarked using the Intel(R) Xeon(R) Gold 6138 CPU (2.00 GHz), with two processing units per pair of MCMC chains.

	N	P	L (kb)	L_{eff} (kb)	**Description**
1	6	16	13.7	1.1	Spiders (Richart 2015 [36])
2	18	16	6.7	0.9	Gallopheasants (Meiklejohn 2016 [37])
3	33	16	5.1	1.5	Birds (McCormack 2013 [38])
4	40	1	3.0	0.6-1.8	*Simulated data* $\times 10$
5	44	3	1.9	0.8	Bark beetles (Cognato 2001 [39])
6	51	6	5.4	1.8	Southern beeches (Sauquet 2011 [40])
7	70	3	2.2	0.9	Caterpillars (Kawahara 2013 [41])
8	94	4	1.8	0.8	Bees (Rightmyer 2013 [42])
9	106	1	0.6	0.3	Songbirds (Moyle 2016 [43])
10	110	1	0.5	0.3	Vertebrates (Fong 2012 [44])

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_{eff} are also specified. The 9 empirical datasets are benchmarked $10\times$ per setting, whereas each of the 10 simulated datasets are benchmarked once per setting.

Comparison of rate parameterisations

We compared the three rate parameterisations described in **Rate parameterisations**. All three settings use the standard BEAST2 clock model operators from Drummond et al. 2006 [1]. *real* and *quant* additionally use the constant-distance tree operators described by Zhang and Drummond 2020. 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**.

Fig. 6 shows that the *real 2006* performs considerably worse than any of the other settings. This is due to the poor sampling of the prior under this setting (ie. low ESS of $p(\theta)$). The failure of *real 2006* thus highlights the appeal of 1) the *cat* or *quant* parameterisations, both of which have trivial contributions to the prior density (ie. uniform priors), and 2) the smarter operators used by *real* (Zhang and Drummond 2020). Due to its computational burden, *real 2006* was not benchmarked for all of the datasets in **Table 2**.

Our results show that the *quant* parameterisation yields the best performance with respect to effective samples per hour. *quant* outperforms *quant 2006*, suggesting that the constant distance operators are effective. Furthermore, *quant* outperforms *real* especially at sampling from the posterior and prior distributions (ie. high ESS/hr for $p(\theta|D)$ and $p(\theta)$). This is most likely because of the uniform prior distribution of rate quantiles.

Overall, *quant* yields a median ESS/hr approximately 150 % faster with respect to sampling the posterior probability, and approximately 30 % faster with respect to sampling branch rates r and clock standard deviation σ, compared with *real*.

Fig 6. Rate parameterisation evaluation. Comparison of ESS/hr (averaged across two independent MCMC chains) with respect to relevant terms – L: likelihood, p: prior density, r: clock rate averaged across all leaves, σ: clock standard deviation. Each point is from one partition sample of the empirical data in **Table 2**.

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{} (ie. Narrow Exchange).

Fig. 7 shows that NER variants which satisfy the genetic distances between nodes B and A (ie. $\mathcal{D}_{AB}$) or between B and C (ie. $\mathcal{D}_{BC}$) 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, 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. 7 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 ie. when clock standard deviation σ 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 $N = 30$) 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. 7** 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 (ie. the operators which are not NER{}). This result is unfortunate however not unexpected,

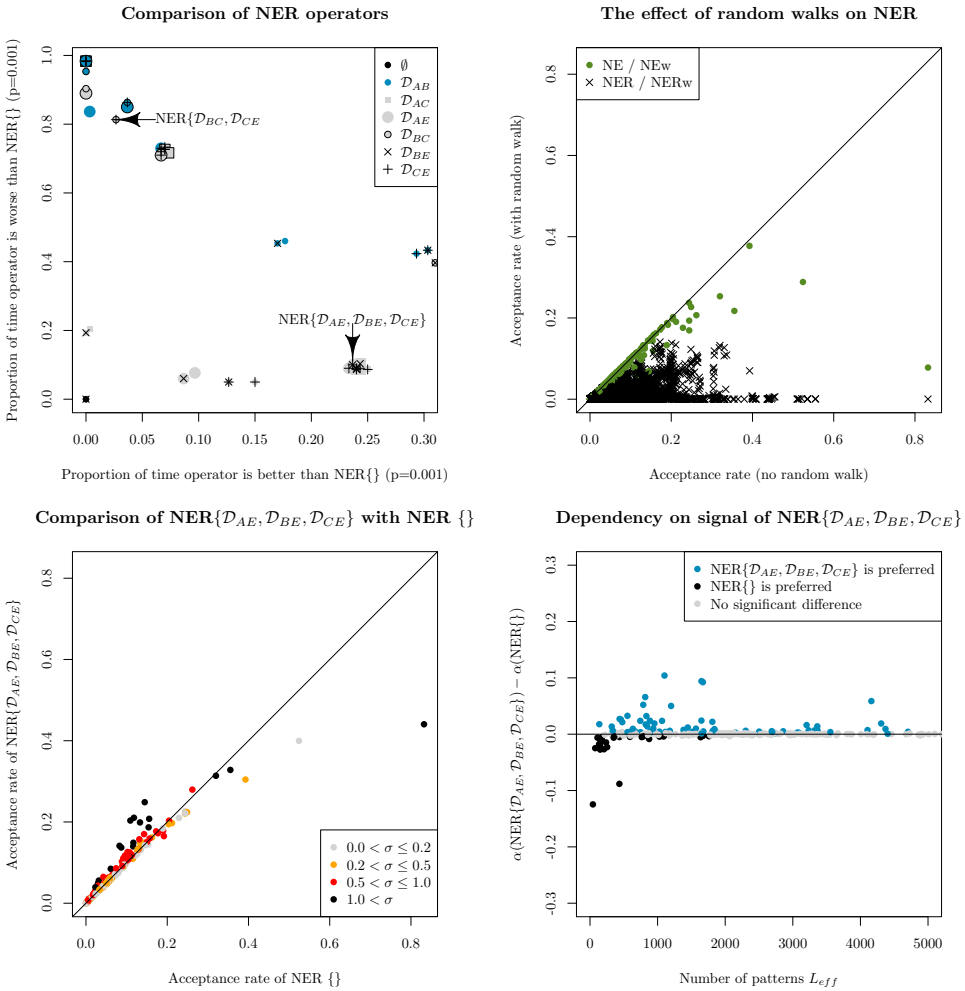

Fig 7. Screening of NER and NERw variants by acceptance rate. Top left: comparison of NER variants with the null operator NER{} (ie. Narrow Exchange). Each of the 48 operators are represented by a single point, uniquely encoded by the point stylings. The number of times each operator is proposed and 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 α of NER$\{\mathcal{D}_{AE}, \mathcal{D}_{BE}, \mathcal{D}_{CE}\}$ and NER{} with the clock model standard deviation σ and the number of patterns L_{eff}. Each point is a single simulation.

and is consistent with Lakner et al. 2008 [11], 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.

Supporting information

S1 Appendix. Rate quantiles. The linear piecewise approximation used in the *quant* parameterisation is described. Tree operators presented by Zhang and Drummond 2020 are extended to the *quant* parameterisation.

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

References

1. Drummond AJ, Ho SY, Phillips MJ, Rambaut A. Relaxed phylogenetics and dating with confidence. PLoS biology. 2006;4(5):e88.

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

3. Rannala B, Yang Z. Inferring speciation times under an episodic molecular clock. Systematic biology. 2007;56(3):453–466.

4. 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.

5. 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.

6. 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.

7. 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.

8. 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.

9. 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.

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. 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.

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

13. 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.

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

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

16. Zhang C, Huelsenbeck JP, Ronquist F. Using Parsimony-Guided Tree Proposals to Accelerate Convergence in Bayesian Phylogenetic Inference. bioRxiv. 2019; p. 778571.

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

18. 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.

19. 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.

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

21. Fragoso TM, Bertoli W, Louzada F. Bayesian model averaging: A systematic review and conceptual classification. International Statistical Review. 2018;86(1):1–28.

22. Lartillot N, Philippe H. Computing Bayes factors using thermodynamic integration. Systematic biology. 2006;55(2):195–207.

23. Xie W, Lewis PO, Fan Y, Kuo L, Chen MH. Improving marginal likelihood estimation for Bayesian phylogenetic model selection. Systematic biology. 2011;60(2):150–160.

24. Parkinson D, Liddle AR. Bayesian model averaging in astrophysics: a review. Statistical Analysis and Data Mining: The ASA Data Science Journal. 2013;6(1):3–14.

25. Barber JJ, Gelfand AE, Silander Jr JA. Modelling map positional error to infer true feature location. Canadian Journal of Statistics. 2006;34(4):659–676.

26. Douglas J, Kingston R, Drummond AJ. Bayesian inference and comparison of stochastic transcription elongation models. PLOS Computational Biology. 2020;16(2):e1006717.

27. Beier BA, Nylander J, Chase MW, Thulin M. Phylogenetic relationships and biogeography of the desert plant genus Fagonia (Zygophyllaceae), inferred by parsimony and Bayesian model averaging. Molecular phylogenetics and evolution. 2004;33(1):91–108.

28. Posada D. jModelTest: phylogenetic model averaging. Molecular biology and evolution. 2008;25(7):1253–1256.

29. Bouckaert RR, Drummond AJ. bModelTest: Bayesian phylogenetic site model averaging and model comparison. BMC evolutionary biology. 2017;17(1):42.

30. 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.

31. Baele G, Li WLS, Drummond AJ, Suchard MA, Lemey P. Accurate model selection of relaxed molecular clocks in Bayesian phylogenetics. Molecular biology and evolution. 2012;30(2):239–243.

32. Gelman A, Rubin DB, et al. Inference from iterative simulation using multiple sequences. Statistical science. 1992;7(4):457–472.

33. Rambaut A, Drummond AJ, Xie D, Baele G, Suchard MA. Posterior summarization in Bayesian phylogenetics using Tracer 1.7. Systematic biology. 2018;67(5):901–904.

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

35. 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.

36. Richart CH, Hayashi CY, Hedin M. Phylogenomic analyses resolve an ancient trichotomy at the base of Ischyropsalidoidea (Arachnida, Opiliones) despite high levels of gene tree conflict and unequal minority resolution frequencies. Molecular Phylogenetics and Evolution. 2016;95:171–182. doi:10.1016/j.ympev.2015.11.010.

37. Meiklejohn KA, Faircloth BC, Glenn TC, Kimball RT, Braun EL. Analysis of a Rapid Evolutionary Radiation Using Ultraconserved Elements: Evidence for a Bias in Some Multispecies Coalescent Methods. Systematic Biology. 2016;65(4):612–627. doi:10.1093/sysbio/syw014.

38. McCormack JE, Harvey MG, Faircloth BC, Crawford NG, Glenn TC, Brumfield RT. A Phylogeny of Birds Based on Over 1,500 Loci Collected by Target Enrichment and High-Throughput Sequencing. PLoS ONE. 2013;8(1):e54848. doi:10.1371/journal.pone.0054848.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. Fong JJ, Brown JM, Fujita MK, Boussau B. A Phylogenomic Approach to Vertebrate Phylogeny Supports a Turtle-Archosaur Affinity and a Possible Paraphyletic Lissamphibia. PLoS ONE. 2012;7(11):e48990. doi:10.1371/journal.pone.0048990.