Quantifying the importance of an inference model in Bayesian phylogenetics

Richèl J.C. Bilderbeek[1,*], Giovanni Laudanno[1], and Rampal S. Etienne[1]

[1]Groningen Institute for Evolutionary Life Sciences, University of Groningen, Groningen, The Netherlands

[*]corresponding author: r.j.c.bilderbeek@rug.nl

December 17, 2019

Summary

1. Phylogenetic trees are current routinely reconstructed from an alignment of character sequences (usually nucleotide sequences). Bayesian tools, such as MrBayes, RevBayes and BEAST2, have gained much popularity over the last decade, as they allow joint estimation of the posterior distribution of the phylogenetic tree and the parameters of the underlying inference model. An important ingredient of these Bayesian approaches is the species tree prior. While in principle the Bayesian framework allows for comparing different species tree priors and hence may elucidate the macroevolutionary processes underlying the species tree, in practice only macroevolutionary models that allow for fast computation of the prior probability are used. An open question is, how accurate the tree estimation is when the real macroevolutionary processes are substantially different from those assumed in the tree prior.

2. Here we present `pirouette`, a free, libre and open-source R package that assesses the inference error made by Bayesian phylogenetics for a given macroevolutionary diversification model. `pirouette` makes use of BEAST2, but its philosophy applies to any Bayesian phylogenetic inference tool.

3. We describe `pirouette`'s usage and the biological scientific question it can answer, including full examples.

4. Last, we discuss the results obtained by the examples and their interpretation.

Keywords: Bayesian model selection, BEAST2, computational biology, evolution, phylogenetics, R, tree prior

1 Introduction

The development of new powerful Bayesian phylogenetic inference tools, such as BEAST [Drummond & Rambaut 2007], MrBayes [Huelsenbeck & Ronquist 2001] or RevBayes [Höhna *et al.* 2016], has been a major advance in constructing phylogenetic trees from character data (usually nucleotide sequences) extracted from extant (but also extinct) organisms, and hence in our understanding of the main drivers and modes of diversification.

BEAST [Drummond & Rambaut 2007] is a typical Bayesian phylogenetics tool, that needs both character data and priors to infer a posterior distribution of phylogenies. Specifically, for the species tree prior - which describes the process of diversification - BEAST has built-in priors such as the Yule [Yule 1925] and (constant-rate) birth-death [Nee *et al.* 1994] models. These simple tree priors are most commonly used, as these have sufficient biological complexity, while being computationally fast. BEAST's successor, BEAST2 [Bouckaert *et al.* 2019], has a package manager, that allows third-party users to extend existing functionalities. For example, one can add novel diversification models by writing a BEAST2 package that contains the likelihood formula of a phylogeny under the novel diversification model, i.e. the prior probability of a species tree. Many such diversification models (and their associated probability algorithms) have been developed, e.g., models in which diversification is time-dependent [Nee *et al.* 1994, Rabosky & Lovette 2008], or diversity-dependent [Etienne *et al.* 2011], or where diversification rates change for specific lineages and their descendants [Etienne & Haegeman 2012, Rabosky 2014, Alfaro *et al.* 2009, Laudanno *et al.* submitted], models that treat speciation as a process that takes time [Rosindell *et al.* 2010][Etienne & Rosindell 2012][Lambert *et al.* 2015], or as a burst of simultaneous branching events [Laudanno *et al.* in preparation], or where diversification rate depends on a trait that has two [Maddison

et al. 2007], or more [FitzJohn 2012] states, even concealed states [Beaulieu & O'meara 2016] or a combination of all these [Herrera-Alsina *et al.* 2018]. Only a few of these diversification models are available as a BEAST2 package.

When a novel diversification model is introduced, its performance in inference should be tested. Part of a model's performance is its ability to recover parameters from simulated data with known parameters (e.g. [Etienne *et al.* 2014]), where ideally the estimated parameter values closely match the known/true values.

Even when a diversification model passes the procedure described above, it is not necessarily used in Bayesian inference. Bayesian phylogenetic inference often requires that the prior probability of the phylogeny according to the diversification model has to be computed millions of times. Therefore, biologically interesting but computationally expensive tree priors are often not implemented, and simpler priors are used instead. This is not necessarily problematic, when the data are very informative, as this will reduce the influence of the tree prior. However, the assumption that tree prior choice is of low importance must first be verified.

There have been multiple attempts to investigate the importance of tree prior choice. For example, recently Sarver et al., [Sarver *et al.* 2019] showed that the choice of tree prior does not substantially affect phylogenetic inferences of diversification rates. Also recently, Duchene et al. [Duchene *et al.* 2018] released a BEAST2 package to assess how well posterior predictive simulations recover a given tree when using the standard diversification models. These studies show how current diversification models compare to one another, but they do not help to assess the importance of a new tree prior.

Here we introduce a method to quantify the importance of a novel tree prior. The method starts with a phylogeny generated by the new model. Next,

nucleotide sequences are simulated on this phylogeny. Then, using the tree priors built-in into BEAST2, a Bayesian posterior distribution of phylogenies is inferred. We then compare the inferred and simulated phylogenies. How to properly perform this comparison forms the heart of our method. Only new diversification models that result in a large discrepancy between inferred and simulated phylogenies will be worth the effort and computational burden to implement a species tree prior for in a Bayesian framework.

Our method is programmed as an R package called `pirouette`. `pirouette` is built on `babette` [Bilderbeek & Etienne 2018], which calls BEAST2 [Bouckaert *et al.* 2019].

2 Description

`pirouette` is written in the R programming language (R Core Team 2013). The goal of `pirouette` is to quantify the importance of a tree prior. It does so by measuring the inference error made for a given reconstructed phylogeny, simulated under a (usually novel) diversification model. We refer to the true model that generated the given tree as the 'generative tree model' p_G. Many tree priors have a parameter setting for which they reduce to a standard tree prior. For example, a protracted birth-death model Etienne & Rosindell 2012 reduces to a standard birth-death model when the speciation-completion rate is infinite [i.e. rate λ in Etienne *et al.* 2014, eqs. (2b) and (2c)]. When benchmarking a novel tree prior, one will typically construct phylogenies for different combinations of the diversification model's parameters, to assess under which scenarios the inference error cannot be neglected. While we recommend many replicate simulations when assessing a novel tree prior, our examples contain only one replicate as they are for illustrative purposes only.

`pirouette` is very flexible and allows the user to specify a wide variety of

custom settings. These settings can be grouped in macro-sections, according to how they operate in the pipeline. We summarize them in Table 1 and Table 2.

Although many possible tests can be performed, we show the usage of `pirouette` by introducing its features gradually, yet ending in quantifying the impact a tree prior has in Bayesian inference.

2.1 pirouette's pipeline

We assume the user has a phylogeny simulated with the new diversification model. The pipeline to assess the error BEAST2 makes in inferring this phylogeny then contains the following steps:

1. from the given phylogeny an alignment is simulated under a known alignment model A;
2. from this alignment, according to the specified inference conditions C, an inference model I is chosen (which may differ from the generative model);
3. the inference model and the alignment are used to infer a posterior distribution of phylogenies;
4. the phylogenies in the posterior are compared with the given phylogeny to estimate the error made, according to the error measure E specified by the user;

The pipeline is visualized in Fig. 1. There is also the option to generate a 'twin tree', that goes through the same pipeline. The utility of this twin tree will be explained below.

The first step simulates a DNA alignment from a given phylogeny (Fig. 1, 1a $\rightarrow$ 2a) using the DNA alignment parameters. The DNA alignment parameters consist of a (DNA) root sequence, a (DNA) mutation rate, a clock model and a nucleotide substitution model. The root sequence is the DNA sequence of

Sub-argument	Description	Possible values
`tree_prior`	Macroevolutionary diversification model	Yule, BD, CBS, CCP, CEP
`clock_model`	Clock for the DNA mutation rates	strict, RLN
`site_model`	Nucleotide substitution model	JC, HKY, TN93, GTR
`mutation_rate`	Pace at which mutation occurs	`mutation_rate` $\in \mathbb{R}_{>0}$
`root_sequence`	DNA sequence at the root of the tree	any combination of a, c, g, t
`model_type`	Criterion to select an inference model	Generative, Candidate
`run_if`	Condition under which an inference model is used	Always, Best candidate
`do_measure_evidence`	Sets whether or not the evidence of the model is to be computed	TRUE, FALSE
`error_fun`	Specifies how to measure the error	nLTT, $\lvert\gamma\rvert$
`burn_in_fraction`	Specifies the percentage of initial posterior trees to discard	`burn_in_fraction` $\in [0, 1]$

Table 1: Most important parameter options. Yule = pure birth model (Yule 1925) BD = birth death (Nee *et al.* 1994), CBS = coalescent Bayesian skyline (Drummond *et al.* 2005), CCP = coalescent constant-population, CEP = coalescent exponential-population, JC = Jukes and Cantor (Jukes *et al.* 1969), HKY = Hasegawa, Kishino and Yano (Hasegawa *et al.* 1985), TN93 = Tamura and Nei (Tamura & Nei 1993), GTR = Generalized time-reversible model (Tavaré 1986) RLN = relaxed log-normal clock model (Drummond *et al.* 2006).

Symbol	Macro-argument	Description
G	Generative model	The full setting to produce BEAST2 input data. Its core features are the tree prior p_G, the clock model c_G and the site model s_G.
A	Alignment model	It includes the parts of the generative model that directly affect the alignment generation, like the clock model c_G and the site model s_G. Additional arguments can be provided, such as the mutation rate and the root sequence.
X_i	i-th candidate experiment	Full setting for a Bayesian inference. It is made by a candidate inference model I_i and its inference conditions C_i.
I	Inference model	Phylogenetic inference model to run BEAST2. Likewise the generative model G, its main components are the tree prior p_I, the clock model c_I and the site model s_I.
C	Inference conditions	Conditions under which I is used in the inference. They are composed by the model type, run condition and whether to measure the evidence.
E	Error measure parameters	Errors measurement setup that can be specified providing an error function to measure the difference between the original phylogeny and the inferred posterior. The initial part of the posterior that is reckoned as not representative can be discarded using a burn-in fraction.

Table 2: Definitions of terms and relative symbols used in the main text and in Fig 1. To run the pipeline A, X and E must be specified. Examples can be found in listings 2, 3 and 5.

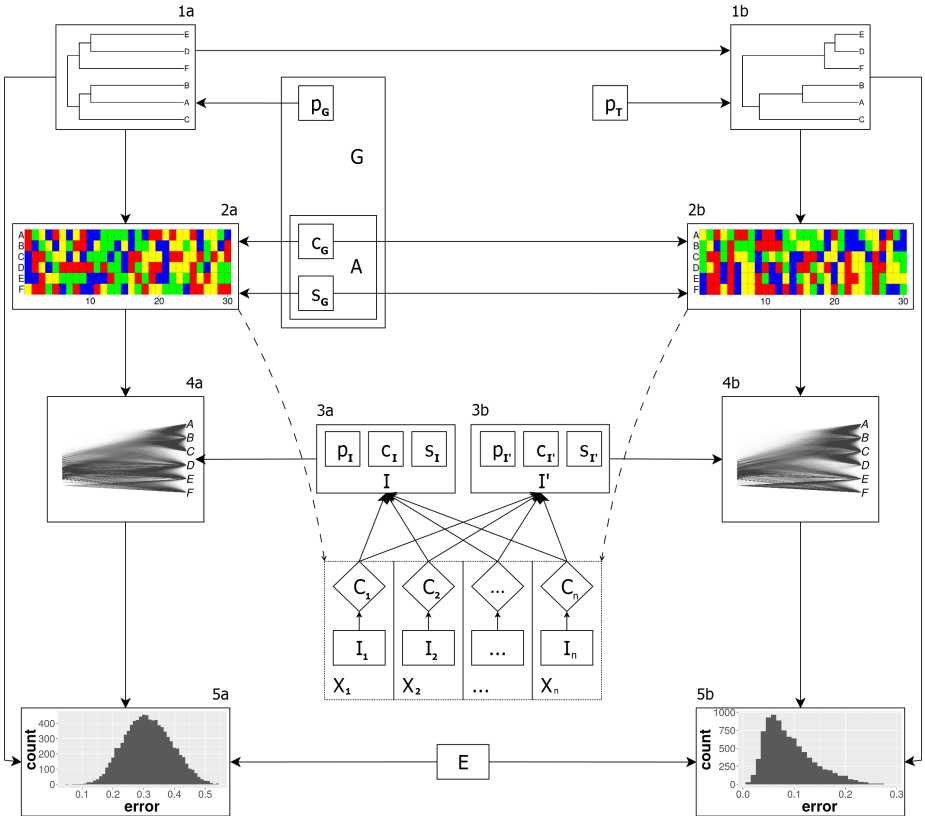

Figure 1: `pirouette` pipeline. The pipeline starts from a phylogeny (1a) simulated by the generative tree model p_G. The phylogeny is converted to an alignment (2a) using the generative alignment model $A = (c_G, s_G)$, composed of a clock and a site model. The user defines one or more experiments. For each candidate experiment X_i (a combination of inference model I_i and condition C_i), if its condition C_i is satisfied (which can depend on the alignment), the corresponding inference model $I = I_i$ is selected to be used in the next step. The inference models (3a) of the selected experiments use the alignment (2a) to each create a Bayesian posterior of (parameter estimates and) phylogenies (4a). Each of the posteriors' trees is compared to the true phylogeny (1a) using the error measure E, resulting in an error distribution (5a). Optionally, for each selected inference model a twin pipeline can be run. A twin phylogeny (1b) can be generated from the original phylogeny (1a) using the twin tree model p_t, selected among standard diversification models; the default option is the standard birth-death model, with parameters estimated from the original phylogeny. A twin alignment (2b) is then simulated from the twin phylogeny using clock model c_G and site model s_G imported from the generative model. The twin alignment has the same number of mutations as the original alignment. The twin pipeline follows the procedure of the main pipeline, resulting in a twin error distribution (5b).

the shared common ancestor, and is set to four different equally-sized mononucleotide blocks by default, as this helps interpreting the resulting alignment. Supported nucleotide substitution model, which we will refer to as site models, are JC, HKY, TN and GTR. Only the strict clock model is currently supported in this step.

The second step (Fig. 1, 3) selects one or more inference models I from a set of inference models $I_1, \ldots, I_n$. We define an experiment X_i as the combination of an inference model I_i and the conditions C_i to actually use it in the inference step. For example, we may require that an inference model (a combination of a tree model, clock model and site model) should include the generative/true tree model. As a second example, we may require that we have selected a set of candidate inference models, of which only the best should be used in the actual inference. In the first example, we specified the condition C_i that this generative model should always be run, whereas in the second example, we specified condition C_i that a candidate model should only be run when it is the best. The 'best' model is defined as the inference model with the highest evidence (a.k.a. marginal likelihood), given the alignment simulated in the previous step. The evidence for an inference model is estimated by nested sampling [Maturana *et al.* 2017], using the `NS` BEAST2 package. We note that scripted use of BEAST2 packages is only possible under Linux and Mac. Windows systems can do the model comparison for shorter DNA sequences using the web interface of `mcbette` [Bilderbeek 2019b].

The third step infers the posterior distributions, using the simulated alignment (Fig. 1, 2a $\rightarrow$ 4a), and the inference models that were selected in the previous step (3). For each selected experiment a posterior distribution is inferred, using the `babette` [Bilderbeek & Etienne 2018] R package which makes use of BEAST2. This step usually takes up most of the pipeline's computation

time.

The fourth step quantifies the inference error made. First the burn-in fraction is removed, i.e. the first phase of the Markov chain Monte Carlo (MCMC) run, which samples an unrepresentative part of state space. By default, `pirouette` removes the first 10% of the posterior. From the remaining posterior `pirouette` creates an error distribution, by measuring the difference between the true tree and each of the posterior trees (Fig. 1, 4a $\rightarrow$ 5a). The default way to quantify the difference between two phylogenies is the nLTT statistic (Janzen *et al.* 2015), but any user-defined error statistic can be used.

2.2 Twinning

An optional step is to use the 'twinning process'. This process, T, encompasses two steps: T_1, that generates a 'twin tree' (Fig. 1, 1b) and T_2, which generates a 'twin alignment' (Fig. 1, 2b). Both twin tree and alignment will be analyzed in the same way as the true tree and alignment.

We define a phylogeny τ as the combination of branching times $\vec{t}$ and topology ψ, and denote as τ_G the phylogeny produced by a (possibly non-standard) generative diversification model, having branching times $\vec{t}_G$ and topology ψ_G.

The first step (T_1) of the twinning process creates a tree τ_T with branching times $\vec{t}_T$ while preserving the original topology ψ_G:

$$\tau_G = (\vec{t}_G, \psi_G) \xrightarrow{T_1} \tau_T = (\vec{t}_T, \psi_G) \tag{1}$$

The default option for the diversification model p_T is the standard birth-death model. It is then possible to use the likelihood function L_T for this diversification model to find the parameters θ_T^* (e.g. speciation and extinction rates, in case of a birth-death model) that maximize this likelihood applied to the true

tree, conditioned on its number of tips n_G:

$$\max[L_T(\theta_T|\tau_G, n_G)] \rightarrow \theta_T^*. \tag{2}$$

We use θ_T^* to simulate a number $n_T = n_G$ of branching times $\vec{t}_T$ for the twin tree τ_T, under the process p_T, while preserving the topology.

The second step (T_2) of the twinning process simulates the twin alignment with the same clock model, site model and mutation rate used to simulate the original alignment. We also impose that, in the twin alignment, the total number of mutations with respect to the root sequence must be the same as in the true alignment in order to keep the information content stored in both the true and twin alignments as similar as possible. We achieve this by simply simulating twin alignments until we obtain one that has the desired number of mutations.

The twin pipeline serves as a control: even when the generating and inference models are identical (as is the case in the twin pipeline), the inferred trees from the posterior distribution will still differ from the true tree, due to stochasticity in producing an alignment and to the MCMC sampling of the posterior. The twin pipeline provides this minimum error, because the generating and inference model match exactly. When comparing the true and twin error distribution, any differences will be due to the fact that true and twin phylogenies are realizations of different processes: one (possibly) non-standard, p_G, and one standard, p_T (see Fig 1).

This can be seen for both the "generative" and "candidate" model types (see Table 1). If the chosen model type is "generative", the tree prior chosen for the twin inference will exactly match the model to generate the tree. In the main pipeline, as the tree model p_G is non-standard, it cannot be used in inference. If, instead, the chosen model type is "candidate", the twin tree model will be included in the pool of examined models during the process of selection of the

inference model.

Finally, if the goal is to evaluate BEAST2's performance on a non-standard tree prior, the last source of stochasticity comes from phylogenies. In fact, a single phylogeny cannot be considered as fully representative of the model. For this reason multiple phylogenies, as well as an equal number of twins, must be considered. Having the error measure normalized (i.e. comprised in the interval $[0, 1]$), it is possible to considerate the aggregated versions of the errors distributions across all the runs. Therefore, if the number of considered phylogenies is high enough, the comparison between the main pipeline's aggregated error distribution and its twin counterpart leads to a fair evaluation of the new tree prior with respect to the baseline error.

3 Installation

`pirouette` will be made available on CRAN from which it can then be easily installed:

```
install.packages("pirouette")
```

Until it is on CRAN, and for the most up-to-date version, one can download and install the package from `pirouette`'s GitHub repository:

```
remotes::install_github("richelbilderbeek/pirouette")
```

To start using `pirouette`, load its functions in the global namespace first:

```
library(pirouette)
```

Because `pirouette` calls BEAST2, BEAST2 must be installed. This can be done from within R, using:

```
install_beast2()
```

For the option to select the best candidate model, `pirouette` needs the "NS" BEAST2 package [Maturana *et al.* 2017]. It can be installed from within R, using:

```
install_beast2_pkg("NS")
```

An overview of `pirouette`'s main functions is shown in Table 3. Their usage is demonstrated in the example code below. All `pirouette`'s functions are documented, have a useful example and sensible defaults.

Name	**Description**	**Listing**
`pir_run`	Run `pirouette`	7
`pir_plot`	Show the `pirouette` results as a plot	8
`create_pir_params`	Create the `pirouette` parameters	6
`create_alignment_params`	Create the alignment parameters	2
`create_twinning_params`	Create the twinning parameters	13
`create_experiment`	Create one experiment	3
`create_error_measure_params`	Create the error measurement parameters	5

Table 3: `pirouette`'s main functions, description and the number of the listing in which it is used.

4 Usage

We show the usage of `pirouette` by gradually introducing its features. First, to get an idea of the baseline error, we measure the Bayesian inference error when we start from a phylogeny generated under a known and standard tree model. Second, to establish that the true tree prior is indeed the best, we also measure the inference error made by a best candidate inference model. Lastly, we quantify the impact of the tree prior in the Bayesian inference. We do so by separating the baseline error from the complete error running the pipeline starting from a tree generated by an unknown or non-standard tree model.

All the figures shown in this section are shown as-is, without any aesthetical modifications. Figures showing the full workflow and tables showing the effective

sample sized (a measure of inference quality) can be found in the supplementary materials.

4.1 The generative and inference models are equal and standard.

`pirouette` quantifies the influence of a new tree prior on BEAST2's inference by measuring the discrepancy between a given/true tree and a posterior distribution of phylogenies obtained as result of the inference process. Due to stochasticity, posterior trees will generally differ from the given phylogeny τ_G, even when the tree prior and alignment model used for inference are the same as those used to generate the alignment. Measuring this difference allows us to know the baseline error of the `pirouette` pipeline. We therefore define as 'standard tree priors' all the tree priors that are available to be used within an inference model (see Table 1).

Now we can formulate the first example research question that `pirouette` can answer: "What is the inference error made on phylogenies created by a standard diversification model?"

In this example we use a standard generative tree model p_G^0, namely the Yule (pure-birth) tree model. We choose to use a small tree with six taxa, to keep the calculations short and the figure more readable. We pick a crown age of ten time units. This value is completely arbitrary, but it ties in with the mutation rate used in simulating an alignment in the next step.

```
phylogeny <- create_yule_tree(n_taxa = 6, crown_age = 10)
```

Listing 1: Create a Yule tree. The resulting tree is shown in Figure 2.

The first step in `pirouette` is to simulate a DNA alignment from the given phylogeny, as described in Subsection 2.1. In this example, the root sequence

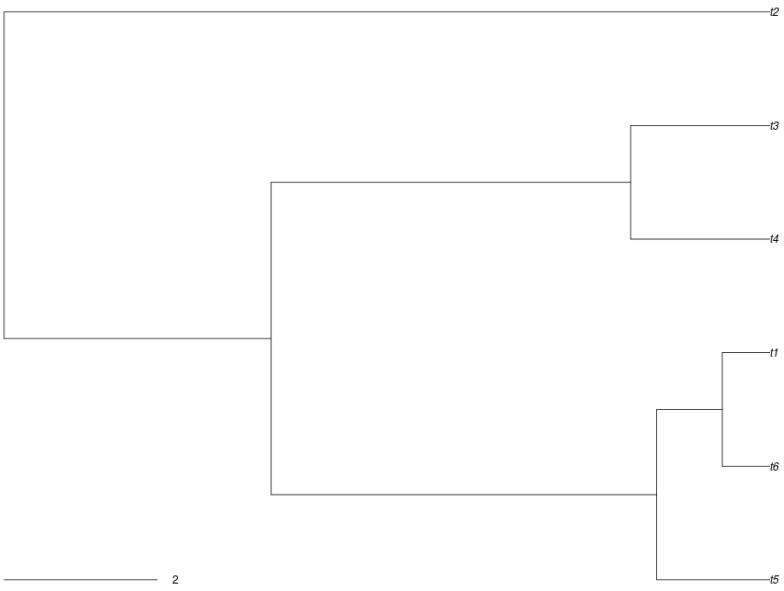

Figure 2: The Yule tree, as created by Listing 1.

consists of four blocks of 250 mononucleotides each, while the per-nucleotide mutation rate is 0.1 mutations per unit time. We use a Jukes-Cantor (JC, Jukes *et al.* 1969) site model and a strict clock model as these are the simplest. A JC site model assumes that mutation rates between nucleotides are equal and constant. A strict clock model assumes that the mutation rates of all lineages are equal and constant.

```
alignment_params <- create_alignment_params(
  sim_tral_fun = get_sim_tral_with_std_nsm_fun(
    mutation_rate = 0.1,
    site_model = create_jc69_site_model()
  ),
```

model_type	run_if	do_measure_evidence	inference model
generative	always	FALSE	JC, strict, Yule

Table 4: Inference conditions and model. JC: Jukes-Cantor site model. strict: strict clock model. Yule: Yule (pure-birth) tree prior.

```
  root_sequence = create_blocked_dna(length = 1000)
)
```

Listing 2: Create an alignment.

As the site and clock models used here are also the defaults, the function arguments can be safely omitted: we just explicitly show them for the sake of clarity.

In the second step we state our experiment. We define an experiment X as a combination of an inference model I and conditions C. In this example we choose I to be the same inference model as the generative one, i.e. the Yule tree prior $p_G^{\emptyset}$ and site and clock models defined in A, respectively Jukes-Cantor and strict clock. We specify in C that the experiment will always be run.

Listing 3 shows how to set up this experiment:

```
generative_experiment <- create_experiment(
  inference_conditions = create_inference_conditions(
    model_type = "generative",
    run_if = "always"
  ),
  inference_model = create_inference_model(
    tree_prior = create_yule_tree_prior(),
    clock_model = create_strict_clock_model(),
    site_model = create_jc69_site_model(),
    mcmc = create_mcmc()
  )
)
```

```

```

Listing 3: Create an experiment with the generative model, that will always be used in the actual inference, using explicit arguments.

Experiments must be bundled in a list to work, even if only one is provided, as in this case:

```
experiments <- list(generative_experiment)
```

Listing 4: Create the experiments. In this case, we use only an experiment with the generative model, that will always be used in the actual inference.

We also need to specify the error measurement parameters E. Here we choose the default E, which has a burn-in fraction of 10% and uses the nLTT statistic to measure the difference between phylogenies. For clarity, we create this setup explicitly here:

```
error_measure_params <- create_error_measure_params(
  error_fun = get_nltt_error_fun(),
  burn_in_fraction = 0.1
)
```

Listing 5: Calling `create_error_measure_params`.

We now have all the needed `pirouette` parameters: the alignment parameters, the experiments and the error measure parameters. These objects need to be bundled in a larger parameter structure, using `create_pir_params`:

```
pir_params <- create_pir_params(
  alignment_params = alignment_params,
  experiments = experiments,
  error_measure_params = error_measure_params
)
```

```
```

Listing 6: Calling `create_pir_params`.

We can finally use the given Yule tree and `pir_params` to measure the inference error made on phylogenies created by a standard diversification model:

```
errors <- pir_run(
  phylogeny = phylogeny,
  pir_params = pir_params
)
```

Listing 7: Calling `pir_run`.

The error distribution can be plotted directly using `pir_plot`:

```
pir_plot(errors)
```

Listing 8: Calling `pir_plot`.

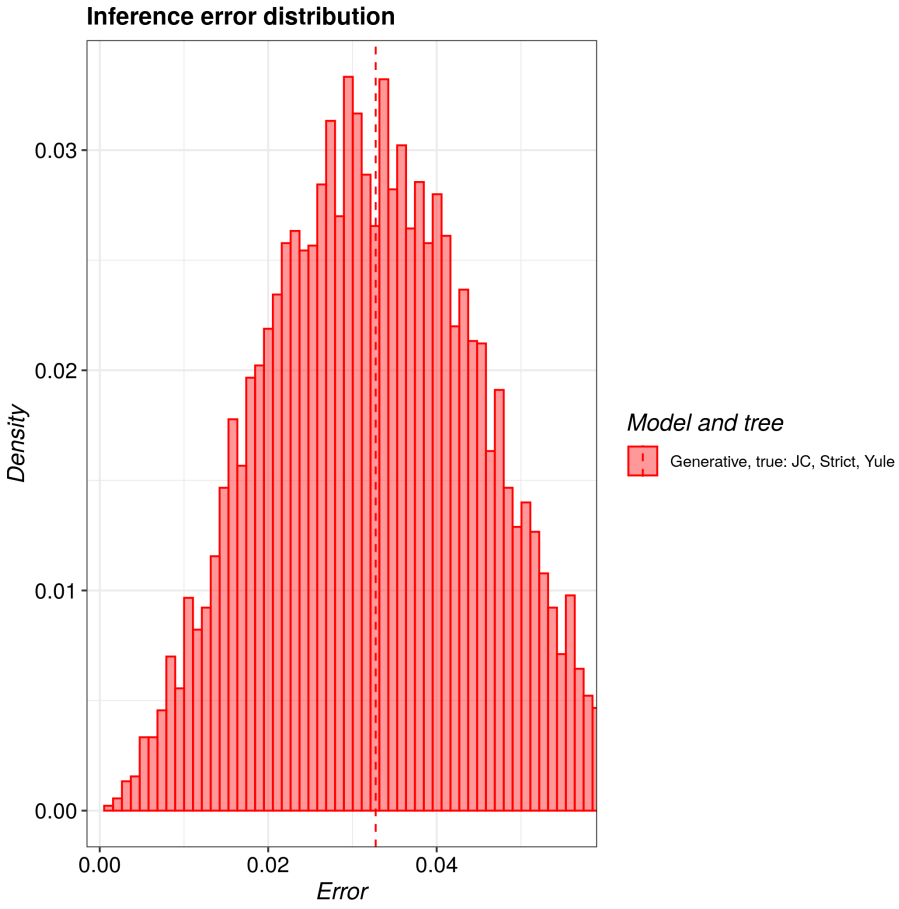

Figure 3: The inference error made when the generative and inference models are the same. The vertical dashed line indicates the median error.

The resulting error distribution, as shown Figure 3, shows the inference error when I matches with the generative model given by A and p_G^0. This error distribution can serve as a control, as it is obtained from a tree of which the generating tree model is standard and known. Obtaining this control is in fact an inherent part of `pirouette`. We call it 'twinning' and we will demonstrate it in section 4.3 below.

model_type	run_if	do_measure_evidence	inference model
generative	always	FALSE	JC, strict, Yule
candidate	best candidate	TRUE	JC, strict, BD
candidate	best candidate	TRUE	JC, strict, CBS
...	...	...	...
candidate	best candidate	TRUE	GTR, RLN, CCP
candidate	best candidate	TRUE	GTR, RLN, CEP

Table 5: Inference conditions and model. JC: Jukes-Cantor site model. strict: strict clock model. Yule: Yule (pure-birth) tree prior. BD: birth-death tree prior. GTR: GTR site model. RLN: relaxed log-normal clock model. CBS: coalescent Bayesian Skyline tree prior. CCP: coalescent constant-population tree prior. CEP: coalescent exponential-population tree prior.

4.2 The inference model may differ from the generative model

In the previous example we selected the inference model I to match with a known generative tree model p_G^0. However, a novel tree prior p_G is by (our) definition non-standard, and hence not part of a standard inference model (i.e. an inference model using a standard tree prior). In such a case, the question is which standard tree prior should be used in the inference. In this example, we do not only re-use our hand-picked inference model, but we also pick the best inference model from a set of inference models, i.e. the model with the highest evidence measured by its marginal likelihood. We show the procedure using `pirouette` to answer a second research question: "What is the inference error made on a novel phylogeny when using the best inference model, in comparison to a hand-picked model?

We will use the same tree as generated in 1, as well as the same alignment parameters as shown in Listing 2.

Here we specify a different set of experiments: we need to state that we already have an experiment for the generative model, as well as that we want all the other inference models to compete. We call the competing models 'can-

didate models'. As model selection is commonly performed on the full list of available candidate models, `pirouette` has a dedicated function for this choice: `create_all_experiments` creates a full set of 40 experiments, containing the inference models of all combinations of 4 site models, 2 clock models and 5 tree priors. All we need to add is to exclude the inference model in the generative experiment:

```
candidate_experiments <- create_all_experiments(
  exclude_model = generative_experiment$inference_model
)
```

Listing 9: Create all 40 candidate experiments, except for the inference model of the generative model.

We combine the generative and all the candidate models into one set of experiments:

```
experiments <- c(
  list(generative_experiment),
  candidate_experiments
)
```

Listing 10: Create a collection of experiments, with 1 generative model, and 39 candidate models.

We can now create the complete `pirouette` parameter set in the usual way (which is the same as Listing 6, but using the defaults):

```
pir_params <- create_pir_params(
  alignment_params = alignment_params,
  experiments = experiments
)
```

Listing 11: Create a `pir_params` with many defaults.

We run `pirouette` (Listing 7) and plot the results (Listing 8) in Figure 4.

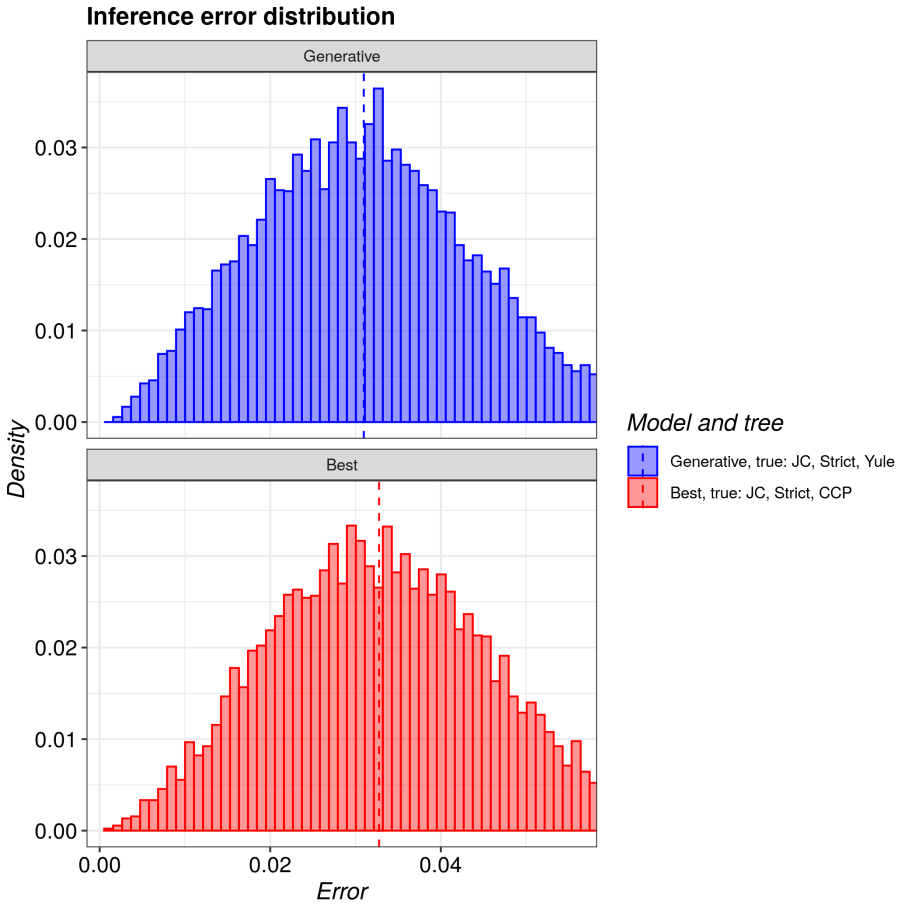

Figure 4: The inference error for both generative and best candidate inference models. Vertical dashed lines show the median error value per distribution.

4.3 The generative model is non-standard

So far we have measured the inference error on a tree generated according to a known (and standard) tree diversification model. The goal of `pirouette` is to measure the expected impact of a novel tree prior. To do so, in this example, we

will use a tree generated by a non-standard tree diversification model, assumed to be closely related to the Yule tree prior. We measure both the baseline error (that serves as a control) and the full inference error, because their difference shows the impact of the tree model. We obtain the baseline error by using a twin tree (see Subsection 2.2) for both the generative and best candidate models. The research question this example answers is: "What is the inference error made from a phylogeny, for both a generative model and the best candidate model?"

We start from the tree generated by a non-standard (and unknown) diversification model, having six taxa and a crown age of ten:

```
phylogeny  <- ape::read.tree(
  text = "(((A:8, B:8):1, C:9):1, ((D:8, E:8):1, F:9):1);"
)
```

Listing 12: A phylogeny generated by an unknown diversification model.

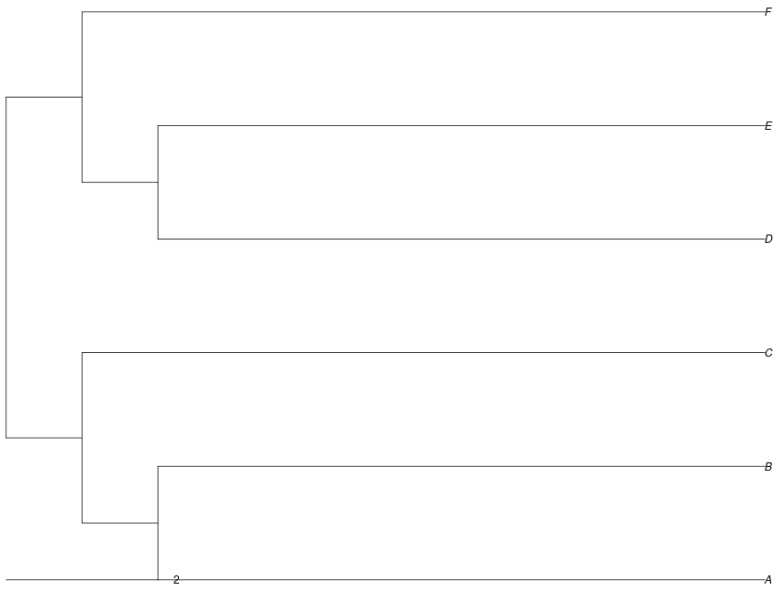

Figure 5: The tree derived from an unknown diversification process, as created by listing 12. The scale overlaps with the lower branch, which is the default behavior for the plotting function used.

Most of the other settings are the same as before: we reuse the alignment parameters (Listing 2), as well as the experiments (Listing 9). This time, however, we enable twinning (see Subsection 2.2), by creating twinning parameters. Creating these parameters is trivial with the default settings. For clarity, however, we explicitly show the most important arguments:

```
twinning_params <- create_twinning_params(
  sim_twin_tree_fun = get_sim_bd_twin_tree_fun(),
  sim_twal_fun = get_sim_twal_with_std_nsm_fun()
```

```
)
```

Listing 13: Create the default twinning parameters.

We combine all the parameters using `create_pir_params`:

```
pir_params <- create_pir_params(
  alignment_params = alignment_params,
  experiments = experiments,
  twinning_params = twinning_params
)
```

Listing 14: Create the default twinning parameters.

We run `pirouette` (Listing 7) and plot the results (Listing 8). The output is shown in Figure 6. While the error distributions using the best or generative model as inference model are very similar, the error distributions of the true tree are substantially larger than those of the twin tree. This is the error made by the mismatch between the generating species tree model and the tree prior used in inference.

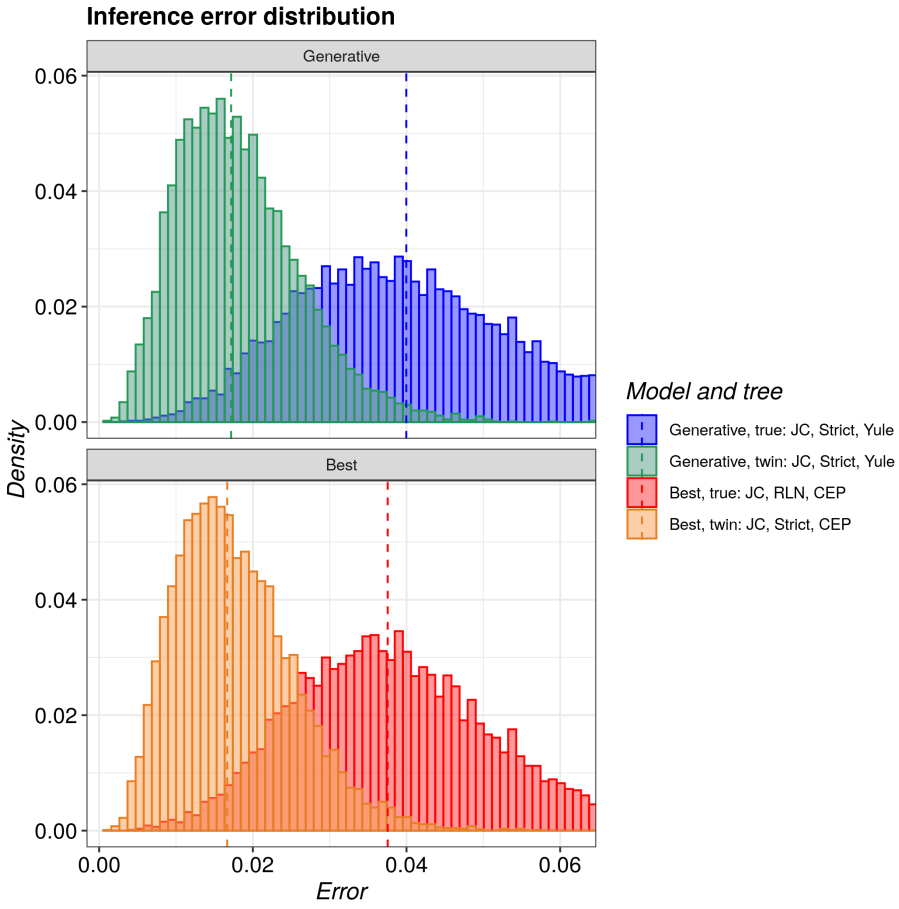

Figure 6: The inference error made for both a generative tree prior and best candidate model compared with the error obtained for the twin tree. Here, the 'twin' tree shows the baseline inference error. Vertical dashed lines show the median error value per distribution.

5 Discussion

We showed how to use `pirouette` to quantify the importance of a tree prior in Bayesian phylogenetics, using the simplest generative tree model possible. In principle any other (more complex) generative tree model can be tested, but we chose to provide the simplest (and fastest to run) examples.

Figure 6 illustrates the primary result of our pipeline: it shows the error distributions for the true tree and the twin tree when either the generating model or the best candidate model is used in inference. The clear difference between the error distributions for the true tree and the twin tree suggests that the choice of tree prior does matter.

We note, however, that all examples used only one original tree, where any speciation process produces a whole range of trees. One tree is not enough to determine the impact of a tree prior on Bayesian inference. However, if the same procedure were repeated and performed on a distribution with a sufficient number of generative trees, it would constitute a quantitative and effective assessment of the quality of the inference. Also a twin tree does not always result in a lower error distribution, as the stochasticity in generating a twin tree will - with very low probability - yield a tree of that same very low probability.

In conclusion, `pirouette` can show the errors to be expected when the tree prior used in inference is different from the generating model. The user can then judge whether or not a new tree prior, tailored on the generative process, is needed. If this is indeed the case, one can implement the novel tree prior as an addition to his/her favorite Bayesian inference tool.

6 pirouette resources

`pirouette` is free, libre and open source software available at `http://github.com/richelbilderbeek/pirouette`, licensed under the GNU General Public License version 3. `pirouette` depends on multiple packages, which are: `ape` (Paradis *et al.* 2004), `babette` (Bilderbeek & Etienne 2018), `becosys` (Bilderbeek 2019a), `DDD` (Haegeman 2018), `devtools` (Wickham & Chang 2016), `dplyr` (Wickham *et al.* 2019), `geiger` (Harmon *et al.* 2008), `ggplot2` (Wickham 2009), `knitr` (Xie 2017), `lintr` (Hester 2016), `magrittr` (Bache & Wickham 2014), `mcbette` (Bilderbeek 2019b), `nLTT` (Janzen 2019), `PBD` (Etienne 2017), `phangorn` (Schliep 2011), `phytools` (Revell 2012), `plyr` (Wickham 2011a), `rappdirs` (Ratnakumar *et al.* 2016), `rmarkdown` (Allaire *et al.* 2017), `Rmpfr` (Maechler 2019), `stringr` (Wickham 2017), `TESS` (Höhna 2013), `testit` (Xie 2014), `testthat` (Wickham 2011b) and `tidyr` (Wickham & Henry 2019).

`pirouette`'s development takes place on GitHub, `https://github.com/richelbilderbeek/pirouette`, which allows submitting bug reports, requesting features, and adding code. To ensure a high quality, `pirouette` uses a continuous integration service, has a code coverage of above 95% and enforces the most commonly used R style guide (Wickham 2015).

`pirouette`'s is extensively documented on its website, its documentation and its vignettes. The `pirouette` website is a good starting point to learn how to use `pirouette`, as it links to tutorials and videos. The `pirouette` package documentation describes all functions and liberally links to related functions. All exported functions show a minimal example as part of their documentation. The `pirouette` vignette demonstrates extensively how to use `pirouette` in a more informally written way.

The code used in this article and more examples that are periodically tested, can be found at `https://github.com/richelbilderbeek/pirouette_examples`.

7 Citation of pirouette

To cite `pirouette` this article from within R, use:

```
> citation("pirouette")
```

8 Acknowledgments

We would like to thank the Center for Information Technology of the University of Groningen for its support and for providing access to the Peregrine high performance computing cluster. We thank the Netherlands Organization for Scientific Research (NWO) for financial support through a VICI grant awarded to RSE.

9 Data Accessibility

All code is archived at `http://github.com/richelbilderbeek/pirouette_article`, with DOI `https://doi.org/12.3456/zenodo.1234567`.

10 Authors' contributions

RJCB, GL and RSE conceived the idea for the package. RJCB created, tested and revised the package. GL provided major contributions to the package. RJCB wrote the first draft of the manuscript, GL and RSE contributed to revisions.

References

Alfaro, M.E., Santini, F., Brock, C., Alamillo, H., Dornburg, A., Rabosky, D.L., Carnevale, G. & Harmon, L.J. (2009) Nine exceptional radiations plus high

turnover explain species diversity in jawed vertebrates. *Proceedings of the National Academy of Sciences*, **106**, 13410–13414.

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J. & Chang, W. (2017) *rmarkdown: Dynamic Documents for R*. R package version 1.8.

Bache, S.M. & Wickham, H. (2014) *magrittr: A Forward-Pipe Operator for R*. R package version 1.5.

Beaulieu, J.M. & O'meara, B.C. (2016) Detecting hidden diversification shifts in models of trait-dependent speciation and extinction. *Systematic biology*, **65**, 583–601.

Bilderbeek, R.J. (2019a) becosys. `https://github.com/richelbilderbeek/becosys` [Accessed: 2019-04-15].

Bilderbeek, R.J. (2019b) mcbette. `https://github.com/richelbilderbeek/mcbette` [Accessed: 2019-01-21].

Bilderbeek, R.J. & Etienne, R.S. (2018) babette: Beauti 2, beast 2 and tracer for r. *Methods in Ecology and Evolution*.

Bouckaert, R., Vaughan, T.G., Barido-Sottani, J., Duchêne, S., Fourment, M., Gavryushkina, A., Heled, J., Jones, G., Kühnert, D., De Maio, N. *et al.* (2019) Beast 2.5: An advanced software platform for bayesian evolutionary analysis. *PLoS computational biology*, **15**, e1006650.

Drummond, A.J., Ho, S.Y., Phillips, M.J. & Rambaut, A. (2006) Relaxed phylogenetics and dating with confidence. *PLoS biology*, **4**, e88.

Drummond, A.J. & Rambaut, A. (2007) Beast: Bayesian evolutionary analysis by sampling trees. *BMC evolutionary biology*, **7**, 214.

Drummond, A.J., Rambaut, A., Shapiro, B. & Pybus, O.G. (2005) Bayesian coalescent inference of past population dynamics from molecular sequences. *Molecular biology and evolution*, **22**, 1185–1192.

Duchene, S., Bouckaert, R., Duchene, D.A., Stadler, T. & Drummond, A.J. (2018) Phylodynamic model adequacy using posterior predictive simulations. *Systematic biology*, **68**, 358–364.

Etienne, R.S. (2017) *PBD: Protracted Birth-Death Model of Diversification*. R package version 1.4.

Etienne, R.S. & Haegeman, B. (2012) A conceptual and statistical framework for adaptive radiations with a key role for diversity dependence. *The American Naturalist*, **180**, E75–E89.

Etienne, R.S., Haegeman, B., Stadler, T., Aze, T., Pearson, P.N., Purvis, A. & Phillimore, A.B. (2011) Diversity-dependence brings molecular phylogenies closer to agreement with the fossil record. *Proc R Soc B*, p. rspb20111439.

Etienne, R.S., Morlon, H. & Lambert, A. (2014) Estimating the duration of speciation from phylogenies. *Evolution*, **68**, 2430–2440.

Etienne, R.S. & Rosindell, J. (2012) Prolonging the past counteracts the pull of the present: protracted speciation can explain observed slowdowns in diversification. *Systematic Biology*, **61**, 204–213.

FitzJohn, R.G. (2012) Diversitree: comparative phylogenetic analyses of diversification in r. *Methods in Ecology and Evolution*, **3**, 1084–1092.

Haegeman, R.S.E..B. (2018) *DDD: Diversity-Dependent Diversification*. R package version 3.7.

Harmon, L., Weir, J., Brock, C., Glor, R. & Challenger, W. (2008) Geiger: investigating evolutionary radiations. *Bioinformatics*, **24**, 129–131.

574 Hasegawa, M., Kishino, H. & Yano, T.a. (1985) Dating of the human-ape split-
575 ting by a molecular clock of mitochondrial dna. *Journal of molecular evolu-*
576 *tion*, **22**, 160–174.

577 Herrera-Alsina, L., van Els, P. & Etienne, R.S. (2018) Detecting the dependence
578 of diversification on multiple traits from phylogenetic trees and trait data.
579 *Systematic biology*.

580 Hester, J. (2016) *lintr: Static R Code Analysis*. R package version 1.0.0.

581 Höhna, S. (2013) Fast simulation of reconstructed phylogenies under global
582 time-dependent birth–death processes. *Bioinformatics*, **29**, 1367–1374.

583 Höhna, S., Landis, M.J., Heath, T.A., Boussau, B., Lartillot, N., Moore, B.R.,
584 Huelsenbeck, J.P. & Ronquist, F. (2016) Revbayes: Bayesian phylogenetic
585 inference using graphical models and an interactive model-specification lan-
586 guage. *Systematic Biology*, **65**, 726–736.

587 Huelsenbeck, J.P. & Ronquist, F. (2001) Mrbayes: Bayesian inference of phylo-
588 genetic trees. *Bioinformatics*, **17**, 754–755.

589 Janzen, T. (2019) nLTT. `https://github.com/richelbilderbeek/nLTT` [Ac-
590 cessed: 2019-04-15].

591 Janzen, T., Höhna, S. & Etienne, R.S. (2015) Approximate bayesian compu-
592 tation of diversification rates from molecular phylogenies: introducing a new
593 efficient summary statistic, the nltt. *Methods in Ecology and Evolution*, **6**,
594 566–575.

595 Jukes, T.H., Cantor, C.R. *et al.* (1969) Evolution of protein molecules. *Mam-*
596 *malian protein metabolism*, **3**, 132.

Lambert, A., Morlon, H. & Etienne, R.S. (2015) The reconstructed tree in the lineage-based model of protracted speciation. *Journal of mathematical biology*, **70**, 367–397.

Laudanno, G., Bilderbeek, R.J. & Etienne, R.S. (in preparation) The error in bayesian phylogenetic reconstruction when speciation co-occurs.

Laudanno, G., Haegeman, B., Rabosky, D.L. & Etienne, R.S. (submitted) Detecting lineage-specific shifts in diversification: A proper likelihood approach. *Systematic Biology*.

Maddison, W.P., Midford, P.E. & Otto, S.P. (2007) Estimating a binary character's effect on speciation and extinction. *Systematic biology*, **56**, 701–710.

Maechler, M. (2019) *Rmpfr: R MPFR - Multiple Precision Floating-Point Reliable*. R package version 0.7-2.

Maturana, P., Brewer, B.J., Klaere, S. & Bouckaert, R. (2017) Model selection and parameter inference in phylogenetics using nested sampling. *arXiv preprint arXiv:170305471*.

Nee, S., May, R.M. & Harvey, P.H. (1994) The reconstructed evolutionary process. *Phil Trans R Soc Lond B*, **344**, 305–311.

Paradis, E., Claude, J. & Strimmer, K. (2004) APE: analyses of phylogenetics and evolution in R language. *Bioinformatics*, **20**, 289–290.

R Core Team (2013) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rabosky, D.L. (2014) Automatic detection of key innovations, rate shifts, and diversity-dependence on phylogenetic trees. *PloS one*, **9**, e89543.

Rabosky, D.L. & Lovette, I.J. (2008) Explosive evolutionary radiations: decreasing speciation or increasing extinction through time? *Evolution: International Journal of Organic Evolution*, **62**, 1866–1875.

Ratnakumar, S., Mick, T. & Davis, T. (2016) *rappdirs: Application Directories: Determine Where to Save Data, Caches, and Logs*. R package version 0.3.1.

Revell, L.J. (2012) phytools: An r package for phylogenetic comparative biology (and other things). *Methods in Ecology and Evolution*, **3**, 217–223.

Rosindell, J., Cornell, S.J., Hubbell, S.P. & Etienne, R.S. (2010) Protracted speciation revitalizes the neutral theory of biodiversity. *Ecology Letters*, **13**, 716–727.

Sarver, B.A., Pennell, M.W., Brown, J.W., Keeble, S., Hardwick, K.M., Sullivan, J. & Harmon, L.J. (2019) The choice of tree prior and molecular clock does not substantially affect phylogenetic inferences of diversification rates. *PeerJ*, **7**, e6334.

Schliep, K. (2011) phangorn: phylogenetic analysis in R. *Bioinformatics*, **27**, 592–593.

Tamura, K. & Nei, M. (1993) Estimation of the number of nucleotide substitutions in the control region of mitochondrial dna in humans and chimpanzees. *Molecular biology and evolution*, **10**, 512–526.

Tavaré, S. (1986) Some probabilistic and statistical problems in the analysis of dna sequences. *Lectures on mathematics in the life sciences*, **17**, 57–86.

Wickham, H. (2009) *ggplot2: elegant graphics for data analysis*. Springer New York.

Wickham, H. (2011a) The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, **40**, 1–29.

Wickham, H. (2011b) testthat: Get started with testing. *The R Journal*, **3**, 5–10.

Wickham, H. (2015) *R packages: organize, test, document, and share your code.* O'Reilly Media, Inc.

Wickham, H. (2017) *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.2.0.

Wickham, H. & Chang, W. (2016) *devtools: Tools to Make Developing R Packages Easier*. R package version 1.12.0.9000.

Wickham, H., François, R., Henry, L. & Müller, K. (2019) *dplyr: A Grammar of Data Manipulation*. R package version 0.8.1.

Wickham, H. & Henry, L. (2019) *tidyr: Easily Tidy Data with 'spread()' and 'gather()' Functions*. R package version 0.8.3.

Xie, Y. (2014) *testit: A Simple Package for Testing R Packages*. R package version 0.4, `http://CRAN.R-project.org/package=testit`.

Xie, Y. (2017) *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.17.

Yule, G.U. (1925) A mathematical theory of evolution, based on the conclusions of dr. jc willis, frs. *Philosophical transactions of the Royal Society of London Series B, containing papers of a biological character*, **213**, 21–87.