Software for Systematics and Evolution

Syst. Biol. 65(4):726–736, 2016

DOI:10.1093/sysbio/syw021
Advance Access publication May 28, 2016

RevBayes: Bayesian Phylogenetic Inference Using Graphical Models and an Interactive Model-Specification Language

SEBASTIAN HÖHNA[1,2,3,4,*], MICHAEL J. LANDIS[1], TRACY A. HEATH[1,5,6], BASTIEN BOUSSAU[1,7], NICOLAS LARTILLOT[7], BRIAN R. MOORE[3], JOHN P. HUELSENBECK[1], AND FREDRIK RONQUIST[8]

[1]*Department of Integrative Biology;* [2]*Department of Statistics, University of California, Berkeley, CA 94720, USA;* [3]*Department of Evolution and Ecology, University of California, Davis, CA 95616, USA;* [4]*Department of Mathematics, Stockholm University, Stockholm, SE-106 91 Stockholm, Sweden;* [5]*Department of Ecology and Evolutionary Biology, University of Kansas, Lawrence, KS 66045, USA;* [6]*Department of Ecology, Evolution and Organismal Biology, Iowa State University, Ames, IA 50011, USA;* [7]*Laboratoire de Biométrie et Biologie Evolutive, Centre National de la Recherche Scientifique, Unité Mixte de Recherche 5558, Université Lyon 1, F-69622 Villeurbanne, France; and* [8]*Department of Bioinformatics and Genetics, Swedish Museum of Natural History, SE-10405 Stockholm, Sweden*

[*]*Correspondence to be sent to: Department of Integrative Biology, University of California, Berkeley, CA 94720, USA; E-mail: Sebastian.Hoehna@gmail.com*

Received 1 April 2015; reviews returned 2 March 2016; accepted 1 March 2016
Associate Editor: David Posada

Abstract.—Programs for Bayesian inference of phylogeny currently implement a unique and fixed suite of models. Consequently, users of these software packages are simultaneously forced to use a number of programs for a given study, while also lacking the freedom to explore models that have not been implemented by the developers of those programs. We developed a new open-source software package, `RevBayes`, to address these problems. `RevBayes` is entirely based on probabilistic graphical models, a powerful generic framework for specifying and analyzing statistical models. Phylogenetic-graphical models can be specified interactively in `RevBayes`, piece by piece, using a new succinct and intuitive language called `Rev`. `Rev` is similar to the `R` language and the `BUGS` model-specification language, and should be easy to learn for most users. The strength of `RevBayes` is the simplicity with which one can design, specify, and implement new and complex models. Fortunately, this tremendous flexibility does not come at the cost of slower computation; as we demonstrate, `RevBayes` outperforms competing software for several standard analyses. Compared with other programs, `RevBayes` has fewer black-box elements. Users need to explicitly specify each part of the model and analysis. Although this explicitness may initially be unfamiliar, we are convinced that this transparency will improve understanding of phylogenetic models in our field. Moreover, it will motivate the search for improvements to existing methods by brazenly exposing the model choices that we make to critical scrutiny. `RevBayes` is freely available at http://www.RevBayes.com. [Bayesian inference; Graphical models; MCMC; statistical phylogenetics.]

INTRODUCTION

Phylogeny estimation is now widely pursued in a Bayesian statistical framework (Rannala and Yang 1996; Larget and Simon 1999; Li et al. 2000; Huelsenbeck et al. 2001; 2002; Holder and Lewis 2003; Ronquist and Deans 2010; Yang and Rannala 2012). The success of the Bayesian approach derives largely from the availability of efficient algorithms that make it practical to compute the joint posterior probability distribution of phylogenetic model parameters (e.g., Markov chain Monte Carlo (MCMC); Metropolis et al. 1953; Hastings 1970), and by the development of computer programs that implement those models and algorithms. Biologists interested in Bayesian inference of phylogeny can now choose among a large number of software packages (Huelsenbeck and Ronquist 2001; Ronquist and Huelsenbeck 2003; Suchard and Redelings 2006; Drummond and Rambaut 2007; Yang 2007; Lartillot et al. 2009; Drummond et al. 2012; Ronquist et al. 2012b; Aberer et al. 2014; Bouckaert et al. 2014; Lewis et al. 2015). Yet, despite the sophistication and quality of the available software, we believe that all of the current Bayesian programs can be improved in several important respects.

First, the number of phylogenetic models available in any single computer program is limited. This forces the user to learn the details of several different computer programs—each with its own idiosyncrasies—to perform the analyses necessary for a study. The patchy implementation of models across software packages is probably a result of the typical life cycle of a phylogenetic model. A model is conceived and described in a paper but may or may not actually be implemented in computer software. A new model typically spends its infancy implemented in special-purpose and quirky software, and may only reach maturity when (or if) it is eventually implemented in a robust software package. As an example of this model life cycle, consider the approach for averaging over substitution models proposed by Huelsenbeck et al. (2004). This model was initially implemented in a computer program that was quite limited in its capabilities; the user could not consider alternative models of rate variation or priors on the branch lengths, etc. The substitution-model averaging approach only

gained traction when it was implemented almost a decade later in the program MrBayes (Ronquist et al. 2012b).

Second, existing software, such as MrBayes (Huelsenbeck and Ronquist 2001; Ronquist and Huelsenbeck 2003; Ronquist et al. 2012b), can be difficult to extend as new models are described. Every computer program has a basic architecture that is developed around the set of models that had been described at the time the program was written. New models, however, might not be compatible with the basic architecture of the program. For example, MrBayes was developed under the assumption that the alignment of DNA sequences is known without error, which makes it difficult to implement models that treat the alignment as a random variable (see e.g., Redelings and Suchard 2005). Similarly, in MrBayes the substitution process is assumed to be homogeneous over branches and sites (although it accommodates variation in substitution rate across sites and allows different models to be applied to subsets of the data). This homogeneity assumption has been questioned under several different circumstances (Galtier and Gouy 1995; Lartillot et al. 2007; Boussau et al. 2008; Groussin et al. 2013). It is possible to modify the program to allow heterogeneity in the substitution process across branches, but only with extensive recoding.

Third, all current phylogeny programs use awkward methods for specifying the assumptions of an analysis (i.e.,the parameters of the phylogenetic model). In general, the user is asked to specify whether a specific parameter is, or is not, part of the model. Hence, model specification in current software is much like throwing the proper sequence of toggle switches in a Lunar Module; the correct sequence of toggles must be thrown to specify any particular model, and each model is represented by a different configuration of toggle positions. This method for specifying models is clumsy even when the number of models implemented in the software is small, but becomes unwieldy as the number and complexity of models increases. More generally, the current approach for phylogenetic model specification limits the range of available models to those imagined by the software developers rather than the collective imagination of all users.

These considerations motivated the development of our new software package, RevBayes, an open-source program written in the C++ language. RevBayes was initially conceived as a major rewrite of the popular Bayesian phylogenetic-inference program MrBayes (Huelsenbeck and Ronquist 2001; Ronquist and Huelsenbeck 2003; Ronquist et al. 2012b). However, RevBayes shares no code with MrBayes and uses an entirely new way of specifying phylogenetic models. Indeed, we devised a language for specifying phylogenetic models that is similar to the R language (R Core Team 2014). RevBayes is a stand-alone software package, but relies heavily on the Nexus Class Library for parsing data (Lewis 2003). The resulting program had only superficial similarity to MrBayes, so we rechristened the program RevBayes (The new name also reconciles a criticism by Joe Felsenstein that Thomas Bayes was a reverend and would have been addressed as such.) to distinguish it from its predecessor.

Similar shortcomings have been identified by other researchers, which led to different solutions implemented in BEAST 2 (Bouckaert et al. 2014), a re-implementation of the popular BEAST program (Drummond and Rambaut 2007; Drummond et al. 2012). General similarities between RevBayes and BEAST 2 include a modular and flexible software design that enables model diversity, complexity, and extensibility. The main differences include the user interface (XML vs. Rev) and our strict adoption of the graphical-model framework. For example, BEAST 2 focuses on method developers by providing a plugin mechanism for adding new features written in Java, whereas RevBayes focuses on high-end users writing new models in Rev (similar to developments in R).

Our development of RevBayes was guided by a few key principles: (i) the program should enable flexible-model specification and implementation; (ii) the program should be easy and intuitive to use, and; (iii) the program should provide fast computation and efficient inference. We describe our solutions to each of the requirements in a separate section below.

The Graphical-Model Representation for Modeling and Inference

RevBayes is designed around the central idea that a phylogenetic model—in fact, any probabilistic model—can be represented as a graph (Jordan 2004). The graphical-model framework provides the flexibility to specify and extend models, and also confers an efficient mathematical foundation for parameter estimation (Höhna et al. 2014). In RevBayes, a probabilistic model is built up in computer memory by executing a series of commands. The user has fine-scale control over the details of the probabilistic model because single commands introduce individual model parameters and also specify their relationships to other parameters. In this way, a probabilistic model is assembled in computer memory as one would use LEGO® bricks. Any type of model can be built as long as the elementary pieces that make up a graphical model—the variables, distributions, and functions—are available.

The graph representing a probabilistic model consists of a set of vertices (often referred to as "nodes" in the phylogenetic literature) corresponding to the variables in the model, connected by edges that depict the dependence relationships among them. In RevBayes, a model graph consists of three types of variables: constant variables, stochastic variables, and deterministic variables. Constant variables represent the fixed parameters of an analysis, such as the parameter values of a prior distribution. Stochastic variables in the graph represent parameters of the model or observations, and are associated with probability

distributions. Finally, deterministic parts in the graphical model represent a transformation of variables. Details of phylogenetic-graphical models are described in Höhna et al. (2014).

Designing the program around the idea of a graphical model has several advantages. First, the explicitness of the graphical-modeling approach has considerable pedagogical value. All of the parameters of the phylogenetic model are exposed, including the parameters associated with the prior probability distributions on the parameters. Programs such as `MrBayes` enable the user to ignore the prior assumptions because the program assigns default priors to all model parameters. By contrast, a graphical-model representation exposes the anatomy of the model (including prior assumptions) to the user.

A second advantage of the graphical-modeling approach is the inherent flexibility. New models can be constructed from existing ones by changing the probability distributions assigned to stochastic variables, or by changing the functions assigned to deterministic variables, or by introducing new relationships among the variables (i.e.,changing the graph structure of the model). An existing model can also be modified by swapping subgraphs, for example, by replacing a pure-birth model with a birth–death model. Finally, a model can be extended by adding another layer to it, for example by introducing a hyperprior distribution for the speciation rate instead of using a fixed value. The only limitation on the types of models that can be built by the user is the number of available "bricks" (i.e.,functions and distributions); `RevBayes` provides a ton of bricks and the ability to easily create new bricks.

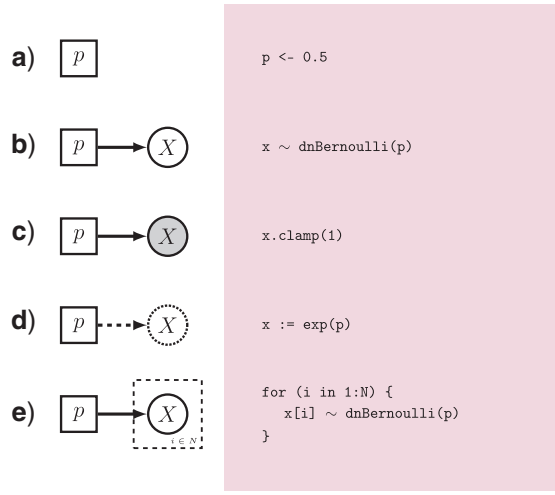


FIGURE 1. Assignment operations for graphical-model components. a) Constant assignment ("`p <- 0.5`"): Assignment of constant variables/parameters in the model. b) Stochastic assignment ("`x ~ dnBernoulli(p)`"): assignment of stochastic variables that are either estimated or observed. c) Observation ("`x.clamp(1)`"): assignment of observation to a stochastic variable. d) Deterministic assignment ("`x := exp(p)`"): assignment of deterministic variables that are transformations of other variables. **e)** Plates/repetitions ("`for (i in 1:N) { ... }`"): identical assignment of *N* variables.

THE Rev LANGUAGE FOR USER INTERACTION

We have developed a new programming language, `Rev`, for interacting with `RevBayes`. `Rev` is suitable for both interactive use and batch processing. Through `Rev`, users define graphical-model components in a succinct and intuitive way. `Rev` is inspired by the `R` (R Core Team 2014) language and the `BUGS` (Lunn et al. 2009) model-specification language; their popularity should reduce the `Rev` learning curve. However, `Rev` differs from these other languages because of the primary focus of `RevBayes` on Bayesian inference of phylogeny, for which `R` or `BUGS` are not compatible. That is, `R` and `BUGS` are designed for statistical inference and visualization of numerical data (e.g., regression analysis and ANOVA). By contrast, the specialized parameter types used in phylogenetic inference—for example trees and nucleotide characters—need entirely different data structures and algorithms for parameter estimation.

We believe that the benefits of a specifically designed `Rev` language for Bayesian phylogenetic inference outweigh the cost of developing a brand new programming language. Our focus while designing `Rev` has been on providing an intuitive and easy-to-learn syntax that also provides methods for automatic error checking (e.g., by using explicit types). Thus, `Rev` prevents users from declaring nonsensical relationships between model variables and their corresponding functions and distributions (e.g., specifying a normal distribution as a prior on tree topologies). Importantly, `Rev` scripts are easily shared with the community, providing a precise description of the details of a phylogenetic analysis that will facilitate replication of the results (Leebens-Mack et al. 2006), while also minimizing the effort required to repeat complex analyses on new data sets. These scripts should be easier to read and understand, and can easily be adapted to incorporate model variants. Moreover, `Rev` itself can easily be extended over time by adding new functions, distributions, data types, and inference algorithms.

In the previous section, we introduced the three components of a graphical model; constant, stochastic, and deterministic variables. Figure 1 illustrates how these model variables are represented in `Rev` code. We will provide a more extensive treatment of the `Rev` language in a forthcoming paper; here we present examples for constructing models using `Rev`.

BENCHMARKS

The success of a program like `RevBayes` depends on its ease of use, the diversity of models it implements, and also its efficiency. One might expect the efficiency of `RevBayes` to be hindered by the generality of its model-specification framework. To address this concern, we compared the efficiency of `RevBayes` to

TABLE 1. Comparing the efficiency of tree-likelihood computation in BEAST, MrBayes, and RevBayes.

Software	HKY	HYK+Γ	GTR	GTR+Γ
BEAST v1.8	65.3	188.4	75.8	213.4
BEAST v1.8—BEAGLE	41.2	105.2	47.5	107.4
MrBayes 3.2	78.2	177.7	76.9	169.9
MrBayes 3.2—BEAGLE	92.5	221.2	91.4	222.7
RevBayes	46.9	161.3	62.5	181.2

Notes: BEAST and MrBayes were run with and without the CPU implementation of the BEAGLE library for fast computation (Ayres et al. 2012). Exactly one substitution model parameter was updated per iteration, ensuring recomputation of the entire tree likelihood. All analyses used the same fixed tree topology and branch lengths under one of four substitution models: HKY (Hasegawa et al. 1985), HKY+Γ (Yang 1994), GTR (Tavaré 1986), or GTR+Γ (Yang 1994). Run times are given in seconds on a MacBookPro with a 3 GHz Intel Core i7 processor for 10^5 iterations on a molecular data set with 12 species and 898 sites.

TABLE 2. Comparing the efficiency of MCMC shortcuts in BEAST, MrBayes, and RevBayes.

Software	NNI	Node-Slide
BEAST v1.8	30.7	42.8
BEAST v1.8—BEAGLE	21.0	28.3
MrBayes 3.2	37.2	38.1
MrBayes 3.2—BEAGLE	42.6	31.9
RevBayes	17.8	23.5

Notes: The GTR substitution model (Tavaré 1986) was fixed but the tree was updated using either the Nearest Neighbor Interchange (NNI) or the Node-Slide move (Lakner et al. 2008; Höhna et al. 2008; Yang 2014). Other test conditions were identical to those described in Table 1.

the two most popular Bayesian phylogenetic software packages: BEAST (Drummond and Rambaut 2007; Drummond et al. 2012; Bouckaert et al. 2014) and MrBayes (Huelsenbeck and Ronquist 2001; Ronquist and Huelsenbeck 2003; Ronquist et al. 2012b). The efficiency of a Bayesian phylogeny inference program can be measured in three ways. First, we can evaluate how fast the software computes the likelihood under a given model. This speed is particularly important because computing the likelihood of a phylogenetic tree is time-consuming, and typically needs to be performed millions of times over the course of a MCMC simulation. Second, we can compare run times for a MCMC simulation, which emphasizes shortcuts taken in the MCMC algorithm to avoid unnecessary calculations. This is particularly critical when few parameters of a model change during an update in the MCMC run, where only a small part of the entire model likelihood needs to be recomputed. Third, we can evaluate the MCMC algorithm itself according to how well and fast it explores parameter space. Here, we focus on the first two efficiency criteria, as the third aspect of efficiency is a property of the algorithm rather than the software. However, we note that RevBayes incorporates new MCMC algorithms—such as slice-sampling (Besag and Green 1993) and the guided-tree proposals described by Höhna and Drummond (2012) to efficiently explore tree space—and can easily incorporate new algorithmic developments. Note that RevBayes currently does not use any external library for fast likelihood computation, for example BEAGLE (Ayres et al. 2012) or PLL (Flouri et al. 2015), but these could be included in the future.

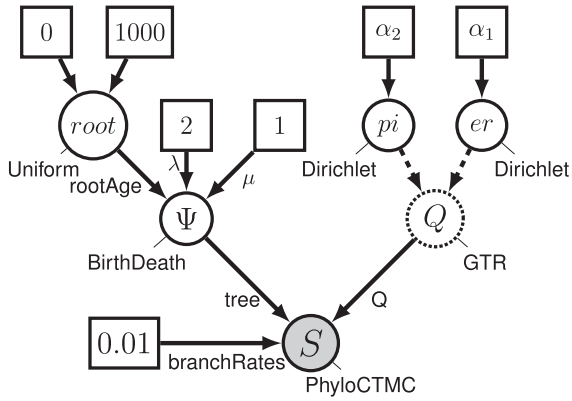
The results demonstrate that RevBayes performs equally well or better than the basic implementations of MrBayes and BEAST in terms of the speed of likelihood computations (Table 1). Only BEAST with BEAGLE outperformed our implementation in RevBayes. The likelihood implementation in MrBayes is actually faster than the implementation of MrBayes with BEAGLE, which is due to the overhead of function calls to the BEAGLE library and the comparably modest speed improvement of BEAGLE for nucleotide substitution models. In terms of MCMC shortcuts, RevBayes outperformed all competing software (Table 2). These benchmarks show that the graphical-model framework—which is generic and thus cannot use shortcuts specifically designed for a particular model—can compete in terms of computational efficiency with the most popular dedicated software used today.

EXAMPLE ANALYSES

Here, we provide examples that are based on tutorials using empirical data, which are available on our website http://www.RevBayes.com. Our objective is to illustrate some of the features implemented in RevBayes, and to demonstrate the flexibility and explicitness of the graphical-model framework. Accordingly, we focus on the specification of the phylogenetic models, omitting specification of the analysis to save space.

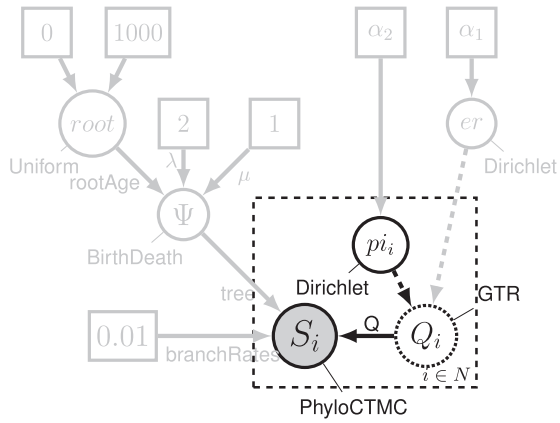
Molecular Phylogenetic Model

We begin with a simple phylogenetic analysis of an unpartitioned molecular data set under the general time-reversible (GTR) substitution model (Tavaré 1986), with a constant-rate birth–death process as the prior distribution on the tree, and a constant substitution rate of 0.01 substitutions per million years per site (e.g., Ho et al. 2007) (Figure 2). Note that the only two variables that are not declared are the *data* and the *taxa*, which are usually provided by the user. The inherent flexibility of this specification is readily apparent: we could, for example, estimate the speciation rate by defining a prior distribution for this parameter, for example, by using *speciation* ∼ *dnExponential(10.0)*. Similarly, we could easily substitute alternative substitution models, such as the HKY substitution model (Hasegawa et al. 1985) by replacing the *fnGTR* function with the *fnHKY* function. Or we could adopt an unconstrained (unrooted) tree by specifying a prior on unrooted trees (e.g., *dnUniformTopology*) and independent, exponentially distributed branch lengths instead of the birth–death process prior. We present some such extensions in the following model descriptions.




```
root ~ dnUniform(0,1000.0)
psi ~ dnBirthDeath( lambda=2.0, mu=1.0,
                    rootAge=root, taxa)


alpha1 <- v(1,1,1,1,1,1)
alpha2 <- v(1,1,1,1)

er ~ dnDirichlet( alpha1 )
pi ~ dnDirichlet( alpha2 )
Q_mol := fnGTR(er, pi)


seq ~ dnPhyloCTMC( tree=psi, Q=Q_mol,
                   branchRates=0.01, type="DNA" )
seq.clamp( data )
```

FIGURE 2. A simple phylogenetic model depicted in graphical-model notation (left) and the corresponding specification in the Rev language (right). In graphical-model notation, constant variables are enclosed in boxes, stochastic variables are enclosed in solid circles, deterministic variables in stippled circles, and observations in shaded circles, with arrows indicating variable dependencies. For example, the root age (*root*) is a random variable described by a uniform prior probability distribution with constant upper and lower bounds (in this case, 0 and 1000), the instantaneous-rate matrix, *Q*, is a deterministic function of the base frequencies and exchangeability rates (*pi* and *er*, respectively), and the observed sequences, *S*, are realizations of the phylogenetic model that are clamped for inference. This model is mirrored in Rev code, where the first two lines create the birth–death process (with fixed speciation and extinction rates; *lambda* = 2.0 and *mu* = 1.0), and a uniform prior distribution on the root age. The following five lines instantiate the instantaneous-rate matrix for the GTR model, where both the base frequencies and exchangeability rates are drawn from flat Dirichlet distributions. Finally, we create the stochastic variable representing the character data drawn from the Phylo-CTMC (continuous-time Markov chain) process and attach (clamp) observations to the variable *seq*.



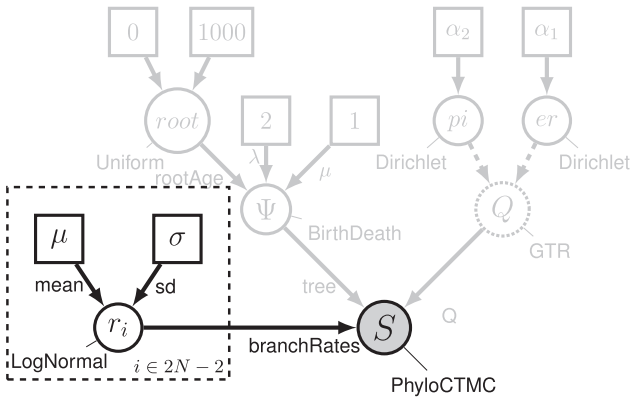

```
root ~ dnUniform(0,1000.0)
psi ~ dnBirthDeath( lambda=2.0, mu=1.0,
                    rootAge=root, taxa)


alpha1 <- v(1,1,1,1,1,1)
alpha2 <- v(1,1,1,1)

er ~ dnDirichlet( alpha1 )
for (i in 1:N) {
   pi[i] ~ dnDirichlet( alpha2 )
   Q_mol[i] := fnGTR(er, pi[i])

   seq[i] ~ dnPhyloCTMC( tree=psi, Q=Q_mol[i],
                         branchRates=0.01, type="DNA" )
   seq[i].clamp( data[i] )
}
```

FIGURE 3. A partitioned-data model depicted in graphical-model notation (left) and the corresponding specification in the Rev language (right). Here, we extend the simple model depicted in Figure 2 by allowing base frequencies, pi_i, to vary across the *N* pre-specified data partitions (identical model components are grayed out). Graphically, this repetition is represented by the enclosure of the replicated variables within a dashed box, which is referred to as a "plate" (other aspects of the graphical-model notation follow the descriptions in Figures 1–2). In Rev, this repetition is specified using loops.

Partitioned-Data Models

It is often important to partition a data set into multiple subsets to capture variation in the substitution process across the alignment (comprising multiple gene/genomic regions, codon positions of protein-coding genes, stem and loop regions of ribosomal genes, etc. Brown and Lemmon 2007). For example, we can extend the previous phylogenetic model by specifying independent base frequencies for each of *N* data subsets (Figure 3).

Again, this partitioned model can be modified in numerous ways, for example, by using independent exchangeability rates for all partitions. The user has complete control over specifying how parameters are shared across data subsets. Furthermore, any combination of substitution models is possible; for instance, we could specify a GTR substitution model for the first partition and an F81 substitution model (Felsenstein 1981) for the second partition, and so on.




```
root ~ dnUniform(0,1000.0)
psi ~ dnBirthDeath( lambda=2.0, mu=1.0,
                    rootAge=root, taxa)

alpha1 <- v(1,1,1,1,1,1)
alpha2 <- v(1,1,1,1)
er ~ dnDirichlet( alpha1 )
pi ~ dnDirichlet( alpha2 )
Q_mol := fnGTR(er, pi)

mu <- 0
sigma <- 1
for (i in 1:(2*N-2)) {
   r[i] ~ dnLognormal( mu, sigma )
}

seq ~ dnPhyloCTMC( tree=psi, Q=Q_mol,
                   branchRates=r, type="DNA" )
seq.clamp( data )
```

FIGURE 4. A relaxed-clock model depicted in graphical-model notation (left) and the corresponding specification in the Rev language (right). Here, we extend the simple model depicted in Figure 2 by allowing substitution rates to vary across branches (identical model components are grayed out). Specifically, we invoke the UCLN branch-rate model (Drummond et al. 2006), which assumes that substitution rates on each of the $2N-2$ branches are independent draws from a shared lognormal distribution, with constant hyperparameters specifying the mean ($\mu = 0$) and standard deviation ($\sigma = 1$). Graphical-model notation follows the descriptions in Figures 1–2.

Relaxed-Clock Models

We can specify relaxed-clock models to accommodate variation in substitution rates across branches. In this example, we allow each branch on the phylogenetic tree to have its own substitution rate that is drawn independently from a shared lognormal distribution; this is the uncorrelated lognormal (UCLN) branch-rate model (Drummond et al. 2006; Figure 4).

The prior distribution on branch rates could just as easily be an exponential distribution or gamma distribution, or, in fact, any probability distribution defined for positive real numbers (Heath and Moore 2014). The graphical-model framework offers considerable flexibility in defining the relaxed-clock model, whether changes in substitution rate occur on branches or at nodes. The rates could be drawn from an autocorrelated process (Thorne et al. 1998; Thorne and Kishino 2002), any type of independent-rates model (Drummond et al. 2006; Lepage et al. 2007; Rannala and Yang 2007), a compound Poisson process (Huelsenbeck et al. 2000), a local-molecular clock model (Yang and Yoder 2003; Drummond and Suchard 2010), a Dirichlet process prior (Heath et al. 2012), or any other process. It is exceptionally easy to specify different relaxed-clock models in RevBayes because the Phylo-CTMC distribution can accommodate any vector of rates and does not restrict how those rates are defined. Accordingly, the user can readily substitute a new relaxed-clock model simply by changing the prior model describing how substitution rates vary across branches, thereby gaining access to the full flexibility of RevBayes.

Gene-Tree Species-Tree Models

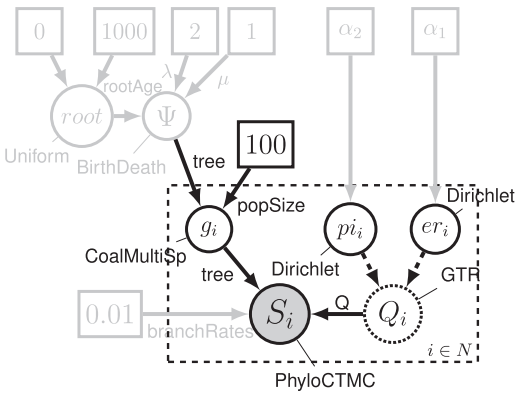
RevBayes allows simultaneous inference of gene trees and species trees. In this example, we use a birth–death process prior on the species tree and a multispecies-coalescent process with constant, homogeneous population size as the prior for each gene tree (Rannala and Yang 2003; Figure 5).

Each gene tree is, in turn, used as a parameter of the Phylo-CTMC for the corresponding sequence data and can have its own substitution process, as defined in the partitioned-data example above. Current variations of this model include branch-specific population sizes. The choice of a prior distribution on the population size is only restricted to positive real numbers but otherwise can be any distribution. Additionally, RevBayes allows you to create any relationship between the gene trees and partition-specific substitution models; for example, data partitions with distinct substitution models can share the same gene trees, or data partitions with distinct gene trees can share the same substitution model.

Discrete Morphological Models

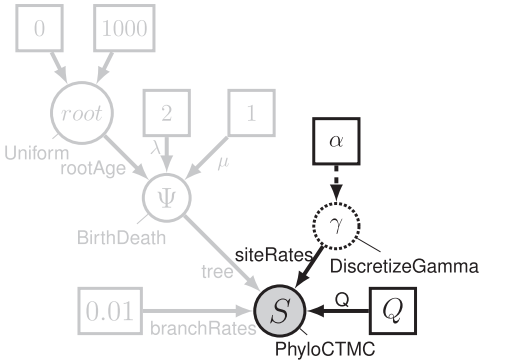
RevBayes provides several models for the analysis of discrete morphological data. Models of molecular evolution and discrete-trait evolution have a similar theoretical basis, with the main considerations being that invariant characters are typically not sampled in morphological data, and that the character states can be flipped without changing the meaning of the trait (i.e.,changing all of the 0s to 1s and 1s to 0s does not alter the information in the discrete-trait data). In this example, we use the Jukes–Cantor instantaneous-rate matrix for binary traits, and accommodate among-trait rate variation using four discrete gamma categories (Yang 1996; Lewis 2001; Harrison and Larsson 2015; Figure 6).

Many of the modeling choices for substitution models also apply to discrete-trait models. For example, morphological data can be partitioned, and rates of morphological evolution across branches can be




```
root ~ dnUniform(0,1000.0)
psi ~ dnBirthDeath( lambda=2.0, mu=1.0,
                    rootAge=root, taxa)


alpha1 <- v(1,1,1,1,1,1)
alpha2 <- v(1,1,1,1)


for (i in 1:N) {
   pi[i] ~ dnDirichlet( alpha2 )
   er[i] ~ dnDirichlet( alpha1 )
   Q_mol[i] := fnGTR(er[i], pi[i])
   g[i] ~ dnCoalMultiSpeciesConst( psi, Ne=100.0, taxon_map )

   seq[i] ~ dnPhyloCTMC( tree=g[i], Q=Q_mol[i],
                         branchRates=0.01, type="DNA" )
   seq[i].clamp( data[i] )
}
```

FIGURE 5. A species-tree model depicted in graphical-model notation (left) and the corresponding specification in the Rev language (right). Here, we extend the simple model depicted in Figure 2 by allowing the topology to vary across N genes (identical model components are grayed out). Specifically, we invoke the multispecies-coalescent process (Rannala and Yang 2003) with constant population size, $N_e = 100$, where we assume each gene has an independent substitution process, Q_i, and gene-tree topology, g_i. Graphical-model notation follows the descriptions in Figures 1–2.



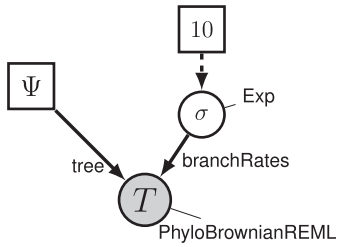

```
root ~ dnUniform(0,1000.0)
psi ~ dnBirthDeath(lambda=2.0, mu=1.0,
                   rootAge=root, taxa)


alpha <- 2
gamma := fnDiscretizeGamma( rate=alpha, shape=alpha, numCats=4 )


Q_disc <- fnJC(2)
seq ~ dnPhyloCTMC( tree=psi, Q=Q_disc, siteRates=gamma,
                   branchRates=0.01, type="Standard" )


seq.clamp( data )
```

FIGURE 6. A discrete-trait model depicted in graphical-model notation (left) and the corresponding specification in the Rev language (right). Here, we modify the simple model depicted in Figure 2 by accommodating discrete morphological traits (identical model components are grayed out). Specifically, we invoke the M_k model (Lewis 2001)—as a special case of the Jukes–Cantor model (Jukes and Cantor 1969)—with a discretized, mean-one gamma model, γ, to accommodate variation in the rate of evolution among traits (Yang 1996), which is controlled by the shape parameter, α. Graphical-model notation follows the descriptions in Figures 1–2.

described using various relaxed-clock models. Thus, in RevBayes we automatically gain a larger variety of models because each model only needs to be implemented once in order for it to be applied in combination with many other models.

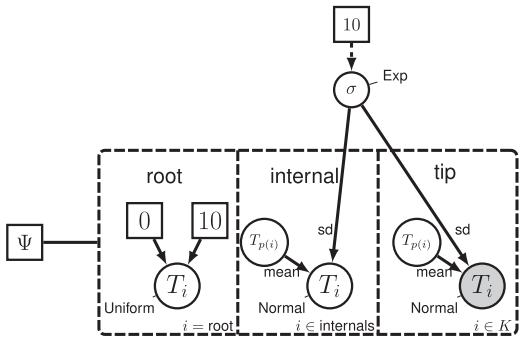
Continuous-Trait Models

Continuous-trait evolution is conceptually different from discrete morphological evolution because it does not rely on an instantaneous-rate matrix. The simplest model of continuous-trait evolution is the Brownian-motion model (Felsenstein 1985). We can instantiate the Brownian-motion model in a way similar to the previous models (Figure 7). The *dnPhyloBrownianREML* (phylogenetic Brownian motion using residual maximum likelihood) is a joint process on the continuous traits at the tips, which is equivalent to a full model (or data-augmented model) using a tree plate (Figure 8). Again, in Rev we use loops to represent any type of plate. The relative superiority of alternative representations of the same model depends on the application. For example, the residual maximum likelihood (REML) approach is faster to compute and allows more efficient mixing if the topology is unknown, whereas the data-augmented model automatically provides estimates for the values at the interior nodes of the tree and is more flexible. There are several extensions to the Brownian-motion model that have been implemented in RevBayes, such as the Ornstein–Uhlenbeck processes (Hansen 1997), Lévy jump processes (Landis et al. 2013b), and the multivariate normal distributed traits (Harvey and Pagel 1991; Lartillot and Poujol 2011) model. Additionally, all




```
psi <- readTrees("myTree.tre")[1]

sigma ~ dnExponential(10.0)

traits ~ dnPhyloBrownianREML( tree=psi, branchRates=sigma )

trais.clamp( data )
```

FIGURE 7. A continuous-trait model depicted in graphical-model notation (left) and the corresponding specification in the Rev language (right). Here we use a fixed phylogeny, *psi*, which is read in from a file. The rate of Brownian motion, *sigma*, is drawn from an exponential distribution with rate parameter 10. The observed traits, *T*, are distributed according to a *dnPhyloBrownianREML*, which is a Brownian-motion process using the phylogenetic-independent contrasts method to integrate over the unobserved states at the internal nodes. Graphical-model notation follows the descriptions in Figures 1–2.




```
psi <- readTrees("myTree.tre")[1]

sigma ~ dnExponential(10.0)

T[2*K-1] ~ dnUniform(0.0,10.0)

for (i in (2*K-2):1) {
   if (psi.isInternal(i)) {
      T[i] ~ dnNormal( T[psi.parent(i)], sd=sigma*sqrt(psi.branchLength(i)) )
   } else {
      T[i] ~ dnNormal( T[psi.parent(i)], sd=sigma*sqrt(psi.branchLength(i)) )
      T[i].clamp(contData.getTaxon(psi.nodeName(i))[1])
   }
}
```

FIGURE 8. An alternative continuous-trait model depicted in graphical-model notation (left) and the corresponding specification in the Rev language (right). Again, the phylogeny, *psi*, is assumed to be known and the rate of the Brownian motion, *sigma*, is drawn from an exponential distribution with rate parameter 10. The root state, $T[2K-1]$, where K is the number of tips, is drawn from a uniform distribution between 0 and 10. The tree-plate notation explicitly specifies that traits evolve according to a normal distribution with mean $T[psi.parent(i)]$ (the trait value of the parent branch) and variance scaled by the branch length, $sigma^2 * psi.branchLength(i)$. Only the values of the tips are clamped. Graphical-model notation follows the descriptions in Figures 1–2.

models and methods that can be applied to molecular data, for example, relaxed-clock models, can be applied to continuous-trait models.

Biogeographic Models

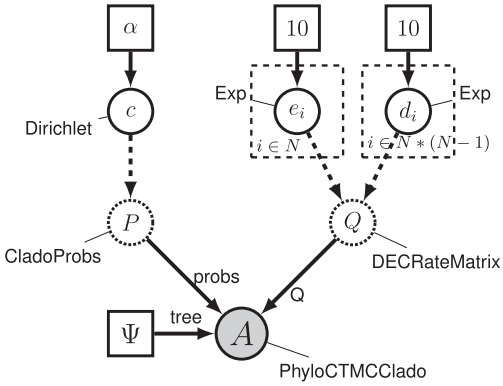
Our final example considers a simple dispersal, extirpation, and cladogenetic (DEC) model for a biogeographic inference (Ree et al. 2005). We assume *iid* dispersal rates between all areas and *iid* extirpation rates within all areas (Figure 9). By explicitly creating each rate parameter, we gain full flexibility to model, for example, distance-dependent dispersal rates. The cladogenic Phylo-CTMC variant additionally integrates over state transitions that coincide with speciation. We have also implemented the data-augmented model described in Landis et al. (2013a), which enables biogeographic analyses for many areas. With RevBayes, one might instead model geographic position as an island-endemic or single-area character as proposed by Sanmartín et al. (2008) and Lemey et al. (2009) by adapting the discrete-morphological models described earlier.

Joint Inference of Combined Data

In the examples above, we described how to design analyses for different types of data. These analyses can be performed independently or jointly, as has been advocated by Nylander et al. (2004) and Ronquist et al. (2012a). Every analysis can be performed jointly if at least one variable in the model is shared. A shared variable will cause the resulting model graph to be connected, which is the only requirement in RevBayes. Note that we used the same variable for the tree (*psi*) for all the examples. Thus, we could jointly infer the phylogeny from molecular data, discrete morphological data, continuous-trait data, and biogeographic data. At the same time, we would accommodate uncertainty in the phylogeny when estimating parameters of the evolutionary process, such as the ancestral morphological states or species ranges.

VALIDATION

The models used in the preceding examples have previously been described and tested in their original publications. Owing to its tremendous flexibility, it is




```
psi <- readTrees("myTree.tre")[1]

for (i in 1:(N*(N-1)))
   d[i] ~ dnExponential(10.0)

for (i in 1:N)
   e[i] ~ dnExponential(10.0)

Q_bg := fnDECRateMatrix(dispersalRates=d, extirpationRates=e)

alpha <- [ 1,1,1,1 ]
c ~ dnDirichlet(alpha)
P_bg := fnCladoProbs(eventProbs=c, numCharacters=N, numStates=2)

area ~ dnPhyloCTMCClado( tree=psi, Q=Q_bg, cladoProbs=P_bg )
area.clamp( data )
```

FIGURE 9. A biogeographic model depicted in graphical-model notation (left) and the corresponding specification in the Rev language (right). Provided rates of dispersal (d) and extirpation (e), Q encodes the instantaneous-rate matrix used to compute transition probabilities for range evolution during anagenesis (along branches). P gives the transition probability of cladogenic events, where the probabilities of event types are given by the distribution, c. Graphical-model notation follows the descriptions in Figures 1–2.

impossible to present a detailed validation of all possible models that can be specified in RevBayes. Instead, we outline our efforts to validate our implementation.

Our general strategy for validating our implementation of models in RevBayes entails several steps: (i) we directly compared the computed probabilities and/or function-return values in the C++ code (low-level comparison) to those of other established software, such as R, MrBayes, Phycas, and BEAST; (ii) we compared the computed probabilities and/or function-return values in small Rev examples (high-level comparison); (iii) we ran analyses on small data sets and compared the results to either MrBayes or BEAST, where applicable.

Most importantly, we implemented an automatic procedure to validate our implementation as follows. A developer or user specifies any model in Rev as if performing an analysis. Next, RevBayes simulates parameter values and data utilizing the imbedded simulation routines. Finally, RevBayes infers parameter estimates from the simulated data and checks how often the true parameters fall into a 90% credible interval. The frequentist expectation of the credible intervals guarantees that 90% of the simulations should recover the true parameters (Huelsenbeck and Rannala 2004). This procedure provides an unprecedented automatic feature to test and validate existing and new implementation in RevBayes.

LIMITATIONS

Understandably, only a subset of all possible models are implemented in RevBayes. Nevertheless, the diversity of models implemented in RevBayes exceeds that of most other programs, such as MrBayes, owing to the immense flexibility for combining models using the Rev language. The addition of many other phylogenetic models will, of course, require extensions of the underlying C++ code. For example, it is currently not possible to specify models for gene-transfer or gene-duplication-and-loss that are available in other more specialized software (Arvestad et al. 2003; Szöllősi et al. 2012; Boussau et al. 2013). As the development of RevBayes continues, such models will become available to users. The fact that not all conceivable models are currenly available in RevBayes, however, should not detract from its primary design strength: the capacity of RevBayes to easily add and extend existing models. We provide an overview of most of the currently implemented models in the RevBayes tutorials, which are available on our website http://www.RevBayes.com.

We note that the availability of a model by itself does not guarantee that it is possible to perform efficient (or even feasible) inference under that model. Instead, clever MCMC algorithms are often crucial for efficient exploration of parameter space, particularly for more complex models (e.g., Vaughan et al. 2014). From this perspective, the design of RevBayes offers several advantages. First, RevBayes provides a number of Monte Carlo algorithms—such as the Metropolis-Hastings (Metropolis et al. 1953; Hastings 1970), and the Metropolis-coupled MCMC (Gilks et al. 1996; Altekar et al. 2004) algorithms—that can be used interchangeably. Second, RevBayes provides a diverse array of proposal mechanisms that can be applied in various combinations to a given parameter to achieve efficient mixing. Third, RevBayes uses adaptive MCMC to automatically tune and optimize the proposal mechanisms (cf., Haario et al. 1999). Slow mixing is a common challenge of MCMC algorithms; RevBayes merely provides a new framework with many features to alleviate these issues.

AVAILABILITY

RevBayes is open source and available from https://github.com/revbayes. It is freely available under GNU General Public License version 3.0. We

are maintaining a website for RevBayes found at http://www.RevBayes.com and a mailing list called *revbayes-users*.

Funding

This work was supported by the Miller Institute for basic research in science [to S.H.]; National Science of Foundation (NSF) [grant DEB-1256993 to T.A.H.]; NSF [grants DEB-DEB-0842181, DEB-0919529, DBI-1356737, and DEB-1457835 to B.R.M.]; and the Swedish Research Council [grant 2011-5622 to F.R.].

Acknowledgments

The authors wish to thank the National Evolutionary Synthesis Center (NESCent) for sponsoring the NESCent Academy Course entitled "Phylogenetic analysis using RevBayes". The participants of this course provided extremely valuable feedback that significantly improved RevBayes and the Rev language. Furthermore, the developers of RevBayes acknowledge generous contributions from: Lars Arvestad, Daniel Ayres, Karen Cranston, Johan Dunfalk, Will Freyman, Laurent Guéguen, Mark Holder, Seraina Klopfstein, Bret Larget, Sibon Li, Ben Liebeskind, Mike May, Emily Jane McTavish, Conor Meehan, Will Pett, Ben Redelings, Felix Reichert, Isabel Sanmartín, Donald Simon, Tanja Stadler, Marc Suchard, Gergely Szöllősi, Paul van der Mark, April Wright, and Chi Zhang. Finally, we want to thank Paul Lewis, David Posada, Frank Anderson, and an anonymous reviewer for helpful comments on the manuscript.

References

Aberer A.J., Kobert K., Stamatakis A. 2014. Exabayes: massively parallel Bayesian tree inference for the whole-genome era. Mol. Biol. Evol. 31:2553–2556.

Altekar G., Dwarkadas S., Huelsenbeck J.P., Ronquist F. 2004. Parallel metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. Bioinformatics 20:407–415.

Arvestad L., Berglund A.-C., Lagergren J., Sennblad B. 2003. Bayesian gene/species tree reconciliation and orthology analysis using MCMC. Bioinformatics 19:i7–i15.

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

Besag J., Green P.J. 1993. Spatial statistics and Bayesian computation. J. Roy. Stat. Soc. B Met. 55:25–37.

Bouckaert R., Heled J., Kühnert D., Vaughan T., Wu C.-H., Xie D., Suchard M.A., Rambaut A., Drummond A.J. 2014. BEAST 2: a software platform for Bayesian evolutionary analysis. PLoS Comput. Biol. 10:e1003537.

Boussau B., Blanquart S., Necsulea A., Lartillot N., Gouy M. 2008. Parallel adaptations to high temperatures in the archaean eon. Nature 456:942–945.

Boussau B., Szöllősi G.J., Duret L., Gouy M., Tannier E., Daubin V. 2013. Genome-scale coestimation of species and gene trees. Genome Res. 23:323–330.

Brown J.M., Lemmon A.R. 2007. The importance of data partitioning and the utility of Bayes factors in Bayesian phylogenetics. Syst. Biol. 56:643–655.

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

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

Drummond A., Suchard M., Xie D., Rambaut A. 2012. Bayesian phylogenetics with beauti and the beast 1.7. Mol. Biol. Evol. 29: 1969–1973.

Drummond A.J., Suchard M.A. 2010. Bayesian random local clocks, or one rate to rule them all. BMC Biol. 8:114.

Felsenstein J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17:368–376.

Felsenstein J. 1985. Phylogenies and the comparative method. Am. Nat. 125:1–15.

Flouri T., Izquierdo-Carrasco F., Darriba D., Aberer A., Nguyen L.-T., Minh B., Von Haeseler A., Stamatakis A.. 2015. The phylogenetic likelihood library. Syst. Biol. 64:356–362.

Galtier N., Gouy M. 1995. Inferring phylogenies from DNA sequences of unequal base compositions. Proc. Natl Acad. Sci. 92:11317–11321.

Gilks W., Richardson S., Spiegelhalter D. 1996. Markov chain Monte Carlo in practice. London: Chapman & Hall/CRC.

Groussin M., Boussau B., Gouy M. 2013. A branch-heterogeneous model of protein evolution for efficient inference of ancestral sequences. Syst. Biol. 62:523–538.

Haario H., Saksman E., Tamminen J. 1999. Adaptive proposal distribution for random walk Metropolis algorithm. Comput. Stat. 14:375–396.

Hansen T.F. 1997. Stabilizing selection and the comparative analysis of adaptation. Evolution 51:1341–1351.

Harrison L.B., Larsson, H.C. 2015. Among-character rate variation distributions in phylogenetic analysis of discrete morphological characters. Syst. Biol. 64:307–324.

Harvey P.H., Pagel M.D. 1991. The comparative method in evolutionary biology, Vol. 239. Oxford: Oxford university press.

Hasegawa M., Kishino H., Yano T. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J. Mol. Evol. 22:160–174.

Hastings W.K. 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57:97–109.

Heath T., Holder M., Huelsenbeck J. 2012. A Dirichlet process prior for estimating lineage-specific substitution rates. Mol. Biol. Evol. 29:939–955.

Heath T.A., Moore B.R. 2014. Bayesian inference of species divergence times. In: Ming-Hui Chen L. K., Lewis P., editors. Bayesian phylogenetics: methods, algorithms, and applications. Sunderland, (MA): Sinauer Associates, p. 487–533.

Ho S.Y., Shapiro B., Phillips M.J., Cooper A., Drummond A.J. 2007. Evidence for time dependency of molecular rate estimates. Syst. Biol. 56:515–522.

Höhna S., Defoin-Platel M., Drummond A.. 2008. Clock-constrained tree proposal operators in Bayesian phylogenetic inference. 8th IEEE International Conference on Bioinformatics and Bioengineering (BIBE 2008), Athens, Greece, Oct 2008.

Höhna S., Drummond A.J. 2012. Guided tree topology proposals for Bayesian phylogenetic inference. Syst. Biol. 61:1–11.

Höhna S., Heath T.A., Boussau B., Landis M.J., Ronquist F., Huelsenbeck J.P. 2014. Probabilistic graphical model representation in phylogenetics. Syst. Biol. 63:753–771.

Holder M., Lewis P. 2003. Phylogeny estimation: traditional and Bayesian approaches. Nat. Rev. Genet. 4:275.

Huelsenbeck J., Larget B., Miller R., Ronquist F. 2002. Potential applications and pitfalls of Bayesian inference of phylogeny. Syst. Biol. 51:673–688.

Huelsenbeck J., Ronquist F. 2001. MRBAYES: Bayesian inference of phylogenetic trees. Bioinformatics 17:754–755.

Huelsenbeck J., Ronquist F., Nielsen R., Bollback J. 2001. Bayesian inference of phylogeny and its impact on evolutionary biology. Science 294:2310–2314.

Huelsenbeck J.P., Larget B., Alfaro M.E. 2004. Bayesian phylogenetic model selection using reversible jump Markov chain Monte Carlo. Mol. Biol. Evol. 21:1123–1133.

Huelsenbeck J.P., Larget B., Swofford D.L. 2000. A compound Poisson process for relaxing the molecular clock. Genetics 154:1879–1892.
Huelsenbeck J.P., Rannala B. 2004. Frequentist properties of Bayesian posterior probabilities of phylogenetic trees under simple and complex substitution models. Syst. Biol. 53:904–913.
Jordan M. 2004. Graphical models. Stat. Sci. 19:140–155.
Jukes T., Cantor C. 1969. Evolution of protein molecules. Mamm. Protein Metab. 3:21–132.
Lakner C., van der Mark P., Huelsenbeck J.P., Larget B., Ronquist F. 2008. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. Syst. Biol. 57:86–103.
Landis M.J., Matzke N.J., Moore B.R., Huelsenbeck J.P. 2013a. Bayesian analysis of biogeography when the number of areas is large. Syst. Biol. 62:789–804.
Landis M.J., Schraiber J.G., Liang M. 2013b. Phylogenetic analysis using lévy processes: finding jumps in the evolution of continuous traits. Syst. Biol. 62:193–204.
Larget B., Simon D. 1999. Markov chain Monte Carlo algorithms for the Bayesian analysis of phylogenetic trees. Mol. Biol. Evol. 16:750–759.
Lartillot N., Brinkmann H., Philippe H. 2007. Suppression of long-branch attraction artefacts in the animal phylogeny using a site-heterogeneous model. BMC Evol. Biol. 7:S4.
Lartillot N., LePage T., Blanquart S.. 2009. Suppression of long-branch attraction artefacts in the animal phylogeny using a site-heterogeneous model. Bioinformatics 25:2286–2288.
Lartillot N., Poujol R. 2011. A phylogenetic model for investigating correlated evolution of substitution rates and continuous phenotypic characters. Mol. Biol. Evol. 28:729–744.
Leebens-Mack J., Vision T., Brenner E., Bowers J.E., Cannon S., Clement M.J., Cunningham C.W., dePamphilis C., deSalle R., Doyle J.J., Eisen J.A., Gu X., Harshman J., Jansen R.K., Kellogg E.A., Koonin E.V., Mishler B.D., Philippe H., Pires J.C., Qiu Y.-L.L., Rhee S.Y., Sjölander K., Soltis D.E., Soltis P.S., Stevenson D.W., Wall K., Warnow T., Zmasek C. 2006. Taking the first steps towards a standard for reporting on phylogenies: minimum information about a phylogenetic analysis (MIAPA). OMICS 10:231–237.
Lemey P., Rambaut A., Drummond A.J., Suchard M.A. 2009. Bayesian phylogeography finds its roots. PLoS Computat. Biol. 5:e1000520.
Lepage T., Bryant D., Philippe H., Lartillot N. 2007. A general comparison of relaxed molecular clock models. Mol. Biol. Evol. 24:2669–2680.
Lewis P.O. 2001. A likelihood approach to estimating phylogeny from discrete morphological character data. Syst. Biol. 50:913–925.
Lewis P.O. 2003. Ncl: a c++ class library for interpreting data files in nexus format. Bioinformatics 19:2330–2331.
Lewis P.O., Holder M.T., Swofford D.L. 2015. Phycas: Software for bayesian phylogenetic analysis. Syst. Biol. 64:525–531.
Li S., Pearl D.K., Doss H. 2000. Phylogenetic tree construction using Markov chain Monte Carlo. J. Am. Stat. Assoc. 95:493–508.
Lunn D., Spiegelhalter D., Thomas A., Best N. 2009. The bugs project: evolution, critique and future directions. Stat. Med. 28:3049–3067.
Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., Teller E. 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21:1087–1092.
Nylander J.A., Ronquist F., Huelsenbeck J.P., Nieves-Aldrey J. 2004. Bayesian phylogenetic analysis of combined data. Syst. Biol. 53: 47–67.
R Core Team. 2014. R: a language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing.
Rannala B., Yang Z. 1996. Probability distribution of molecular evolutionary trees: A new method of phylogenetic inference. J. Mol. Evol. 43:304–311.
Rannala B., Yang Z. 2003. Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. Genetics 164:1645–1656.
Rannala B., Yang Z. 2007. Inferring speciation times under an episodic molecular clock. Syst. Biol. 56:453–466.
Redelings B., Suchard M. 2005. Joint Bayesian estimation of alignment and phylogeny. Syst. Biol. 54:401–418.
Ree R.H., Moore B.R., Webb C.O., Donoghue M.J. 2005. A likelihood framework for inferring the evolution of geographic range on phylogenetic trees. Evolution 59:2299–2311.
Ronquist F., Deans A. 2010. Bayesian phylogenetics and its influence on insect systematics. Annu. Rev. Entomol. 55:189–206.
Ronquist F., Huelsenbeck J. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572–1574.
Ronquist F., Klopfstein S., Vilhelmsen L., Schulmeister S., Murray D.L., Rasnitsyn A.P. 2012a. A total-evidence approach to dating with fossils, applied to the early radiation of the hymenoptera. Syst. Biol. 61:973–999.
Ronquist F., Teslenko M., van der Mark P., Ayres D.L., Darling A., Höhna S., Larget B., Liu L., Suchard M.A., Huelsenbeck J.P. 2012b. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. Syst. Biol. 61:539–542.
Sanmartín I., van der Mark P., Ronquist F. 2008. Inferring dispersal: a Bayesian approach to phylogeny-based island biogeography, with special reference to the Canary Islands. J. Biogeogr. 35:428–449.
Suchard M.A., Redelings B.D. 2006. Bali-phy: simultaneous Bayesian inference of alignment and phylogeny. Bioinformatics 22:2047–2048.
Szöllősi G.J., Boussau B., Abby S.S., Tannier E., Daubin V. 2012. Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations. Proc. Natl Acad. Sci. 109:17513–17518.
Tavaré S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. In: Miura R.M., editor. Some mathematical questions in biology DNA sequence analysis, Vol. 17. American Mathematical Society, Providence (RI), p. 57–86.
Thorne J., Kishino H., Painter I.S. 1998. Estimating the rate of evolution of the rate of molecular evolution. Mol. Biol. Evol. 15:1647–1657.
Thorne J.L., Kishino H. 2002. Divergence time and evolutionary rate estimation with multilocus data. Syst. Biol. 51:689–702.
Vaughan T.G., Kühnert D., Popinga A., Welch D., Drummond A.J. 2014. Efficient Bayesian inference under the structured coalescent. Bioinformatics 30:2272–2279.
Yang Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. J. Mol. Evol. 39:306–314.
Yang Z. 1996. Among-site rate variation and its impact on phylogenetic analyses. Trends in Ecol. & Evol. 11:367–372.
Yang Z. 2007. PAML 4: phylogenetic analysis by maximum likelihood. Mol. Biol. Evol. 24:1586–1591.
Yang Z. 2014. Molecular evolution: A statistical approach. Oxford, UK: Oxford University Press.
Yang Z., Rannala B. 2012. Molecular phylogenetics: principles and practice. Nat. Rev. Genet. 13:303–314.
Yang Z., Yoder A.D. 2003. Comparison of likelihood and Bayesian methods for estimating divergence times using multiple gene loci and calibration points, with application to a radiation of cute-looking mouse lemur species. Syst. Biol. 52:705–716.