Version dated: June 27, 2014

Probabilistic Graphical Model Representation in Phylogenetics
Supplementary information

Sebastian Höhna[1,2], Tracy A. Heath[3,4], Bastien Boussau[3,5], Michael J. Landis[3], Fredrik Ronquist[6] and John P. Huelsenbeck[3,7]

[1]*Department of Mathematics, Stockholm University, Stockholm, SE-106 91 Stockholm, Sweden;*

[2]*Department of Evolution and Ecology, University of California, Davis, Storer Hall, One Shields Avenue, Davis, CA 95616, USA;*

[3]*Department of Integrative Biology, University of California, Berkeley, CA, 94720, USA;*

[4]*Department of Ecology and Evolutionary Biology, University of Kansas, Lawrence, KS, 66045, USA;*

[5]*Bioinformatics and Evolutionary Genomics, Université de Lyon, Villeurbanne, France;*

[6]*Department of Bioinformatics and Genetics, Swedish Museum of Natural History, SE-10405 Stockholm, Sweden*

[7]*Department of Biological Science, King Abudlaziz University, Jeddah, Saudi Arabia*

Corresponding author: Sebastian Höhna, Department of Mathematics, Stockholm University, Stockholm, SE-106 91 Stockholm, Sweden; E-mail: sebastian.hoehna@gmail.com.

Additional phylogenetic graphical models

A model of continuous character evolution

Graphical models are high-level representations that do not depend on details of the model. As a result, similar models will have similar representations. We provide here the example of a Brownian motion model of the evolution of continuous characters to convey this point (Felsenstein 1985). We use the same 5 species phylogeny as before. Figure S.1 shows that the graphical representation of this model is very similar to that of the former binary model, notably because the structure of the phylogenetic tree is still very obvious, and because a plate indicates replication over several characters. Only a few details differ between figures Figs. 4 and S.1: they describe peculiarities of the model of character evolution. In particular, the ancestral state at the root for the Brownian motion model is drawn uniformly between values *min* and *max*, and the parameter δ for the variance of the model is drawn from an exponential distribution with rate λ.

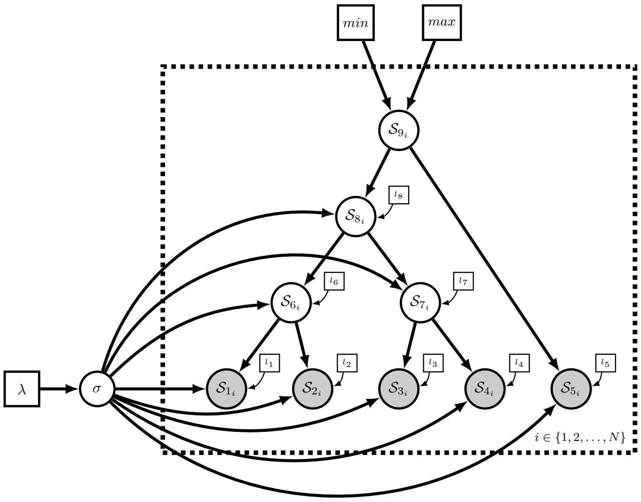

Figure S.1: Graphical model of continuous trait evolution example. This model assumes that the tree structure (topology) is known and is explicitly shown by the model graph. Additionally, it is assumed that the effective branch lengths $\{l_1, \ldots, l_8\}$ are known. The continuous trait values of the root (S_{9_i}) are drawn from a uniform distribution between fixed *min* and *max* values. Every other continuous trait value, for internal nodes S_{6_i}, S_{7_i} and S_{8_i} and tip nodes $\{S_{1_i}, \ldots, S_{5_i}\}$, evolve under a Brownian motion with scaling parameter σ. The scaling parameter σ is shared for all traits and has an exponential prior with known rate λ.

A mixture model for rate variation among sites

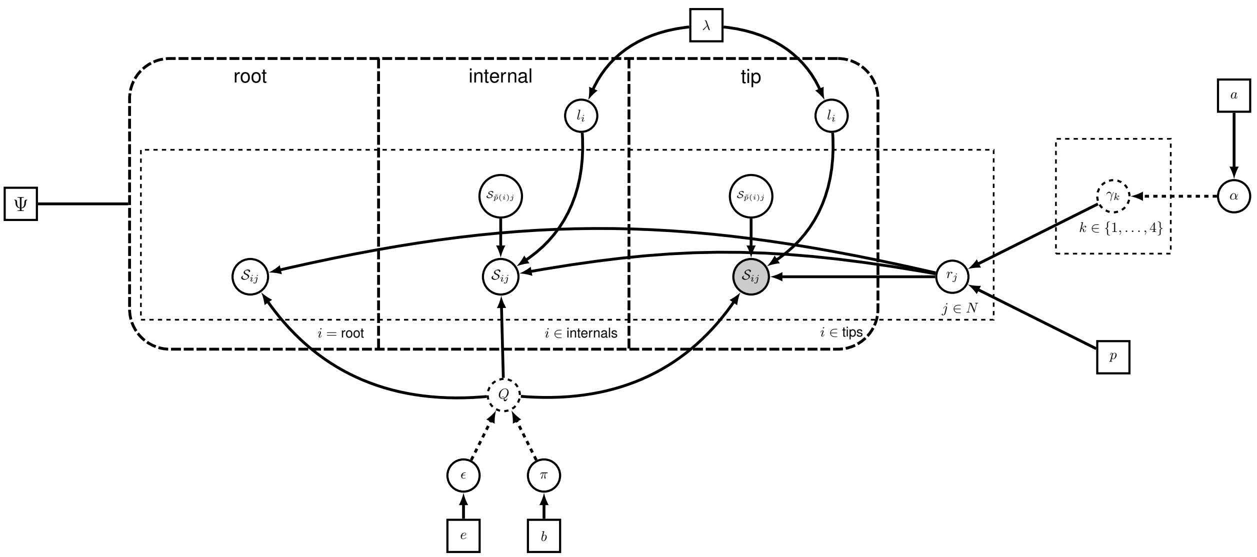

Figure S.2: Graphical model representation of among site rate variation model (GTR+Γ). In this example the tree topology Ψ is known (constant). We assume again an exponential prior distribution on the branch lengths l_i with known rate λ and general time reversible rate (GTR) matrix with base frequencies π and exchangeability rates ϵ both drawn from a dirichlet distribution with parameters b and e, respectively. The rate variation across sites is modeled by a mixture of four rate categories, each with probability p and rate multiplier γ_k. The rate multipliers are computed from the quantiles of a gamma distribution with rate and shape parameters set to the same value, α, to ensure that the mean of the rate multipliers is 1.0.

Mixture models are very common in phylogenetics. As an example, we present the model graph for the GTR model with rate variation across sites modeled by a discrete gamma distribution with four rate categories (Yang 1994; 1996). One way of understanding this model is to consider the rate of each site drawn from a mixture of four different rates. The mixture distribution is represented by a multinomial distribution with four possible outcomes, the four possible rates, with equal probabilities p. The rate multiplier of each category is computed by the quantiles ($q \in \{0.125, 0.375, 0.625, 0.875\}$) of a gamma distribution with rate parameter α and shape parameter α ($\gamma_i = \text{qgamma}(q_i, \text{shape}{=}\alpha, \text{rate}{=}\alpha)$). Note, that it is standard practice to set the rate and shape parameters to be equal to guarantee that the average of the rate multipliers is 1.0. The overall rate or amount of evolution is typically controlled by other parameters in the model, such as the branch length parameters in our case.

In general, mixture models can be represented by a multinomial distribution and the category one observation belongs to can be obtained via an indicator variable. The actual mapping to the mixture category can be integrated over by summation of the probabilities of being in each category or marginalized over within the MCMC algorithm. The graphical model framework is not restricted to any of these methods and can be applied to many types of mixture models, for example infinite mixture models like the Dirichlet Process Prior model (Huelsenbeck et al. 2006; Heath 2012; Heath et al. 2012).

A autocorrelated relaxed clock model

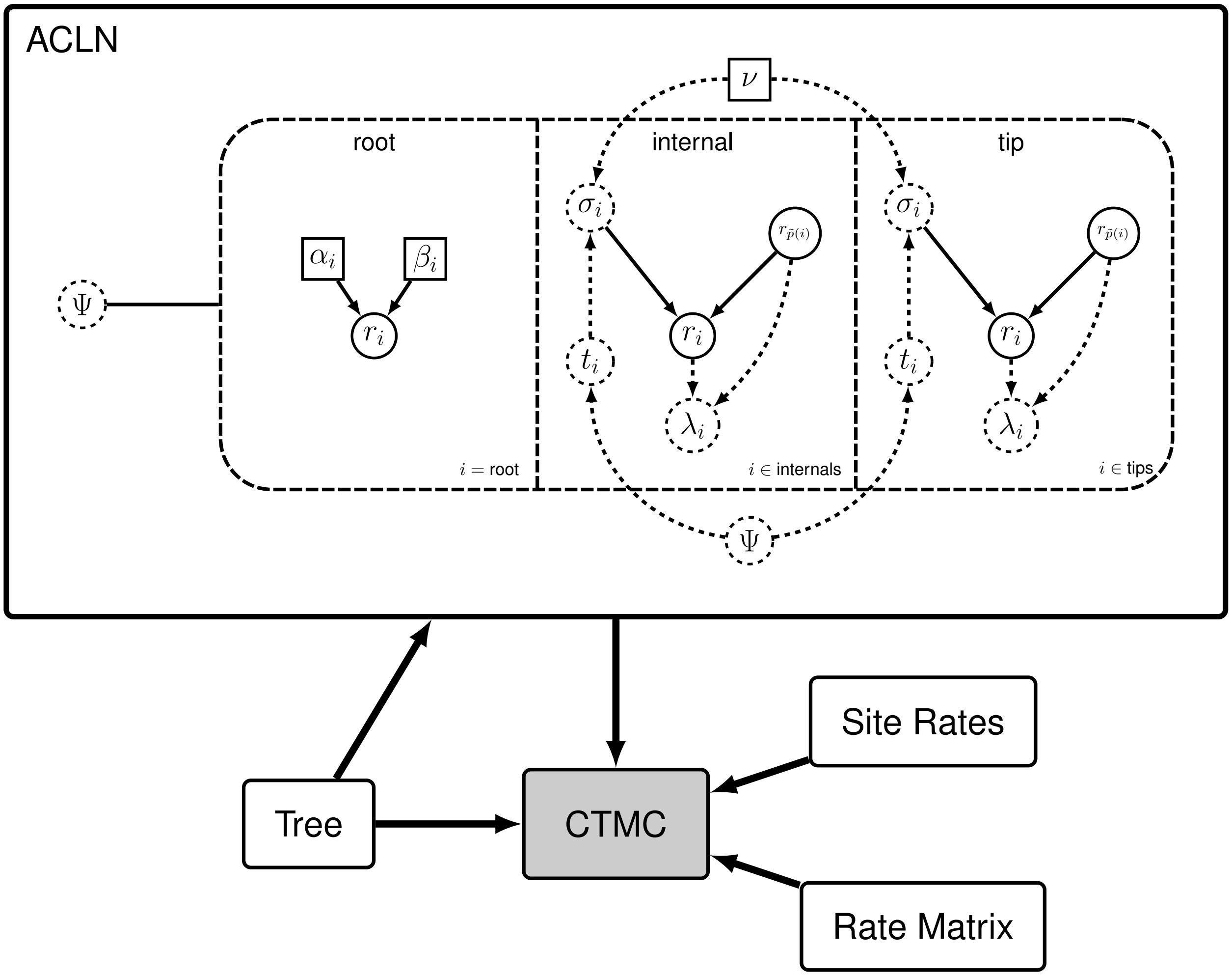

Figure S.3: The graphical model representation of the autocorrelated log-normal (ACLN) relaxed-clock model (Thorne et al. 1998; Kishino et al. 2001; Thorne and Kishino 2002). Here, the tree plate partitions the nodes into the root, tips, and internals. The ACLN model assumes that the substitution rate at each node is drawn from a log-normal distribution that is centered on the parent node $r_{\tilde{p}(i)}$ and the rates at each node change according to a geometric Brownian motion model. The variance σ_i of the log-normal distribution is determined by the time duration of the branch leading to node i and the Brownian motion parameter ν. We denote the rate along the branch as λ_i, which is a deterministic node, where the value is equal to $\frac{(r_{\tilde{p}(i)}+r_i)}{2}$.

Relaxed-clock models are commonly used in analyses seeking to date species divergences and here we provide an example of a graphical module for the autocorrelated relaxed clock model (Fig. S.3; Thorne et al. 1998; Kishino et al. 2001; Thorne and Kishino 2002). The autocorrelated relaxed clock specifies that the clock rates evolve under a geometric Brownian motion ($r_i \sim \text{lnorm}(\text{mean}{=}r_{\tilde{p}(i)}, \text{variance}{=}t_i\nu)$). The tree plate enables a compact representation of this relaxed clock but still emphasizes that the clock rates depend on the tree topology. Furthermore, the graphical model representation illustrates that the parameter of interest is the rate along the branch, which we denote as λ_i. The product of the branch rate and time duration, $\lambda_i \times t_i$, is the branch length – a component of the tree likelihood. Under the ACLN model, this parameter is a deterministic node, where the value is the average of the rates subtending the branch (Fig. S.3). Alternative relaxed-clock models, like the uncorrelated log-normal (UCLN; Drummond et al. 2006) or Dirichlet process prior (Heath et al. 2012) describe the branch rates directly and under these models λ_i is a stochastic node.

RevBayes: Implementation and Examples

Software Implementation

BUGS (Bayesian inference using Gibbs sampling) is a very popular software package for Bayesian inference (Lunn et al. 2000; Sturtz et al. 2005; Lunn et al. 2009; 2012). It defines its own modeling language, the BUGS language, which is entirely based on graphical model concepts. A model is specified by setting up the dependency structure of the variables, both deterministic and stochastic, in a special model definition file. This file is compiled into a model graph, and once the data and initial values are read in, the posterior probability distribution can be estimated using Gibbs sampling or, more recently, the Metropolis-Hastings algorithm. The focus in BUGS is on linear models, even though a few other model types are also available.

Unfortunately, BUGS is not suited for PhyloGMs. PhyloGMs include a number of variable types and probability distributions that are not implemented in BUGS, such as tree topologies, instantaneous rate matrices, and continuous time Markov chains. The domain-specific PhyloGM objects also put special demands on the computational machinery, such as efficient belief propagation in tree plates and effective MCMC samplers of tree topologies (Lakner et al. 2008; Höhna and Drummond 2012). Furthermore, most PhyloGMs include a graph learning problem, in that part of the graph structure (the topology of the phylogenetic tree) is considered a random variable. Currently, such inference problems are foreign to BUGS. Finally, PhyloGMs are considerably larger than most other graphical models, raising significant challenges in handling the objects in a manner that allows fast computation and leaves a small memory footprint.

The limitations of BUGS motivate an independent software implementation for PhyloGMs. We provide such an implementation in *RevBayes* (www.RevBayes.net). The software will be presented in more detail elsewhere but we briefly outline it here. RevBayes provides a command-line interface for interactive analyses, much like the widely used statistical software package *R* (R Core Team 2013). However, unlike R and BUGS, RevBayes allows users to interactively construct complex graphical models, step by step, and it supports all the objects needed to build PhyloGMs. Like BUGS, the specification of a model closely mirrors its visual graphical-model representation. The language used by RevBayes, *Rev*, combines features of the R and BUGS languages with those of popular object-oriented programming languages. The similarities in syntax between Rev and R are intended to help users with previous R experience to learn the language quickly.

Examples

In the following section we have listed the two *baculum* examples, a non-phylogenetic example assuming no dependence structure between species and a phylogenetic example using the phylogenetic tree as structural dependence.

A non-phylogenetic example.— In this example we have observations on the presence and absence of a baculum for five species. We model this by a Bernoulli distribution and estimate the parameter p. This example is given in the Rev language and thus can be run in *RevBayes*. It is available separately as a runnable Rev source file too.

```
# set the prior parameters
alpha <- 1      # this creates a constant variable with value 1
beta <- 1

# create the stochastic variable for the parameter of the binomial distribution
p ~ beta(alpha,beta)    # this creates a stochastic variable drawn from
                        # a beta distribution with parameters alpha and beta

# create the data
data <- [1,1,1,0,0]     # this creates a vector

for (i in 1:data.size()) {
  x[i] ~ bernoulli(p)   # this creates a stochastic variable drawn from
                        # a Bernoulli distribution with parameter p
  # attach/clamp the data
  x[i].clamp(data[i])
}

# create the model from the DAG
mymodel <- model(p)



# create the moves/proposals that change the parameters
# of the model during the MCMC
moves[1] <- mSlide(p, delta=0.2, weight=1.0)

monitors[1] <- modelmonitor(filename= "GraphicalModels_Example_1a.log",
printgen=10, separator = " ")
monitors[2] <- screenmonitor(printgen=10, separator = " ", p)



mymcmc <- mcmc(mymodel, monitors, moves)

# If you choose more or different proposals,
# or different weights for the proposals,
# then the number of proposals changes per iteration.
# Currently there is only one proposal with weight 1.0.
mymcmc.burnin(generations=2000,tuningInterval=100)
mymcmc.run(generations=200000)

result <- readTrace("GraphicalModels_Example_1a.log")
result[5]
```

A phylogenetic example

The second example includes all 274 mammalian species included by dos Reis et al. (2012). The data are read in from file. This examples assumes an underlying phylogenetic model and specifies the evolution of the presence/absence of the baculum by a continuous time Markov model. Note, that we assume a different root frequency of the baculum than the stationary frequency of the continuous time Markov model (see Fig. 3.b). The example is also available separately as a runnable Rev source file.

```
# Read the data.
# The readCharacter function returns a vector of matrices.
# We just take the first one.
D <- readCharacterData("data/baculumData01.nex")[1]

# Get some useful variables from the data
nSites <- D.nchar()[1]

# Create the parameter for the root frequencies.
# Instead of a beta distribution we use the Dirichlet distribution
# because the data type needs to be a simplex.
# Set a flat prior.
rf_prior <- [1,1]
# Create the random variable for the root frequencies
rf ~ dirichlet(rf_prior)

# Create a move/proposal that changes the root-frequencies during the MCMC.
moves[1] <- mSimplexElementScale(rf, alpha=10.0, tune=true, weight=2.0)



# Now let us create the random variables for the continuous time Markov process
# that changes the absent/present state along the tree.
# We use the F81 rate matrix again with a Dirichlet distribution
# as the prior on the base-frequencies.
bf_prior <- [1,1]
# Create the random variable for the base-frequencies.
bf ˜ dirichlet(bf_prior)
# Construct the rate matrix.
Q := F81(bf)

# Create a move/proposal that changes the base-frequencies during the MCMC.
moves[2] <- mSimplexElementScale(bf, alpha=10.0, tune=true, weight=2.0)



# We use a fixed tree (dos Reis et al.) read from a file.
tau <- readTrees("data/mammalia_dosReis.tree")[1]

# Just use the default clock rate. (We could also omit this parameter.)
clockRate <- 1.0

# Construct a random variable for the sequence evolution model.
seq ˜ substModel(tree=tau, Q=Q, branchRates=clockRate,
                      rootFrequencies=rf, nSites=nSites, type="Standard")

# Attach the data.
seq.clamp(D)



# Create the model from the DAG.
mymodel <- model(Q)



monitors[1] <- modelmonitor(filename= "GraphicalModels_Example_2.log",
                              printgen=10, separator = " ")
monitors[2] <- screenmonitor(printgen=10, separator = " ", bf, rf)
```

```
mymcmc <- mcmc(mymodel, monitors, moves)

# If you choose more or different proposals, or different weights for the proposals,
# then the number of proposals changes per iteration.
# Currently there are only two proposal with weight 2.0 each.
mymcmc.burnin(generations=2000,tuningInterval=100)
mymcmc.run(generations=200000)

result <- readTrace("GraphicalModels_Example_2.log")
result[5]
```

The estimated root frequency is rf $= 0.503$ with HPD$= \{0.07, 0.91\}$ and the estimated equilibrium frequency is rf $= 0.48$ with HPD$= \{0.38, 0.58\}$. Although there is a slight difference in the parameter estimates, our assumption of a different root frequency is not supported by the data.

REFERENCES

dos Reis, M., J. Inoue, M. Hasegawa, R. J. Asher, P. C. Donoghue, and Z. Yang. 2012. Phylogenomic datasets provide both precision and accuracy in estimating the timescale of placental mammal phylogeny. Proceedings of the Royal Society B: Biological Sciences 279:3491–3500.

Drummond, A., S. Ho, M. Phillips, and A. Rambaut. 2006. Relaxed Phylogenetics and Dating with Confidence. PLoS Biol 4:e88.

Felsenstein, J. 1985. Phylogenies and the comparative method. American Naturalist 125:1–15.

Heath, T. A. 2012. A hierarchical bayesian model for calibrating estimates of species divergence times. Systematic biology 61:793–809.

Heath, T. A., M. T. Holder, and J. P. Huelsenbeck. 2012. A Dirichlet process prior for estimating lineage-specific substitution rates. Molecular biology and evolution 29:939–55.

Höhna, S. and A. J. Drummond. 2012. Guided Tree Topology Proposals for Bayesian Phylogenetic Inference. Systematic Biology 61:1–11.

Huelsenbeck, J. P., S. Jain, S. W. D. Frost, and S. L. K. Pond. 2006. A Dirichlet process model for detecting positive selection in protein-coding DNA sequences. Proceedings of the National Academy of Sciences 103:6263–6268.

Kishino, H., J. L. Thorne, and W. Bruno. 2001. Performance of a divergence time estimation method under a probabilistic model of rate evolution. Molecular Biology and Evolution 18:352–361.

Lakner, C., P. van der Mark, J. P. Huelsenbeck, B. Larget, and F. Ronquist. 2008. Efficiency of markov chain monte carlo tree proposals in bayesian phylogenetics. Systematic Biology 57:86–103.

Lunn, D., C. Jackson, D. J. Spiegelhalter, N. Best, and A. Thomas. 2012. The BUGS book: A practical introduction to Bayesian analysis vol. 98. CRC Press.

Lunn, D., D. Spiegelhalter, A. Thomas, and N. Best. 2009. The BUGS project: Evolution, critique and future directions. Statistics in medicine 28:3049–3067.

Lunn, D. J., A. Thomas, N. Best, and D. Spiegelhalter. 2000. WinBUGS-a bayesian modelling framework: concepts, structure, and extensibility. Statistics and computing 10:325–337.

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

Sturtz, S., U. Ligges, and A. E. Gelman. 2005. R2winbugs: a package for running winbugs from r. Journal of Statistical software 12:1–16.

Thorne, J. and H. Kishino. 2002. Divergence time and evolutionary rate estimation with multilocus data. Systematic Biology 51:689–702.

Thorne, J., H. Kishino, and I. S. Painter. 1998. Estimating the rate of evolution of the rate of molecular evolution. Molecular Biology and Evolution 15:1647–1657.

Yang, Z. 1994. Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites: approximate methods. Journal of Molecular evolution 39:306–314.

Yang, Z. 1996. Among-site rate variation and its impact on phylogenetic analyses. Trends in Ecology & Evolution 11:367–372.