Phylogenetic Inference using `RevBayes`

Phylogenies and the mutlivariate comparative method

Nicolas Lartillot and Sebastian Höhna

1 Introduction

The subject of the comparative method is the analysis of trait evolution at the macroevolutionary scale. In a comparative context, many different questions can be addressed: tempo and mode of evolution, correlated evolution of multiple quantitative traits, trends and bursts, changes in evolutionary mode correlated with major key innovations in some groups, etc (for a good introduction see Harvey and Pagel 1991).

In order to correctly formalize comparative questions, the underlying phylogeny should always be explicitly accounted for. This point is clearly illustrated, in particular, by the independent contrasts method (Felsenstein 1985b; Huelsenbeck and Rannala 2003). Practically speaking, the phylogeny and the divergence times are usually first estimated using a separate phylogenetic reconstruction software. In a second step, this time-calibrated phylogeny is used as an input to the comparative method. Doing this, however, raises a certain number of methodological problems:

- the uncertainty about the phylogeny (and about divergence times) is ignored
- the traits themselves may have something to say about the phylogeny
- the rate of substitution, and more generally the parameters of the substitution process, can also be seen as quantitative traits, amenable to a comparative analysis.

All these points are not easily formalized in the context of the step-wise approach mentioned above. Instead, what all this suggests is that phylogenetic reconstruction, molecular dating and the comparative method should all be considered jointly, in the context of one single overarching probabilistic model.

Thanks to its modular structure, `RevBayes` represents a natural framework for attempting this integration. The aim of the present tutorial is to guide you through a series of examples where this integration is achieved, step by step. It can also be considered as an example of the more general perspective of *integrative modeling*, which can be recruited in many other contexts.

2 Data and files

We provide several data files which we will use in this tutorial. You may want to use your own data instead. In the **`data`** folder, you will find the following files

- **`primates_cytb.nex`**: Alignment of the *cytochrome b* subunit from 23 primates representing 14 of the 16 families (*Indriidae* and *Callitrichidae* are missing).

- **primates_lhtlog.nex**: 2 life-history traits (endocranial volume (ECV), body mass; each for males and females separately) for 23 primate species (taken from the Anage database, De Magalhaes and Costa 2009). The traits have been log-transformed.
- **primates.tree**: A time calibrated phylogeny of the same 23 primates.

3 Univariate Brownian evolution of quantitative traits

As a first preliminary exercise, we wish to reconstruct the evolution of body mass in primates and, in particular, estimate the body mass of their last common ancestor. For this, we will assume that the logarithm of body mass follows a simple univariate Brownian motion along the phylogeny. In a first step, we will ignore phylogenetic uncertainty: thus, we will assume that the Brownian process describing body mass evolution runs along a fixed time-calibrated phylogeny (with fixed divergence times), such as specified in the file **primates.tree**.

→ You may want to take the time to visualize the tree given in **primates.tree** as well as the matrix of quantitative traits specified by the **primates_lhtlog.nex** file, before going into the modeling work described below.

3.1 The model and the priors

A univariate Brownian motion $x(t)$ is parameterized by its starting value at the root of the phylogeny $x(0)$ and a rate parameter σ. This rate parameter tunes the amplitude of the variation per unit of time. Specifically, along a given time interval $(0, T)$, the value of X at time T is normally distributed, with mean $x(0)$ and variance $\sigma^2 T$:

$$x(T) \quad \sim \quad \text{Normal}\left(x(0), \sigma^2 T\right).$$

Concerning σ, we can formalize the idea that we are ignorant about the *scale* (the order of magnitude) of this parameter by using a log-uniform prior:

$$\sigma \quad \sim \quad \frac{1}{\sigma}.$$

Concerning the initial value $x(0)$ of the Brownian process at the root of the phylogeny. Alternatively, you may want to specify a normal distribution as the prior distribution on the root value if you have some prior information.

Finally, the tree topology ψ is, as mentioned above, fixed to some externally given phylogeny. The entire model is now specified: tree ψ, variance σ and Brownian process $x(t)$:

$$\begin{aligned} \sigma &\sim \frac{1}{\sigma}, \\ x(0) &\sim \text{Uniform}, \\ x(t) \mid \Psi, \sigma &\sim \text{Brownian}\left(x(0),\, \psi,\, \sigma\right). \end{aligned}$$

Conditioning the model on empirical data by clamping $x(t)$ at the tips of the phylogeny, we can then run a MCMC to sample from the joint posterior distribution on σ and x. Once this is done, we can obtain posterior means, medians or credible intervals for the value of body mass or other life-history traits for specific ancestors.

3.2 Programming the model in RevBayes

The problem of continuous trait evolution —just as for discrete trait evolution— along a phylogeny is that we do not know the values of the traits at the internal nodes. That means, that we need to treat the states at the internal nodes as additional parameters of the model. For discrete characters we use the sum-product (a.k.a. pruning) algorithm (Felsenstein 1981) to analytically integrate over all possible states at the internal nodes. For continuous characters (traits) similar methods have been proposed. In RevBayes you have three main ways of specifying this model and running an analysis on it. The three approaches are: (1) phylogenetic independent contrasts using the reduced likelihood (REML), (2) Brownian motion using a phylogenetic covariance matrix, and (3) a full Brownian motion model using data augmentation. Each of these approaches has there advantages and disadvantages as will be explained below. Nevertheless, all approaches give the same results in terms of rate estimation.

3.3 Phylogenetic Independent Contrasts using the reduced likelihood (REML)

The reduced or restricted maximum likelihood (REML) method computes the probability of observing the continuous character at the tips by an analytical solution to integrate over the internal states (Felsenstein 1985a). This analytical solution is very fast to compute and thus can be applied to large phylogenies and/or many independent characters. However, the REML method looses the information about the location of the root state and thus you cannot infer which state the root or other internal nodes have.

You do not need to understand the algorithm but we provide a sketch of the idea behind REML to give you some insights. REML compute the values at the internal nodes as the phylogenetic contrasts $x_k = x_i - x_j$ where x_i and x_j are the values of the child nodes in the phylogeny. Then, we can compute the probability of observing the contrast x_k using the probability density of a normal distribution with mean $\mu = 0$ and standard deviation $\sigma = \sqrt{\nu_i + \delta_i + \nu_j + \delta_j}$ where ν_i and ν_j are the (scaled) branch lengths leading to node i and j respectively. δ is the additional uncertainty that is propagated through the phylogeny and is compute by $\delta_k = ((\nu_i + \delta_i) * (\nu_j + \delta_j))/(\nu_i + \delta_i + \nu_j + \delta_j)$. These computations are done for you in the RevBayes distribution called **dnPhyloBrownianREML**.

In the directory **RevBayes_scripts/** you will find a script called **primatesMass_REML.Rev**. This script implements the univariate Brownian model described above. Instead of re-typing the content of script entirely in the context of an interactive RevBayes session, you can instead run the script directly:

```
source("RevBayes\_scripts/primatesMass_REML.Rev")
```

This script essentially reformulates what has been explained in the last subsection and serves as an example solution for you. For the later section you need to adjust the script.

Let us go through the script step by step in the Rev language. First, load the trait data:

```
contData <- readContinuousCharacterData("data/primates_lhtlog.nex")
```

If you type you will see that the continuous character data matrix contains several characters (columns).

```
contData

   Continuous character matrix with 23 taxa and 11 characters
   ===========================================================
   Origination:                primates_lhtlog.nex
   Number of taxa:             23
   Number of included taxa:    23
   Number of characters:       11
   Number of included characters: 11
   Datatype:                   Continuous
```

Since we only want the body mass (of females) we exclude all but the third character

```
contData.excludeAll()
contData.includeCharacter(3)
```

Next, load the time-tree from file. Remember that we use in this first simple example a fixed tree that we assume is known without uncertainty.

```
treeArray <- readTrees("data/primates.tree")
psi <- treeArray[1]
```

As usual, we start be initializing some useful helper variables. For example, we set up a counter variable for the number of moves that we already added to our analysis. This will make it much easier if we extend the model or analysis to include additional moves or to remove some moves.

```
mi = 0
```

Then, we define the overall rate parameter σ which we assign a (truncated) log-uniform prior. Note that it is more efficient in Bayesian inference to specify a uniform prior and then to transform the parameter which we will use here:

```
logSigma ~ dnUniform(-5,5)
sigma := 10^logSigma
```

Since the rate of trait evolution **`logSigma`** is a stochastic variable and we want to estimate it, we need to add a sliding move on it

```
moves[++mi] = mvSlide(logSigma, delta=1.0, tune=true, weight=2.0)
```

Next, define a random variable from the univariate Brownian-Phylo-REML process, which we will call **`logmass`**:

```
logmass ~ dnPhyloBrownianREML(psi, branchRates=1.0, siteRates=sigma, nSites=1)
```

Now, condition the Brownian model on empirically observed values for body mass in the extant taxa.

```
logmass.clamp( contData )
```

The model is now entirely specified and we can create a model object containing the entire model graph by providing it with only one of our model variables, *e.g.,***`sigma`**.

```
mymodel = model(sigma)
```

To see what it happing during the MCMC let us make a screen monitor that tracks the rate **`sigma`**

```
monitors[1] = mnScreen(printgen=10, sigma)
```

Additionally, we'll use a file monitor that does the same thing, but directly stores the values into a file:

```
monitors[2] = mnFile(filename="output/primates_mass_REML.log", printgen=10, separator = TAB,
    sigma)
```

We can finally create a mcmc, and run it for a good 100 000 cycles after we did a burnin phase of 10 000 iterations:

```
mymcmc = mcmc(mymodel, monitors, moves)
mymcmc.burnin(generations=10000,tuningInterval=500)
mymcmc.run(100000)
```

Exercises

- Run the model.
- using **`Tracer`**, visualize the posterior distribution on the rate parameter **`sigma`**
- calculate the 95% credible interval for the rate of evolution of the log of body mass (σ)

3.4 Phylogenetic covariance matrix

The second method to we will use creates a phylogenetic covariance matrix. The phylogenetic covariance matrix method integrates over the states at the internal nodes as well but uses instead a multivariate normal distribution. The key advantage is that this method provides information about the root state since it models the root state as an additional parameter of the model. The disadvantage is that it is very computationally intensive. That means, that the phylogenetic covariance matrix approach may take long for very large data sets (at least in its current implementation).

→ Copy the file **primatesMass_REML.Rev**, name it for example **primatesMass_Cov.Rev** and start editing it.

In the previous example, the REML approach, we did not specify a parameter for the state at the root. In this exercise, we need this additional parameter. Let us use a uniform prior distribution on the logarithm of the root mass.

```
rootlogmass ~ dnUniform(-100,100)
```

Next, we'll specify a sliding move that proposes new values for the **rootlogmass** randomly drawn from a window centered around the current value.

```
moves[++mi] = mvSlide(rootlogmass,delta=10,tune=true,weight=2)
```

Finally, we need to substitute the **dnPhyloBrownianREML** by **dnPhyloBrownianMVN** to use the phylogenetic covariance matrix approach.

```
logmass ~ dnPhyloBrownianMVN(psi, branchRates=1.0, siteRates=sigma, rootStates=rootlogmass,
    nSites=1)
```

This will automatically connect the parameters of the model together.

Additionally, we now have the extra parameter **rootlogmass** which we want to monitor. Thus we need to replace the file monitor and use instead.

```
monitors[2] = mnFile(filename="output/primates_mass_Cov.log", printgen=10, separator = TAB,
    sigma, rootlogmass)
```

→ Don't forget to change the output file names in the monitors, otherwise your old analyses files will be overwritten.

Exercises

- Run the analysis.

- Using **Tracer**, visualize the posterior distribution on the rate parameter **sigma** and the **rootlogmass**
- How does the posterior distribution of **sigma** looks compared with the first analysis?
- Calculate the 95% credible interval for the rate of evolution of the log of body mass (σ) and the **rootlogmass**

3.5 Data augmentation

The third method to we will use creates a phylogenetic covariance matrix.

→ Copy the file **primatesMass_REML.Rev**, name it for example **primatesMass_DA.Rev** and start editing it.

In the previous example, the REML approach, we did not specify a parameter for the state at the root. In this exercise, we need this additional parameter. Let us use a uniform prior distribution on the logarithm of the root mass.

```
rootlogmass ~ dnUniform(-100,100)
```

Next, we'll specify a sliding move that proposes new values for the **rootlogmass** randomly drawn from a window centered around the current value.

```
moves[++mi] = mvSlide(rootlogmass,delta=10,tune=true,weight=2)
```

```
numNodes = psi.nnodes()
numTips = psi.ntips()
```

```
# univariate Brownian process along the tree
# parameterized by sigma
for (i in (numNodes-1):(numTips+1) ) {
  logmass[i] ~ dnNormal( logmass[psi.parent(i)], sd=sigma*sqrt(psi.branchLength(i)) )

  # moves on the Brownian process
  moves[++mi] = mvSlide( logmass[i], delta=10, tune=true ,weight=2)
}
```

```
for (i in numTips:1 ) {
  logmass[i] ~ dnNormal( logmass[psi.parent(i)], sd=sigma*sqrt(psi.branchLength(i)) )

  # condition Brownian model on quantitative trait data (second column of the dataset)
  logmass[i].clamp(contData.getTaxon(psi.nodeName(i))[1])
}
```

Finally, we need to substitute the **dnPhyloBrownianREML** by **dnPhyloBrownianMVN** to use the phylogenetic covariance matrix approach.

```
logmass ~ dnPhyloBrownianMVN(psi, branchRates=1.0, siteRates=sigma, rootStates=rootlogmass,
    nSites=1)
```

This will automatically connect the parameters of the model together.

Additionally, we now have the extra parameter **rootlogmass** which we want to monitor. Thus we need to replace the file monitor and use instead.

```
monitors[2] = mnModel(filename="output/primates_mass_DA.log", printgen=10, separator = TAB)
```

→ Don't forget to change the output file names in the monitors, otherwise your old analyses files will be overwritten.

Exercises

- Run the analysis.
- Using **Tracer**, visualize the posterior distribution on the rate parameter **sigma** and the **rootlogmass** and the internal states.
- How does the posterior distribution of **sigma** looks compared with the first and second analysis?
- Calculate the 95% credible interval for the rate of evolution of the log of body mass (σ) and the **rootlogmass**. Have they changed?

4 Correlated evolution of multiple traits

Next, we would like to estimate the correlation between the $K = 3$ life-history traits given in the **plac40lhtlog.nex** file, while properly taking into account phylogenetic inertia. To do so, we will assume that the traits jointly evolve along the phylogeny as a *multivariate* Brownian process. We will estimate the *covariance matrix* of this process and assess the empirical support in favor of positive or negative correlations between pairs of traits in terms of posterior probabilities of having positive or negative entries in this covariance matrix. At this stage of the tutorial, we will again ignore phylogenetic uncertainty.

4.1 The model and the priors

A multivariate Brownian process $X(t)$, of dimension K (here $K = 3$), is entirely parameterized by its starting value ($X(0)$ at the root of the phylogeny, which a vector of dimension K) and a $K \times K$ symmetric positive matrix (the covariance matrix), which we will call Σ. A positive entry between two traits, say $\Sigma_{12} > 0$, means that when trait 1 increases, trait 2 also tends to increase. Conversely, a negative entry means that the two traits tend to undergo variation in opposite directions. As for the diagonal entries (e.g. Σ_{11}), they represent the variance per unit of time (i.e. the rate of evolution) of each trait considered marginally, thus very much like σ^2 (*not* σ) in the univariate model of the previous section.

On Σ, we will assume an inverse-Wishart prior:

$$\Sigma \quad \sim \quad W^{-1}(\Sigma_0, d),$$

where Σ_0 is a multiple of the identity matrix (i.e. $\Sigma_0 = \kappa I_K$), for some positive real number κ. Using a prior centered on a diagonal matrix means that we want to be indifferent with respect to either positive or negative correlations among traits. As for the parameter κ, it will set the amplitude of the variation per unit of time of the traits. Since we have no idea about the scale of this parameter, we can use a log-uniform prior:

$$\kappa \quad \sim \quad \frac{1}{\kappa}.$$

This completes our model:

$$\begin{aligned} \kappa \quad &\sim \quad \frac{1}{\kappa}, \\ \Sigma \mid \kappa \quad &\sim \quad W^{-1}(\Sigma_0 = \kappa I_K,\, d = K + 2), \\ X(0) \quad &\sim \quad \text{Uniform}, \\ X(t) \mid X(0), \Psi, \Sigma \quad &\sim \quad \text{Brownian}(X(0),\, \psi,\, \Sigma). \end{aligned}$$

As in the univariate case, we can then clamp X at the tips of the phylogeny and sample from the joint distribution over the parameters of the model by MCMC. Once this is done, we can estimate marginal posterior probabilities (e.g. for positive or negative covariance among traits) and infer ancestral traits.

Programming the model in RevBayes

You may find it convenient to program this multivariate model by first duplicating the script of the univariate model:

```
cp placentaliaMass.Rev placentaliaTraits.Rev
```

Then, you can edit the new script, **placentaliaTraits.Rev**, and introduce the modifications that would change the univariate model into its multivariate counterpart.

In the following, only those aspects of the multivariate model that differ from the univariate case are outlined. Essentially, instead of a univariate Brownian motion parameterized by a scalar parameter, you now need to:

define κ:

```
kappa ~ \text{dnLogUniform}(min=0.001,max=1000)
```

define the number of degrees of freedom as $d = K + 2$:

```
df <- nTraits+2
```

define the covariance matrix Σ as inverse Wishart:

```
Sigma ~ dnInvWishart(dim=nTraits, kappa=kappa, df=df)
```

define the multivariate Brownian process:

```
X ~ dnBrownianMultiVariate(psi,Sigma)
```

condition the Brownian model on quantitative trait data. This needs to be done separately for each trait:

```
for (i in 1:nTraits) {
       X.clampAt(contData,i,i)
}
```

Here, we give twice the index **i** to the **clampAt** function: the first corresponds to the entry of the Brownian process, and the second one to the column of the data matrix. In some cases (as we will see below), the Brownian process and the data matrix may not be of same dimension, and therefore, it will be useful to be able to specify arbitrary maps between them.

The model is now entirely specified. We can define the moves on its parameters.

push a scaling move on κ:

```
moves[++mi] = mvScale(kappa, lambda=2.0, tune=true, weight=3.0)
```

a sliding move on the Brownian process

```
moves[++mi] = mvMultivariateRealNodeValTreeSliding(process=X, lambda=10, tune=true,weight=100)
```

a global translation move on the Brownian process (component-wise, that is, a random global translation across the entire phylogeny is applied to one trait taken at random):

```
moves[++mi] = mvMultivariateRealNodeValTreeTranslation(process=X, lambda=1, tune=true, weight
    =1)
```

finally, a conjugate Gibbs move for Σ: as it turns out, conditional on κ and the Brownian process X, it is possible to directly resample Σ from its conditional posterior distribution (Lartillot and Poujol 2011). In `RevBayes`, this is implemented as follows:

```
moves[index] <- mvConjugateInverseWishartBrownian(sigma=Sigma, process=X, kappa=kappa, df=df,
    weight=1)
```

Before creating the model, we need to define a few summary statistics, which we want to track during MCMC, either to monitor convergence or for obtaining interesting outputs. First, suppose you are specifically interested in the covariance and the correlation coefficient associated with the joint variation of body-size (trait 2) and longevity (trait 3). You may also be interested in the *partial* correlation coefficient between body mass and longevity, i.e. while controlling for variation in age at sexual maturity. These three quantities can be singled out and named as follows:

the covariance:

```
cov23 := Sigma.covariance(2,3)
```

the correlation coefficient:

```
cor23 := Sigma.correlation(2,3)
```

the variance per unit of time of, say, log body mass, which is given by the diagonal entry:

```
var2 := Sigma.covariance(2,2)
```

we can also get all correlation coefficients into a single vector (you can skip this part during the session and leave it as homework):

```
# initialize a running index
corrindex = 1
# loop over all pairs of traits
for (i in 1:(nTraits-1)) {
    for (j in (i+1):nTraits) {
        correl[corrindex] := Sigma.correlation(i,j)
        ++corrindex
    }
}
```

we could be interested in tracking several summary statistics also for the Brownian motion, in particular the mean along the tree, separately for each trait:

```
for (i in 1:nTraits)  {
        meanX[i] := X.mean(i)
}
```

After creating the model, all these new variables (**cor12**, **correl**, **meanX**, etc) can be monitored, along with the other parameters of the model:

create the model

```
mymodel <- model(kappa)
```

make a screen monitor that tracks correlation coefficients and mean Brownian values:

```
monitors[1] = mnScreen(printgen=10, correl, meanX)
```

a file monitor that does the same thing, but directly into a file:

```
monitors[2] = mnFile(filename="output/plactraits.trace", printgen=10, separator = " ", correl,
    meanX)
```

a file monitor for Σ:

```
monitors[3] = mnFile(filename="output/plactraits.cov", printgen=10, separator = " ", Sigma)
```

a file monitor for the ancestral reconstruction of traits:

```
monitors[4] = mnFile(filename="output/plactraits.traits", printgen=10, separator = " ", X)
```

and a general model monitor:

```
monitors[5] = mnModel(filename="output/plactraits.log", printgen=10, separator = " ")
```

We can finally create the mcmc and run it:

```
mymcmc = mcmc(mymodel, monitors, moves)
mymcmc.burnin(generations=100,tuningInterval=100)
mymcmc.run(100000)
```

Exercises

- using **`Tracer`**, visualize the posterior distribution on the correlation coefficient between mass and longevity.
- estimate the posterior mean, median and 95% credible interval for this correlation coefficient.
- does the credible interval overlap 0? What does that say about the empirical support for the correlation between body mass and longevity?
- what proportion of the variation in longevity among placental mammals is explained by body mass?

5 Accounting for uncertainty in divergence times

Starting from the model implemented in the last section, we now want to account for phylogenetic uncertainty. As first pointed out by **?**, this can easily be done in a Bayesian framework, through the use of a joint model combining sequence data and quantitative traits. Specifically:

- two data sets are loaded: one for sequence data and one for quantitative traits
- a tree is defined (here, with a uniform prior, but this could be a birth death or anything else)
- a Brownian model is defined over the tree (just as described in the previous section)
- the Brownian model is conditioned on the quantitative trait data
- a substitution model is defined over the same tree
- the substitution model is conditioned on the molecular sequence data.

Instead of remaining fixed to a pre-defined value, the tree should now be moved during the MCMC. Ideally, we would like to move both the toplogy and the divergence times. Mixing over tree topologies under a Brownian model is relatively challenging, however (it works, but it requires rather long MCMC runs). For that reason, in the following, we will mix over divergence times only, under the constraint of a fixed tree topology. The features of the model that would need to be modified in order to also mix over topologies will nevertheless be indicated. You may want try them after the workshop.

5.1 Programming the model in RevBayes

Implementing this joint model in RevBayes is just a matter of adding the following features to the model defined in the previous section (after duplicating the script):

instead of having a fixed tree, we should now define a **random** tree. We could use a birth death prior, whose speciation and extinction rates are themselves endowed with some diffuse exponential prior:

```
speciation ~ dnExp(0.1)
extinction ~ dnExp(0.1)
sampling_fraction := 0.01  # 40 out of the ~ 4000 placental mammals
psi ~ dnBDP(lambda=speciation, mu=extinction, rho=sampling_fraction, rootAge=1.0, nTaxa=nTaxa,
    names=names)
```

we still want to work on a fixed, pre-specified, tree topology (thus, the birth-death prior will be used here only for averaging over uncertainty about divergence times):

```
treeArray <- readTrees("data/chronoplac40.tree")
fixedTree <- treeArray[1]
psi.setValue(fixedTree)
```

create a substitution model, just like what you probably did in previous sessions. In a first step, you can use a simple GTR model, without any rate variation, neither among sites nor among branches.

load the sequence data matrix specified in **data/plac40_4fold.nex** and condition (or clamp) the substitution model to this dataset.

in the moves section, you should add moves for divergence times:

```
moves[++mi] = mvSubtreeScale(psi, weight=5.0)
moves[++mi] = mvNodeTimeSlideUniform(psi, weight=10.0)
```

you would add topology moves here (again, only in a second step):

```
moves[++mi] = mvNNI(psi, weight=5.0)
moves[++mi] = mvFNPR(psi, weight=5.0)
```

finally, you should add moves for the parameters of the substitution model.

Note that, here, we do not have included any fossil information: we are merely doing *relative* dating. We will see at the end of this tutorial how fossil information can be integrated.

Write this model and make sure that it runs when you give it to **RevBayes**. Once this is the case, don't spend too much time analyzing the results and quickly turn to the model introduced in the next section.

6 Autocorrelated relaxed molecular clock

In the previous model, no consideration was given to the problem of rate variation among lineages – we bluntly used a strict clock. This is of course problematic, in particular at the phylogenetic scale considered here (placental mammals), where we know that there is substantial rate variation. In addition, we know that substitution rates across branches are *auto-correlated* in the present case: typically, entire orders, such as rodents, are fast evolving, whereas other orders like Cetartiodactyla are slowly evolving. In other words, nearby lineages along the phylogeny tend to be characterized by similar substitution rates.

You have perhaps already seen an autocorrelated relaxed clock model in the molecular dating session (ACLN). You could easily recruit it in the present context (a good exercise to try after the workshop: modify the model suggested in the previous section so as to replace the strict clock by the ACLN model).

Here, however, we will derive the autocorrelated clock in a slightly different way. This derivation will be less straightforward, but more useful for what we want to do next. Specifically, we will first model the logarithm of the instant substitution rate as a Brownian motion, just like we did for body mass in section 3. Then, we will exponentiate this Brownian process and take branch-specific averages, which we will finally plug into the substitution model as the **branchRates** argument.

6.1 Programming the model in RevBayes

Compared to the model described in the last section, you should:

delete the **clockRate** variable

based on what you have done with body mass in section 3, you should be able to create a univariate Brownian process, which you could call **lograte**

you can then exponentiate and average this Brownian process over branches using **expBranchTree**:

```
branchrates := expBranchTree(tree=psi, process=lograte)
```

plug these rates into the PhyloCTMC object, as the branchRates parameter vector.

condition the model on the sequence and trait data and run the program.

Again, write this model by duplicating and adapting the last script that you have written. Make sure that the model runs when given to RevBayes, before turning to the next model now introduced.

7 Rates, dates and traits

We have just seen that the logarithm of the substitution rate can be seen as a quantitative trait. But then, this raises one further obvious question: why considering the substitution rate and the quantitative traits as separate Brownian motions? Why not instead considering them as a joint multivariate Brownian process? Doing so would have one major advantage: the correlated evolution of rates and traits will be automatically estimated, as a by-product of the model.

To do so, we just need to define a multivariate Brownian process of dimension 4. By convention, we will consider that the first dimension of this process corresponds to the log of the substitution rate, while the other 3 dimensions of the process (2 to 4) will map to the quantitative traits defined by the data matrix (Lartillot and Poujol 2011).

7.1 Programming the model in RevBayes

You now have all the tools to implement this model entirely by yourself, except for one little detail: you now need to exponentiate one specific component of a multivariate process (as opposed to exponentiating a univariate process, as we did in the previous section). Thus, assuming that **X** is your 4-dimensional process, you need to tell the **expBranchTree** function that you want to exponentiate the first component of the process (with the **traitIndex=1** option):

```
branchrates := expBranchTree(tree=psi, process=x, traitIndex=1)
```

Also, be careful with the mapping of the quantitative traits: you need to map trait i to entry $i+1$ of the Brownian process:

```
for (i in 1:nTraits) {
       X.clampAt(contData,i+1,i)
}
```

7.2 Exercises

- write the model and run it on the placental example
- investigate the correlation between substitution rate and life-history traits
- multiple regression: controlling for body mass, do you still get some support for a correlation between longevity and substitution rate variation?
- conversely, controlling for longevity, do you get supported correlations of the substitution rate and body mass (or with age at sexual maturity?)
- compare the credible interval obtained here for the body mass of the last common ancestor of placentals with what was obtained in the very first model (section 3).

8 A comparative analysis of variation in GC content

Apart from the overall substitution rate, any other aspect of the substitution process (transition-transversion ratio, dN/dS, equilibrium frequencies, etc) could in principle display variation among lineages. These various aspects of the substitution process could therefore be modeled exactly like the substitution rate, i.e. as Brownian processes – or as components of a multivariate Brownian process. In this section, we will focus on compositional variation, and more particularly variation in equilibrium GC content between species.

We first start with a simple T92 model of sequence evolution:

$$Q = \begin{pmatrix} \begin{array}{c|cccc} & A & C & G & T \\ \hline A & - & \frac{\gamma}{2} & \kappa\,\frac{\gamma}{2} & \frac{1-\gamma}{2} \\ C & \frac{1-\gamma}{2} & - & \frac{\gamma}{2} & \kappa\,\frac{1-\gamma}{2} \\ G & \kappa\,\frac{1-\gamma}{2} & \frac{\gamma}{2} & - & \frac{1-\gamma}{2} \\ T & \frac{1-\gamma}{2} & \kappa\,\frac{\gamma}{2} & \frac{\gamma}{2} & - \end{array} \end{pmatrix}$$

The model has two parameters: the transition-transversion rate κ and the equilbrium GC content γ. In the following, we will assume that κ is constant across the tree (although unknown, and thus endowed with a diffuse prior). In contrast, γ will be allowed to vary among lineages, jointly with the overall substitution rate. Technically, since γ is strictly between 0 and 1, its log-it transform $\ln \frac{\gamma}{1-\gamma}$ will range over the entire real line. Therefore, we could propose that the log-it transform of γ evolves according to a Brownian motion.

Putting everything together, we will therefore propose a multivariate Brownian motion $X(t)$, of dimension $K+2$, where K is the number of quantitative traits, and such that:

$$\begin{aligned} X_1(t) &= \ln r(t) \\ X_2(t) &= \ln \frac{\Gamma(t)}{1-\Gamma(t)} \\ k = 1..K, \quad X_{k+2}(t) &= \ln C_k(t) \end{aligned}$$

where $r(t)$ is the instant substitution rate and $\gamma(t)$ the instant equilibrium GC composition and $C_k(t)$ is the kth. quantitative trait. Equivalently, we may re-write this as follows:

$$\begin{aligned} r(t) &= e^{X_1(t)} \\ \gamma(t) &= \frac{e^{X_2(t)}}{1+e^{X_2(t)}} \\ \ldots \end{aligned}$$

In other words, the instant rate of substitution $r(t)$ is the exponential of the first component $X_1(t)$ of the Brownian process (as above), while the instant equilibrium GC $\gamma(t)$ is the *hyperbolic tangent* of the second component $X_2(t)$ of the Brownian process.

There is a slight complication here: in a non-homogeneous model, independently of the rate matrices across branches, we also need to specify the nucleotide frequencies from which the sequence at the root of the tree is sampled. We will call this frequency vector π, and we will put a Dirichlet prior on π.

This model has been described in Lartillot (2013).

Programming the model in RevBayes

Assuming that **X** is the multivariate Brownian process:

as above, define the branch rates as the exponential of the first component:

```
branchrates := expBranchTree(tree=psi, process=X, traitIndex=1)
```

define the branch equilibrium GC as the hyperbolic tangent of the second component:

```
branchGC := tanhBranchTree(tree=psi, process=X, traitIndex=2)
```

for $k = 1..K$, map trait k onto entry $k + 2$ of X:

```
for (k in 1:nTraits) {
       X.clampAt(contData,k+2,k)
}
```

define the transition-transversion ratio; usually, this ratio is of the order of 1-10, so we will use an exponential prior of mean 10:

```
tstv ~ dnExp(0.1)
```

define a vector of branch-specific T92 substitution matrices:

```
branchMatrices := t92GCBranchTree(tree=psi,branchGC=branchGC,tstv=tstv)
```

create a Dirichlet-distributed vector of equilibrium frequencies over nucleotides:

```
bf <- v(1,1,1,1)
pi ~ dnDirichlet(bf)
```

finally, create the substitution model:

```
seq ~ dnPhyloCTMC(tree=psi, Q=branchMatrices, rootFrequencies=pi, branchRates=branchrates,
    nSites=nSites, type="DNA")
```

Exercises

- program the model in `RevBayes`
- run the model on the placental dataset
- investigate the correlation between GC and body mass

- how do you explain these correlations?
- run the model on the archaeal rRNA dataset `archaea.nex`, using temperature as the trait. In that case, no phylogeny is provided, so you may try to run the model without any constraint on the topology.
- assess the correlation between GC and temperature
- how much of the variation in GC is explained by temperature?
- what could be the underlying biological cause?

References

De Magalhaes, J. and J. Costa. 2009. A database of vertebrate longevity records and their relation to other life-history traits. Journal of evolutionary biology 22:1770–1774.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: A maximum likelihood approach. Journal of Molecular Evolution 17:368–376.

Felsenstein, J. 1985a. Confidence limits on phylogenies: an approach using the bootstrap. Evolution 39:783–791.

Felsenstein, J. 1985b. Phylogenies and the comparative method. American Naturalist Pages 1–15.

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

Huelsenbeck, J. P. and B. Rannala. 2003. Detecting correlation between characters in a comparative analysis with uncertain phylogeny. Evolution 57:1237–1247.

Lartillot, N. 2013. Interaction between selection and biased gene conversion in mammalian protein-coding sequence evolution revealed by a phylogenetic covariance analysis. Molecular Biology and Evolution 30:356–368.

Lartillot, N. and R. Poujol. 2011. A phylogenetic model for investigating correlated evolution of substitution rates and continuous phenotypic characters. Molecular Biology and Evolution 28:729–744.

Version dated: July 10, 2016