Phylogenetic Inference using `RevBayes`

Species tree estimation from multiple genes using concatenation

Bastien Boussau and Sebastian Höhna

1 Overview: Gene tree-species tree models

Ever since **?**, researchers have acknowledged that phylogenies reconstructed from homologous gene sequences could differ from species phylogenies. As molecular sequences accumulated, the link between gene trees and species trees started to be modeled. The first models were based on parsimony, and aimed for instance at reconciling a gene tree with a species tree by minimizing the number of events of gene duplication and gene loss. In the past dozen years, probabilistic models have been proposed to reconstruct gene trees and species trees in a rigorous statistical framework. Models and algorithms have quickly grown in complexity, to model biological processes with increasing realism, to accommodate several processes at the same time, or to handle genome-scale data sets. In this overview we will not detail these models, and we invite the interested reader to take a look at recent reviews (*e.g.,* (**?**)).

1.1 Processes of discord

There are several reasons why a gene tree may differ from a species tree. Of course, a gene tree may differ from the species tree just because a mistake was made during the analysis of the gene sequences, at any point in a pipeline going from the sequencing itself to the gene tree reconstruction. Such a mistake would produce an incorrect gene tree. Here we do not mean this kind of discord, but rather discord that comes from a real biological process that generates true gene histories that differ from true species histories. These processes include gene duplication, gene loss, gene transfer (used loosely here to also include reticulation, hybridization between species), and incomplete lineage sorting (Fig. 1). In this tutorial we focus on Incomplete lineage sorting, which will be discussed in more details in the following subsection.

Fig. 1 suggests that for all processes the gene tree can be seen as the product of a branching process operating inside the species tree. Indeed, all processes are modeled as some type of birth-death process running along the species tree. For duplication/loss models, birth correspond to gene duplication events, and death to gene loss events. Transfers can be added to the model by introducing another type of birth, with a child lineage appearing in another branch of the species tree. Incomplete lineage sorting is also modeled with a birth-death type of model, the coalescent. All these models can be made heterogeneous, for instance by allowing different sets of parameters for different branches of the species tree. This is useful to model differences in rates of duplication, loss or transfer among species, or to model different effective population sizes in a species tree. In `RevBayes` so far only models of incomplete lineage sorting have been implemented (models of duplication and loss and transfer will soon be added). Thanks to `RevBayes` modular design, there is quite a lot of flexibility in specifying the model, for instance by associating different parameters to different branches of the species tree, or by combining the gene tree-species tree model to other types of models, for instance models of trait evolution, or models of relaxed molecular clock.

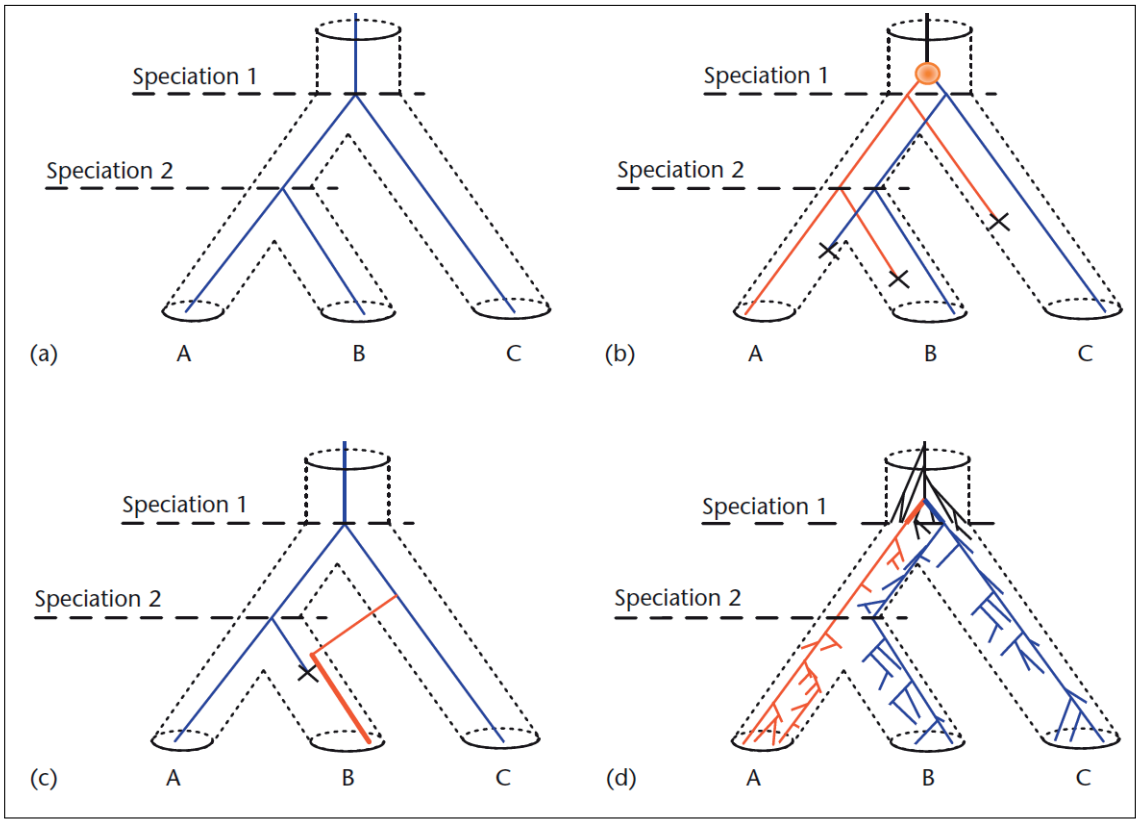

Figure 1: The processes of discord. The species tree is represented as a tubular structure. Gene trees are blue and red lines running along the species trees. a) A gene tree that perfectly matches the species tree. b) The gene tree and the species tree differ because of gene duplications and losses. c) The gene tree and the species tree differ because of gene transfer and gene loss. d) The gene tree and the species tree differ because of incomplete lineage sorting. [Replicated from Fig. 2 in Boussau et al. (2009).]

1.2 Gene tree discordance is a problem for species tree reconstruction

There have been several approaches to species tree reconstruction: concatenation and supertree approaches, which have been used for quite some time now, and more recently methods that rely on gene tree-species tree models.

1. Concatenation simply consists in taking all gene alignments, concatenating them into one super alignment, and then analyzing it as if it were a single gene sequence. More sophisticated approaches allow different partitions for different genes, but the main assumption at the heart of this approach is that all sites of all genes have evolved according to the same species tree. This assumption is often not correct because all the processes of discord presented above conspire to make gene trees different from the species tree. In practice, this matters: simulation studies have found that in the presence of incomplete lineage sorting, in some particular areas of the parameter space, concatenation will often return an incorrect species tree (Leaché and Rannala 2011). Concatenation may also be a questionable approach in prokaryotic phylogenetics, where the quest for a tree of life has been difficult, to the point that some doubted that one could find a meaningful species tree representing vertical descent. Nonetheless, the concatenation approach may be fairly robust to lateral gene transfers, as it returns good species trees (arguably better than small subunit or large subunit rRNA trees) in a range of prokaryotic groups (**?**).

2. Supertree approaches differ from concatenation notably by discarding sequence information once individual gene trees have been built. Contrary to concatenation approaches that combine individual gene alignments, supertree approaches combine individual gene trees to obtain a species tree. Most supertree methods are not based on an explicit model of the processes causing discordance between gene trees and species tree (although there are exceptions, notably modelling incomplete lineage sorting, see below). Instead, they aim at finding a tree that would best describe the distribution of gene trees, according to some fairly arbitrary criterion. In practice, these methods have been found to provide reasonable results in many cases, but in simulations they are usually less accurate than concatenation.

3. Methods that rely on gene tree-species tree models appear very promising as they explicitly model the processes of discord. The advantage of these models is that we account for processes that we know have taken a part in generating the data, thus possibly improving the accuracy and robustness of our inferences. Further, these models can be combined with *e.g.,* models of sequence evolution, models of co-evolution between gene trees, or models of trait evolution. However, these models are computationally challenging to use, because they require estimating jointly gene trees, species trees, and other parameters that entertain strong correlations. As a consequence, in many gene tree-species tree models, devising a well-mixing MCMC strategy can be problematic.

2 Concatenating genes to model species evolution

The simplest model to estimate a species tree when you have several genes is to concatenate your genes and to assume that all gene trees are exactly equal to the species tree. You thus assume that there is no discordance between gene trees in the hope that the combined information in all genes will provide a reliable signal for the species tree.

The exercises assume you have a working installation of RevBayes. In this introductory tutorial, we will apply the concatenated model to 10 gene alignments from 23 primate species. We will assume that:

- The species tree is drawn from a constant birth-death process.
- All genes share the same tree, that is, both the topology and branch lengths.
- Each gene has its own set of substitution model and clock parameters which are drawn from a shared prior distribution.
- Gene sequences are evolved according to an HKY model with gamma distributed rate variation among sites and a strict global clock.
- Here, we run an MCMC on this model, using data from 10 genes in 23 mammalian species.

Scripts are all placed in *tutorials/RB_GeneConcatenation_Tutorial/RevBayes_scripts/*.

1. Open RevBayes

2. Let's load all 10 gene alignments.

```
# read in each data matrix together which will create a vector of objects
data = readDiscreteCharacterData("data/merged.nex")

# Now we get some useful variables from the data. We need these later on.
num_loci = data.size()
# get the number of species
n_species <- data[1].ntaxa()
# get the taxon information (e.g. the taxon names)
taxa <- data[1].taxa()
n_branches <- 2 * n_species - 1 # number of branches in a rooted tree

# We set our move index
mi = 0
```

3. We specified a constant-rate birth-death process as our prior on the species tree. The birth-death process has a speciation and extinction rate as its parameters. We will use here a transformation and specify priors on the speciation rate and relative extinction rate. Additionally, we calibrate the tree by assuming that the crown age of primates is around 75 MYA. Thus, we specify a normal distribution with mean 75 and standard deviation 2.5 as the prior on the root age. Since the root age can only be a positive real number we truncate the normal distribution at 0.

```
# Specify a prior on the diversification and turnover rate
speciation ~ dnGamma(2,2)
relativeExtinction ~ dnBeta(1,1)

# Now transform the diversification and turnover rates into speciation and
    extinction rates
extinction := speciation * relativeExtinction

# Specify a prior on the root age (our informed guess is about ~75 mya)
# Note that we use a truncated normal distribution because the root age must be
    positive
root ~ dnNormal(mean=75,sd=2.5,min=0.0, max=Inf)

sampling_fraction <- 23 / 450 # we sampled 23 out of the ~ 450 primate species

# create some moves that change the stochastic variables
# Moves are sliding and scaling proposals
moves[++mvi] = mvSlide(diversification,delta=1,tune=true,weight=2)
moves[++mvi] = mvSlide(relativeExtinction,delta=1,tune=true,weight=2)
moves[++mvi] = mvScale(diversification,lambda=1,tune=true,weight=2)
moves[++mvi] = mvScale(relativeExtinction,lambda=1,tune=true,weight=2)
```

```
moves[++mvi] = mvSlide(root,delta=1,tune=true,weight=0.2)


# construct a variable for the tree drawn from a birth-death process
psi ~ dnBDP(lambda=speciation, mu=extinction, rootAge=root, rho=sampling_fraction,
     taxa=taxa )

moves[++mvi] = mvNarrow(psi, weight=5.0)
moves[++mvi] = mvNNI(psi, weight=1.0)
moves[++mvi] = mvFNPR(psi, weight=3.0)
moves[++mvi] = mvGPR(psi, weight=3.0)
moves[++mvi] = mvSubtreeScale(psi, weight=3.0)
moves[++mvi] = mvNodeTimeSlideUniform(psi, weight=15.0)
```

4. Now that we have a species tree, which we assume is shared exactly for all genes. That is, we assume each gene evolves under exactly the same tree, and thus each gene tree is equivalent to the species tree. Nevertheless, we assume that each gene evolves at a different rate and with its own substitution model parameters. Here we will assume for simplicity that every gene evolves under a global strict clock but has its own independent clock rate. We assume that the logarithm of the clock rate is uniformly distribution, thus we specify in effect a log-uniform prior distribution. This prior assumption means that we put the same prior probability on values of each magnitude, e.g., values between 0.0001 and 0.001 have the same prior probability as values between 10 and 100.

```
for ( i in 1:num_loci ) {
   log_clock_rate[i] ~ dnUniform(-8,4)
   clock_rate[i] := 10^log_clock_rate[i]

   moves[++mvi] = mvSlide(log_clock_rate[i], weight=1.0)
}
```

5. Next we need our model for the substitution process. Hence, we just need to define the substitution matrix. We use a single HKY matrix that will apply to all sites per gene. Additionally, we assume that sites evolve according to one of four possible rates, where each rate corresponds to a quantile from a gamma distribution.

```
for ( i in 1:num_loci ) {

    #### specify the HKY substitution model applied uniformly to all sites of a
       gene
    kappa[i] ~ dnLognormal(0,1)
    moves[++mvi] = mvScale(kappa[i],weight=1)

    pi_prior[i] <- v(1,1,1,1)
    pi[i] ~ dnDirichlet(pi_prior[i])
    moves[++mvi] = mvSimplexElementScale(pi[i],weight=2)
```

```
    #### create a deterministic variable for the rate matrix
    Q[i] := fnHKY(kappa[i],pi[i])

    #### create the rates to model the gamma distributed rate variation among
        sites.
    alpha_prior[i] <- 0.05
    alpha[i] ~ dnExponential( alpha_prior[i] )
    gamma_rates[i] := fnDiscretizeGamma( alpha[i], alpha[i], 4, false )

    # add moves for the stationary frequencies, exchangeability rates and the
        shape parameter
    moves[++mvi] = mvScale(alpha[i],weight=2)

}
```

6. Finally, we can create our distribution for the character evolution. We will use the common **PhyloCTMC** distribution, which is a continuous time Markov process along a phylogenetic tree. We create a **seq** variable and attach/clamp each gene to one of the **seq** variables.

```
for ( i in 1:num_loci ) {
    # the sequence evolution model
    seq[i] ~ dnPhyloCTMC(tree=psi, Q=Q[i], branchRates=clock_rate[i], siteRates=
        gamma_rates[i], type="DNA")

    # attach the data
    seq[i].clamp(data[i])
}
```

7. Now we have defined all the bricks of the model, and create our model object from it.

```
# We get a handle on our model.
# We can use any node of our model as a handle, here we choose to use the topology
    .
mymodel = model(psi)
```

8. Finally, we need to perform inference under the model, using the data.

```
# Monitors to check the progression of the program
monitors[1] = mnScreen(printgen=100, root)
monitors[2] = mnModel(filename="output/primates_concatenation_root_calibration",
    printgen=10, separator = TAB)
```

```
monitors[3] = mnFile(filename="output/primates_concatenation_root_calibration",
    printgen=10, separator = TAB, psi)

# Here we use a plain MCMC. You could also set nruns=2 for a replicated analysis
# or use mcmcmc with heated chains.
mymcmc = mcmc(mymodel, monitors, moves)

# This should be sufficient to obtain enough MCMC samples
mymcmc.burnin(generations=3000,tuningInterval=100)
mymcmc.run(generations=10000)
```

9. Now we can perform some post-run analyses.

```
# Now, we will analyze the tree output.
# Let us start by reading in the tree trace
treetrace = readTreeTrace("output/primates_concatenation_root_calibration",
    treetype="clock")
# and get the summary of the tree trace
treetrace.summarize()

mapTree(treetrace,"output/primates_concatenation_root_calibration")
```

References

Boussau, B., L. Guéguen, and M. Gouy. 2009. A mixture model and a hidden markov model to simultaneously detect recombination breakpoints and reconstruct phylogenies. Evolutionary bioinformatics online 5:67.

Leaché, A. D. and B. Rannala. 2011. The accuracy of species tree estimation under simulation: a comparison of methods. Systematic Biology 60:126–137.

Version dated: July 10, 2016