Using mbd

Giovanni Laudanno and Richel J.C. Bilderbeek

2018-10-02

1. Introduction

`mbd` is a package to support multiple birth model. This model accounts for a diversification process in which multiple birth (at different nodes) are allowed to occur at the same time.

The basic framework is described in Etienne et al. 2012[1]. The main purpose of the package is to provide a tool to estimate, given a phylogenetic tree, the likelihood for a set of three parameters: λ,μ and q. Here:

- λ is the sympatric speciation rate;
- μ is the extinction rate;
- ν is the multiple allopatric speciation trigger rate;
- q is the single-lineage speciation probability.

Those parameters may be inferred through a maximum likelihood approach using the function mbd_ml.

2. Setup

We will need to load the package:

```
library(mbd)
```

Also, we'll set the random number generator seed to a value, so that this vignette always produces the same results:

```
set.seed(14)
```

2. Simulating an MBD tree

First we set the parameters of the MBD speciation model:

```
lambda <- 0.2 # sympatric speciation rate
mu <- 0.15 # extinction rate;
nu <- 2.0 # multiple allopatric speciation trigger rate
q <- 0.1 # single-lineage speciation probability
```

3.1. Classic interface

Then we set some basic parameters to create a simulated tree:

```
crown_age <- 1
sim <- mbd_sim(
  pars = c(lambda, mu, nu, q),
  soc = 2, # Use a crown age
```

[1] Etienne, R.S., Haegeman, B., Stadler, T., Aze, T., Pearson, P.N., Purvis, A., Phillimore, A.B.: Diversity-dependence brings molecular phylogenies closer to agreement with the fossil record. Proceedings of the Royal Society of London B: Biological Sciences 279(1732), 1300–1309 (2012)

```r
  age = crown_age,
  cond = 1 # Condition on non-extinction
)
```

These are additional auxilary parameters such as `soc` (equal to 1 if we want a stem, or 2 if we want a crown), `cond` (let us condition the tree on the survival of crown species) and `crown_age` (being the total age of the tree).

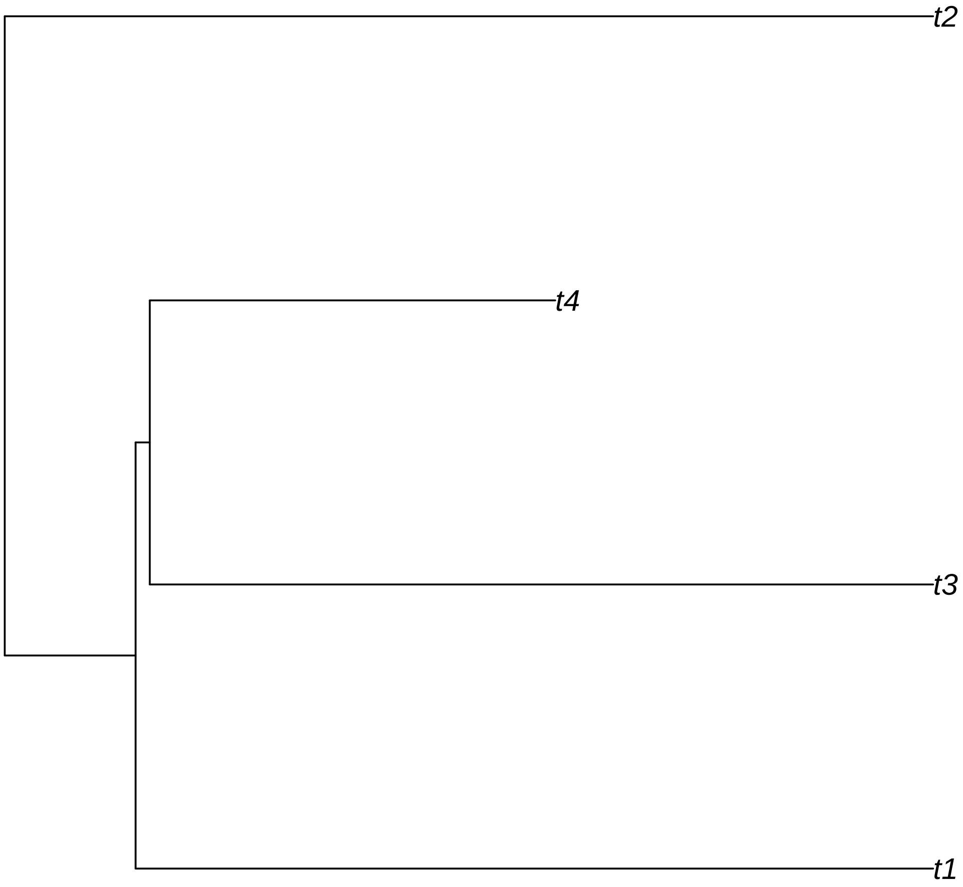
3.2. Checked interface

```r
sim <- mbd_sim_checked(
  mbd_params = create_mbd_params(lambda = lambda, mu = mu, nu = nu, q = q),
  crown_age = crown_age,
  conditioned_on = "non_extinction"
)
```

4. Showing the results

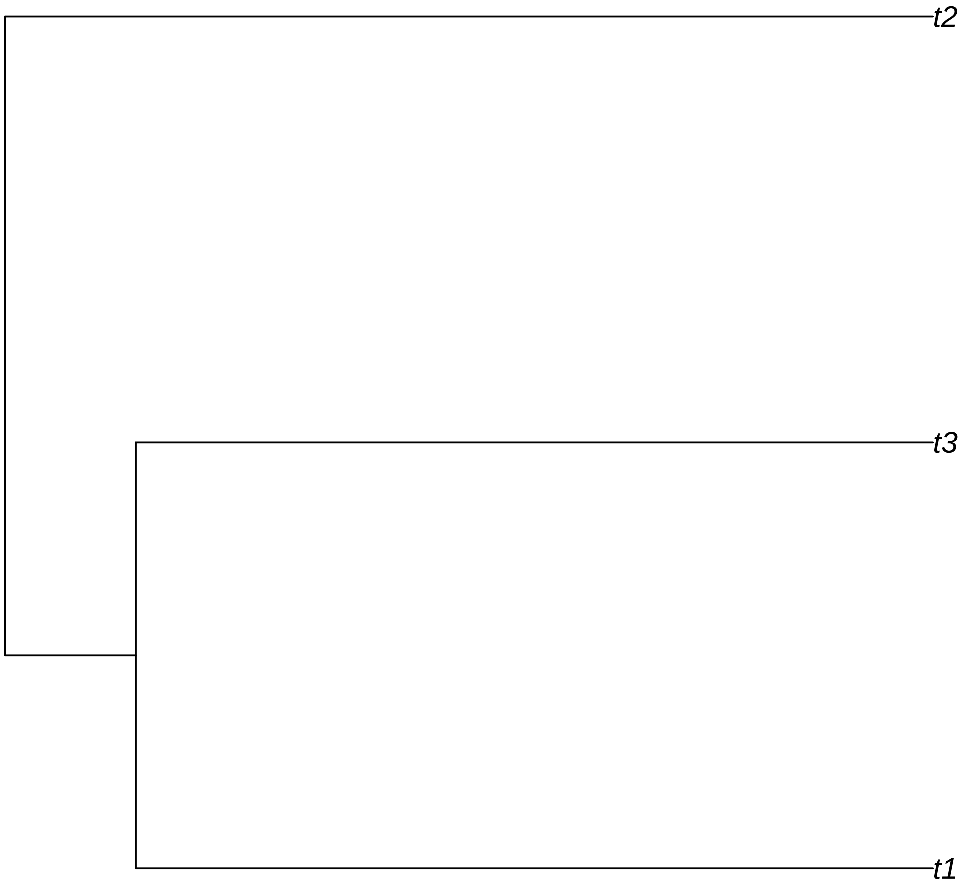
The `sim` list contain different objects. You can visualize the full tree and keep track of species that die before present time (missing species):

```r
graphics::plot(sim$tas)
```



If you want you can also show the reconstructed tree, obtained cutting the kinks from the previous tree:

```
graphics::plot(sim$tes)
```



The dataframe containing all the information related to this tree is labeled as L (or L-table). You can access it by typing:

```
knitr::kable(head(sim$L))
```

In this table:

- the first column are the times at which species are born;
- the second contains labels of the species' parents; positive and negative values only indicate whether the species belongs to the left or right crown lineage;
- the third contains the labels of the daughter species themselves; positive and negative values only indicate whether the species belongs to the left or right crown lineage;
- the fourth contains the extinction times of the species. If this is equal to -1, then the species is still extant.

To recover the vector of branching times use:

```
knitr::kable(head(sim$brts))
```

x
-1.0000000
-0.8590596

In this vector values repeated more than once indicate that a multiple birth process has taken place.

5. Likelihood estimation

To estimate the likelihood of a phylogeny, one needs

- parameter estimates
- the branching times (topology is irrelevant)

In this case, we'll use

- the parameter estimates that we used to generate the tree
- the branching times of the tree we generated

The function `mbd_loglik` shows the likelihood of the parameters having generated the branching times:

```
mbd::mbd_loglik(
  pars = c(lambda, mu, nu, q),
  brts = sim$brts,
  soc = 2, # Crown age
  cond = 1  # Non-extinction
)
```

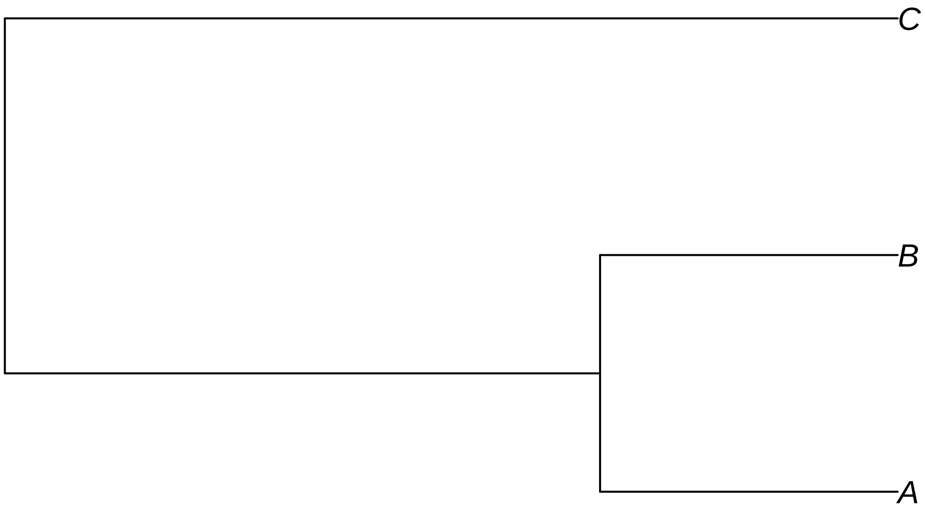
```
## [1] 2.291128
```

6. Maximum likelihood estimation

You can also ask for an optimization of the likelihood function using the function `mbd_ml` either considering a variation of all the parameters (λ, μ, ν, q) or any their subset of your choice.

Because this is a heavy calculation, we'll use a simpler tree:

```
phylogeny <- ape::read.tree(text = "((A:1, B:1):2, C:3);")
ape::plot.phylo(phylogeny)
```



```
brts <- ape::branching.times(phylogeny)
```

```
lambda <- 0.3 # sympatric speciation rate
mu <- 0.1 # extinction rate
nu <- 0.11 # multiple allopatric speciation trigger rate
q <- 0.15 # single-lineage speciation probability
```

6.1 Classic interface

If you want for example to maximize the likelihood only for the parameter q:

```
idparsopt <- 4 # Only optimize the fourth parameter, q
idparsfix <- c(1, 2, 3) # Fix all parameters except q
parsfix <- c(lambda, mu, nu) # Use the known values for the fixed parameters
initparsopt <- q # Set an initial guess for q
out <- mbd_ml(
  brts = brts,
  initparsopt = initparsopt,
  idparsopt = idparsopt,
  parsfix = parsfix,
  idparsfix = idparsfix,
  soc = 2, # Crown age
  cond = 1 # Conditioned on non-exitnction
)
```

```
## You are optimizing q
## You are fixing lambda mu nu
## Optimizing the likelihood - this may take a while.
## The loglikelihood for the initial parameter values is -1.652305
##
##  Maximum likelihood parameter estimates: lambda: 0.3 mu: 0.1 nu: 0.11 q: 0.00100021296751783
##  Maximum loglikelihood: -1.598731
```

```
knitr::kable(out)
```

lambda	mu	nu	q	loglik	df	conv
0.3	0.1	0.11	0.0010002	-1.598731	1	0

6.2. More intuitive interface

If you want for example to maximize the likelihood only for the parameter q, using a more type-safe interface, call `mbd_calc_max_lik`:

```
out <- mbd_calc_max_lik(
  branching_times = brts,
  init_param_values = create_mbd_params(
    lambda = lambda, mu = mu, nu = nu, q = q
  ),
  fixed_params = create_mbd_params_selector(
    lambda = TRUE, mu = TRUE, nu = TRUE
  ),
  estimated_params = create_mbd_params_selector(
    q = TRUE
  ),
  init_n_species = 2,
  conditioned_on = "non_extinction"
)
```

```
## 1 0.15 -1.65230526698096 initial
## 2 0.135287846481876 -1.6471619350163 expand
## 3 0.106964656964657 -1.63716628437282 expand
## 4 0.0543564356435637 -1.61828093270909 expand
## 5 0.00652173913043386 -1.60077089117342 reflect
## 6 0.00652173913043386 -1.60077089117342 contract inside
## 7 0.00652173913043386 -1.60077089117342 contract inside
## 8 0.00652173913043386 -1.60077089117342 contract inside
## 9 0.00652173913043386 -1.60077089117342 contract inside
## 10 0.00367577756833083 -1.59971968990733 reflect
## 11 0.00367577756833083 -1.59971968990733 contract inside
## 12 0.00225882352941082 -1.59919593953927 reflect
## 13 0.00225882352941082 -1.59919593953927 contract inside
## 14 0.00155184575593604 -1.59893452559888 reflect
## 15 0.00155184575593604 -1.59893452559888 contract inside
## 16 0.00119873075566952 -1.59880393377478 reflect
## 17 0.00119873075566952 -1.59880393377478 contract inside
## 18 0.0010222666118315 -1.59873866662868 reflect
## 19 0.0010222666118315 -1.59873866662868 contract inside
## 20 0.0010222666118315 -1.59873866662868 contract inside
## 21 0.0010222666118315 -1.59873866662868 contract inside
## 22 0.00100021296751783 -1.59873050958342 reflect
## 23 0.00100021296751783 -1.59873050958342 contract inside
## 24 0.00100021296751783 -1.59873050958342 contract inside
## 25 0.00100021296751783 -1.59873050958342 contract inside
## 26 0.00100021296751783 -1.59873050958342 contract inside
## 27 0.00100021296751783 -1.59873050958342 contract inside
## Optimization has terminated successfully.
```

```r
knitr::kable(out)
```

lambda	mu	nu	q	loglik	df	conv
0.3	0.1	0.11	0.0010002	-1.598731	1	0