

Generating outputs from TreeDucken for approximate Bayesian computation - an example implementation

We use the TreeDucken package for simulating co-phylogenies. The outputs of this model includes two trees, one for the host and other for the symbiont, and the association between the tips of the two trees in the format of a matrix.

We first visualize the co-phylogeny with the TreeDucken package itself (the last line is commented out to omit the plot).

```
# For R/4.4.0}  
  
# install.packages("htmlwidgets")  
# install.packages("devtools")  
library(devtools)
```

Loading required package: usethis

```
# install_version("apTreeshape", version = "1.5-0.1")  
library(apTreeshape)
```

Loading required package: ape

```
Registered S3 method overwritten by 'apTreeshape':  
  method      from  
  is.binary.phylo ape
```

Attaching package: 'apTreeshape'

The following object is masked from 'package:ape':

is.binary.phylo

```
# install_version("treeduck", version = "1.1.0")
library(treeduck)
library(phytools)
```

Loading required package: maps

```
set.seed(54)

# host speciation
lambda_H <- rexp(n=1)

# host extinction rate
mu_H <- 0.3

# cospeciation rate
lambda_C <- rexp(n=1)

# time
time <- 2

# symbiont speciation rate
lambda_S <- rexp(n=1)

# symbiont extinction rate
mu_S <- 0.3

lambda_total_H <- lambda_H + lambda_C
lambda_total_S <- lambda_S + lambda_C

# calculate the expected number of tips for the host and symbiont trees
H_tips <- ave_tips_st(lambda = lambda_total_H, mu = mu_H, t = time)
S_tips <- ave_tips_st(lambda = lambda_total_S, mu = mu_S, t = time)

cophy_obj <- sim_cophyBD(hbr = lambda_H,
                        hdr = mu_H,
                        sbr = lambda_S,
                        sdr = mu_S,
```

```

                                cosp_rate =lambda_C,
                                host_exp_rate = 0.0,
                                time_to_sim = time,
                                numbsim = 1)

# visualization using treeDucken, with imperfections
# plot.cophy(cophy_obj[[1]], col = "orange", lty = "dotted", lwd = 2)

```

Alternatively, the co-phylogeny can be visualize using phytools (the last line is commented out to omit the plot).

```

# visualization using ape or phytools, transformation of the association matrix needed
ass_mat_trans <- NULL
ass_mat <- cophy_obj[[1]][[3]]
for(i in 1:nrow(ass_mat)){
  for (j in 1:ncol(ass_mat)){
    if (ass_mat[i,j] == 1)

        ass_mat_trans <- rbind(ass_mat_trans, c(rownames(ass_mat)[i], colnames(ass_mat)[j]))

  }
}

# using ape / phytools
cophylo_plot <- cophylo(cophy_obj[[1]][[1]], cophy_obj[[1]][[2]], assoc = ass_mat_trans)

```

Rotating nodes to optimize matching...
Done.

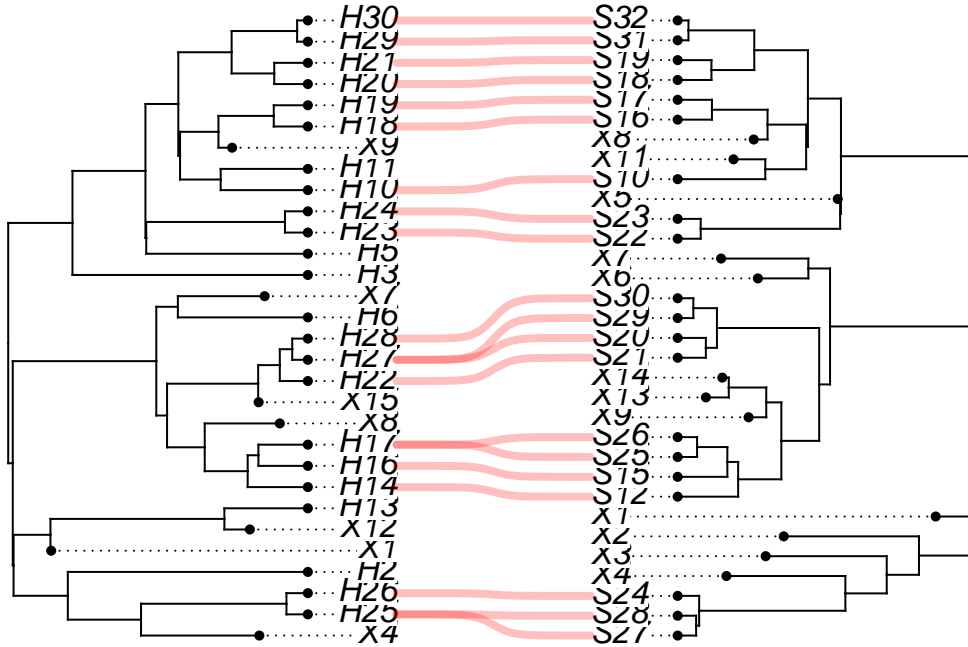
```
#plot(cophylo_plot)
```

However, it turns out base ape does a good job plotting co-phylogenies as well, which is shown here.

```

plot(cophylo_plot, link.type="curved", link.lwd=4,
     link.lty="solid", link.col=make.transparent("red", 0.25))

```



To use the model for ABC, we slightly modify the way the model is parameterized. Instead of directly specifying the values of host and symbiont extinction rates, we define each of these two extinction rates as a fraction (between 0 and 1) of the corresponding total speciation rate. We do this because TreeDucken returns an error when host extinction rate (μ_H) exceeds host total speciation rate ($\lambda_H + \lambda_C$). In contrast, it does not return an error when symbiont extinction rate (μ_S) exceeds symbiont total speciation rate ($\lambda_S + \lambda_C$), but we avoid this anyway since a symbiont clade undergoing faster extinction than speciation is less likely to persist to today.

Before constructing different models for model selection, it is good to have an intuitive understanding of the additive effects of different processes, what processes are optional or required for TreeDucken to return outputs. So here are some tests.

We start with the simplest scenario where there is only co-speciation and no other processes involved - this should result in perfect congruence between the two phylogenies.

```
cophy_obj <- NULL

# time
time <- 2

# host speciation rate
lambda_H <- 0#rexp(n=1)
```

```

# symbiont speciation rate
lambda_S <- 0#rexp(n=1)

# cospeciation rate
lambda_C <- rexp(n=1)

# host expansion rate
exp_H <- 0#rexp(n=1)

# host turnover and extinction rates
mu_H_frac <- 0#runif(1, min = 0, max = 1)
mu_H <- mu_H_frac * (lambda_H + lambda_C)

# symbiont turnover and extinction rates
mu_S_frac <- 0#runif(1, min = 0, max = 1)
mu_S <- mu_S_frac * (lambda_S + lambda_C)

lambda_total_H <- lambda_H + lambda_C
lambda_total_S <- lambda_S + lambda_C

# calculate the expected number of tips for the host and symbiont trees
H_tips <- ave_tips_st(lambda = lambda_total_H, mu = mu_H, t = time)
S_tips <- ave_tips_st(lambda = lambda_total_S, mu = mu_S, t = time)

cophy_obj <- sim_cophyBD(hbr = lambda_H,
                        hdr = mu_H,
                        sbr = lambda_S,
                        sdr = mu_S,
                        cosp_rate = lambda_C,
                        host_exp_rate = exp_H,
                        time_to_sim = time,
                        numbsim = 1)

# visualization using ape, transformation of the association matrix needed
ass_mat_trans <- NULL
ass_mat <- cophy_obj[[1]][[3]]
for(i in 1:nrow(ass_mat)){
  for (j in 1:ncol(ass_mat)){
    if (ass_mat[i,j] == 1)

      ass_mat_trans <- rbind(ass_mat_trans, c(rownames(ass_mat)[i], colnames(ass_mat)[j]))
  }
}

```

```

}
}

cophylo_plot <- cophylo(cophy_obj[[1]][[1]], cophy_obj[[1]][[2]], assoc = ass_mat_trans)

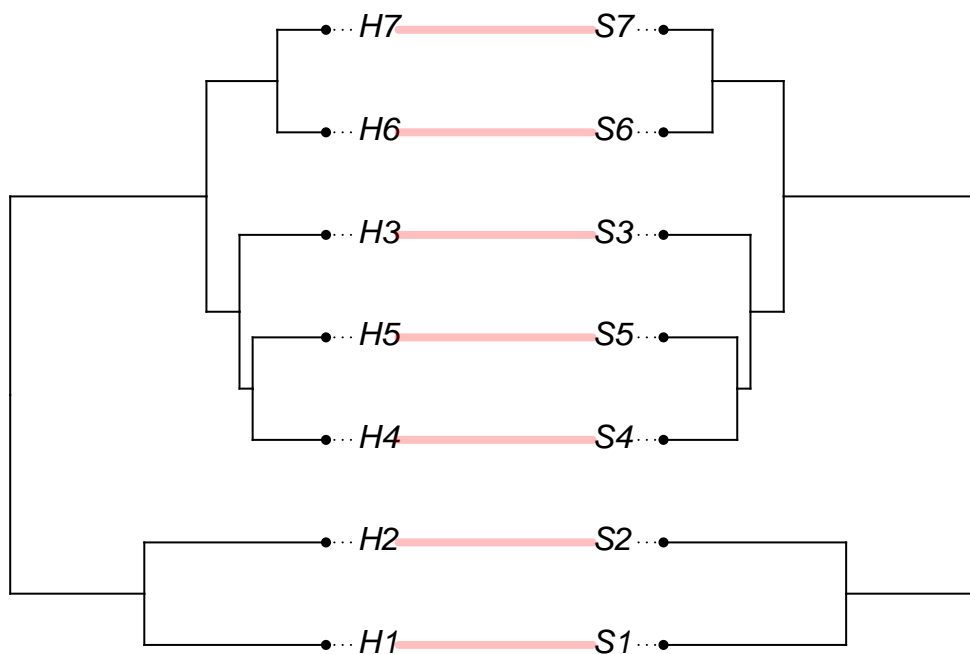
```

Rotating nodes to optimize matching...
Done.

```

#plot(cophylo_plot)
plot(cophylo_plot, link.type="curved", link.lwd=4,
      link.lty="solid", link.col=make.transparent("red", 0.25))

```



Simulate multiple co-phylogenetic systems.

Quantify the summary statistics for the co-phylogenetic systems and then visualize the multi-dimensional joint distributions of (a) the parameters used and (b) the summary statistics.

```

# Load necessary libraries
library(GGally)

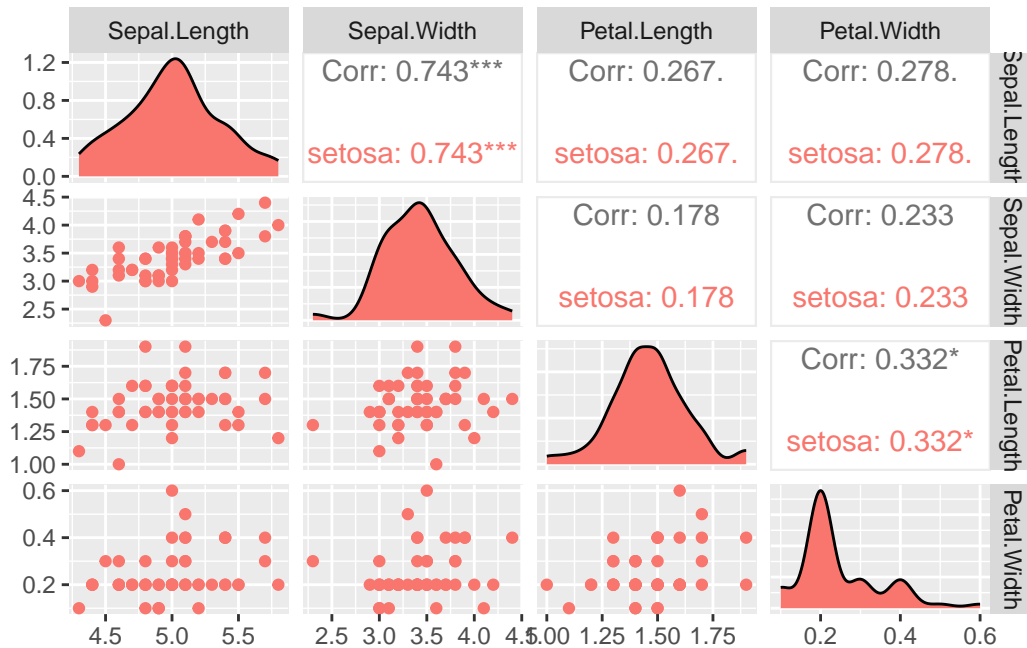
```

Loading required package: ggplot2

Registered S3 method overwritten by 'GGally':
 method from
 +.gg ggplot2

```
library(ggplot2)

# Create a pair plot
ggpairs(iris[iris$Species=='setosa',], columns = 1:4, aes(color = Species))
```



Here it may be necessary to do a sanity check to make sure the simulated co-phylogenies are