Bioinformatics - Computer Lab 3

Group 7: Lennart Schilling (lensc874), Thijs Quast (thiqu264), Mariano Maquieira Mariani (marma330)

06-12-2018

Question 1

Answer Using the script http://ape-package.ird.fr/APER/APER2/SylviaWarblers.R we obtained the Sylvia warblers phylogeny: the file sylvia_nj_k80.tre is in this folder.

```
library(ape)
```

Question 1.1

Explain all the steps in the script required to obtain the phylogeny and trait data. We willget the sequence data from GenBank and align them, and read the ecological data from a file to obtain the phylogeny:

1- We create a vector starting with Z73494, and then the values AJ5345, appending 26,27,28,etc...

```
x <- paste("AJ5345", 26:49, sep = "")
x <- c("Z73494", x)
```

We connect to the GenBank database, and get sequences using accession numbers given as arguments (As a result we get an object of type DNAbin. (the object will have 25 dna sequences stored)

```
sylvia.seq <- read.GenBank(x)
```

We have a list with 25 sequences. 23 of them have 1143 nucleotides, and 2 have 1041. (It needs an alignment operation which is done with Clustal).

3- We use 'clustal' which aligns a set of nucleotide sequences. (We also aligned using mafft and the result was identical).

```
sylvia.clus <- clustal(sylvia.seq)
```

4-We save in taxa.sylvia the names of all the species

```
taxa.sylvia <- attr(sylvia.seq, "species")
names(taxa.sylvia) <- names(sylvia.seq)
taxa.sylvia[1] <- "Sylvia_atricapilla"
taxa.sylvia[24] <- "Sylvia_abyssinica"
```

5- We create a function that reroots a phylogenetic tree with 'AJ534526' as the outgroup

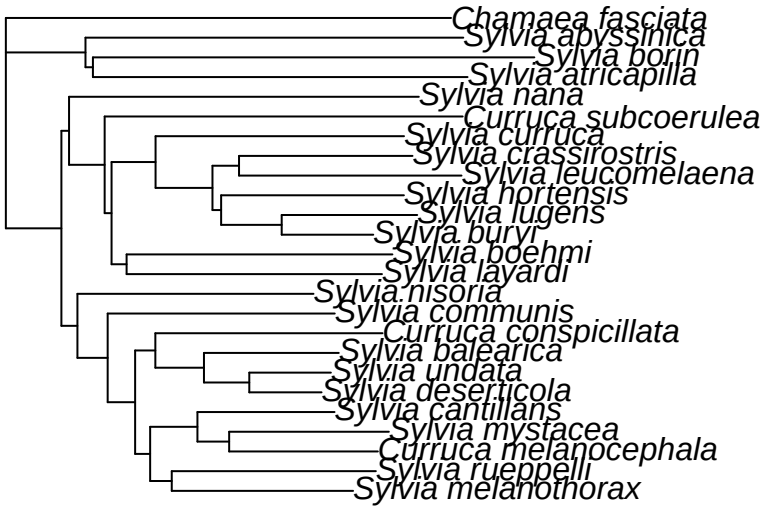
```
f <- function(xx) root(nj(dist.dna(xx, p=TRUE)), "AJ534526")
```

6- We reroot using our aligned sequence

```
tr <- f(sylvia.clus)
nj.est <- tr
```

7- We relabel from the 'criptic' names to the useful species names.

```
nj.est$tip.label <- taxa.sylvia[tr$tip.label]
write.tree(nj.est, "sylvia_nj_k80.tre")
plot(nj.est)
```



Finally we get the trait data from http://ape-package.ird.fr/APER/APER2/sylvia_data.txt

```
sylvia.eco <- read.table("sylvia_data.txt")
```

We drop the outgroup species (Chamaea fasciata) for which we have no ecological data:

```
nj.est <- drop.tip(nj.est, "Chamaea_fasciata")
```

Question 1.2

Sorting the data so its rows are in the same order as the tip labels of the tree

```
DF <- sylvia.eco[nj.est$tip.label, ]
```

Evolution: migration is realted to geographical range:

```
table(DF$geo.range, DF$mig.behav)
```

```
##
##            long resid short
##   temp        0     4     0
##   temptrop    9     0     2
##   trop        0     6     0
```

Consider different models (parameter model). Report on the results and interpret the estimated rates and their standard errors. Analyze the discrete (type=discrete) geographical range variable (DF$geo.range)

```
syl.er <- ace(DF$geo.range, nj.est, type = "d")
syl.er
```

```
##
##     Ancestral Character Estimation
##
## Call: ace(x = DF$geo.range, phy = nj.est, type = "d")
##
##     Log-likelihood: -19.94478
```

```
##
## Rate index matrix:
##          temp temptrop trop
## temp        .        1    1
## temptrop    1        .    1
## trop        1        1    .
##
## Parameter estimates:
##  rate index estimate std-err
##           1   5.9888  2.2301
##
## Scaled likelihoods at the root (type '...$lik.anc' to get them for all nodes):
##       temp   temptrop       trop
## 0.02488517 0.90928439 0.06583044
```

symmetrical model: transition rates will change from state to state. But transitions between two given states have equal rates in both directions

```
syl.sym <- ace(DF$geo.range, nj.est, type="d", model="SYM")
```

```
## Warning in sqrt(diag(solve(h))): NaNs produced
```

```
syl.sym
```

```
##
##     Ancestral Character Estimation
##
## Call: ace(x = DF$geo.range, phy = nj.est, type = "d", model = "SYM")
##
##     Log-likelihood: -18.40128
##
## Rate index matrix:
##          temp temptrop trop
## temp        .        1    2
## temptrop    1        .    3
## trop        2        3    .
##
## Parameter estimates:
##  rate index estimate std-err
##           1   3.9839  2.3194
##           2   0.0000     NaN
##           3  11.8850  5.9166
##
## Scaled likelihoods at the root (type '...$lik.anc' to get them for all nodes):
##        temp    temptrop        trop
## 0.002427204 0.772384704 0.225188093
```

The symmetrical model is the best one.

We compare likelihood:

```
anova(syl.er, syl.sym)
```

```
## Likelihood Ratio Test Table
##   Log lik. Df Df change Resid. Dev Pr(>|Chi|)
## 1  -19.945  1
## 2  -18.401  3         2      3.087     0.2136
```

model: only the transitions temp to temptrop to trop. We create a matrix mod

```
mod <- matrix(0, 3, 3)
mod[2, 1] <- mod[1, 2] <- 1
mod[2, 3] <- mod[3, 2] <- 2

syl.mod <- ace(DF$geo.range, nj.est, type="d", model=mod)
syl.mod
```

```
##
##     Ancestral Character Estimation
##
## Call: ace(x = DF$geo.range, phy = nj.est, type = "d", model = mod)
##
##     Log-likelihood: -18.40128
##
## Rate index matrix:
##          temp temptrop trop
## temp        .        1    0
## temptrop    1        .    2
## trop        0        2    .
##
## Parameter estimates:
##  rate index estimate std-err
##           1   3.9839  2.3970
##           2  11.8850  5.9998
##
## Scaled likelihoods at the root (type '...$lik.anc' to get them for all nodes):
##        temp    temptrop        trop
## 0.002427204 0.772384708 0.225188087
```

we compare them using their AIC values:

```
sapply(list(syl.er, syl.sym, syl.mod), AIC)
```

```
## [1] 41.88955 42.80257 40.80257
```

custom model has the smallest AIC.

Question 2

Within this assignment, we are working with different libraries we have to load and attach. Since the *carni70* data set will be used to fit the five phylogenetic comparative models in 2.2, it will be also read into the R environment.

```
# loading necessary libraries
library(ape)
library(mvSLOUCH)
library(ouch)
library(ade4)
library(ggplot2)
library(cluster)
library(mvMORPH)

# reading data
```

```r
data(carni70)
data = carni70$tab
size = data$size
range = data$range
```

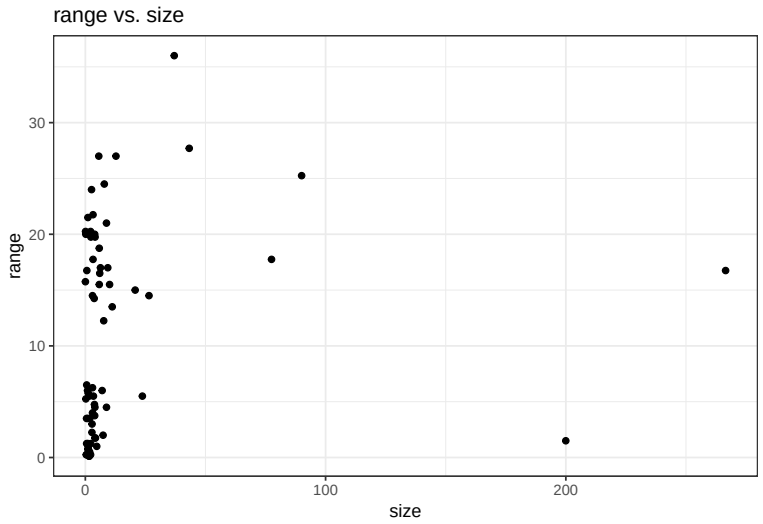
Question 2.1

In the first step, we explore the data. *carni70* is a list of two elements. The second element represents the data.

```r
str(data)
```

```
## 'data.frame':    70 obs. of  2 variables:
##  $ size : num  37.01 2.59 3.2 7.9 3.99 ...
##  $ range: num  36 24 21.75 24.5 1.75 ...
```

The data frame includes 70 observations and two columns. For each observation, namend by the name of carnivora species, the first column reports the bodysize and the second column represents the geographic range.

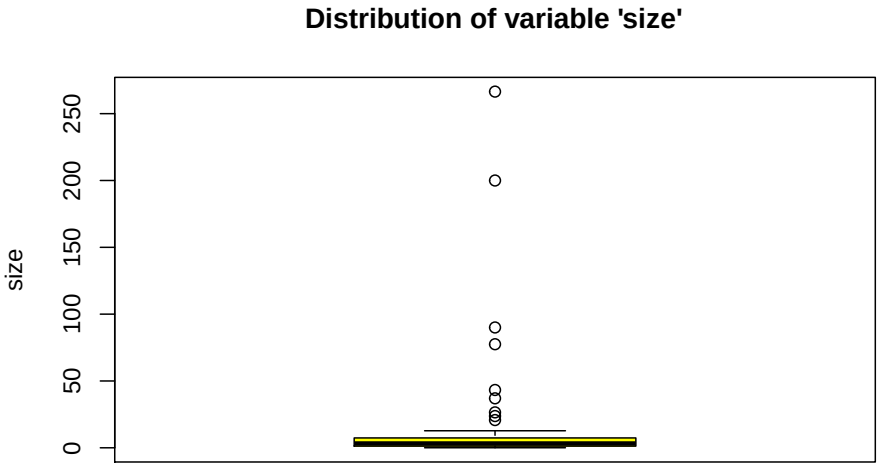


Plotting the data using *size* as the x-axis and *range* as the y-axis, it already becomes clear that for the *size*-variable there are some outliers with very high values. Also, there seem to be two clusters within the data. The first cluster seems to be characcterized by small *size*- and *range*-values. The second cluster seems to be defined by small *size*-values and higher *range*-values.

Distribution of variable *size*

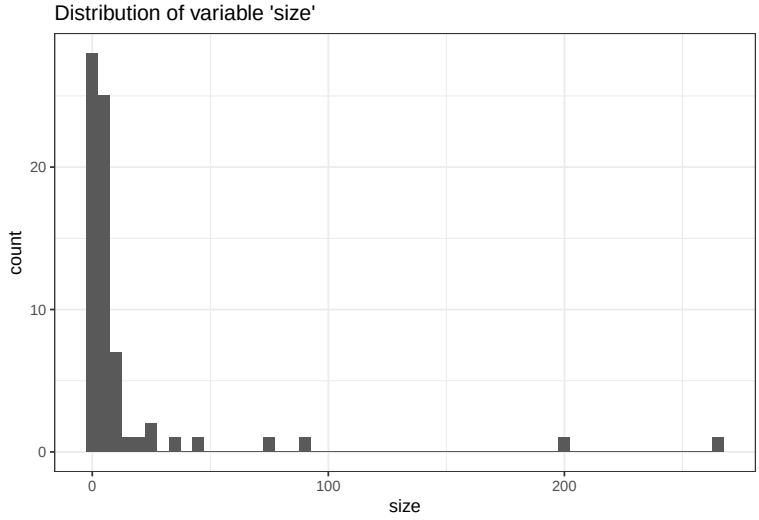
The following table delivers some descriptive statistics related to the distribution of the *size*-variable.

Min.	0.04000
1st Qu.	1.28250
Median	3.20000
Mean	14.28771
3rd Qu.	7.29250
Max.	266.50000



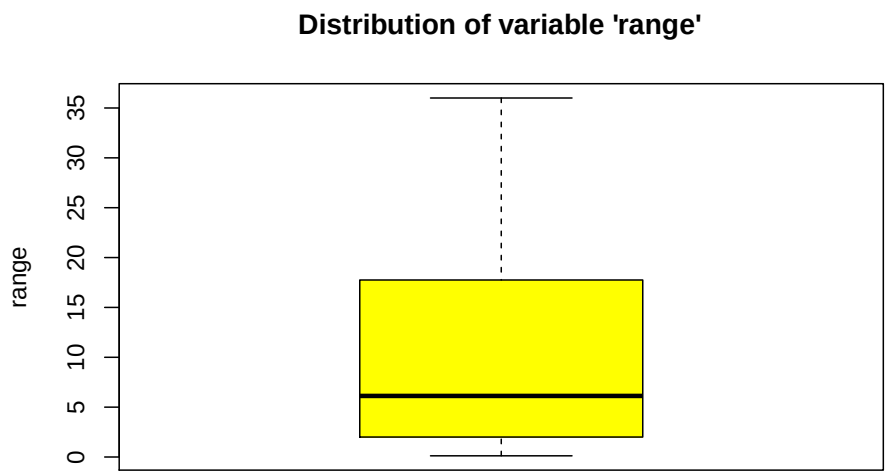
The shape of the boxplot for the *size*-variable indicates that most of the data points are characterized by very similar *size*-values around the median of 3.2. This shows that for most of the observations, the variance related to the *size*-variable is pretty low. However, there are a few significant outliers in the data which are represented by the small circles in the plot.

These outliers can be also observed in the following histogram.

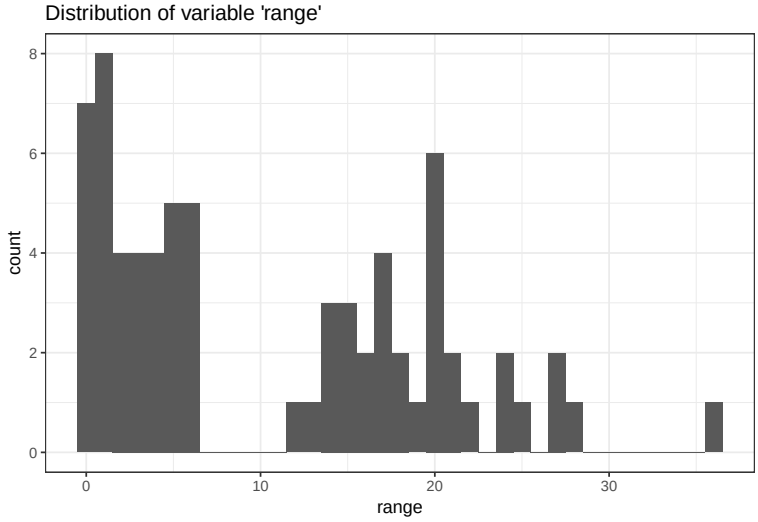


Distribution of variable *range*

Min.	0.12000
1st Qu.	2.06250
Median	6.12500
Mean	10.72057
3rd Qu.	17.75000
Max.	36.00000

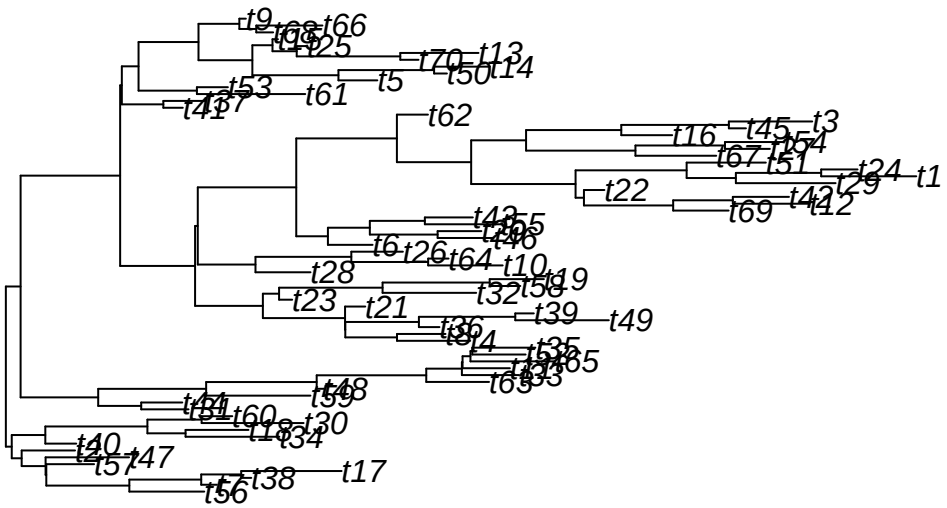


In contrast to the *size*-variabke, the boxplot shows clearly that the *range*-variable does not include any outliers, because the maximum value of 36 does not exceed $Q3 + IQR * 1,5 (= 41.28)$. The following histogram also indictaes, that the spread of the data is much lower than for the *size*-variable.



Question 2.2

```
# creating random tree in ouch format with same size of data
tree = rtree(n=70)
plot(tree)
```




```
myoutree = ape2ouch(tree)
```

2.2.1 Independent Brownian motions

```
# Both traits evolve as independent Brownian motions.
brown_size <- BrownianMotionModel(myoutree, data = size)
brown_range <- BrownianMotionModel(myoutree, data = range)
```

2.2.2 Correlated Brownian motions

```
# The traits evolve as a correlated Brownian motion
# usage of created random tree
phyltree = myoutree
# Correct the names of the internal node labels
phyltree@nodelabels[1:(phyltree@nnodes-phyltree@nterm)]<-as.character(
  1:(phyltree@nnodes-phyltree@nterm))
# Correct the names of the end node labels
phyltree@nodelabels[(phyltree@nnodes-phyltree@nterm+1):phyltree@nnodes]<-as.character(rownames(data))
### And finally try to recover the parameters of the Brownian motion.
correlation <- cor(data$size, data$range)
BMestim<-BrownianMotionModel(phyltree, data, M.error = correlation)
```

2.2.3 Independent Ornstein-Uhlenbeck processes

```
# Both traits evolve as independent Ornstein-Uhlenbeck processes.
ou_size <- ouchModel(myoutree, data = size)
```

```
## [1] "Starting point of heuristic search procedure : "
##        A      Syy
## 1.872120 1.273041
```

```
ou_range <- ouchModel(myoutree, data = range)
```

```
## [1] "Starting point of heuristic search procedure : "
##         A       Syy
## 1.2818904 0.1649841
```

2.2.4 Bivariate Ornstein-Uhlenbeck processes

```
# usage of created random tree
tree4 = tree
# Correct the names of the end node labels
tree4$tip.label = as.character(rownames(data))
# fit model
trait_OUM <- mvOU(tree4, data, model = "OUM")
```

```
## No selective regimes mapped on the tree, only a OU1 model could be estimated
## successful convergence of the optimizer
## a reliable solution has been reached
##
## -- Summary results --
## LogLikelihood:    -609.5493
## AIC:      1239.099
## AICc:     1240.804
## 10 parameters
##
## Estimated theta values
## ----------------------
##                  size         range
## theta_0 1.468404e+06 -4.425259e+07
## theta_1 1.221751e+01  1.087729e+01
##
## ML alpha values
## ----------------------
##             size      range
## size   1.2075140 -0.8102434
## range -0.8102434 25.5985627
##
## ML sigma values
## ----------------------
##            size     range
## size  3917.9042 -800.6864
## range -800.6864 4015.3261
```

2.2.5 Size evolves as Brownian Motion, range evolves as Ornstein-Uhlenbeck process

```
# All of the internal nodes have to be uniquely named
myoutree@nodelabels[1:(myoutree@nnodes-myoutree@nterm)] =
  as.character(1:(myoutree@nnodes-myoutree@nterm))
```

```
myoutree@nodelabels[(myoutree@nnodes-myoutree@nterm+1):myoutree@nnodes] =
  as.character(rownames(data))
# fit model
brown_ou <- mvslouchModel(myoutree, data = data[,c(2,1)], kY = 1) # second column = Brownian
```

```
## [1] "Starting point of heuristic search procedure : "
##          A        Syy
## -0.1808913  0.1997172
```

```
# comparison of the models
# Independent Brownian motions:
brown_size$ParamSummary$aic
```

```
## [1] 688.1079
```

```
brown_range$ParamSummary$aic
```

```
## [1] 545.1826
```

```
# Correlated Brownian motions:
BMestim$ParamSummary$aic
```

```
## [1] 1234.841
```

```
# Independent Ornstein-Uhlenbeck processes
ou_size$FinalFound$ParamSummary$aic
```

```
## [1] 689.2891
```

```
ou_range$FinalFound$ParamSummary$aic
```

```
## [1] 512.8511
```

```
# Bivariate Ornstein-Uhlenbeck processes
trait_OUM$AIC
```

```
## [1] 1239.099
```

```
# Size evolves as Brownian Motion, range evolves as Ornstein-Uhlenbeck process
brown_ou$FinalFound$ParamSummary$aic
```

```
## [1] 1201.609
```

In principle want the lowest values for the AIC score. For this we need to transform the AIC scores for the two independent models.

```
k <- brown_size$ParamSummary$dof
n <- nrow(data)
aic <- (2*k) - 2*(brown_size$ParamSummary$LogLik + brown_range$ParamSummary$LogLik)
aic_correction_numerator <- (2*(k^2) + 2*k)
aic_correction_denominator <- (n + n) - k - k - 1
aic_new <- aic + aic_correction_numerator/aic_correction_denominator
```

```
k_ou <- ou_size$FinalFound$ParamSummary$dof
aic_ou <- (2*k) - 2*(ou_size$FinalFound$LogLik + ou_range$FinalFound$LogLik)
aic_ou_numerator <- (2*(k_ou^2) + 2*k_ou)
aic_ou_denominator <- (n + n) - k_ou - k_ou - 1
ou_new <- aic_ou + aic_ou_numerator/aic_ou_denominator
```

```
# comparison of the models
# Independent Brownian motions:
aic_new
```

```
## [1] 1229.379
```

```
# Correlated Brownian motions:
BMestim$ParamSummary$aic
```

```
## [1] 1234.841
```

```
# Independent Ornstein-Uhlenbeck processes
ou_new
```

```
## [1] 1194.321
```

```
# Bivariate Ornstein-Uhlenbeck processes
trait_OUM$AIC
```

```
## [1] 1239.099
```

```
# Size evolves as Brownian Motion, range evolves as Ornstein-Uhlenbeck process
brown_ou$FinalFound$ParamSummary$aic
```

```
## [1] 1201.609
```

After transforming the AIC codes in a proper way in order to compare them. We conclude that the most suitable model is the one with the lowest AIC score. Which, in our case is when the traits evolve as two independent Ornstein-Uhlenbeck processes. This gives us an AIC score of 1230.38