Lab 2 Bioinformatics

Masinde, Maria, Jasleen, Mathew

2018-12-27

Preliminary

Packages used in this exercise.

```
#--- ape ---
library(ape)

#--- seqinr ---
library(seqinr)

#--- simulation ---
library(phangorn)

#--- msa ---
library(msa)

library(markovchain)
```

Question 1.

There are gaps in the sequence in each of the accession sequences. Points of **puRines (R)**, **pYrimidines (Y)**, **aMino groups bases (M)** and **strong interaction (S)**. Adenine have the highest composition percentage at each of the accession numbers. Area of strong interaction only occurs at accession number "FJ356747". Overall, these imply the sequences are incomplete. In the simulations gaps and all other non ACTG are removed.

The length of the sequnces at the first five accessions are as below.

```
## $JF806202
## [1] 1081
##
## $HM161150
## [1] 2934
##
## $FJ356743
## [1] 3132
##
## $JF806205
## [1] 1093
##
## $JQ073190
## [1] 1597
```

First five accession base compositions:

The first five accession GC are as follows:

```
#--- GC content of each ---
lapply(lizard_seqs, GC)[1:5]
```

```
## $JF806202
## [1] 0.446339
##
## $HM161150
## [1] 0.4437062
##
## $FJ356743
## [1] 0.4448288
##
## $JF806205
## [1] 0.4542744
##
## $JQ073190
## [1] 0.4348711
```

Question 1.1

Artificial DNA simulation; each nucleotide randomly and independently drawn from base frequencies.

```r
bcomps <- lapply(lizard_seqs, table)

#--- simulation fxn ---

set.seed(12345)
sims <- function(x){

  x <- lapply(x, function(x){x[c("a","c","g","t")]})
  # x is base compositions

  # return a list of sequences
  art_sims <- list()

  # simulations
  for (i in 1:length(x)) {
    art_sims[[paste("synthetic", i, sep = "_")]] <- sample(
      names(x[[i]]),
      size = sum(x[[i]]),
      replace = TRUE,
      prob = x[[i]]/sum(x[[i]])
    )
  }

  return(art_sims)
}

sim1_seq <- sims(bcomps)
```

The first five base frequencies are presented below. We observed that in most of the simulated sequence the a's had higher frequencies while in the origin sequence all the accessions had a's with highest frequency.

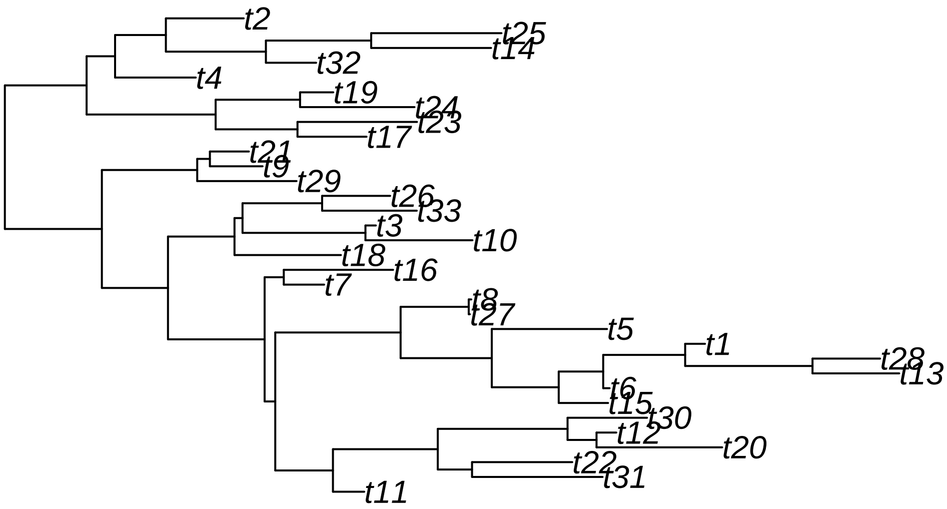
```
## $synthetic_1
##
##   a   c   g   t
## 262 200 272 263
##
```

```
## $synthetic_2
##
##   a   c   g   t
## 851 583 594 681
##
## $synthetic_3
##
##   a   c   g   t
## 912 592 666 721
##
## $synthetic_4
##
##   a   c   g   t
## 260 222 240 284
##
## $synthetic_5
##
##   a   c   g   t
## 455 288 343 388
```

```
#--- saving simulation as fasta file
ape::write.dna(sim1_seq, file ="sim1_seq.fasta", format = "fasta", append =FALSE, nbcol = 6, colsep = "
```

Question 1.2

```
#---- creating a phylogenetic tree with 33 nodes
set.seed(12345)
tree <- rtree(n = 33)

plot(tree)
```



We opted to simulate sequences using a custom transition matrix Q where probabiliy of mutation is 0.1. Base frequencies of each of the accessions are used in the simulation. The DNA sequences are saved as a fasta file **sim2_seq.fasta**.

```
#simulating the sequence

Q_mat <- matrix(data= c(0.9,0.1,0.1,0.1,0.1,0.9,0.1,0.1,0.1,0.1,0.9,0.1,0.1,0.1,0.1,0.9),4,4)

base_f <- as.vector(bcomps[[1]][c("a","c","g","t")])/sum(bcomps[[1]][c("a","c","g","t")])

data <- simSeq(tree, l = 1000, type="DNA", bf=base_f, Q=Q_mat)

counter = 1
sim2_seq = list()
for (i in 1:length(data))
{
  sim2_seq[[paste("synthetic2", i, sep = "_")]] <- as.character(data)[counter:(counter+1000)]
  counter = counter + 1000
}
```

A sample of the base compostion on the simulated sequences is shown below. The base frequencies are almost similar to the original sequence with a's and t's having higher frequencies.

```
## $synthetic2_1
##
##   a   c   g   t
## 323 225 237 216
##
```

```
## $synthetic2_2
##
##   a   c   g   t
## 302 227 204 268
##
## $synthetic2_3
##
##   a   c   g   t
## 290 211 267 233
##
## $synthetic2_4
##
##   a   c   g   t
## 263 224 243 271
##
## $synthetic2_5
##
##   a   c   g   t
## 311 198 234 258
```

Question 2

Question 2.1

Basic statistics on GC content.

```
gc1 <- lapply(lizard_seqs, GC)
gc1[1:5]
```

```
## $JF806202
## [1] 0.446339
##
## $HM161150
## [1] 0.4437062
##
## $FJ356743
## [1] 0.4448288
##
## $JF806205
## [1] 0.4542744
##
## $JQ073190
## [1] 0.4348711
```

```
# GC content of sim1_seq
gc2 <- lapply(sim1_seq, GC)
gc2[1:5]
```

```
## $synthetic_1
## [1] 0.4734203
##
## $synthetic_2
## [1] 0.4344777
##
## $synthetic_3
## [1] 0.4351435
```

```
##
## $synthetic_4
## [1] 0.4592445
##
## $synthetic_5
## [1] 0.4280868
```

```r
# omit na from sim2_seq
sim2_seq <- lapply(sim2_seq, na.omit)
gc3 <- lapply(sim2_seq, GC)
gc3[1:5]
```

```
## $synthetic2_1
## [1] 0.4615385
##
## $synthetic2_2
## [1] 0.4305694
##
## $synthetic2_3
## [1] 0.4775225
##
## $synthetic2_4
## [1] 0.4665335
##
## $synthetic2_5
## [1] 0.4315684
```

AT content for original lizard sequence first 5. Computed as 1 - GC content.

```r
at1 <- list()

for (i in 1:length(gc1)) {
  at1[[i]] <- 1 -gc1[[i]]
}

at1[1:5]
```

```
## [[1]]
## [1] 0.553661
##
## [[2]]
## [1] 0.5562938
##
## [[3]]
## [1] 0.5551712
##
## [[4]]
## [1] 0.5457256
##
## [[5]]
## [1] 0.5651289
```

AT content for independence sequence simulation.

```r
at2 <- list()
for (i in 1:length(gc2)) {
  at2[[i]] <- 1 - gc2[[i]]
}

at2[1:5]
```

```
## [[1]]
## [1] 0.5265797
##
## [[2]]
## [1] 0.5655223
##
## [[3]]
## [1] 0.5648565
##
## [[4]]
## [1] 0.5407555
##
## [[5]]
## [1] 0.5719132
```

AT content of simulated sequence from tree

```
at3 <- list()
for (i in 1:length(gc3)) {
  at3[[i]] <- 1 - gc3[[i]]
}

at3[1:5]
```

```
## [[1]]
## [1] 0.5384615
##
## [[2]]
## [1] 0.5694306
##
## [[3]]
## [1] 0.5224775
##
## [[4]]
## [1] 0.5334665
##
## [[5]]
## [1] 0.5684316
```

For all the sequences AT content was greater than 50%.

Base compositions:

```
lapply(lizard_seqs, table)[1:5]
```

```
## $JF806202
##
##        a   c   g   t   y
##  83 289 202 243 263   1
##
## $HM161150
##
##        a   c   g   t
## 225 844 575 627 663
##
## $FJ356743
##
##       a   c   g   t
## 241 905 607 679 700
##
## $JF806205
##
##       a   c   g   r   t   y
##  84 286 211 246   2 263   1
##
## $JQ073190
##
##       a   c   g   t
## 123 451 293 348 382
```

```r
# base composition of sim2_seq
lapply(sim1_seq, table)[1:5]
```

```
## $synthetic_1
##
##   a   c   g   t
## 262 200 272 263
##
## $synthetic_2
##
##   a   c   g   t
## 851 583 594 681
##
## $synthetic_3
##
##   a   c   g   t
## 912 592 666 721
##
## $synthetic_4
##
##   a   c   g   t
## 260 222 240 284
##
## $synthetic_5
##
##   a   c   g   t
## 455 288 343 388
```

```r
# base compostion of sim1_seq
lapply(sim2_seq, table)[1:6]
```

```
## $synthetic2_1
##
##   a   c   g   t
## 323 225 237 216
##
## $synthetic2_2
##
##   a   c   g   t
## 302 227 204 268
##
## $synthetic2_3
##
##   a   c   g   t
## 290 211 267 233
##
## $synthetic2_4
##
##   a   c   g   t
## 263 224 243 271
##
## $synthetic2_5
##
##   a   c   g   t
## 311 198 234 258
##
## $synthetic2_6
##
##   a   c   g   t
## 279 198 243 281
```

We used the translate function from package **seqinr** to obtain the protein sequences of each of the three data sets. We noticed that the simulated sequence using independence model the accessions had higher number of stop codons compared to the original sequences data sets. Some accessions had as high as 73 stop codons.

```
p_seq1 <- lapply(lizard_seqs, translate)

p_seq2 <- lapply(sim2_seq, translate)

p_seq3 <- lapply(sim1_seq, translate)
```

Question 2.2

Markov chains were fitted for the lizards sequences and the artificial sequences, from the transition matrix obtained we have understood that the markov chain is of order 1. Thus, the current state only depends on the immediate previous state.

```
markov1 <-markovchainFit(sim1_seq)

markov2 <-markovchainFit(sim2_seq)

markov_initial <-markovchainFit(lizard_seqs)
```

```
#Align sequences

Lizard_seq_align <- msa(readDNAStringSet("lizard_seqs.fasta"))
```

```
## Warning in .Call2("fasta_index", filexp_list, nrec, skip, seek.first.rec, :
## reading FASTA file lizard_seqs.fasta: ignored 5451 invalid one-letter
## sequence codes
```

```
## use default substitution matrix
```

```
sim1_seq_align <- msa(readDNAStringSet("sim1_seq.fasta"))
```

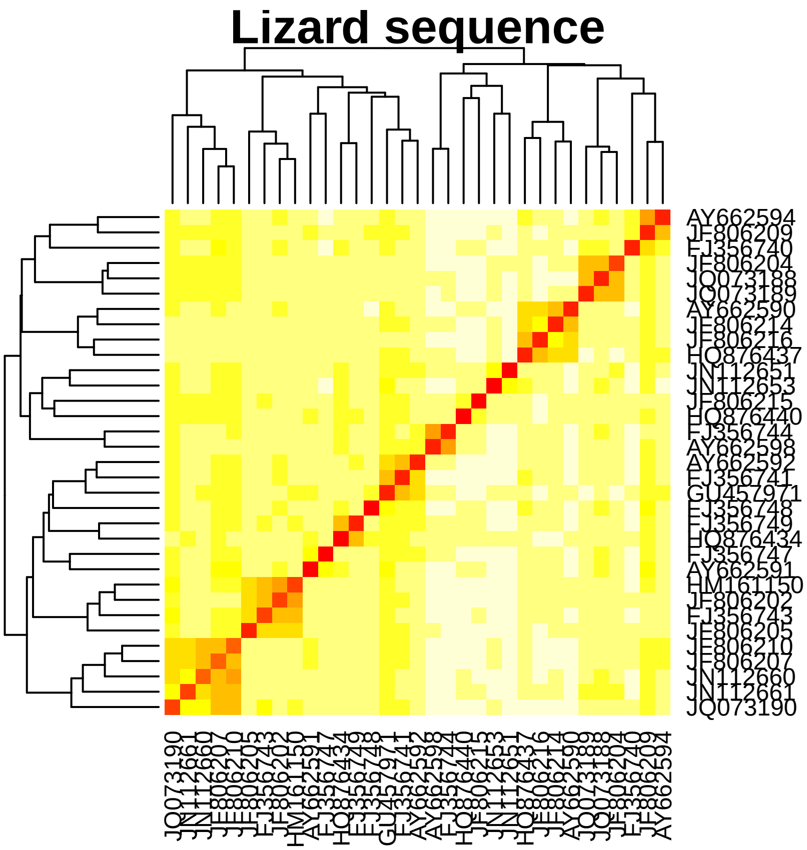
```
## Warning in .Call2("fasta_index", filexp_list, nrec, skip, seek.first.rec, :
```

```
## reading FASTA file sim1_seq.fasta: ignored 5448 invalid one-letter sequence
## codes

## use default substitution matrix
```

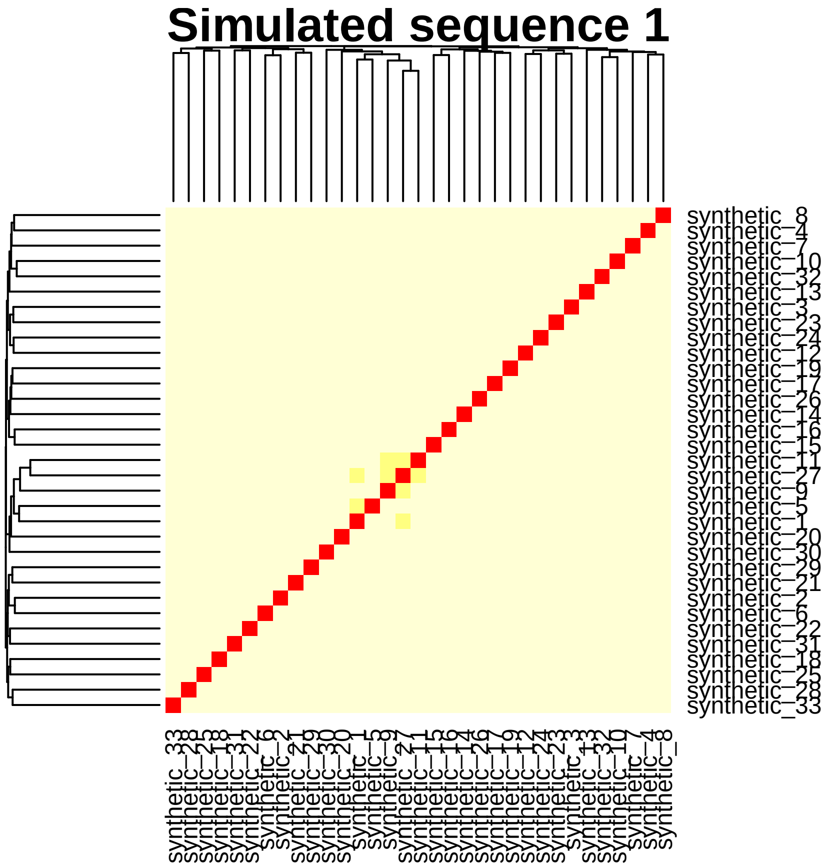
```
sim2_seq_align <- msa(readDNAStringSet("sim2_seq.fasta"))
```

```
## Warning in .Call2("fasta_index", filexp_list, nrec, skip, seek.first.rec, :
## reading FASTA file sim2_seq.fasta: ignored 2772 invalid one-letter sequence
## codes

## use default substitution matrix
```

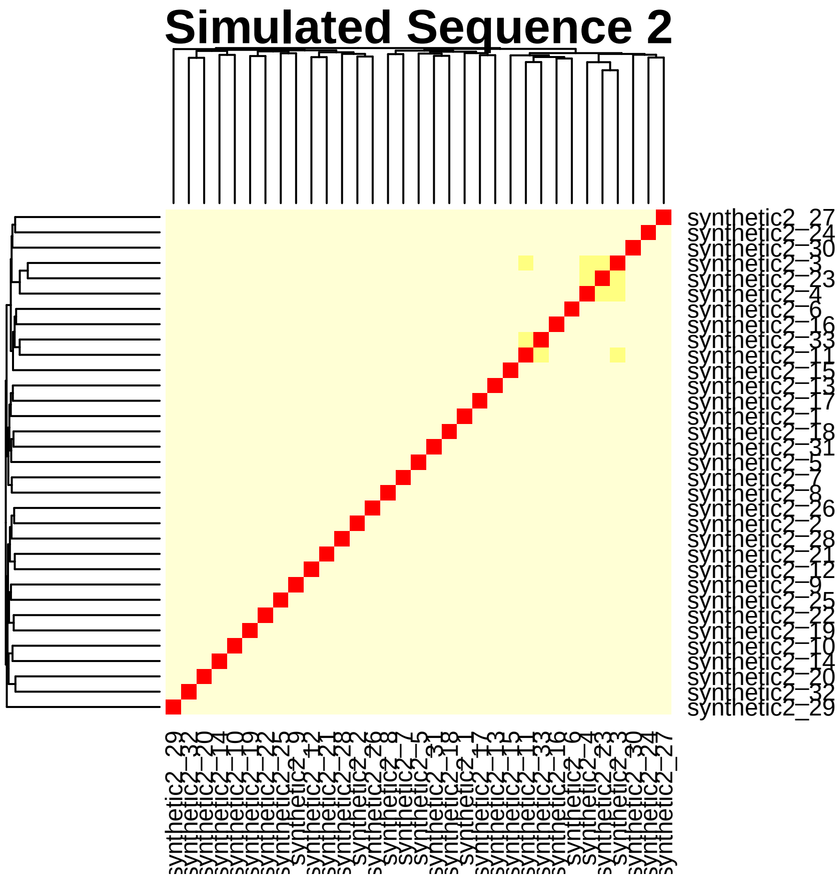
```
Lizard_seq_align_c <- msaConvert(Lizard_seq_align, type="seqinr::alignment")
sim1_seq_align_c <- msaConvert(sim1_seq_align, type="seqinr::alignment")
sim2_seq_align_c <- msaConvert(sim2_seq_align, type="seqinr::alignment")


dist_mat1 <- as.matrix(dist.alignment(Lizard_seq_align_c, matrix = "identity"))
dist_mat2 <- as.matrix(dist.alignment(sim1_seq_align_c, matrix = "identity"))
dist_mat3 <- as.matrix(dist.alignment(sim2_seq_align_c, matrix = "identity"))


heatmap(dist_mat1, main = "Lizard sequence")
```



```r
heatmap(dist_mat2, main = "Simulated sequence 1")
```



```r
heatmap(dist_mat3, main = "Simulated Sequence 2")
```



From the heat map of the lizard sequence we observe clusters on the anti-diagonal.The clusters in this heatmap represent species that are having similar traits. The heat maps of the simulated sequences does not have distinguishable clusters, this can be attributed to the fact that these sequences have no biological significance since they are randomly generated from the original sequences.

Question3

```
#fun <- function(x) as.phylo(hclust(dist.dna(x), "average")) # upgma() in phangorn
#tree <- fun(Lizard_seq_align_d)
#bstrees <- boot.phylo(tree, Lizard_seq_align_d, fun, trees = TRUE)$trees

f <- function(x) upgma(x)

tree <- f(dist_mat1)
bstrees <- boot.phylo(tree, dist_mat1, f, trees = TRUE)$trees
```

```
##
Running bootstraps:       100 / 100
## Calculating bootstrap values... done.
```

```
sim1_tree <- f(dist_mat2)
bstrees_sim1 <- boot.phylo(sim1_tree, dist_mat2, f, trees = TRUE)$trees
```

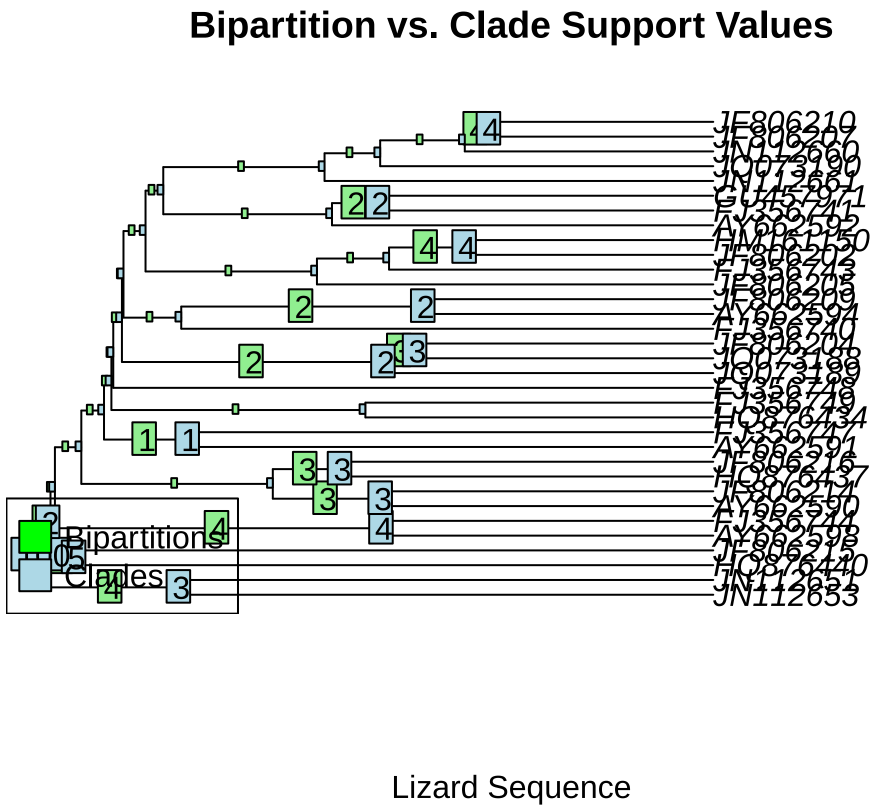
```
##
Running bootstraps:       100 / 100
## Calculating bootstrap values... done.
```

```r
sim2_tree <- f(dist_mat3)
bstrees_sim2 <- boot.phylo(sim2_tree, dist_mat3, f, trees = TRUE)$trees
```

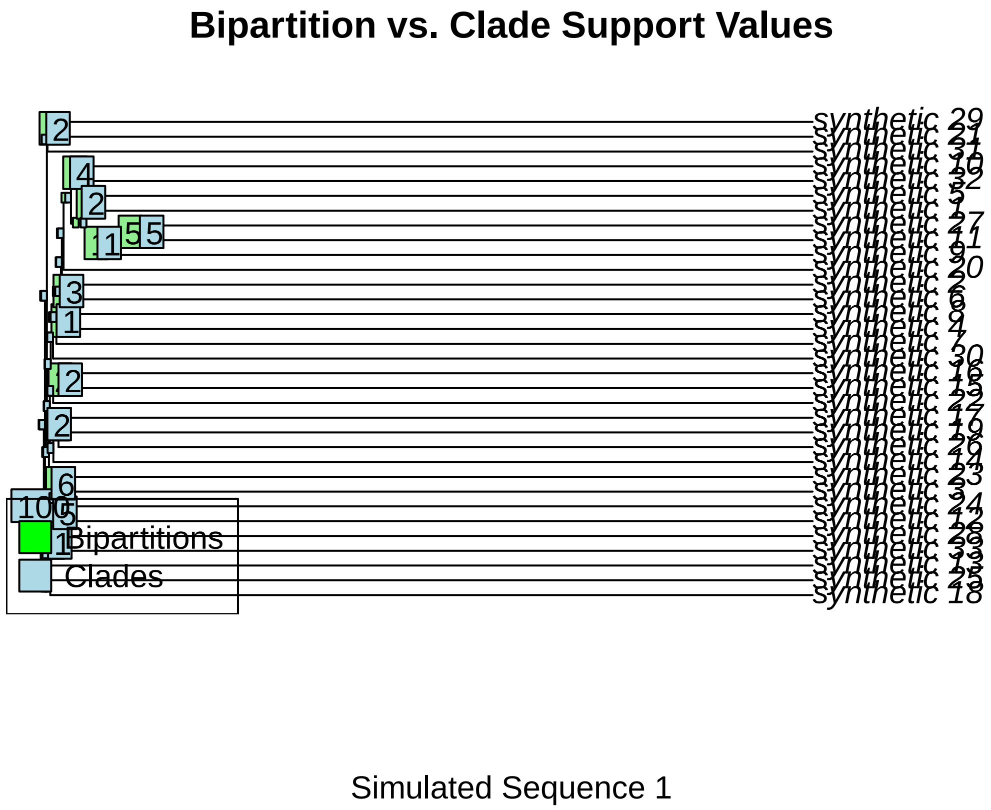
```
##
Running bootstraps:       100 / 100
## Calculating bootstrap values... done.
```

```r
# clads

clad_l_seq <- prop.clades(tree, bstrees, rooted = TRUE)
clad_sim1 <- prop.clades(sim1_tree, bstrees_sim1, rooted = TRUE)
clad_sim2 <- prop.clades(sim2_tree, bstrees_sim2, rooted = TRUE)

# bipartitions

boot <- prop.clades(tree, bstrees)
layout(1)
par(mar = rep(2, 4))

boot_sim1 <- prop.clades(sim1_tree, bstrees_sim1)
layout(1)
par(mar = rep(2, 4))

boot_sim2 <- prop.clades(sim2_tree, bstrees_sim2)
layout(1)
par(mar = rep(2, 4))
```

```r
#plot of main DNA
plot(tree, main = "Bipartition vs. Clade Support Values", sub = "Lizard Sequence")
drawSupportOnEdges(boot)
nodelabels(clad_l_seq)
legend("bottomleft", legend = c("Bipartitions", "Clades"), pch = 22,
       pt.bg = c("green", "lightblue"), pt.cex = 2.5)
```

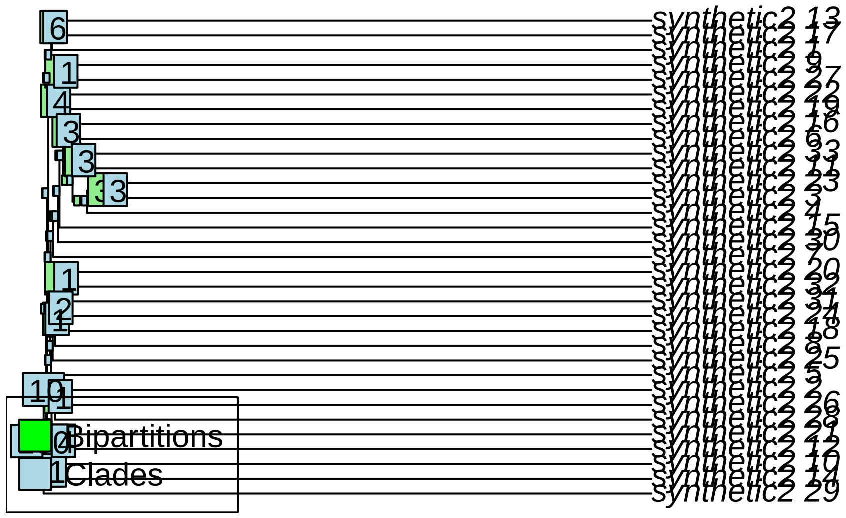


```r
#Plot synthetic sequence 1
plot(sim1_tree, main = "Bipartition vs. Clade Support Values", sub = "Simulated Sequence 1")
drawSupportOnEdges(boot_sim1)
nodelabels(clad_sim1)
legend("bottomleft", legend = c("Bipartitions", "Clades"), pch = 22,
       pt.bg = c("green", "lightblue"), pt.cex = 2.5)
```



```
#plot synthetic sequence 2
plot(sim2_tree, main = "Bipartition vs. Clade Support Values", sub = "Simulated Sequence 2")
drawSupportOnEdges(boot_sim2)
nodelabels(clad_sim2)
legend("bottomleft", legend = c("Bipartitions", "Clades"), pch = 22,
       pt.bg = c("green", "lightblue"), pt.cex = 2.5)
```

Bipartition vs. Clade Support Values



Simulated Sequence 2

We have used UPGMA function and passed distance matrices into it to construct the phylogenitic trees. The trees generated from simulated sequences are not similar to the tree generated from the lizard sequence. We used boot.phylo to bootstrap the trees and then function prop.clades was used to measure the clades. The clade values in the plot show the number of bootstrap trees (out of 100) that share the same clade as with the input tree.

```
treedist(
  tree1 = tree,
  tree2 = sim1_tree,
  check.labels = FALSE
)
```

```
##      symmetric.difference   branch.score.difference
##                 54.000000                  1.907072
##           path.difference quadratic.path.difference
##                104.747315                 13.470256
```

```
treedist(
  tree1 = tree,
  tree2 = sim2_tree,
  check.labels = FALSE
)
```

```
##      symmetric.difference   branch.score.difference
##                  54.00000                   1.91895
##           path.difference quadratic.path.difference
##                 104.40307                  13.55005
```

```r
treedist(
  tree1 = sim1_tree,
  tree2 = sim2_tree,
  check.labels = FALSE
)
```

```
##      symmetric.difference   branch.score.difference
##               46.00000000                0.04878851
##           path.difference quadratic.path.difference
##               77.44675590                0.17277232
```

```r
comparePhylo(tree, sim1_tree, plot = FALSE, force.rooted = FALSE,
             use.edge.length = FALSE)
```

```
## => Comparing tree with sim1_tree.
## Both trees have the same number of tips: 33.
## Tips in tree not in sim1_tree : FJ356741, AY662592, GU457971, AY662591, FJ356747, JF806202, HM161150
## Tips in sim1_tree not in tree : synthetic_11, synthetic_27, synthetic_9, synthetic_1, synthetic_5, sy
## Both trees have the same number of nodes: 32.
## Both trees are rooted.
## Both trees are ultrametric.
## 32 clades in tree not in sim1_tree.
## 32 clades in sim1_tree not in tree.
## Branching times of clades in common between both trees: see ..$BT
## (node number in parentheses).
##
## $BT
##   tree sim1_tree
## 1   ()        ()
```

```r
comparePhylo(tree, sim2_tree, plot = FALSE, force.rooted = FALSE,
             use.edge.length = FALSE)
```

```
## => Comparing tree with sim2_tree.
## Both trees have the same number of tips: 33.
## Tips in tree not in sim2_tree : FJ356741, AY662592, GU457971, AY662591, FJ356747, JF806202, HM161150
## Tips in sim2_tree not in tree : synthetic2_3, synthetic2_23, synthetic2_4, synthetic2_11, synthetic2_
## Both trees have the same number of nodes: 32.
## Both trees are rooted.
## Both trees are ultrametric.
## 32 clades in tree not in sim2_tree.
## 32 clades in sim2_tree not in tree.
## Branching times of clades in common between both trees: see ..$BT
## (node number in parentheses).
##
## $BT
##   tree sim2_tree
## 1   ()        ()
```

```r
comparePhylo(sim1_tree, sim2_tree, plot = FALSE, force.rooted = FALSE,
             use.edge.length = FALSE)
```

```
## => Comparing sim1_tree with sim2_tree.
## Both trees have the same number of tips: 33.
## Tips in sim1_tree not in sim2_tree : synthetic_11, synthetic_27, synthetic_9, synthetic_1, synthetic_
## Tips in sim2_tree not in sim1_tree : synthetic2_3, synthetic2_23, synthetic2_4, synthetic2_11, synthe
## Both trees have the same number of nodes: 32.
## Both trees are rooted.
## Both trees are ultrametric.
## 32 clades in sim1_tree not in sim2_tree.
## 32 clades in sim2_tree not in sim1_tree.
## Branching times of clades in common between both trees: see ..$BT
## (node number in parentheses).
##
## $BT
##   sim1_tree sim2_tree
## 1        ()        ()
```

We used treedist function to calculate the distance between the original tree and simulated tree and as expected we find considerable difference in distance measures. We also used comparePhylo function which uses values of similarity/dissimilarity. The number of tips in all the trees are the same (equal to 33), but the branches and structure of the trees are significantly different.