Lab2_report

Question 1: DNA sequence acquisition and simulation

In this question an analysis of 3 nucleotide sequences is performed. One sequence is downloaded from GenBank database, and saved into fatsa file. Here we load thev data set:

```
lizard_seq_data = read.FASTA("files/lizard_seqs.fasta")
str(lizard_seq_data)
```

```
## List of 33
##  $ JF806202: raw [1:998] 18 28 28 88 ...
##  $ HM161150: raw [1:2709] 88 88 18 88 ...
##  $ FJ356743: raw [1:2891] 28 88 88 88 ...
##  $ JF806205: raw [1:1009] 48 28 28 88 ...
##  $ JQ073190: raw [1:1474] 28 28 88 28 ...
##  $ GU457971: raw [1:1091] 18 48 18 88 ...
##  $ FJ356741: raw [1:2891] 28 88 88 88 ...
##  $ JF806207: raw [1:1004] 48 28 28 48 ...
##  $ JF806210: raw [1:1044] 18 48 48 48 ...
##  $ AY662592: raw [1:2920] 28 88 88 88 ...
##  $ AY662591: raw [1:2816] 48 28 18 88 ...
##  $ FJ356748: raw [1:2895] 28 88 88 88 ...
##  $ JN112660: raw [1:2748] 88 88 88 48 ...
##  $ AY662594: raw [1:2837] 28 88 88 88 ...
##  $ JN112661: raw [1:2748] 88 88 88 48 ...
##  $ HQ876437: raw [1:1004] 28 18 48 28 ...
##  $ HQ876434: raw [1:1000] 18 28 28 88 ...
##  $ AY662590: raw [1:2825] 28 88 48 88 ...
##  $ FJ356740: raw [1:2849] 28 88 88 88 ...
##  $ JF806214: raw [1:1008] 48 28 28 88 ...
##  $ JQ073188: raw [1:1472] 28 28 28 28 ...
##  $ FJ356749: raw [1:2893] 28 88 88 88 ...
##  $ JQ073189: raw [1:1447] 28 28 88 28 ...
##  $ JF806216: raw [1:1000] 18 28 28 88 ...
##  $ AY662598: raw [1:2837] 28 88 88 88 ...
##  $ JN112653: raw [1:2506] 88 88 88 48 ...
##  $ JF806204: raw [1:1005] 28 18 28 28 ...
##  $ FJ356747: raw [1:2891] 28 88 88 88 ...
##  $ FJ356744: raw [1:2891] 28 88 88 88 ...
##  $ HQ876440: raw [1:1060] 18 48 18 48 ...
##  $ JN112651: raw [1:2738] 28 88 28 18 ...
##  $ JF806215: raw [1:1003] 18 28 28 88 ...
##  $ JF806209: raw [1:931] 48 28 28 88 ...
##  - attr(*, "class")= chr "DNAbin"
```

```
print(lizard_seq_data)
```

```
## 33 DNA sequences in binary format stored in a list.
##
## Mean sequence length: 1982.879
##    Shortest sequence: 931
##     Longest sequence: 2920
```

```
##
## Labels:
## JF806202
## HM161150
## FJ356743
## JF806205
## JQ073190
## GU457971
## ...
##
## Base composition:
##     a     c     g     t
## 0.312 0.205 0.231 0.252
## (Total: 65.44 kb)
```

```
lizards_accession_numbers = names(lizard_seq_data)
```

Q1.1

In this part we simulate DNA sequences with the same base composition and sequences length as the one from the GenBank:

```
artificialDNA_simulator <- function()
{
  simulated_seq <- list()

  #simulation of sequence
  for (i in 1:length(lizard_seq_data))
  {
    dna_seq_length <- as.numeric(lengths(lizard_seq_data[i]))
    compos_prob <- base.freq(lizard_seq_data[i])
    nucleotides <- c("a","c","g","t")
    sim_seq <- sample(nucleotides,dna_seq_length,rep=TRUE,prob = compos_prob)
    simulated_seq[i] <- list(sim_seq)

  }

  #convert a seqeunce in Dna sequence
  simulated_seq <- as.DNAbin(simulated_seq)
  names(simulated_seq) <- lizards_accession_numbers

  return(simulated_seq)
}

simulated_seq_1 <- artificialDNA_simulator()
write.dna(simulated_seq_1, file ="files/simulated_seqs_1.fasta",
          format = "fasta", append =FALSE,
          nbcol = 6, colsep = " ",
          colw = 10)
```

```
## Base composition of the data from GenBank:
```

```
##         a         c         g         t
## 0.3121454 0.2052325 0.2307222 0.2518999
```

```
## Base composition of the simulated sequences:
```

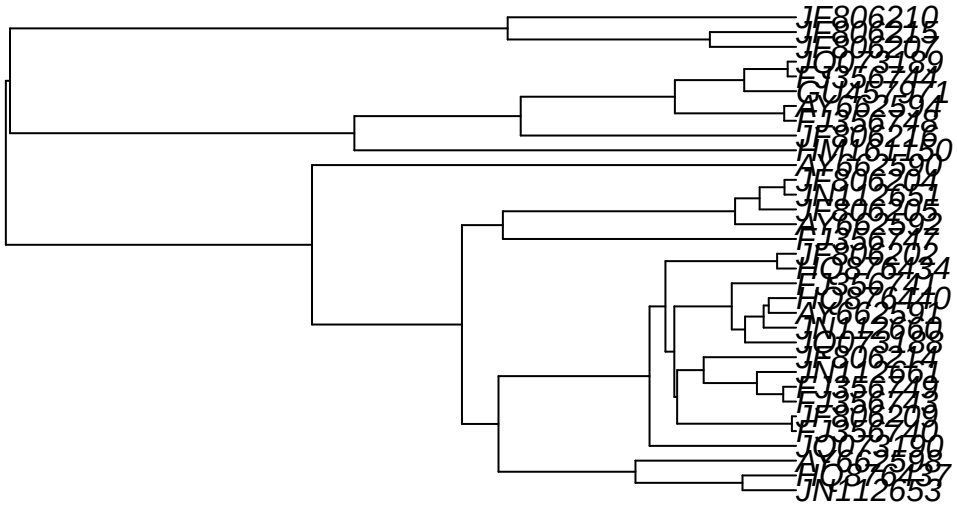
```
##         a         c         g         t
## 0.3116528 0.2051043 0.2320776 0.2511653
```

We can see that the original data, and the simulated data have quite similar base compositions.

Q1.2

In this part we simulate another set of DNA data with similar characteristics as the original one. Now we use a simulated tree with 33 nodes from the original data and package phangorn to do this:

```
#simulated_seq_2 <- artificialDNA_simulator()
# Simulate a tree with 33 nodes
phylo_tree<- TreeSim::sim.bd.taxa(n=33, numbsim=1, lambda=1, mu=0)[[1]]
phylo_tree$tip.label <- lizards_accession_numbers
plot(phylo_tree, yaxt = "n")
```



We need to create a transition matrix, so we use the full dataset to compute that:

```
#building up a Q - transition matrix

DNA_list <- unlist(as.character(lizard_seq_data),use.names = FALSE)
DNA_unique <- c("a","c","g","t")
matrix <-  matrix(0,
                  ncol = length(DNA_unique),
                  nrow = length(DNA_unique))

#transition matrix
for (i in 1:(length(DNA_list) - 1)) {
  index_of_i <- DNA_unique == DNA_list[i]
  index_of_i_plus_1 <- DNA_unique == DNA_list[i + 1]
  matrix[index_of_i, index_of_i_plus_1] = matrix[index_of_i, index_of_i_plus_1] + 1
}


Q <- matrix / rowSums(matrix)
#Q_lower <- Q[lower.tri(Q)]
```

```r
# Other parameters that are needed for phangorn package function
bf <-  base.freq(lizard_seq_data)
size <- round(mean(lengths(lizard_seq_data)))
```

```r
#Simulate sequences for a given evolutionary tree.
simulated_tree_seq = simSeq(x = phylo_tree,
       type= "DNA",
       Q=Q,
       l = size,
       bf = bf)

# change numbers into letters
states = c("a","c","g","t")

simulation_result = list()
for (i in 1:length(simulated_tree_seq)) {

  sequence = simulated_tree_seq[[i]]
  #looping over each sequence character and change the state number into state letter
  for (seq_index in 1:length(sequence)) {
    state_index = as.numeric(sequence[seq_index])
    sequence[seq_index] = states[state_index]
  }
  simulation_result[[i]] = sequence
}

names(simulation_result) = lizards_accession_numbers
simulated_seq_2 <- as.DNAbin(simulation_result)
write.dna(simulated_seq_2, file ="files/simulated_seqs_2.fasta",
          format = "fasta", append =FALSE,
          nbcol = 6, colsep = " ",
          colw = 10)
```

```
## Base composition of the data from GenBank:
```

```
##         a         c         g         t
## 0.3121454 0.2052325 0.2307222 0.2518999
```

```
## Base composition of the simulated sequences Q1.1:
```

```
##         a         c         g         t
## 0.3116528 0.2051043 0.2320776 0.2511653
```

```
## Base composition of the simulated sequences Q1.2:
```

```
##         a         c         g         t
## 0.3098611 0.2039457 0.2309326 0.2552606
```

The base compositions of all simulated sequences are quite similar to the original one.

Question 2: Sequence analysis

Q2.1

```
## Individual base composition of sequence 1
```

```
## The data from the GenBank:
##         a         c         g         t
## 0.2898696 0.2026078 0.2437312 0.2637914
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2665331 0.2094188 0.2464930 0.2775551
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3141704 0.1996974 0.2385275 0.2476046
##
## Individual base composition of sequence 2
## The data from the GenBank:
##         a         c         g         t
## 0.3115541 0.2122554 0.2314507 0.2447398
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3167220 0.2137320 0.2314507 0.2380952
## The simulated data in Q1.2:
##         a         c         g         t
## 0.2919818 0.2123046 0.2294503 0.2662632
##
## Individual base composition of sequence 3
## The data from the GenBank:
##         a         c         g         t
## 0.3130405 0.2099620 0.2348668 0.2421308
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3085438 0.2044275 0.2466275 0.2404012
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3106404 0.1981846 0.2314675 0.2597075
##
## Individual base composition of sequence 4
## The data from the GenBank:
##         a         c         g         t
## 0.2842942 0.2097416 0.2445328 0.2614314
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2824579 0.2071358 0.2507433 0.2596630
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3192133 0.2022189 0.2218860 0.2566818
##
## Individual base composition of sequence 5
## The data from the GenBank:
##         a         c         g         t
## 0.3059701 0.1987788 0.2360923 0.2591588
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3175034 0.2028494 0.2320217 0.2476255
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3146747 0.2002017 0.2274332 0.2576904
##
```

```
## Individual base composition of sequence 6
## The data from the GenBank:
##         a         c         g         t
## 0.2988084 0.2071494 0.2364803 0.2575619
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2786434 0.2135655 0.2593951 0.2483960
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3151790 0.1936460 0.2299546 0.2612204
##
## Individual base composition of sequence 7
## The data from the GenBank:
##         a         c         g         t
## 0.3137323 0.2068488 0.2338291 0.2455898
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3130405 0.2092701 0.2310619 0.2466275
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3020676 0.2107917 0.2193646 0.2677761
##
## Individual base composition of sequence 8
## The data from the GenBank:
##         a         c         g         t
## 0.2968127 0.2001992 0.2340637 0.2689243
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2739044 0.2031873 0.2480080 0.2749004
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3061019 0.1870903 0.2400403 0.2667675
##
## Individual base composition of sequence 9
## The data from the GenBank:
##         a         c         g         t
## 0.2998084 0.1992337 0.2346743 0.2662835
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2959770 0.1973180 0.2289272 0.2777778
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3040847 0.2239032 0.2289460 0.2430661
##
## Individual base composition of sequence 10
## The data from the GenBank:
##         a         c         g         t
## 0.3157534 0.2071918 0.2304795 0.2465753
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3318493 0.1972603 0.2301370 0.2407534
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3156833 0.2037317 0.2239032 0.2566818
```

```
##
## Individual base composition of sequence 11
## The data from the GenBank:
##         a         c         g         t
## 0.3121449 0.2041903 0.2325994 0.2510653
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2982955 0.1981534 0.2485795 0.2549716
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3136662 0.2107917 0.2208775 0.2546646
##
## Individual base composition of sequence 12
## The data from the GenBank:
##         a         c         g         t
## 0.3177408 0.2061677 0.2293832 0.2467082
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3226252 0.2086356 0.2362694 0.2324698
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3197176 0.1971760 0.2148260 0.2682804
##
## Individual base composition of sequence 13
## The data from the GenBank:
##         a         c         g         t
## 0.3187773 0.2034207 0.2241630 0.2536390
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3191412 0.1994178 0.2194323 0.2620087
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3050933 0.2107917 0.2284418 0.2556732
##
## Individual base composition of sequence 14
## The data from the GenBank:
##         a         c         g         t
## 0.3146384 0.2035273 0.2320988 0.2497354
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3126542 0.2065562 0.2273528 0.2534367
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3156833 0.1921331 0.2254160 0.2667675
##
## Individual base composition of sequence 15
## The data from the GenBank:
##         a         c         g         t
## 0.3196064 0.2033528 0.2255831 0.2514577
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3271470 0.2005095 0.2237991 0.2485444
## The simulated data in Q1.2:
##         a         c         g         t
```

```
## 0.3005547 0.2057489 0.2516389 0.2420575
##
## Individual base composition of sequence 16
## The data from the GenBank:
##         a         c         g         t
## 0.2921236 0.2123629 0.2382851 0.2572283
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2798805 0.1942231 0.2430279 0.2828685
## The simulated data in Q1.2:
##         a         c         g         t
## 0.2975290 0.2067574 0.2309632 0.2647504
##
## Individual base composition of sequence 17
## The data from the GenBank:
##         a         c         g         t
## 0.2902903 0.2052052 0.2412412 0.2632633
## The simulated data in Q1.1:
##     a     c     g     t
## 0.285 0.216 0.248 0.251
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3207262 0.2007060 0.2314675 0.2471004
##
## Individual base composition of sequence 18
## The data from the GenBank:
##         a         c         g         t
## 0.3203540 0.2042478 0.2269027 0.2484956
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3097345 0.2017699 0.2346903 0.2538053
## The simulated data in Q1.2:
##         a         c         g         t
## 0.2960161 0.2203732 0.2385275 0.2450832
##
## Individual base composition of sequence 19
## The data from the GenBank:
##         a         c         g         t
## 0.3170475 0.2003515 0.2274165 0.2551845
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3148473 0.1944542 0.2228852 0.2678133
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3015633 0.2087746 0.2420575 0.2476046
##
## Individual base composition of sequence 20
## The data from the GenBank:
##         a         c         g         t
## 0.2946429 0.2043651 0.2341270 0.2668651
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2996032 0.1944444 0.2400794 0.2658730
## The simulated data in Q1.2:
```

```
##         a         c         g         t
## 0.2980333 0.2178517 0.2370146 0.2471004
##
## Individual base composition of sequence 21
## The data from the GenBank:
##         a         c         g         t
## 0.3172554 0.1956522 0.2248641 0.2622283
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3023098 0.1793478 0.2302989 0.2880435
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3045890 0.2022189 0.2344932 0.2586989
##
## Individual base composition of sequence 22
## The data from the GenBank:
##         a         c         g         t
## 0.3219225 0.2074689 0.2285615 0.2420470
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3318355 0.2232976 0.2163844 0.2284825
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3081190 0.2037317 0.2334846 0.2546646
##
## Individual base composition of sequence 23
## The data from the GenBank:
##         a         c         g         t
## 0.3144044 0.1966759 0.2250693 0.2638504
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3026952 0.2073255 0.2301313 0.2598480
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3151790 0.1951589 0.2274332 0.2622289
##
## Individual base composition of sequence 24
## The data from the GenBank:
##     a     c     g     t
## 0.296 0.214 0.237 0.253
## The simulated data in Q1.1:
##     a     c     g     t
## 0.282 0.199 0.241 0.278
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3101362 0.2138174 0.2188603 0.2571861
##
## Individual base composition of sequence 25
## The data from the GenBank:
##         a         c         g         t
## 0.3175890 0.2054988 0.2284103 0.2485019
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3285160 0.1882270 0.2266479 0.2566091
```

```
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3192133 0.2087746 0.2370146 0.2349975
##
## Individual base composition of sequence 26
## The data from the GenBank:
##         a         c         g         t
## 0.3144453 0.2110934 0.2306464 0.2438148
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3028731 0.2194733 0.2250599 0.2525938
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3111447 0.2027231 0.2304589 0.2556732
##
## Individual base composition of sequence 27
## The data from the GenBank:
##         a         c         g         t
## 0.3094527 0.2059701 0.2218905 0.2626866
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3014925 0.1920398 0.2288557 0.2776119
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3202219 0.2047403 0.2259203 0.2491175
##
## Individual base composition of sequence 28
## The data from the GenBank:
##         a         c         g         t
## 0.3160083 0.2065142 0.2286902 0.2487872
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3275683 0.2175718 0.2251816 0.2296783
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3227433 0.1941503 0.2334846 0.2496218
##
## Individual base composition of sequence 29
## The data from the GenBank:
##         a         c         g         t
## 0.3207612 0.2020761 0.2280277 0.2491349
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3362158 0.1968177 0.2331373 0.2338291
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3141704 0.1961674 0.2304589 0.2592032
##
## Individual base composition of sequence 30
## The data from the GenBank:
##         a         c         g         t
## 0.2943396 0.2037736 0.2377358 0.2641509
## The simulated data in Q1.1:
##         a         c         g         t
```

```
## 0.3141509 0.2009434 0.2367925 0.2481132
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3015633 0.2037317 0.2344932 0.2602118
##
## Individual base composition of sequence 31
## The data from the GenBank:
##         a         c         g         t
## 0.3200585 0.2071611 0.2232371 0.2495433
## The simulated data in Q1.1:
##         a         c         g         t
## 0.3199416 0.2319211 0.2063550 0.2417823
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3242562 0.2012103 0.2228946 0.2516389
##
## Individual base composition of sequence 32
## The data from the GenBank:
##         a         c         g         t
## 0.2961117 0.2033898 0.2352941 0.2652044
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2871386 0.1924227 0.2562313 0.2642074
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3071104 0.1961674 0.2365103 0.2602118
##
## Individual base composition of sequence 33
## The data from the GenBank:
##         a         c         g         t
## 0.2932331 0.2030075 0.2406015 0.2631579
## The simulated data in Q1.1:
##         a         c         g         t
## 0.2685285 0.2223416 0.2588614 0.2502685
## The simulated data in Q1.2:
##         a         c         g         t
## 0.3045890 0.2047403 0.2430661 0.2476046
```

The GC contents are:

```
## [1] "In the data from the GenBank :  0.435954678206089"
```

```
## [1] "In the simulated data in Q1.1 :  0.437181936272637"
```

```
## [1] "In the simulated data in Q1.2 :  0.434878283592353"
```

AT contents are:

```
comp_true <- base.freq(lizard_seq_data)
cg_true <- sum(comp_true[c(1,4)])/sum(comp_true)
comp_s1 <- base.freq(simulated_seq_1)
cg_s1 <- sum(comp_s1[c(1,4)])/sum(comp_s1)
comp_s2 <- ape::base.freq(simulated_seq_2)
cg_s2 <- sum(comp_s2[c(1,4)])/sum(comp_s2)

paste("In the data from the GenBank : ",cg_true)
```

```
## [1] "In the data from the GenBank :  0.564045321793911"
```

```
paste("In the simulated data in Q1.1 : ",cg_s1)
```

```
## [1] "In the simulated data in Q1.1 :  0.562818063727363"
```

```
paste("In the simulated data in Q1.2 : ",cg_s2)
```

```
## [1] "In the simulated data in Q1.2 :  0.565121716407647"
```

The nucleotide sequences were transformed into amino acid sequences using the EMBOSS Transeq function, and saved to a fasta file:

```
amino_lizard_seq <- read.fasta("files/lizard_amino_seq.fasta")
amino_sim1_seq <- read.fasta("files/sim_aminio_seqs_1.fasta")
amino_sim2_seq <- read.fasta("files/sim_aminio_seqs_2.fasta")
```

WHAT DO WE COMMENT ABOUT THE STOP CODONS???

```
stop_code_count <- function(amino_acid_seq){

  stop_count = c()
  for (i in 1:length(amino_acid_seq)) {

    sequence = amino_acid_seq[[i]]
    stop <- sequence[which(sequence == "*")]
    stop_count[i] = length(stop)
  }
  return(stop_count)
}
df = data.frame(original_seq= stop_code_count(amino_lizard_seq),
           sim1 = stop_code_count(amino_sim1_seq),
           sim2 = stop_code_count(amino_sim2_seq)
          )

knitr::kable(df)
```

original_seq	sim1	sim2
16	28	39
0	51	41
78	54	38
33	16	45
0	37	52
38	23	35
79	61	31
38	14	37
41	13	44
78	60	41
0	58	45
81	73	39
0	50	41
82	73	35
0	71	36
0	16	30
19	21	38
85	63	33
87	53	36

original_seq	sim1	sim2
37	23	45
0	26	40
75	70	43
0	23	48
18	21	29
75	53	46
0	48	40
0	22	38
81	58	25
76	70	34
38	21	37
61	50	41
17	21	36
36	16	40

For some sequences the stop codons are more in true in sequence than in simulated sequences but there are some sequences for which the stop codon does not appear at all in the original sequence but it appears many times in the simulated sequences. It could be possible because the nucleotides forming the stop codon might have been drawn more in simulated sequences.

Q2.2

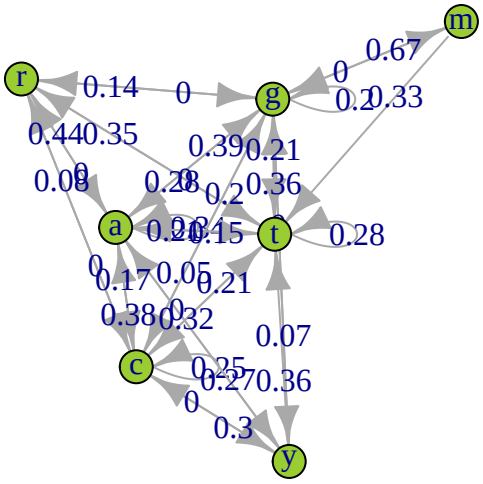
In this part we do markov chain analysis of the data sets. We combine all the sequences from each dataset to seperate lists, and use the *markovchainFit* function with bootstrapping to estimate the Markov models. This function fits the first order markov model to all three datasets.

```
lizard_seq <- unlist(as.character(lizard_seq_data),use.names = FALSE)
sim1_seq <- unlist(as.character(simulated_seq_1),use.names = FALSE)
sim2_seq <- unlist(as.character(simulated_seq_2),use.names = FALSE)

lizard_seq_mc <-  markovchainFit(lizard_seq,method = "bootstrap", nboot = 3,
                            parallel = TRUE)


sim1_seq_mc <-  markovchainFit(sim1_seq,method = "bootstrap",
                               nboot = 3,
                              parallel = TRUE)

sim2_seq_mc <-  markovchainFit(sim2_seq,method = "bootstrap", nboot = 3,
                               parallel = TRUE)
```

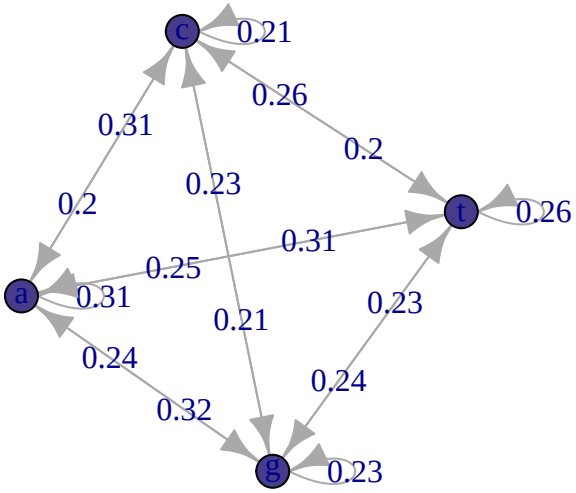
We see that the original data set from GenBank contains a small amount of additional "nucleotides": they are called "y", "m", and "r".

These are the nucleotides which we are not sure about where Y indicates unsurity in A or G while R shows ambiguity in C or T and M appears when there is no surity at all.

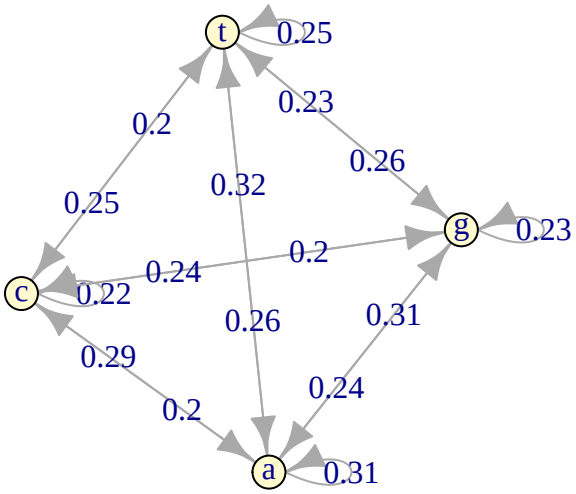
```
plot(lizard_seq_mc$estimate)
```




```
plot(sim1_seq_mc$estimate)
```




```
plot(sim2_seq_mc$estimate)
```




```
# Results from the zero order
    # barplot(zeroOrderFreq,col=1:4,
    #         main="Compositional bias of each nucleotide\nZero Order Markov Chain",
```

```
    #         xlab="Base",
    #         ylab="Base proportion")
```

In the first order Markov chains we assume that the nucleotide X_n only depends on a previous nucleotide, namely X_{n-1}. This is not a reasonable assumption, as DNA (or RNA) sequences contain the codons: the triplets of the nulceotides that determine what amino acids will be produced from it. Hence, it may be more reasonable that the most important dependences are between nucleotides in these triplets. In the lecture it was mentioned that there is some research supporting that a sixth order Markov Chain models perform better. That implies that the nth nucleotide depends on what were the previous 5 nucleotides.

Q2.3

In this part the three DNA collections of the sequences are aligned using the *msa* package. Furthermore the distances between the alignments are calculated. They are used to produce the heatmaps

```
lizard_seq_msa = msa("files/lizard_seqs.fasta", method = "ClustalW",type="dna")
```

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

```
sim1_seq_msa <- msa("files/simulated_seqs_1.fasta", type="dna")
```

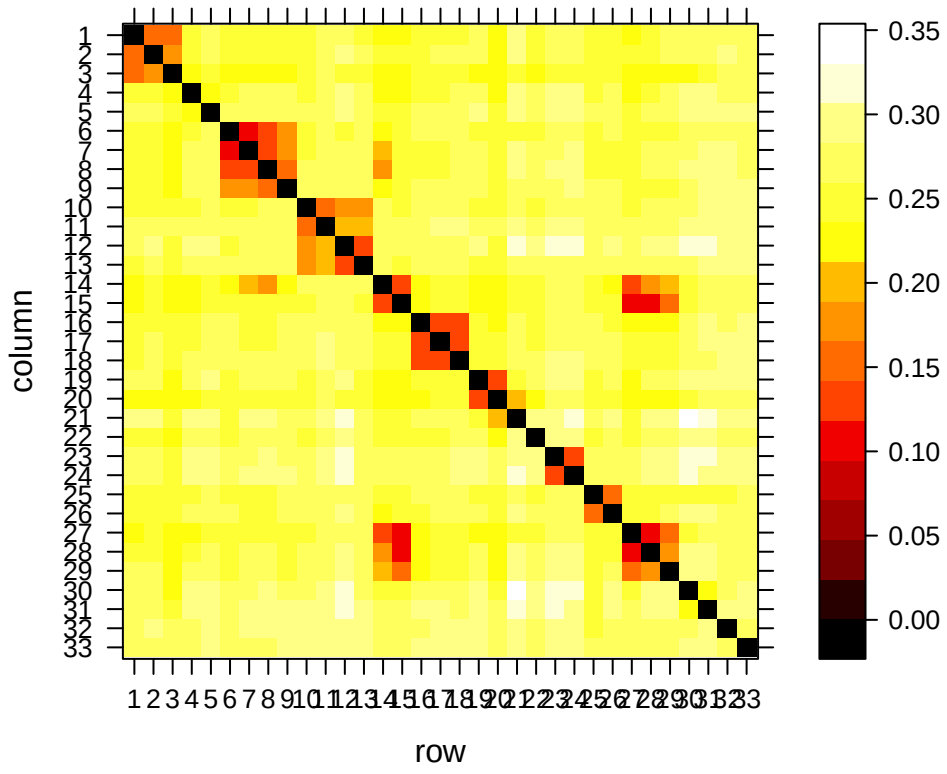
```
## use default substitution matrix
```

```
sim2_seq_msa <- msa("files/simulated_seqs_2.fasta", type="dna")
```

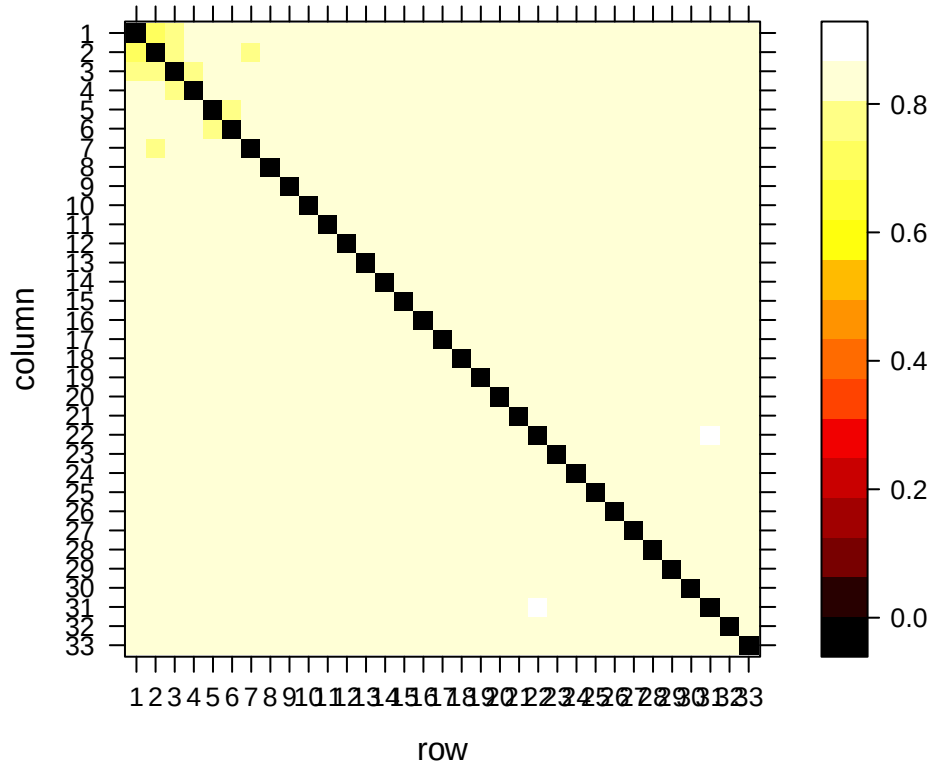
```
## use default substitution matrix
```

```
#multiple alignment object in an object of class align as defined by the bios2mds package
lizard_seq_align <- msaConvert(lizard_seq_msa, type="seqinr::alignment")
sim1_seq_align <- msaConvert(sim1_seq_msa, type="seqinr::alignment")
sim2_seq_align <- msaConvert(sim2_seq_msa, type="seqinr::alignment")


#distance matrix
lizard_seq_dist <- dist.alignment(lizard_seq_align, matrix = "identity")
sim1_seq_dist <- dist.alignment(sim1_seq_align, matrix = "identity")
sim2_seq_dist <- dist.alignment(sim2_seq_align, matrix = "identity")

lm = as.matrix(lizard_seq_dist)
sim1_m = as.matrix(sim1_seq_dist)
sim2_m = as.matrix(sim2_seq_dist)
```

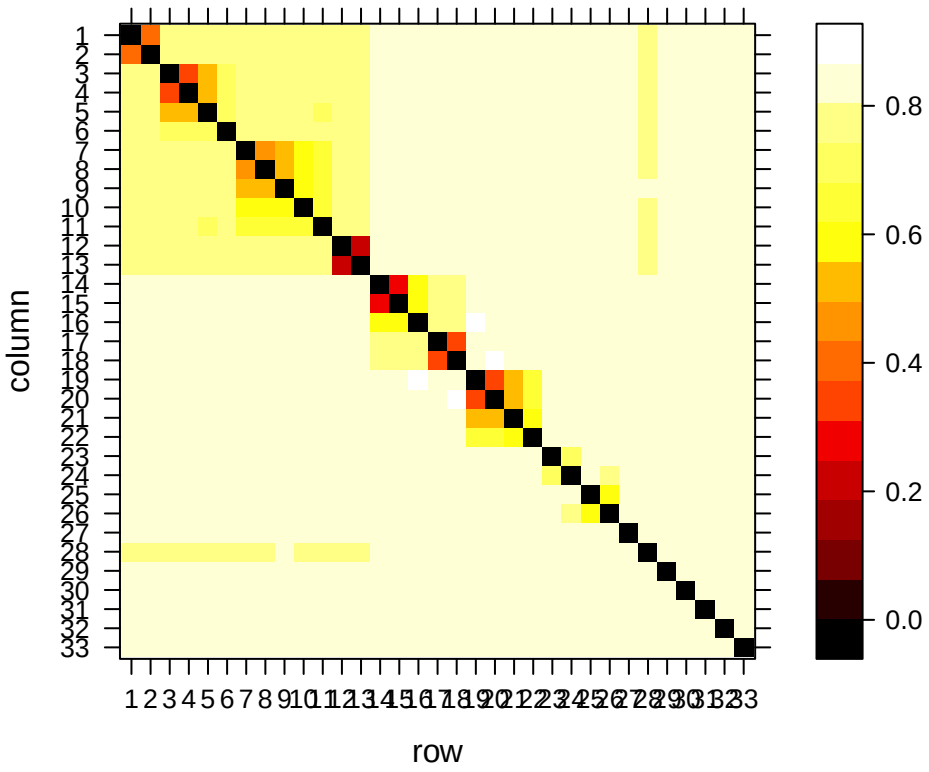
```
# heatmaps
new.palette=colorRampPalette(c("black","red","yellow","white"),space="rgb")
levelplot(lm[1:ncol(lm),ncol(lm):1],col.regions=new.palette(20))
```




```
levelplot(sim1_m[1:ncol(sim1_m),ncol(sim1_m):1],col.regions=new.palette(20))
```




```
levelplot(sim2_m[1:ncol(sim2_m),ncol(sim2_m):1],col.regions=new.palette(20))
```



The plots illustrate how much overlaying the sequences have in each dataset. The smaller the value (the darker the colour), the more nucleotides are matching between the corresponding sequences of the DNA. Therefore, the sequences from the data from GenBank has much more overlaying parts then the two randomly generated sequences. This result is expected as sequencing of a DNA produces "fragments" of the real DNA sequence, and those "fragments" do contain the same bits of the original DNA. The artificially simulated DNA sequences did not actually go through a process of the sequencing, hence they are way less likely to randomly contain long matching parts in their sequences.

Question 3: Phylogeny reconstruction

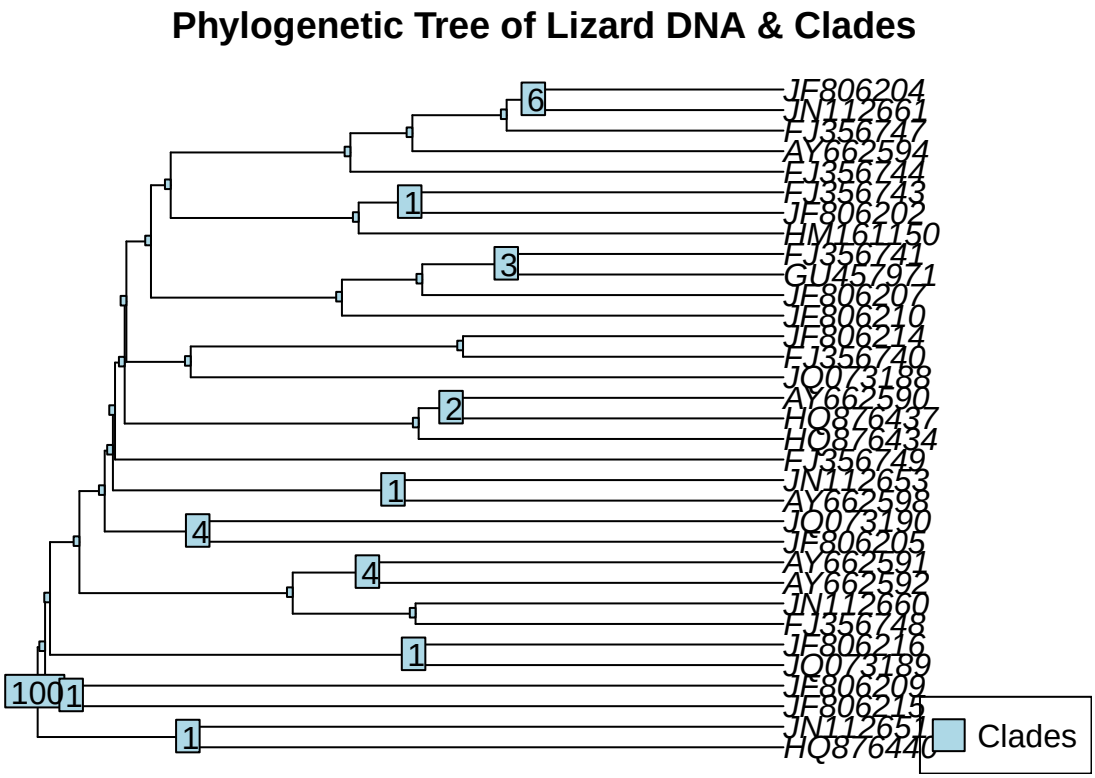
Q3.1

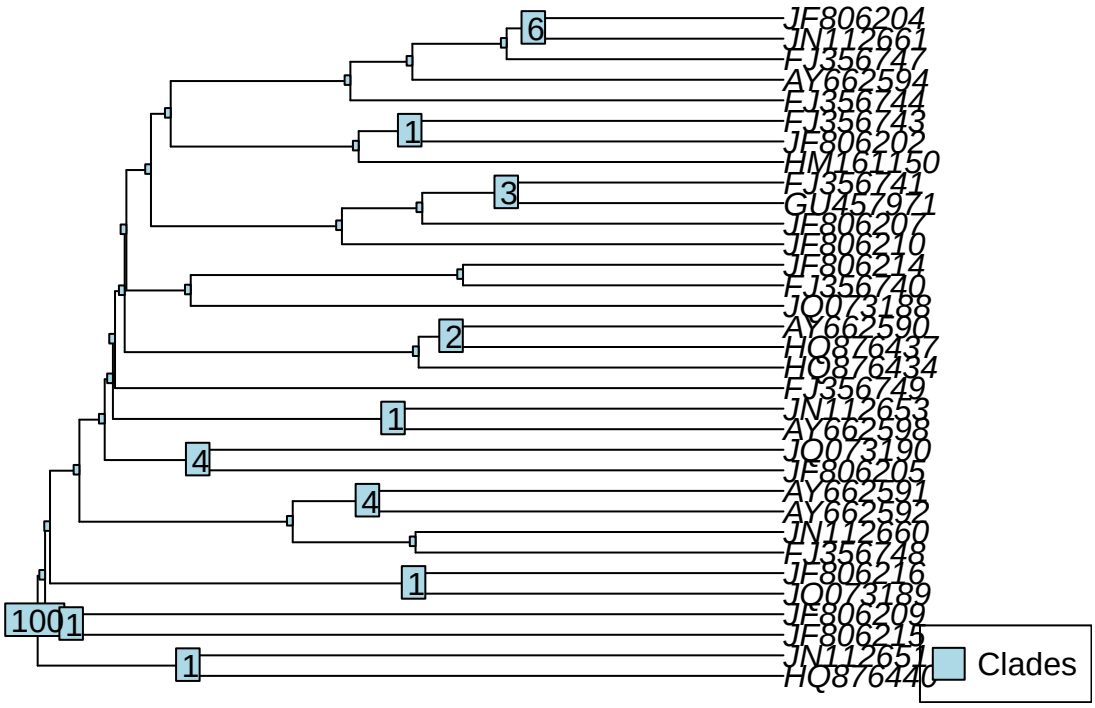
```
library(phangorn)


upgma_tree <- function(x){
  tree <- upgma(x)
  tree$tip.label <- lizards_accession_numbers
  return(tree)
}


lizard_seq_tree <- upgma_tree(lizard_seq_dist)
#plot(lizard_seq_tree, main="Phylogenetic Tree of Lizard DNA")
```

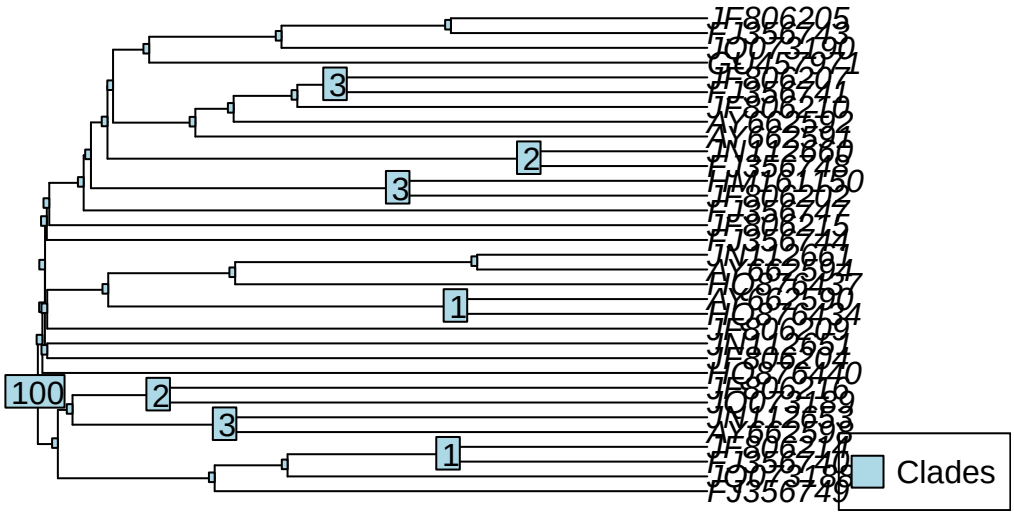
```
sim1_seq_tree <- upgma_tree(sim1_seq_dist)
#plot(sim1_seq_tree, main="Phylogenetic Tree of Lizard DNA sim1")


sim2_seq_tree <- upgma_tree(sim2_seq_dist)
#plot(sim2_seq_tree, main="Phylogenetic Tree of Lizard DNAs sim2")


boot_lizard <-  boot.phylo(lizard_seq_tree, lm, FUN=upgma_tree, trees  = TRUE,quiet = TRUE)
sim1_boot     <- boot.phylo(sim1_seq_tree, sim1_m, FUN=upgma_tree, trees  = TRUE,quiet = TRUE)
sim2_boot     <- boot.phylo(sim2_seq_tree, sim2_m, FUN=upgma_tree, trees  = TRUE,quiet = TRUE)
```

Extracting Clades

```
lizard_clade = prop.clades(lizard_seq_tree,boot_lizard$trees,rooted = TRUE)
sim1_clade = prop.clades(sim1_seq_tree,sim1_boot$trees,rooted = TRUE)
sim2_clade = prop.clades(sim2_seq_tree,sim2_boot$trees,rooted = TRUE)
```







Q3.2

```
library(phangorn)
path_diff_sim1_lizard <- path.dist(sim1_seq_tree,lizard_seq_tree)
path_diff_sim2_lizard <- path.dist(sim2_seq_tree,lizard_seq_tree)
path_diff_sim1_sim2 <- path.dist(sim1_seq_tree,sim2_seq_tree)
```

```
felenstein_diff_sim1_lizard <- KF.dist(sim1_seq_tree,lizard_seq_tree)
felenstein_diff_sim2_lizard <- KF.dist(sim2_seq_tree,lizard_seq_tree)
felenstein_diff_sim1_sim2 <- KF.dist(sim1_seq_tree,sim2_seq_tree)

result = data.frame(path_difference=c(path_diff_sim1_lizard,
                                      path_diff_sim2_lizard,
                                      path_diff_sim1_sim2),
                    felenstein_distance=c(felenstein_diff_sim1_lizard,
                                          felenstein_diff_sim2_lizard,
                                          felenstein_diff_sim1_sim2)
                    )
rownames(result) = c("sim1 vs lizard","sim2 vs lizard","sim1 vs sim2")
knitr::kable(result)
```

	path_difference	felenstein_distance
sim1 vs lizard	98.40732	1.906936
sim2 vs lizard	115.75837	1.320883
sim1 vs sim2	84.62860	1.090855

Geodesic Distance (Distance between edeges)

```
library(distory)
#distance between edeges using geodesic
geodesic_dist = dist.multiPhylo(list(lizard_seq_tree,sim1_seq_tree,sim2_seq_tree), method = "geodesic")
geodesic_dist = as.matrix(geodesic_dist)
geodesic_dist
```

```
##          1        2        3
## 1 0.000000 1.909169 1.352303
## 2 1.909169 0.000000 1.101143
## 3 1.352303 1.101143 0.000000
```

It can be seen from the path distance, Kuhner Felensteins distance and edge distance that tree from simulation 1 differs more from expected lizard tree than simulation 2. It can also be observed that both simulations are closer to each other that is in terms of distance.

```
comp1 <- comparePhylo(lizard_seq_tree,sim1_seq_tree)
comp2 <- comparePhylo(lizard_seq_tree,sim2_seq_tree)
comp12 <- comparePhylo(sim1_seq_tree,sim2_seq_tree)
```

```
comp1$messages
```

```
## [1] "=> Comparing lizard_seq_tree with sim1_seq_tree."
## [2] "Both trees have the same number of tips: 33."
## [3] "Both trees have the same tip labels."
## [4] "Both trees have the same number of nodes: 32."
## [5] "Both trees are rooted."
## [6] "Both trees are ultrametric."
## [7] "30 clades in lizard_seq_tree not in sim1_seq_tree."
## [8] "30 clades in sim1_seq_tree not in lizard_seq_tree."
## [9] "Branching times of clades in common between both trees: see ..$BT\n(node number in parentheses)
```

```
comp2$messages
```

```
## [1] "=> Comparing lizard_seq_tree with sim2_seq_tree."
## [2] "Both trees have the same number of tips: 33."
## [3] "Both trees have the same tip labels."
## [4] "Both trees have the same number of nodes: 32."
## [5] "Both trees are rooted."
## [6] "Both trees are ultrametric."
## [7] "26 clades in lizard_seq_tree not in sim2_seq_tree."
## [8] "26 clades in sim2_seq_tree not in lizard_seq_tree."
## [9] "Branching times of clades in common between both trees: see ..$BT\n(node number in parentheses)
```

```
comp12$messages
```

```
## [1] "=> Comparing sim1_seq_tree with sim2_seq_tree."
## [2] "Both trees have the same number of tips: 33."
## [3] "Both trees have the same tip labels."
## [4] "Both trees have the same number of nodes: 32."
## [5] "Both trees are rooted."
## [6] "Both trees are ultrametric."
## [7] "28 clades in sim1_seq_tree not in sim2_seq_tree."
## [8] "28 clades in sim2_seq_tree not in sim1_seq_tree."
## [9] "Branching times of clades in common between both trees: see ..$BT\n(node number in parentheses)
```

The comparision in terms of tips,nodes,label and clades also implies that simulation 2 is more similar to lizard tree as it has 27 different clades from expected lizard tree while simulation 1 has 28 different clades