MARKOW MODELS

Ricard Garcia & Antonio Arcas

2023-11-16

```
library(seqinr)
setwd("/home/anarpo22/Documents/Bioinformatics2/statistics/Homework_9_markovmodels")

sars_cov <- read.fasta(file="SARSCov2.fasta")
raTG13 <- read.fasta(file="RaTG13.fasta")
```

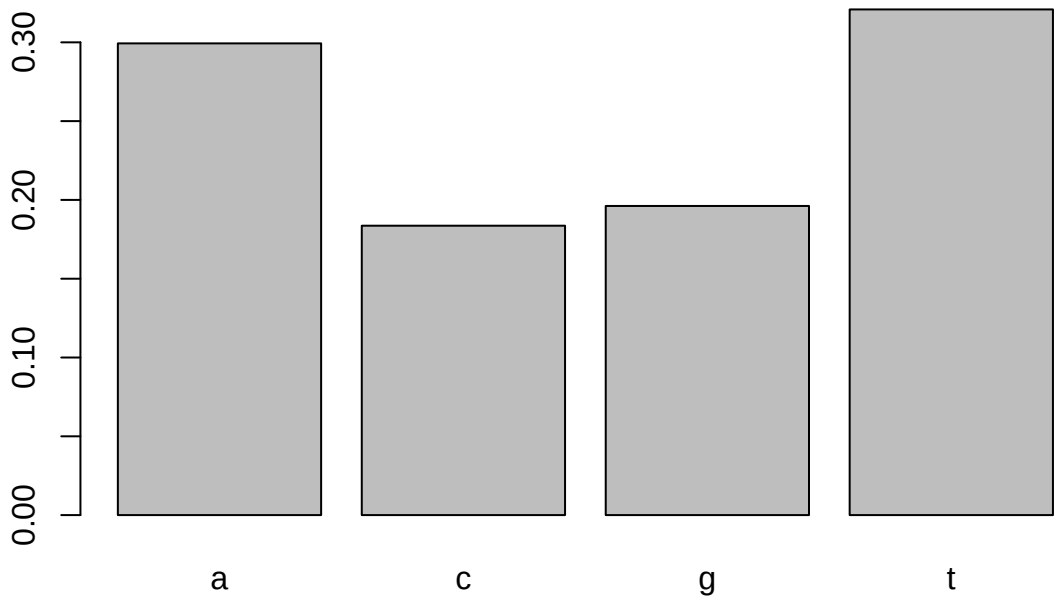
1- Download the genome of SARS-Cov-2 sequenced in Catalonia in 15/03/2020, accession number MT359865, and the genome of the virus RaTG13 of the bats Rhinolophus afinis, accession number MN996532.

```
sars<-sars_cov[[1]]
probability_nuc_sars <- table(sars)/length(sars)
probability_nuc_sars
```

2- Calculate the frequency of each nucleotide for each virus and draw Bar Plots. Which nucleotide is more frequent? Calculate the GC content and the AT content. Note that you have fitted the sequences to multinomial models: compute their BIC statistics

```
## sars
##         a         c         g         t
## 0.2993309 0.1836400 0.1961526 0.3208765
```

```
barplot(probability_nuc_sars)
```




```
gc_content_sars <- GC(sars)
cat("GC content is:",gc_content_sars)
```

```
## GC content is: 0.3797926
```

```
at_content_sars <-1 - gc_content_sars
cat("AT content is:",at_content_sars)
```

```
## AT content is: 0.6202074
```

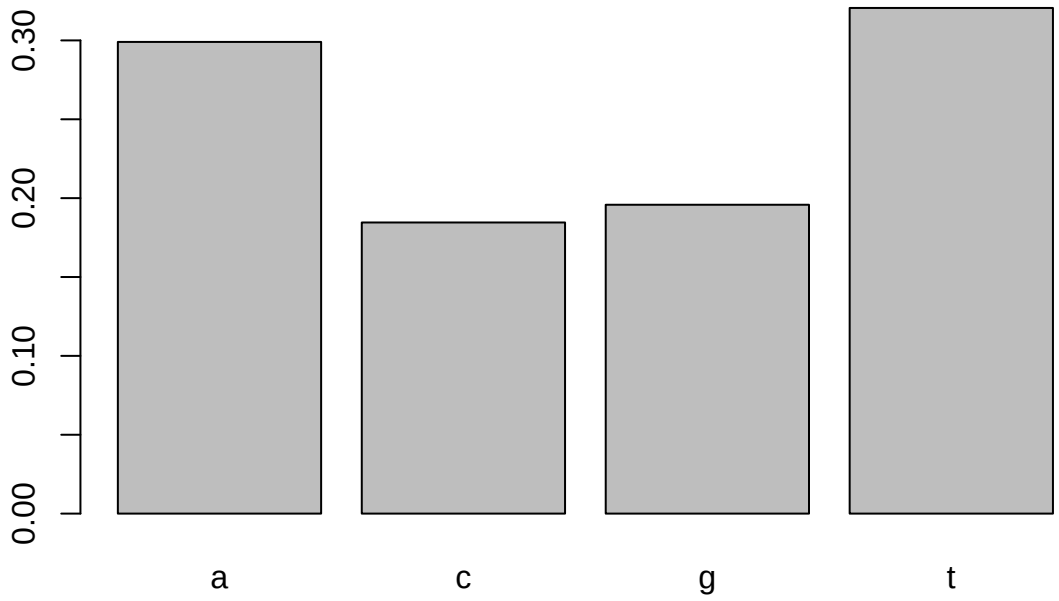
```
#BIC of sars multinomial model:
n=length(sars)
par=3
c=count(sars,1)
p=count(sars,1, freq=T)
BIC_sars_0 <- -2*sum(c*log(p)) + par*log(n)
BIC_sars_0
```

```
## [1] 81124.23
```

```
raTG<- raTG13[[1]]
probability_nuc_raTG13 <- table(raTG)/length(raTG)
probability_nuc_raTG13
```

```
## raTG
##         a         c         g         t
## 0.2990454 0.1845587 0.1958131 0.3205828
```

```
barplot(probability_nuc_raTG13)
```




```
gc_content_raTG <- GC(raTG)
cat("GC content is:",gc_content_raTG)
```

```
## GC content is: 0.3803718
```

```
at_content_raTG <-1 - gc_content_raTG
cat("AT content is:",at_content_raTG)
```

```
## AT content is: 0.6196282
```

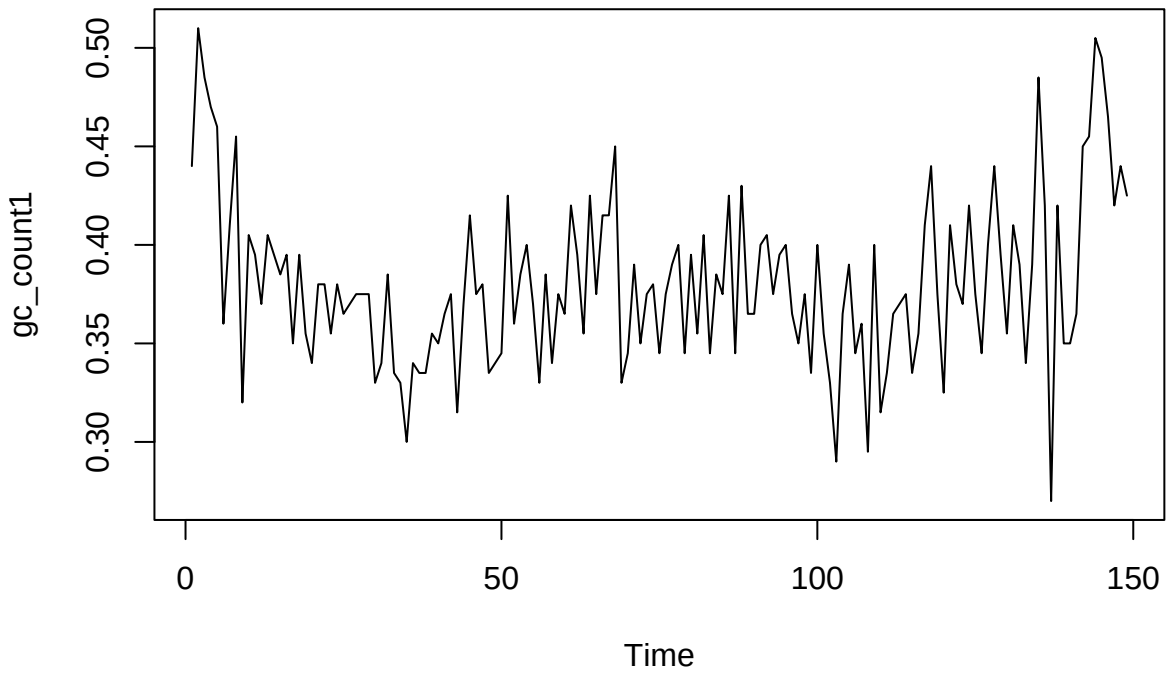
```
#BIC of raTG multinomial model:
n=length(raTG)
par=3
c=count(raTG,1)
p=count(raTG,1, freq=T)
BIC_raTG_0 <- -2*sum(c*log(p)) + par*log(n)
BIC_raTG_0
```

```
## [1] 81048.55
```

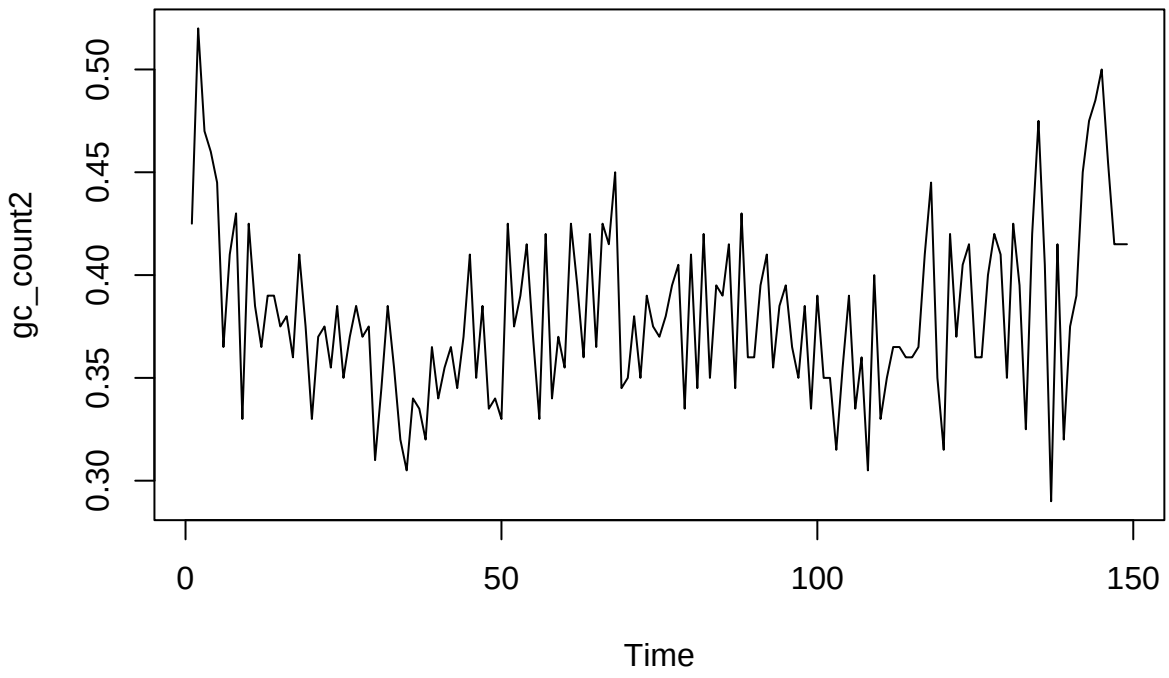
3- Do a sliding window analysis of the GC content for each virus, that is, to study the variation in GC content within the genome sequence: - calculate the GC content of chunks with length 200 (window size=200) - plot the resulting frequencies

```
k<-length(sars)%/%200
#we will loop 149 times
gc_count1<-numeric(k)
```

```
for (x in 1:k){
  a= 200*(x-1)+1
  b=a+199
  gc_count1[x]=GC(sars[a:b])
}
#Plot the frequencies
ts.plot(gc_count1)
```




```
r<-length(raTG)%/%200
#we will loop 149 times
gc_count2<-numeric(r)
for (p in 1:r){
  a= 200*(p-1)+1
  b=a+199
  gc_count2[p]=GC(raTG[a:b])
}
#Plot the frequencies
ts.plot(gc_count2)
```



Cannot execute function lines(gc_count1) because rises an error.

```
dinucleotides_sars <- count(sars,2)
dinucleotides_raTG <- count(raTG,2)

zscore(sars,modele="base")
```

4- Compute the number of occurrences of the dinucleotides for each virus. Which are over or under-represented?

```
##
##          aa          ac          ag          at          ca          cc
##   5.5152528  12.3968156  -0.4745183 -15.3083538  14.3866854  -4.6673136
##          cg          ct          ga          gc          gg          gt
## -24.0271924  10.2013541  -4.5777423   3.4369545  -2.0544844   3.3939744
##          ta          tc          tg          tt
## -13.4684933 -11.2083086  22.1481892   3.6783485
```

```
  #UNDEREPRESENTED --> AT, GA, CG, CC, GG, TA, TC
  #OVEREPRESENTED --> AA, AC, CA, GC, CT, GT, TG, TT
zscore(raTG,modele="base")
```

```
##
##          aa          ac          ag          at          ca          cc
##   5.1138578  12.6861592  -0.6103881 -15.0619656  13.6311485  -5.6103192
##          cg          ct          ga          gc          gg          gt
## -23.9823404  11.6887220  -3.9871389   3.0868662  -1.7893802   2.8731090
```

```
##          ta          tc          tg          tt
## -12.9492126 -10.4325505  22.0614148   2.6254669
```

```
  #UNDEREPRESENTED --> AT, GA, CG, CC, TA, TC
  #OVEREPRESENTED --> AA, AC,CA,GC,CT,GT,TG,TT
```

```
#TRANSITION PROBABILITY MATRICES SARS
a=count(sars,2)
c=matrix(a,4,4,byrow = TRUE)
M<- c[,]/(c[,1]+c[,2]+c[,3]+c[,4])
LIMITING_SARS <- M%*%M%*%M%*%M%*%M%*%M%*%M%*%M%*%M%*%M%*%M%*%M
LIMITING_SARS
```

5- Fit both sequences to Markov chain models. Estimate their transition probability matrices. Compute the BIC of the models and find the steady-state (or limiting) distributions.

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.2993074 0.1836462 0.1961591 0.3208873
## [2,] 0.2993074 0.1836462 0.1961591 0.3208873
## [3,] 0.2993074 0.1836462 0.1961591 0.3208873
## [4,] 0.2993074 0.1836462 0.1961591 0.3208873
```

```
BIC_sars_1 <- -2*sum(c*log(M)+12*log(length(sars)-1))
BIC_sars_1
```

```
## [1] 75808.79
```

```
#TRANSITION PROBABILITY MATRICES raTG
b=count(raTG,2)
d=matrix(b,4,4,byrow = TRUE)
N <- d[,]/(d[,1]+d[,2]+d[,3]+d[,4])
LIMITING_raGT <- N%*%N%*%N%*%N%*%N%*%N%*%N%*%N%*%N%*%N%*%N%*%N
LIMITING_raGT
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.2990538 0.1845338 0.1958229 0.3205895
## [2,] 0.2990538 0.1845338 0.1958229 0.3205895
## [3,] 0.2990538 0.1845338 0.1958229 0.3205895
## [4,] 0.2990538 0.1845338 0.1958229 0.3205895
```

```
BIC_raTG_1 <- -2*sum(d*log(N)+12*log(length(raTG)-1))
BIC_raTG_1
```

```
## [1] 75753.8
```

```
#transition matrix for sars
xt=factor(sars[3:n])
xt_1=factor(sars[2:(n-1)])
xt_2=factor(sars[1:(n-2)])
cts=table(xt_1:xt_2,xt)
transition_matrix_sars2 <- cts[,]/(cts[,1]+cts[,2]+cts[,3]+cts[,4])
#BIC
n=length(sars)-2; par=48
c=table(xt_1:xt_2,xt)
p=c[,]/(c[,1]+c[,2]+c[,3]+c[,4])
```

```
BIC_sars_2=-2*sum(c*log(p)) + par*log(n)
BIC_sars_2
```

6- Fit both sequences to second order Markov chain models, compute their BIC and compare the results with those obtained in 5. Which models are better?

```
## [1] 79887.33
```

```
#transition matrix for raTG
xt=factor(raTG[3:n])
xt_1=factor(raTG[2:(n-1)])
xt_2=factor(raTG[1:(n-2)])
cts2=table(xt_1:xt_2,xt)
transition_matrix_raTG2 <- cts[,]/(cts[,1]+cts[,2]+cts[,3]+cts[,4])
#BIC
n=length(raTG)-2; par=48
c=table(xt_1:xt_2,xt)
p=c[,]/(c[,1]+c[,2]+c[,3]+c[,4])
BIC_raTG_2=-2*sum(c*log(p)) + par*log(n)
BIC_raTG_2
```

```
## [1] 79889.67
```

```
#comparison:
#sars markov1 vs sars markov2:
BIC_sars_1
```

```
## [1] 75808.79
```

```
BIC_sars_2
```

```
## [1] 79887.33
```

```
#raTG markov1 vs raTG markov2:
BIC_raTG_1
```

```
## [1] 75753.8
```

```
BIC_raTG_2
```

```
## [1] 79889.67
```

```
#The simple markov models are best because they have a lower BIC score.
```

We have to choose the model with lowest BIC.

```
raw_sequence <- c("g","t","g","t","g","c","t","c","a","g","t","t","g","a","a","a","a","t","c","c","c","t
result <- c()
legend <- c('a'=1,'c'=2,'g'=3,'t'=4)
n <- length(raw_sequence)-1
for (i in seq(n)){
  from <- legend[raw_sequence[i]]
  to <- legend[raw_sequence[i+1]]
  result[i] <- log(M[from,to]/N[from,to])
}
sum(result)
```

7- Consider the following fragment of a DNA sequence: "g" "t" "g" "t" "g" "c" "t" "c" "a" "g" "t" "t" "g" "a" "a" "a" "a" "t" "c" "c" "c" "t" "t" "g" "t" "c" "a" "a" "c" "a" "t" "c"

“t” “a” “g” “g” “t” “c” “t” “t” “a” “t” “c” “a” “c” “a” “t” “c” “a” “c” “a”. Decide using the log-likelihood method to which virus this sequence belongs, SARS-Cov_2 or RaTG13.

```
## [1] -0.04506527
```

It’s more probable that the sequence belongs to raTG13, because the log-likelihood is negative.