Exercise 2: Balin Lin, Haoyuan Li

Code

Hide

```
install.packages("seqinr")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
library(seqinr)
```

Hide

```
aedesaegypi=read.fasta(file="/Users/mac06/Code/MBD/Excercise1_2/aedesaegypti.fasta")
```

Hide

```
aedesaegypi=aedesaegypi[[1]]
```

Hide

```
length(aedesaegypi)
```

```
[1] 310827022
```

Hide

```
# table(aedesaegypi)/length(aedesaegypi)
```

1. Do a sliding window analysis of the GC content, that is, to study the variation in GC content within the genome sequence:

a. calculate the GC content of chunks with length 200 and length 1000 (window sizes=200 and 1000)

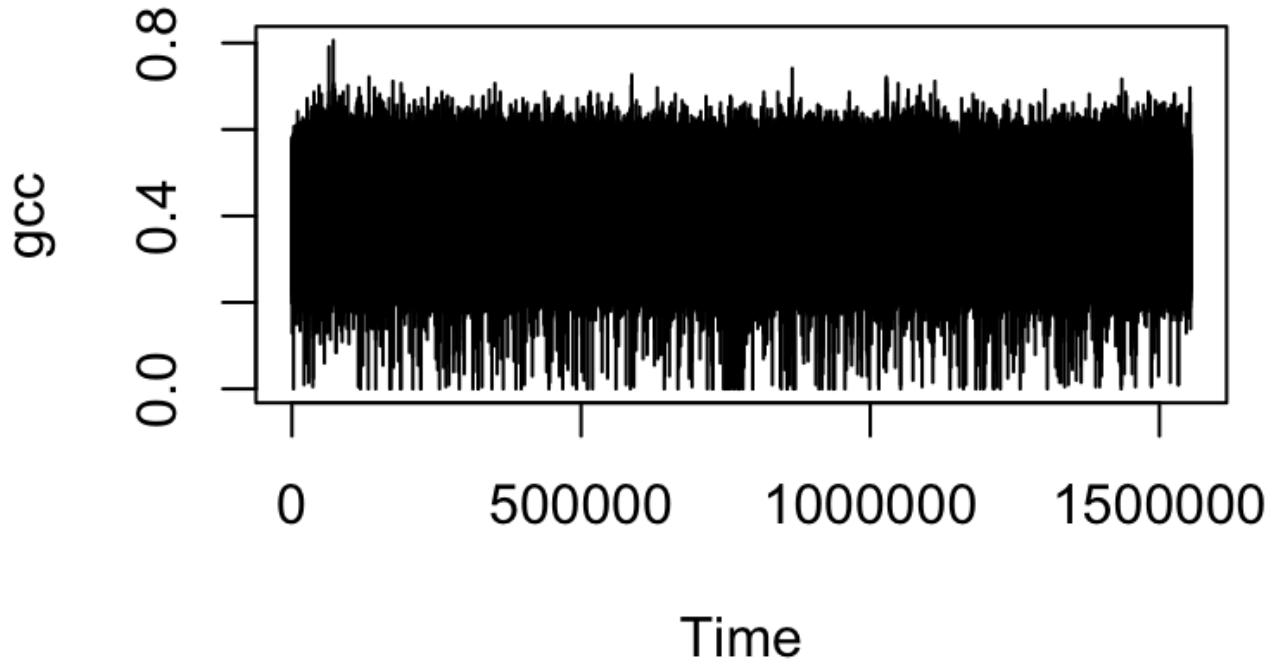
Hide

```
n=length(aedesaegypi); m=200; k=n%/%m
gcc=numeric(k)
id200 = 0
m200 = 0
id1000 = 0
m1000 = 0

for(i in 1:k){
  a=(i-1)*m+1; b=a+m
  gcc[i]=GC(aedesaegypi[a:b])
  if(m200 < gcc[i]){
    id200 = i
    m200 = gcc[i]
  }
}
```

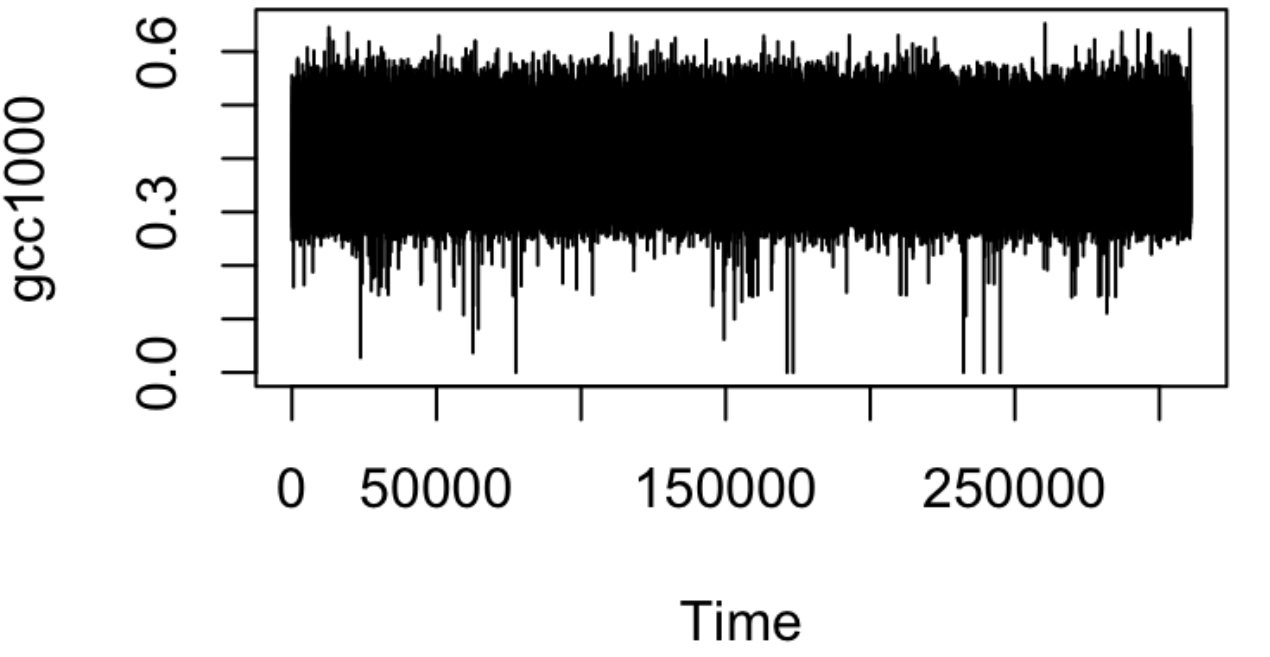
Hide

```
ts.plot(gcc)
```



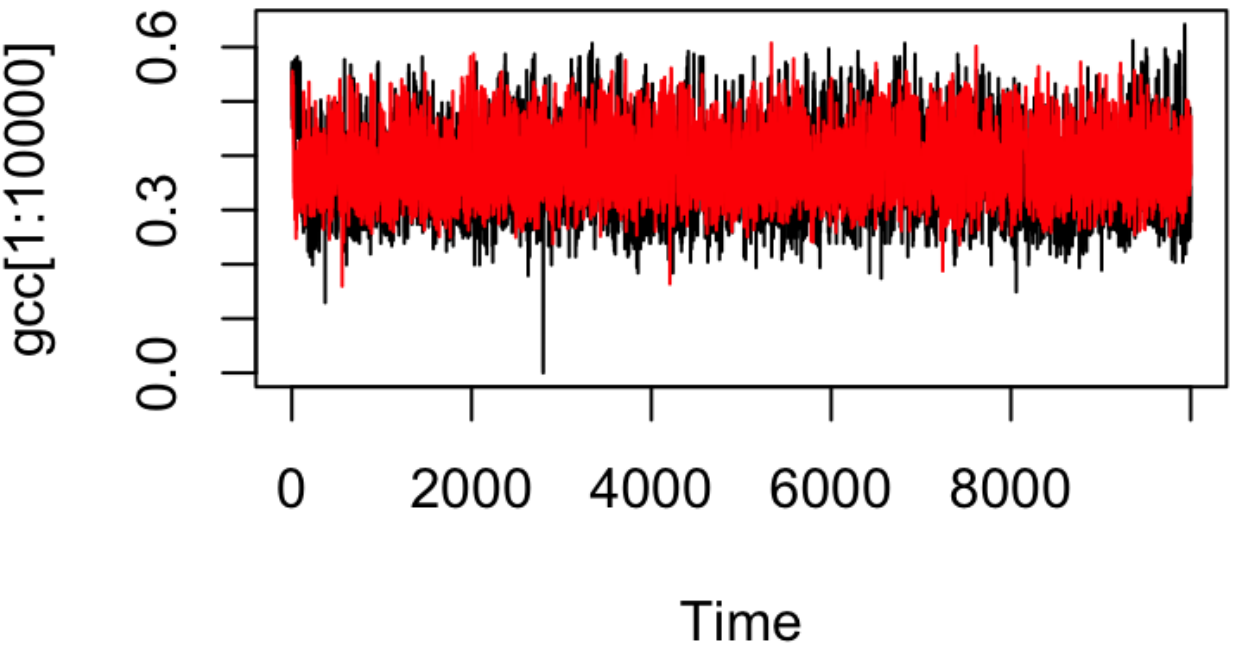
Hide

```
ts.plot(gcc1000)
```



Hide

```
ts.plot(gcc[1:10000])
lines(gcc1000[1:10000],col="red")
```



b. find the maximum GC content for each window size and plot the GC content around the point (± 1000) where the maximum is reached.

Hide

```
print("windows size 200")
```

```
[1] "windows size 200"
```

Hide

```
print("idx:")
```

```
[1] "idx:"
```

Hide

```
print(id200)
```

```
[1] 71611
```

Hide

```
print("max:")
```

```
[1] "max:"
```

Hide

```
print(m200)
```

```
[1] 0.8059701
```

Hide

```
print("windows size 1000")
```

```
[1] "windows size 1000"
```

Hide

```
print("idx:")
```

```
[1] "idx:"
```

Hide

```
print(id1000)
```

```
[1] 260435
```

Hide

```
print("max:")
```

```
[1] "max:"
```

Hide

```
print(m1000)
```

```
[1] 0.6523477
```

2. Fit the genome sequence to a Multinomial and to a Markov chain model. Estimate its corresponding probabilities and transition probability matrix. Compute also the BIC and decide which model is better.

Hide

```
count_mm = count(aedesaegypi,1,freq=T)
```

Hide

```
print(count_mm)
```

```
        a         c         g         t
0.3068436 0.1932460 0.1931294 0.3067811
```

Hide

```
a = count(aedesaegypi, 2)
```

The smaller the value of the statistic BIC the better the fit of the model.

Hide

```
n=length(aedesaegypi); par=3
c=count(aedesaegypi,1)
p=count(aedesaegypi,1,freq=T)
BIC=-2*sum(c*log(p)) + par*log(n)
print("Multinomial")
```

```
[1] "Multinomial"
```

Hide

```
print(BIC)
```

```
[1] 845583590
```

Hide

```
n=length(aedesaegypi)-1; par=12
a=count(aedesaegypi,2)
c = matrix(a, 4, 4, byrow=TRUE, dimnames = list(c("A","C","G","T"), c("A","C","G","T"
)))
p=c[,]/(c[,1]+c[,2]+c[,3]+c[,4])
BIC=-2*sum(c*log(p)) + par*log(n)
print("Markov chain")
```

```
[1] "Markov chain"
```

Hide

```
print(BIC)
```

```
[1] 838882703
```

Markov model is better!

3. Consider again sliding windows of length 50 and calculate the GC content and the presence/absence of the trinucleotid “aaa”. Is there any relationship between the presence of “aaa” and the GC content? What is the probability of “aaa” for a chunk with a GC content of 0.51 ? Plot the estimated probability of “aaa” against the GC content.

Hide

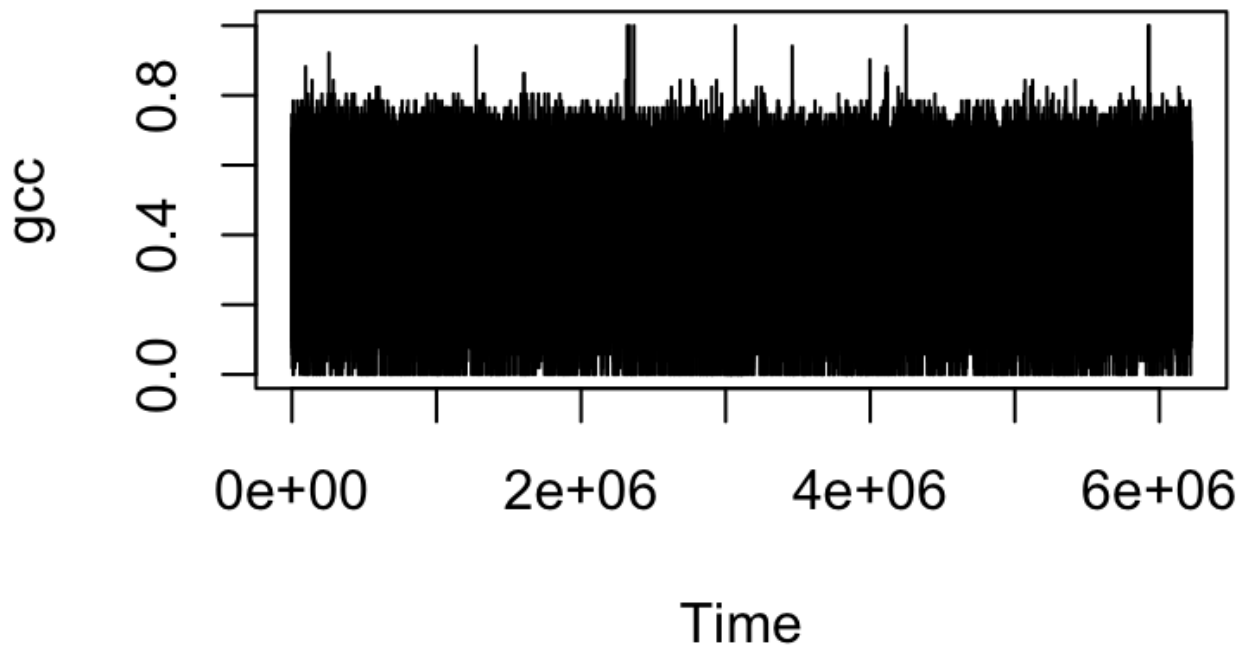
```
vec <- c()
num = 0

n=length(aedesaegypi); m=50; k=n%/%m
gcc=numeric(k)

for(i in 1:k){
  a=(i-1)*m+1; b=a+m
  gcc[i]=GC(aedesaegypi[a:b])
  num = num + 1
  if(!is.na(gcc[i]) && round(gcc[i], digits = 2) == 0.51){
    aaa = count(aedesaegypi[a:b], 3, freq=T)['aaa']
    vec <- c(vec, aaa)
  }
}
```

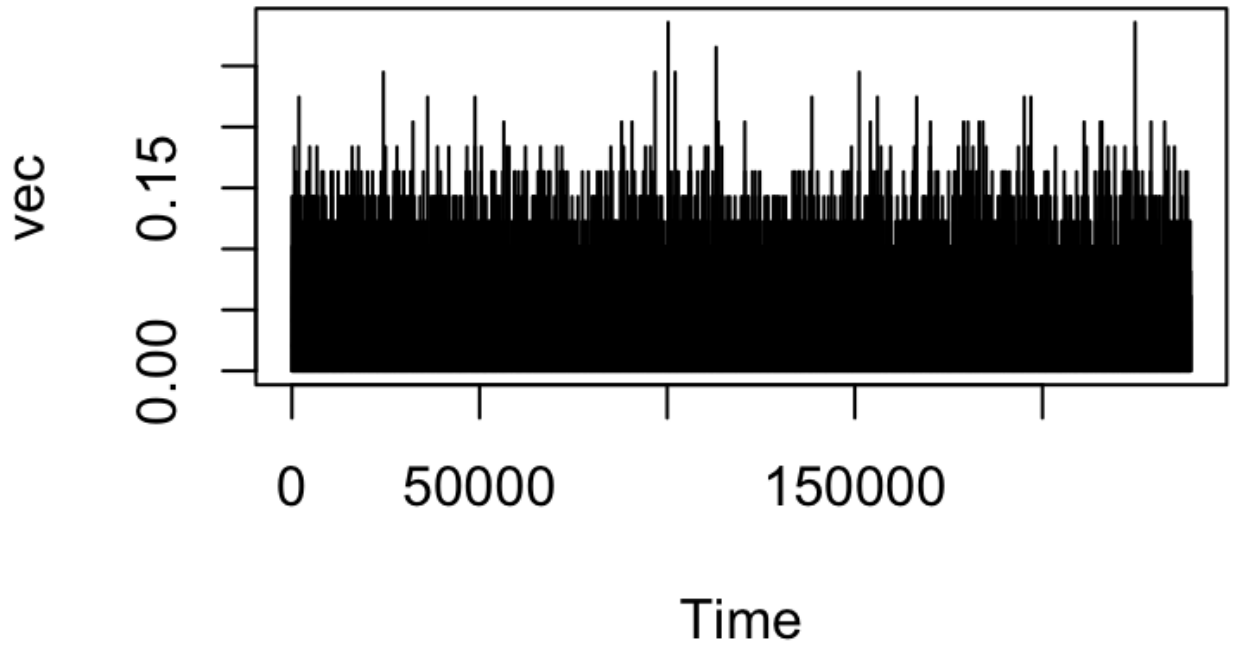
Hide

```
ts.plot(gcc)
```



Hide

```
ts.plot(vec)
```



4. Consider again sliding windows of length 50 and calculate the GC content and the counts of the trinucleotid “aaa”. Is there any relationship between the mean of the counts of “aaa” and the GC content? What is the predicted mean of the counts of “aaa” for a chunk with a GC content of 0.4 ? Plot the estimated mean of the counts of “aaa” against the GC content.

Hide

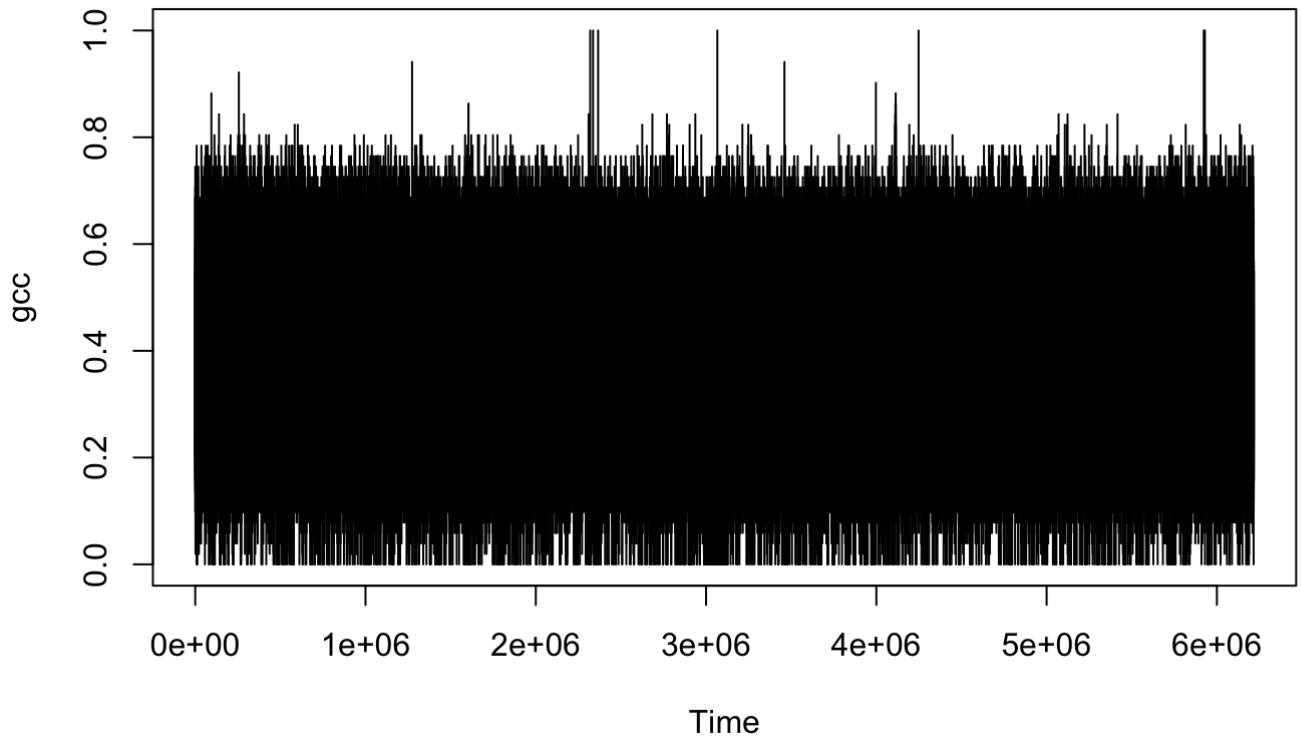
```
vec <- c()
num = 0

n=length(aedesaegypi); m=50; k=n%/%m
gcc=numeric(k)

for(i in 1:k){
  a=(i-1)*m+1; b=a+m
  gcc[i]=GC(aedesaegypi[a:b])
  num = num + 1
  if(!is.na(gcc[i]) && round(gcc[i], digits = 2) == 0.4){
    aaa = count(aedesaegypi[a:b], 3, freq=T)['aaa']
    vec <- c(vec, aaa)
  }
}
```

Hide

```
ts.plot(gcc)
```



Hide

```
ts.plot(vec)
```