# Data Bases for Bioinformatics

Jerusalem College of Technology

R Programming

# Statistical Analysis with R

1. Finding a gene sequence. Retrieving and saving its FASTA file.

> library("rentrez")

Warning message:

package ‘rentrez’ was built under R version 3.4.3

> seq<-entrez\_fetch(db="nuccore",id=508550,rettype="fasta")

> cat(seq,sep="\n")

>L34291.1 Pisum sativum ribulose-1,5 bisphosphate carboxylase large subunit N-methyltransferase (rbcMT) mRNA, complete cds

ACAACACAAAAGAAAAGCGTATTATCACAAAACAAAACCAAGAACTAGAAACCAGAAAATGGCTACTATC

TTTTCCGGAGGTTCAGTTTCTCCCTTTCTTTTTCACACCAACAAGGGTACATCTTTTACACCCAAAGCTC

CAATTCTTCATCTCAAGAGATCTTTCTCTGCAAAATCAGTAGCCTCTGTAGGAACCGAACCATCACTGTC

TCCAGCAGTTCAAACCTTCTGGAAGTGGCTACAGGAAGAAGGTGTCATCACTGCAAAGACCCCAGTGAAA

GCTAGTGTGGTCACAGAAGGTTTAGGATTGGTTGCACTTAAGGACATTTCTAGGAATGATGTTATTTTGC

AGGTACCAAAAAGGCTGTGGATAAATCCAGATGCAGTTGCAGCTTCAGAGATTGGGAGAGTGTGCAGTGA

GTTGAAGCCATGGTTGTCTGTTATACTCTTTCTTATAAGAGAGAGGTCAAGGGAAGATTCTGTTTGGAAG

CACTATTTTGGTATTCTGCCACAGGAAACTGATTCTACTATATATTGGTCAGAGGAAGAGCTTCAAGAGC

TTCAAGGTTCTCAACTTTTGAAAACAACAGTGTCTGTGAAAGAATATGTGAAGAATGAATGTTTGAAACT

AGAACAAGAAATCATTCTCCCTAATAAGCGGCTTTTTCCGGATCCTGTGACGCTGGATGACTTCTTTTGG

GCATTTGGAATTCTCAGATCAAGGGCGTTTTCTCGCCTTCGCAATGAAAATCTGGTTGTGGTTCCAATGG

CAGACTTGATTAACCACAGTGCAGGAGTTACTACAGAGGATCATGCTTATGAAGTTAAAGGAGCAGCTGG

CCTTTTCTCTTGGGATTACCTATTTTCCTTAAAGAGCCCCCTTTCCGTCAAGGCCGGAGAACAGGTATAT

ATACAATATGATTTGAACAAAAGCAATGCAGAGTTGGCTCTAGACTACGGTTTCATTGAACCAAATGAAA

ATCGACATGCATACACTCTGACGCTGGAGATATCTGAGTCGGACCCTTTTTTTGATGACAAACTAGACGT

TGCTGAGTCCAATGGTTTTGCTCAGACAGCGTACTTTGACATCTTCTATAATCGCACTCTTCCACCTGGA

TTGCTTCCATATCTGAGACTTGTAGCGCTAGGGGGTACCGACGCTTTCTTATTGGAATCACTGTTCAGAG

ACACCATATGGGGTCATCTTGAGTTGTCTGTCAGCCGTGACAATGAGGAGCTACTATGCAAAGCCGTTCG

AGAAGCCTGCAAATCTGCCCTTGCTGGTTATCATACAACCATTGAACAGGATCGCGAGTTGAAAGAAGGA

AATCTAGATTCAAGGCTTGCAATAGCAGTTGGAATAAGAGAAGGGGAAAAGATGGTCCTGCAGCAAATTG

ACGGGATCTTCGAGCAGAAAGAATTGGAGTTGGACCAGTTAGAGTATTATCAAGAAAGGAGGCTCAAGGA

TCTTGGACTTTGCGGAGAAAATGGCGATATCCTTGGAGACCTAGGAAAATTCTTCTAATCTTGCAGGAAA

ATTCTTCTAATCTTGCAGGAAGCATTTCAACCTGTTAAAGATACACTGTTGTTTACAAATGGAGTCTTCT

GAGACGTACGATGCCATGATTTTGCAATCAATCTTAAGAGGATCGTGATCAATTTTGACTCTGGAGTCTG

GACCAATCCATTACATGCTTGAAGTTTGTAAAGAGGAAAATGTAATGTGTGAAATATAAATTACACTTCT

GTACTGGTGATTATTTATAAAGCAGTTGACCATTATTATTAC

> write(seq, file="mySeq.fasta")

> install.packages("seqinr")

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/seqinr\_3.4-5.zip'

Content type 'application/zip' length 3937655 bytes (3.8 MB)

downloaded 3.8 MB

package ‘seqinr’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\Shirly\AppData\Local\Temp\Rtmpy6ppxc\downloaded\_packages

> library("seqinr")

Warning message:

package ‘seqinr’ was built under R version 3.4.3

> myFile <- read.fasta(file="mySeq.fasta")

> mySequence<-myFile[[1]]

1. calculateing the AT content of a genome.

|  |
| --- |
| > count<-count(mySequence, 2)  > count  aa ac ag at ca cc cg ct ga gc gg gt ta tc tg tt  175 92 142 128 124 61 33 123 147 74 95 86 90 115 132 174  > AT<-function(seq)  + {  + count<-count(seq, 2)  + fat<-count[[4]]/sum(count)  +  + }  > result <- AT(mySequence)  > print(result)  [1] 0.07146845 |
|  |
| |  | | --- | | > | |

1. Analyzing the AT content variation of a specific DNA sequence on a changing window size.

> GC\_window<-function(mySequence,sizeWindow)

+ {

+ starts <- seq(1,length(mySequence), by = sizeWindow)

+ n <- length(starts)-1

+ for (i in 1:n){

+ chunk <- mySequence[starts[i]:(starts[i]+(sizeWindow-1))]

+ chunkGC <- GC(chunk)

+ print(chunkGC)

+ }

+ chunkGCs <- numeric(n)

+ for (i in 1:n){

+ chunk <- mySequence[starts[i]:(starts[i]+(sizeWindow-1))]

+ chunkGC <- GC(chunk)

+ print(chunkGC)

+ chunkGCs[i] <- chunkGC

+ }

+ plot(starts[1:n],chunkGCs,type="b", main=paste("Window size: ",sizeWindow ),xlab="Nucleotide start position",ylab="GC content")

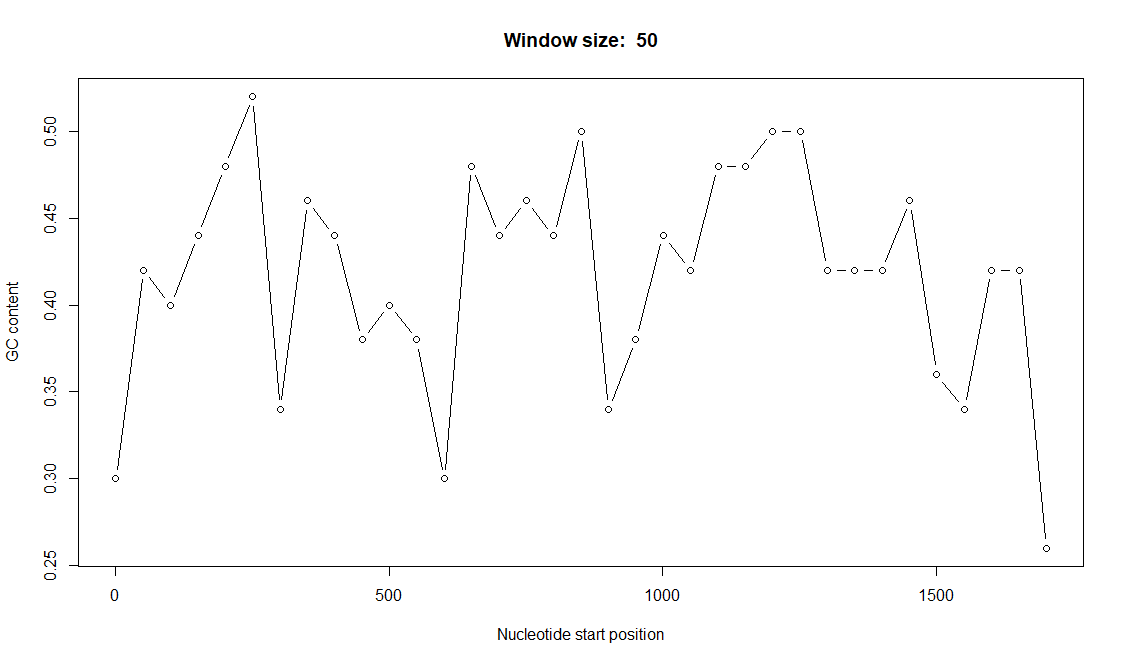
+ }

> window <- readline(prompt="Enter a window size: ")

Enter a window size: 50

> window<- as.integer(window)

> result <- GC\_window(mySequence,window)

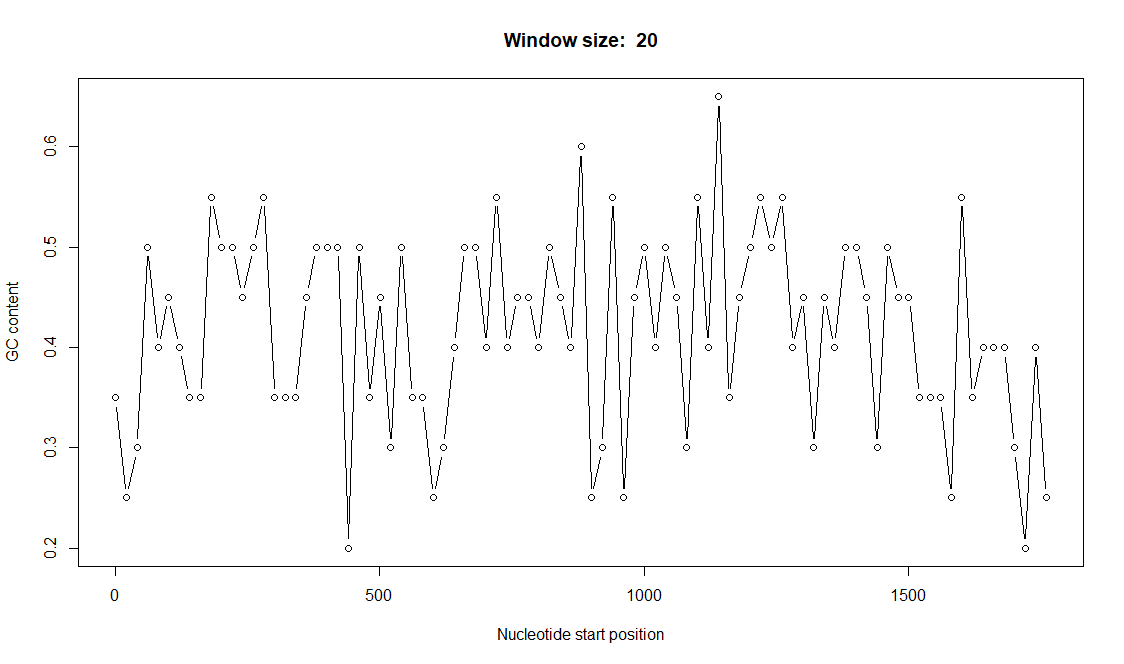


> window <- readline(prompt="Enter a window size: ")

Enter a window size: 20

> window<- as.integer(window)

> result <- GC\_window(mySequence,window)



1. Function that calculate all of the 2-letters sequences representation states (under/ over) of a sequence.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | > representation<-function(seq)  + {  + c1<-count(seq, 1)  + c2<-count(seq, 2)  + letter=c('a','c','g','t')  + len=length(c1)  + flag<-0  + for(i in 1:len)  + {  +  + for(j in 1:len)  + {  + flag<-flag+1  + f1 = count(seq, 1)[[i]]/sum(count(seq, 1))  + f2 = count(seq, 1)[[j]]/sum(count(seq, 1))  + f12 = count(seq, 2)[[flag]]/sum(count(seq, 2))  + title=paste(letter[i],letter[j],":")  + t=gsub( " ", "", title)  + result=paste(t,(f12/(f1\*f2)))  + if ((f12/(f1\*f2))>1) {  + type<-"over represented:"  + } else {  + type<-"under represented:"  + }  + result=paste(type,result)  + print(result)  +  + }  +  + }  +  +  + }  >  > result2<-representation(mySequence)  [1] "over represented: aa: 1.08810273542155"  [1] "under represented: ac: 0.898189265509877"  [1] "over represented: ag: 1.17941984345181"  [1] "under represented: at: 0.83636376429338"  [1] "over represented: ca: 1.21060292307853"  [1] "under represented: cc: 0.935099979986875"  [1] "under represented: cg: 0.430370213446369"  [1] "over represented: ct: 1.26193948728395"  [1] "over represented: ga: 1.22094871117898"  [1] "under represented: gc: 0.965072599849434"  [1] "over represented: gg: 1.0540274562696"  [1] "under represented: gt: 0.750640379403701"  [1] "under represented: ta: 0.588068271768783"  [1] "over represented: tc: 1.17986212225735"  [1] "over represented: tg: 1.15214569861963"  [1] "over represented: tt: 1.19477980381673" | |  | | |  | | --- | |  | | |
|  |
| |  | | --- | | > | |