**2.1**

In this question, is large and is small; therefore, the distribution should be Poisson.

From the goodness-of-fit summary, we accept the null hypothesis that the underlying process is Poisson. The table comparison and rootgram plot can also confirm this.

A picture containing bird

Description automatically generated

A close up of a map

Description automatically generated

R Code:

*#2.1*

*#generate data*

*sequence\_length <- 1000*

*mutation\_rate <- 10^-4*

*trial\_size <- 10000*

*l <- replicate(trial\_size,{sum(rbinom(sequence\_length,1,mutation\_rate))})*

*#fit data*

*library("vcd")*

*f <- goodfit( l, "poisson")*

*#check goodness of Poisson fit*

*summary(f)*

*rootogram(f, main="Poisson Rootgram",xlab = "",ylab="Frequency", rect\_gp = gpar(fill = "chartreuse4"))*

*#simulate many Poisson trials using the fitted lambda parameter*

*lambda <- f$par*

*simulated = rpois(trial\_size,lambda[[1]])*

*table(l)*

*table(simulated)*

**2.2**

A screenshot of a cell phone

Description automatically generated

Based on the plot above, it is evident that the maximum likelihood estimation for is 7 for the maximum of 25 independent identically distribution uniform random variable.

The theoretical justification is:

Let be an independent identically distributed uniform random variable, , and

Let be an infinitesimally small number:

Therefore, as increases, the maximum converges to

Let’s increase the number of trials from 100 to 10000, and plot the distribution of maxima again:

A screenshot of a cell phone

Description automatically generated

R Code:

*#2.2*

*#generate data*

*generator <- function(n=25,min=0,max=7){*

*return(max(runif(n,0,7)))*

*}*

*B = 100*

*l <- replicate(B,generator())*

*#plot data*

*hist(l,xlab="",main="Distribution Of Maxima")*

**2.3**

a.

The fact that 20 amino acids have redundant expressions due to 64 codon spellings is verified below.

A picture containing bird

Description automatically generated

b.

The “PerThous” variable refers to the frequency that a codon would appear every thousand codons.

It can be computed from the command “(mtb$Number/sum(mtb$Number))\*1000”

c.

the strongest bias belongs to the isoleucine amino acid. There are three codon spellings for isoleucine and the greatest bias for ATC is 46.3%.

A picture containing bird

Description automatically generated

R Code:

*#2.3*

*#a*

*mtb = read.table("~/Desktop/M\_tuberculosis.txt",header=TRUE)*

*table(mtb$AmAcid)*

*table(mtb$Codon)*

*#b*

*(mtb$Number/sum(mtb$Number))\*1000*

*#c*

*library(dplyr)*

*bias\_transform <- function(t = mtb){*

*new\_mtb <- t %>%*

*group\_by(AmAcid) %>%*

*mutate(freq = Number/sum(Number), redundant = length(factor(Codon))) %>%*

*mutate(bias = abs(freq-(1/redundant)))*

*return(new\_mtb[which.max(new\_mtb$bias),])}*

*bias\_transform()*

**2.4**

Question 2.4 was completed with help from a tutorial on “Biostring” posted by the Stanford University: <https://web.stanford.edu/class/bios221/labs/biostrings/lab_1_biostrings.html>

a.

To see the complete sequence, use the “as.character(staph[i])” expression.

A screenshot of a cell phone

Description automatically generated

b.

Herein we use the built-in function “alphabetFrequency” from the package “Biostring” for fixed window analysis

A close up of a map

Description automatically generated

c.

Herein we use the built-in function “letterFrequencyInSlidingWindow” from the package “Biostring” for sliding window analysis

A screenshot of a social media post

Description automatically generated

A screenshot of a social media post

Description automatically generated

d.

We could plot the GC fraction along the window sequence.

Here we can see the plot from part (d) roughly follows that of (b).

A drawing of a person

Description automatically generated

R Code:

*#2.4*

*library("Biostrings")*

*#a*

*staph = readDNAStringSet("~/Desktop/staphsequence.ffn.txt", "fasta")*

*staph[1:3]*

*#b*

*library(Biostrings)*

*staph <- readDNAStringSet("~/Desktop/staphsequence.ffn.txt", "fasta")*

*window <- 100*

*l <- length(staph[[1]])*

*start <- (c(1:as.integer(l/window))-1)\*window*

*end <- start + window*

*view <- Views(staph[[1]],start=start,end=end)*

*gc\_fixed\_window <- rowSums(alphabetFrequency(view)[, c(2,3)]/window)*

*plot(gc\_fixed\_window, type = 'l')*

*#c*

*window <- 100*

*GC\_content <- letterFrequencyInSlidingView(staph[[1]], window, c("G","C"))*

*GC\_content*

*GC\_fraction <- GC\_content/window*

*GC\_fraction*

*#d*

*GC\_roll <- rowSums(GC\_fraction)*

*plot(GC\_roll, type = 'l',ylab="GC %",xlab="Window")*

**2.5**

is flat, hence the name “uniform” distribution.

is flat at its central region then curves up at its tails.

Using formula , we see that and have the same mean at 0.5 while the rest of the s have the same mean at 0.25.

A screenshot of a cell phone

Description automatically generated

R Code:

*s <- seq(0,1,by=0.005)*

*d\_10\_30 <- dbeta(s, 10, 30)*

*d\_20\_60 <- dbeta(s, 20, 60)*

*d\_50\_150 <- dbeta(s, 50, 150)*

*d\_1\_1 <- dbeta(s, 1, 1)*

*d\_h\_h <- dbeta(s, 1/2, 1/2)*

*plot(s,d\_10\_30,type="l",lty=2,ylab="",ylim=c(0,15),col="red")*

*lines(s,d\_20\_60,col="green",lty=2)*

*lines(s,d\_50\_150,col="blue",lty=2)*

*lines(s,d\_1\_1, col="purple",lty=2)*

*lines(s,d\_h\_h, col="orange",lty=2)*

*legend( "topright", c("(10,30)","(20,60)","(50,150)","(1,1)","(1/2,1/2)"),text.col=c("red","green","blue","purple","orange"))*

**2.6**

The prior distribution for the textbook example is and its posterior distribution given is .

I chose the prior distribution to be and plotted its posterior against that from the textbook example above in a QQ plot.

A close up of a map

Description automatically generated

As expected, they are not matched because the theoretical posterior distribution should be , quite different from .

R Code:

*#2.6*

*#the posterior distribution from textbook where alpha = 50, beta = 350*

*posterior\_theta\_textbook = rbeta(n = 1e6, 90, 610)*

*#my own posterior distribution generated from alpha = 20, beta= 50*

*rtheta = rbeta(100000, 20, 50)*

*y = vapply(rtheta, function(th) {rbinom(1, prob = th, size = 300)}, numeric(1))*

*posterior\_theta\_new = rtheta[ y == 40 ]*

*qqplot(posterior\_theta\_new, posterior\_theta\_textbook, type = "l", asp = 1)*

*abline(a = 0, b = 1, col = "blue")*