Determining the location of clusters of palindromes to find a potential replication site of the human Cytomegalovirus

Fiona Moss

### Summary of Findings:

A careful look at the data, strongly impresses the researcher that there is an imminent need to identify the existence of unusually dense clusters (Candidate Regions) of complementary palindromes as these are potential replication sites for the virus.

My analysis indicates that, the first candidate region with the Largest Cluster of palindromes (12 palindromes) is located between 92526 and 93601. The second and the third candidate regions(with 8 palindromes) are located between 194111 - 195835 and 195032 - 196992 respectively.

Therefore, if we consider the bin size of 2000, the bin to be considered to find candidate regions is as follows: 1. First Candidate Region: 92001 - 94000 2. Second Candidate Region: 194001 - 196000 3. Third Candidate Region: 195001 - 197000

### Analysis of Data:

The data for this study is present in the following location:

# The file where the data is located  
cmv <- read.csv("http://www.uwyo.edu/buerkle/compbio/statlabs/data/hcmv.data.txt")$location

These potential replication sites can be found using the following analyses:

1. Location and count using Sliding Window Analysis
2. Spacing

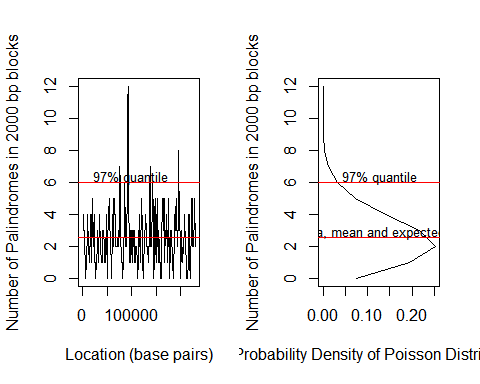
\* Location and Count:

Graphical methods can be used to find the location of clusters of palindromes in the DNA along with the count of the number of palindromes in each cluster in the DNA.

One such method is the sliding window analysis. Since in a histogram, bins are independent of each other, a tight cluster of palindromes can be split between two intervals. To overcome this problem, Sliding Window Analysis can be used to generate counts for overlapping blocks of locations in the CMV genome with a block size of 2000 and increments of 1000.

Since Poisson Distribution gives the probability of occurrence of palindromes in each interval, we can compare the graphical results of the sliding window analysis with the Poisson Distribution as follows:

# To plot both the graphs beside each other  
par(mfrow = c(1,2))  
# Sliding Window Analysis to find the number of Palindromes in 2000 base-pair blocks  
slide.window <- function(seqdata, total.length, blocksize = 500, incr = 100){  
 block.right <- seq(blocksize,total.length, by = incr)  
 block.left <- block.right - blocksize + 1  
 block.mid <- (block.right + block.left) / 2  
 nblocks <- length(block.right)  
 count <- numeric(nblocks)  
 rate <- numeric(nblocks)  
   
 for(i in 1:nblocks) {  
 focal <- seqdata[seqdata > block.left[i] & seqdata < block.right[i]]  
 count[i] <- length(focal)  
 rate[i] <- count[i] / blocksize  
 # alternatively  
 # count[i] <- sum(seqdata>block.left[i] & seqdata < block.right[i])  
 }  
 data.frame(count, rate, block.left, block.mid, block.right)  
}  
  
#Plotting the Sliding Window for Location of palindromes versus the count of palindromes in that location with bin size 2000  
plot(count ~ block.mid, data = slide.window(cmv, 230000, blocksize = 2000, incr = 1000), xlab = "Location (base pairs)", ylab = "Number of Palindromes in 2000 bp blocks", type = "l")  
  
abline(h = qpois(0.97, lambda = 2000\*296/229354), col = 'red')  
text(x = 100000, y = qpois(0.97, lambda = 2000\*296/229354) + 0.3, labels = "97% quantile", cex = 0.8)  
abline(h = 2000\*296/229354, col = 'red')  
  
# Plotting the Probability Density of Poisson Distribution  
plot(dpois(0:12, lambda = 2000\*296/229354), 0:12, type = 'l', ylab = "Number of Palindromes in 2000 bp blocks", xlab = "Probability Density of Poisson Distribution")  
abline(h = qpois(0.97, lambda = 2000\*296/229354), col = 'red')  
abline(h = 2000\*296/229354, col = 'red')  
text(x = 0.13, y = (2000\*296/229354) + 0.3, labels = "lambda, mean and expected value", cex = 0.8)  
text(x = 0.13, y = qpois(0.97, lambda = 2000\*296/229354) + 0.3, labels = "97% quantile", cex = 0.8)



From the results of the sliding window analysis, it can be found that there is an unusual number of palindromes above the 97% quantile.

The maximum count using sliding window analysis along with the location of the count is as follows:

data = slide.window(cmv, 230000, blocksize = 2000, incr = 1000)  
data[which(data$count == max(data$count) | data$count == 8),]

## count rate block.left block.mid block.right  
## 93 12 0.006 92001 93000.5 94000  
## 195 8 0.004 194001 195000.5 196000  
## 196 8 0.004 195001 196000.5 197000

Therefore, with a block size of 2000, the largest cluster consists of 12 palindromes and is located in the block: 92001-94000

# Palindrome location in the interval with the largest number of palindromes in a cluster   
largest <- cmv[cmv >= 92001 & cmv < 94000]  
#largest

The locations of palindromes in the largest cluster (first candidate region) are:

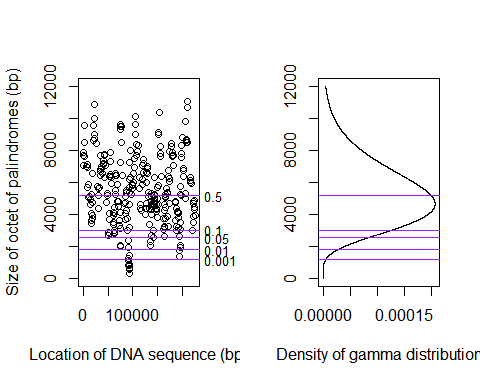
# Palindrome location in the intervals with the second largest number of palindromes in a cluster   
second1 <- cmv[cmv >= 194001 & cmv < 196000]   
second2 <- cmv[cmv >= 195001 & cmv < 197000]  
#second1  
#second2

The locations of palindromes in the second most largest candidate regions(second and the third candidate regions) are:

\* Spacing:

The spacing between clusters can be found using gamma distribution. Here a lag of 7 is considered to find the distance between octets of palindromes. Here, quantiles of location and size of the octet have been compared to density of gamma distribution and size of the octet using qgamma and dgamma respectively.

# To plot both the graphs beside each other  
par(mfrow = c(1,2))  
plot(cmv[1:(296-7)], diff(cmv, lag=7), xlab = "Location of DNA sequence (bp)", ylab = "Size of octet of palindromes (bp)", ylim = range(0, 12000))  
  
# Plotting various quantiles  
abline(h = qgamma(c(0.001, 0.01, 0.05, 0.1, 0.5), rate = 296/229354, shape = 7), col = 'purple')  
mtext(text = "0.001", side = 4, at = qgamma(0.001, rate = 296/229354, shape = 7), las = 1, cex = 0.8, line = 0.25)  
mtext(text = "0.01", side = 4, at = qgamma(0.01, rate = 296/229354, shape = 7), las = 1, cex = 0.8, line = 0.25)  
mtext(text = "0.05", side = 4, at = qgamma(0.05, rate = 296/229354, shape = 7), las = 1, cex = 0.8, line = 0.25)  
mtext(text = "0.1", side = 4, at = qgamma(0.1, rate = 296/229354, shape = 7), las = 1, cex = 0.8, line = 0.25)  
mtext(text = "0.5", side = 4, at = qgamma(0.5, rate = 296/229354, shape = 7), las = 1, cex = 0.8, line = 0.25)  
  
# Plotting the density of Gamma Distribution  
plot(dgamma(0:12000, shape = 7, rate = 296/229354), 0:12000, xlab = "Density of gamma distribution", ylab = '', type = 'l')  
abline(h = qgamma(c(0.001, 0.01, 0.05, 0.1, 0.5), rate = 296/229354, shape=7), col = 'purple')



# Location of smallest octet of palindromes  
difference <- diff(cmv, lag = 7)  
location <- as.matrix(cbind(1:(296-7), difference))  
colnames(location) <- c("order", "size")  
minimum <- min(location[,2])  
position <- location[,1][location[,2] == min(location[,2])]

The distance between an octet of palindromes in a cluster is lesser than any 8 palindromes considered randomly.

Hence, the location at which the distance between the octets of palindromes is most minimum the First Candidate region. Therefore, the location of the palindromes in this octet (first candidate region) is

### Accuracy of results:

In my opinion, this line of analysis should yield the right results because of the following reasons:

1. I have used the suitable scientific statistical methods and tested the results if they are statistically significant, I feel that there is reason enough for me to end up with highly reliable results. The following steps were followed in this study.

* Checked to find out if the homogeneous Poisson process can be followed for the analysis, by making sure that the characteristics of Homogeneous Poisson process exist.
* Then the lambda rate or the rate of hits per unit area, were estimated and goodness of fit made sure of. Only in the case of a good fit can we find out if there is an excess of palindromes.
* We can also use different different properties of the Poisson process to test the data in different ways.
* The study has been extended by constructing Sliding bin plot in order to calculate palindrome counts for overlapping intervals.

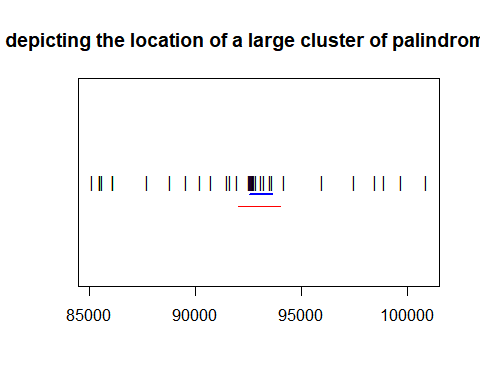
1. Both the analyses used (Count, Length and Space) have shown the same location for the first candidate region thereby proving the accuracy of the leading candidate results.

### Robustness of choice of lag and window size:

Considering the model to be a Homogenous Poisson Process:

1. I have tested the analyses with different bin sizes and increments but the location of the candidate regions hasn't changed considerably. This can be proved by plotting the graph showing the location of palindromes in the DNA. It can be seen that the estimation of the location of the first candidate region is fairly accurate irrespective of the bin size. Similar results have been found when the window size was changed.

myhist<-hist(cmv, breaks=seq(0,230000, by=2000), plot = F)  
mycounts<- data.frame(loc=myhist$breaks[-1],counts=myhist$counts)  
stripchart(cmv[cmv <= mycounts$loc[which.max(mycounts$counts)+4] &  
 cmv > mycounts$loc[which.max(mycounts$counts)-1-4] ], pch="|")  
title("Graph depicting the location of a large cluster of palindromes in DNA")  
segments(mycounts$loc[which.max(mycounts$counts)], 0.9,  
 mycounts$loc[which.max(mycounts$counts)-1], 0.9, col="red")  
# Range of the locations of palindromes  
myrange<-range(cmv[cmv <= mycounts$loc[which.max(mycounts$counts)] &  
 cmv > mycounts$loc[which.max(mycounts$counts)-1]])  
segments(myrange[1], 0.95, myrange[2], 0.95, col="blue", lwd=2)



2. For a proper analysis of spacings between the clusters, an appropriate lag is necessary. Since Palindromes shorter than 10 letters were ignored a very small lag wouldn't give the optimal result. Similarly, as the largest palindrome is of size 18, a larger lag also couldn't be considered. I have tested the analyses with different lags(7,8,9) and it didn't produce any significant change in the location of the candidade regions. 3. The rate was initially unknown but the emperical average number of hits per unit interval were used to estimate the rate. Lambda is a measure of rate. The rate that I considered was: ((bin size \* Total number of palindromes)/Length of the DNA) which produced accurate results.