Homework 3

Student 1 and Student 2

2024-11-5

0. Contribution Statement

Student 1

Student 1 mainly worked on questions 1, 3, and 5.

Student 2

Student 2 mainly worked on questions 2 and 4, the advanced analysis, and formatting the Rmd filee.

Introduction

Data

This data is from a publication by Chee et al. and describes the DNA sequence of CMV. A CMV DNA molecule has 229,354 complementary pairs of letters or base pairs. These scientists are in search of special patterns in the virus' DNA that contains instructions for its reproduction: replication. They discovered a total of 296 palindromic sequences, each of which are at least 10 pairs long.

Objective

The goals of this report are to investigate the distribution of genes among this DNA sequence.

Basic Analysis

Question 1: RANDOM SCATTER

Methods

Here, we are using `sample.int` to conduct 1 simulation assuming a uniform spread.

```
N = 229354
n = 296
set.seed(10)
random_1 <- as.vector(sample.int(N, size=n, replace=FALSE)) # locations uniformly randomly generated
```

We're interested in seeing how simulations vary when repeated many times, so we used `replicate` to take 100 simulations of a DNA sequence, with each simulation containing 296 palindrome sites.

QUANTITATIVE ANALYSIS:

We will calculate the average distances between palindrome sites for each simulation to determine whether the palindrome sites are randomly distributed or if there are patterns or clusters.

```
# For each simulation, we will calculate the differences between the data points (palindrome sites)
average_distances <- sapply(simulations, function(simulation){
    sorted_simulation <- sort(simulation) # We sorted the simulation to get correct distances
    distances <- diff(sorted_simulation)
    mean(distances)
})
```

Now, we have a list of the average distances between palindrome sites in a DNA sequence. We will compute the mean and variance of the average distances between the palindrome sites to provide a central measure of approximation.

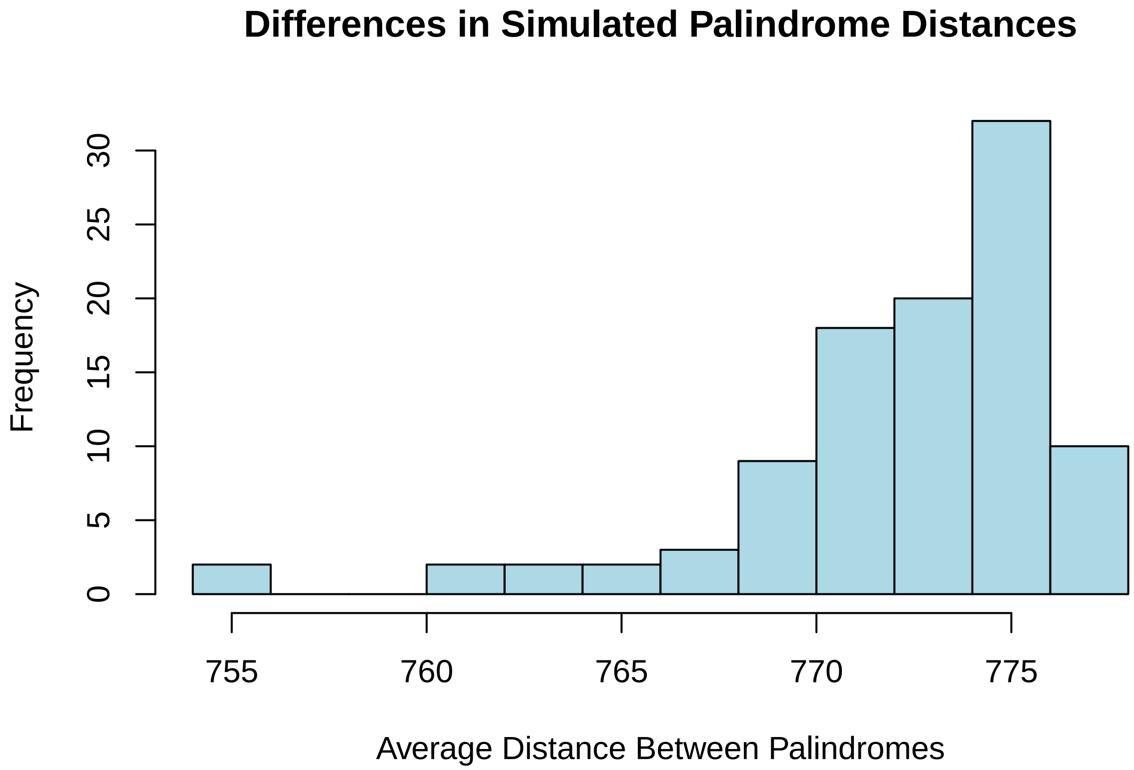
```
# On average, palindrome sites tend to be about 771.755 bases apart in simulated datasets.
mean(average_distances)
```

```
## [1] 772.1922
```

```
# The variance of the average distances, about 14.677, tells us there is moderate variability in the av
var(average_distances)
```

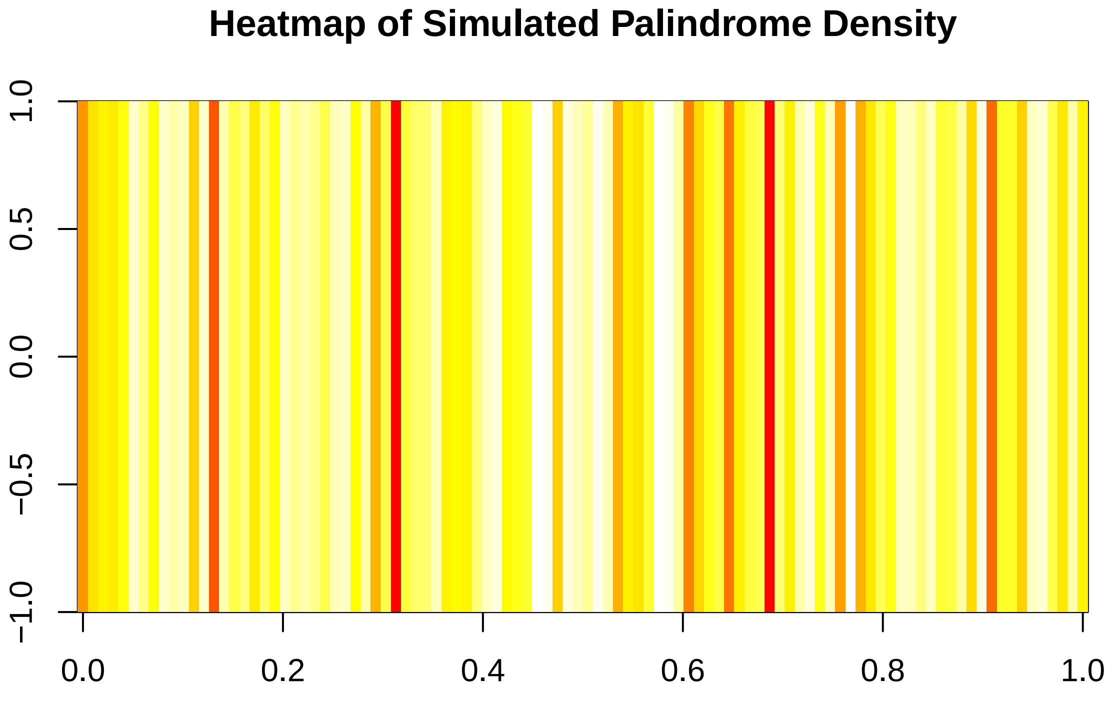
```
## [1] 18.65551
```

QUALITATIVE ANALYSIS:



The histogram of the average distance between palindromes is left-skewed, which suggests few small gaps between palindromes sites. The average distances appear to mostly be clustered between 770 and 780, but we also have palindromes distances that were as low as 755. This variation can indicate that the palindromes may not be uniformly spread and have clustering at particular locations, which was confirmed by the quantitative analysis above. This histogram is also unimodal, suggesting most palindrome sites are mostly spaced apart within a certain range.

```
image(matrix(average_distances, ncol=1), col=heat.colors(max(average_distances)), main="Heatmap of Simul
```



We can visualize the spread of distances between palindromes in a heat map. The palindromes are generally uniformly spread out but experience clustering in certain areas.

Analysis

We will conduct a **KS Analysis** to test whether the distribution of palindrome sites in our simulated dataset is statistically similar to the real dataset.

- **H0:** The distribution of simulated distances and real distances are the same.
- **H1:** The distribution of simulated and real distances differ. We meet the conditions to perform this test because the samples are independent and the data is continuous. We don't have the "real data", so we will take a sample and treat it like the "real data".

```r
real_data_distances <- diff(sort(sample(1:N, n)))
# KS test
ks_test <- ks.test(average_distances, real_data_distances)
```

```
## Warning in ks.test.default(average_distances, real_data_distances): p-value
## will be approximate in the presence of ties
```

Conclusion

With a p-value of p < 2.2 * 10^(-16), we can reject the null hypothesis, and there is a significant difference between the random scatters and real data. However, our inferences in the accuracy of this conclusion are limited because we simply drew another sample and treated it as the "real data", but our "real data", may not necessarily look like this.

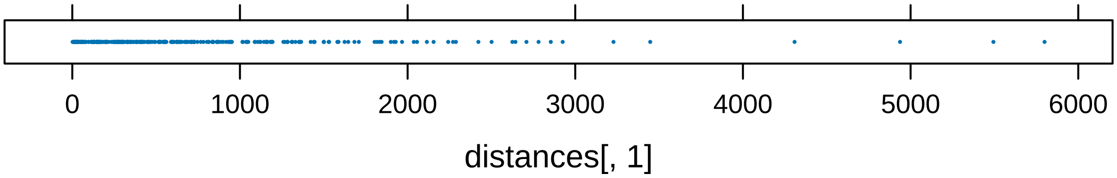
Question 2: LOCATIONS & SPACING

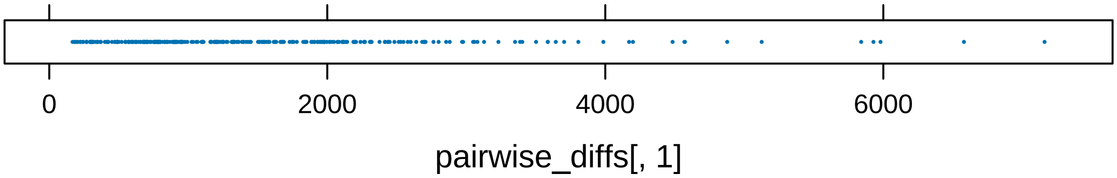
To analyze the patterns in the sampled data, we first calculated the specified differences and sums of distances.

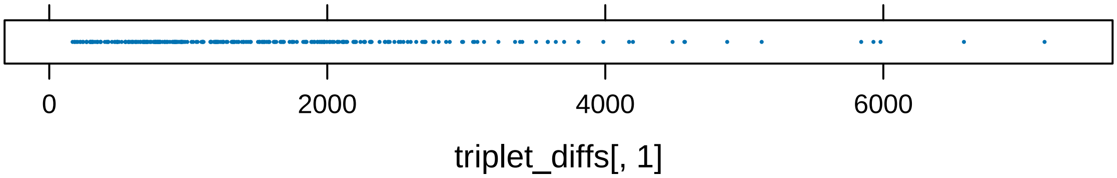
```r
set.seed(10)
simulations <- replicate(num_sim, sample(1:N, n, replace = FALSE), simplify = FALSE)
# spacing between consecutive pairs
distances <- sapply(simulations, function(simulation){
    sorted_simulation <- sort(simulation) # We sorted the simulation to get correct distances
    distances <- diff(sorted_simulation)
})
# sums of palindrome pairs
pairwise_diffs <- sapply(seq(1, ncol(distances) - 1, by = 2), function(i) rowSums(distances[, i:(i+1)]))
# sums of palindrome triplets
triplet_diffs <- sapply(seq(1, ncol(distances) - 1, by = 3), function(i) rowSums(distances[, i:(i+1)]))
```

Analysis

Next, we graphed each of the distances on a line plot to view the distribution of each data point. For simplicity, we are only displaying the first sample of each distance distribution.







Conclusion

After inspecting each graph, we noticed that the distribution of genes is centralized within the first 1000 locations for the simple distances distribution, and the first 3000 locations for both the paired and tripled sum distances distributions. The distribution of genes tend to taper off past these locations, with few genes being located past location 4000.

Question 3: COUNTS

Methods

We will make our list of all possible DNA locations, ranging from 1 to the 229354, denoted by N.

```r
DNA <- 1:N
# INVESTIGATE REGIONS OF LENGTH 1000
split_DNA <- split(DNA, ceiling(seq_along(DNA) / 1000))
# function to count number of palindromes per region (observed counts)
count_per_region <- function(region, palindrome_locations){
  sum(region %in% data$location)
}
counts_1000 <- sapply(split_DNA, count_per_region, data$location)
```

Analysis

REGIONS OF SIZE 1,000 We will conduct a chi-squared test, goodness of fit to compare the number of palindromes when each region has 1,000 bases.

- **H0:** There is no difference between the number of palindromes in regions of length 1,000 and number of palindromes that we would expect from uniform random scatter.
- **H1:** There is a difference between the number of palindromes in regions of length 1,000 and the number of palindromes we would expect from a uniform random scatter.

First, we will check the conditions to make sure we can conduct this test.

- **Observed Counts**: Yes, we have observed counts of the number of palindromes.
- **Independence**: The counts of the palindromes in each chunk are independent since they are non-overlapping.
- **Expected Frequency > 5**: When we have regions of length 1000 each, we would expect each region to have 296 / 230 chunks = 1.287 palindromes per chunk. We will still perform the chi-squared test, but it may not be accurate, so we need to be careful. **

```r
# number of palindromes we would expect for each region
expected <- 296 / 230 # total palindromes / # chunks
expected_vec <- rep(expected, 230)
expected_prob <- expected_vec / sum(expected_vec)

# perform a chi-squared test to compare observed counts with expected values
chi_squared_test <- chisq.test(counts_1000, p = expected_prob)
```

```
## Warning in chisq.test(counts_1000, p = expected_prob): Chi-squared
## approximation may be incorrect
```

With a p-value of $0.004 < 0.05$, we reject the null hypothesis, and there is a difference between the number of palindromes in regions of length 1000 and a uniform random scatter.

Now, we will repeat this process but with larger region sizes. REGIONS OF SIZE 5,000

```r
split_DNA_5k <- split(DNA, ceiling(seq_along(DNA) / 5000))
counts_5000 <- sapply(split_DNA_5k, count_per_region, data$location)
expected_5k <- 296 / 46 # total palindromes / # chunks
```

Now that we have expected counts of 296 / 46 = 6.435, which is > 5, we meet the condition of having E_i >= 5, for all i. The rest of the conditions are still satisfied, as described above.

```
expected_vec_5k <- rep(expected_5k, 46)
expected_prob_5k <- expected_vec_5k / sum(expected_vec_5k)
chi_squared_test_5k <- chisq.test(counts_5000, p = expected_prob_5k)
```

Our p-value from this chi-squared test is now 0.02, which is < 0.05. We still reject the null hypothesis, and there is a difference between the number of palindromes in regions of length 1000 and a uniform random scatter. We will conduct this statistical test one last time with regions of size 10,000. REGIONS OF SIZE 10,000

```
split_DNA_10k <- split(DNA, ceiling(seq_along(DNA) / 10000))
counts_10000 <- sapply(split_DNA_10k, count_per_region, data$location)
expected_10k <- 296 / 23 # All conditions for the chi-squared test have been satisfied, described previ
expected_vec_10k <- rep(expected_10k, 23)
expected_prob_10k <- expected_vec_10k / sum(expected_vec_10k)
chi_squared_test_10k <- chisq.test(counts_10000, p = expected_prob_10k)
```

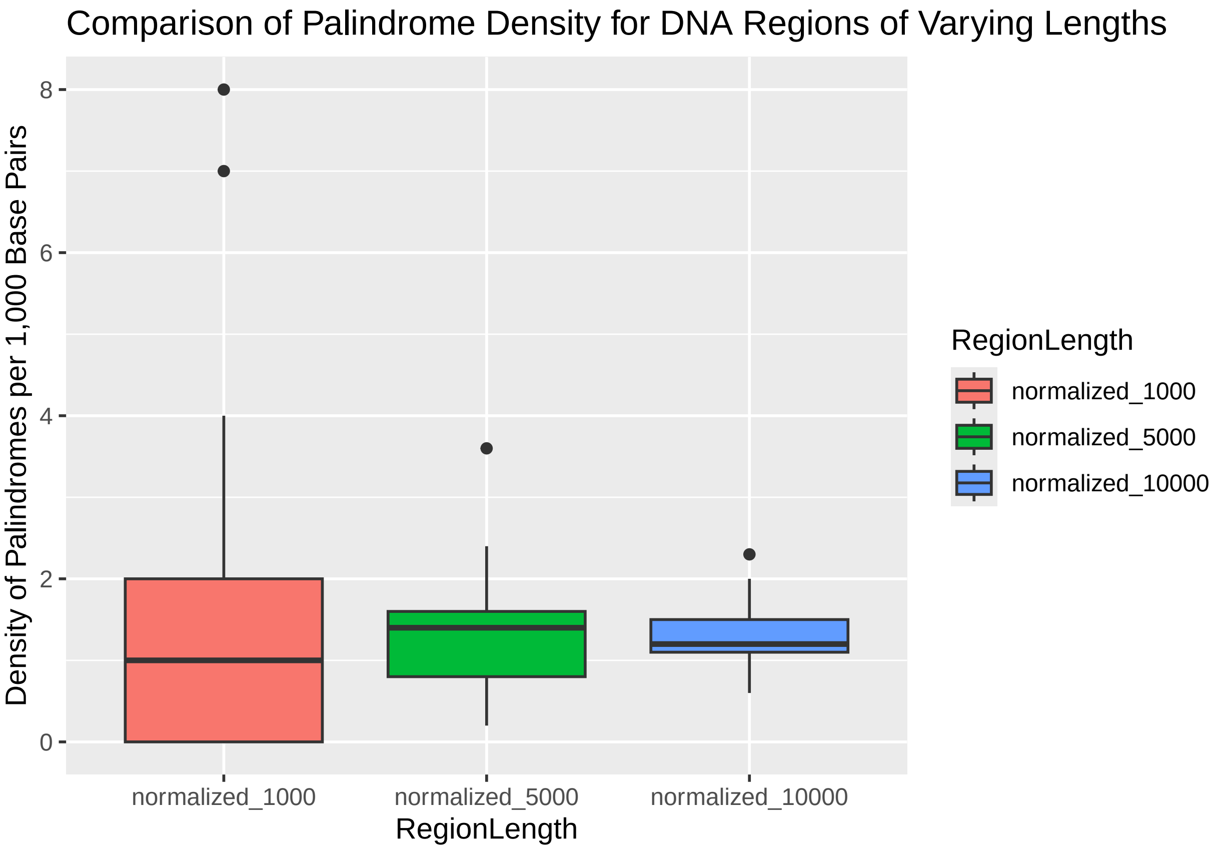
Our p-value went up once again! With a p-value of 0.317, p > 0.05. We fail to reject the null hypothesis, and there is no significant difference between the counts of palindromes in our non-overlapping regions of equal length with the counts we would expect from a uniform random scatter.

GRAPHICAL ANALYSIS: To compare the counts of palindromes across different regions, we need to normalize them since the regions are of different lengths. We will find the number of palindromes per 1000 base pairs.

```
library(ggplot2)
library(forcats)
#
normalized_1000 <- counts_1000 / 1000 * 1000
normalized_5000 <- counts_5000 / 5000 * 1000
normalized_10000 <- counts_10000 / 10000 * 1000

palindrome_counts <- data.frame(
  RegionLength = factor(rep(c('normalized_1000', 'normalized_5000', 'normalized_10000'),
                            times = c(length(normalized_1000), length(normalized_5000), length(normalize
  Count = c(normalized_1000, normalized_5000, normalized_10000)
)
# reorder DataFrame
palindrome_counts$RegionLength <- factor(palindrome_counts$RegionLength, levels = c('normalized_1000',

ggplot(palindrome_counts, aes(x = RegionLength, y = Count, fill = RegionLength)) +
  geom_boxplot() + ylab('Density of Palindromes per 1,000 Base Pairs') +
  labs(title = 'Comparison of Palindrome Density for DNA Regions of Varying Lengths')
```



In the graphs above, we are comparing the density of the palindromes for DNA regions of lengths 1,000; 5,000; and 10,000. The shorter DNA regions have notably more variance compared to the DNA sequences of a shorter length. The shorter sequences also have more outliers of a higher density, indicating a greater likelihood of having palindromes compared to longer sequences; this is supported by our quantitative analysis.

Conclusion

When our region of non-overlapping DNA is shorter, we are sampling less DNA and are more likely to think there is a significant difference of counts of palindromes in our DNA sequence compared to a uniform random scatter. As we increased our DNA region length, our p-value also went up. As we observe more DNA consecutively, we see that the pattern of palindromes does not really occur at such a high rate as it appears to in the smaller regions. A good analogy is that you are more likely to find palindromes in shorter words (eg 'mom', 'dad') than a full sentence (eg "Was it a car or a cat I saw?").

Question 4: THE BIGGEST CLUSTER

Methods

Analysis

Conclusion

Question 5: ADVICE TO BIOLOGIST TO FIND ORIGIN OF REPLICATION

METHODS

We would suggest the following to a biologist to find the origin of replication.

1. [OPTIONAL] Verify that palindromes indicate a greater likelihood of the origin of replication occurring nearby with other DNA sequences.
2. Split the DNA into smaller regions/samples to make it more manageable since DNA sequences can be very long. However, we have to be careful about this because if our regions of DNA are too small, we may cut off larger sequences of DNA palindromes that are significant and not be able to identify them.
3. 1. SLIDING WINDOW APPROACH: We could use a sliding window of a particular length, where each window slides one base at a time. We could check if each sequence matches its reverse component.
   2. HASH TABLE FOR DNA SAMPLE REGIONS: Create a hash table for each DNA sample region we took that shows the possible substrings and their corresponding palindromes, including the string of the entire length of the DNA sample.
   3. SUFFIX TREES: We would recommend searching for palindromes through a data structure that would allow for efficient pattern matching, such as suffix trees.

ANALYSIS

Here, we will analyze each step we are suggesting to the biologist to find the origin of replication of DNA.

1. We would first have to confirm whether palindromes truly suggest a greater likelihood of the origin of replication occurring nearby. It would be useful to verify this with other DNA sequences and conduct a statistical test to study whether this difference is significant.
2. We plan to divide the DNA into smaller chunks for more manageable analysis, but this comes with the *risk* of inadvertently missing a palindrome if a palindrome sequence is long and biologically significant, cutting cut off by the sample size.
3. 1. **SLIDING WINDOW APPROACH**
      - PRO: Effective and easy to implement for short and medium length palindromes; is not computationally expensive since we are only inspecting whatever is happening within the sliding window
      - CON: may miss key palindromes if the palindrome in question is longer than the sliding window we are using. This approach is also more useful if we know the exact length of the palindrome we expect to see. It will also be unable to find overlapping patterns in DNA sequences.
   2. **HASH TABLE FOR DNA SAMPLE REGIONS:**
      - PRO: Can find palindromes very quickly once the substring and its reverse complement have been identified through the hash table
      - CON: Takes a lot of memory to build the hash table for every substring and its reverse complement. This is also limited to exact matches, so if only part of a palindrome is present in a DNA sample sequence, the hash table will not capture that.
   3. **SUFFIX TREES:**
      - PRO: Works well with long sequences, has efficient pattern matching, and is optimized for finding long, overlapping, or variable-length sequences of palindromes in DNA
      - CON: Building a suffix tree may be take up a lot of memory, especially for larger sequences. It is also very complex to implement a suffix tree compared to the other two proposed methods.

CONCLUSION

Considering our pros and cons, it seems like the best option we should recommend to a biologist to find the origin of replication of DNA is to implement suffix trees. However, as in all the methods we have suggested above, this requires 1) confirming whether palindromes are truly a sign of the origin of replication in DNA and 2) ensuring we take sample sizes of DNA of a reasonable length so that we don't inadvertently ignore a palindrome that is longer than our sample sequence.

Advanced Analysis