CASE STUDY 3: SEARCH FOR THE UNUSUAL CLUSTER IN THE PALINDROMES

Code

Question

In this paper, we will search for unusual clusters of complementary palindromes. The overarching research question is: “How do we find clusters of palindromes? How do we determine whether a cluster is just a chance occurrence or a potential replication site? Based on our analysis, we will then provide recommendations to biologists who are about to start experimentally searching for the origin of replication.

Setup

Hide

```
locations <- read.table('hcmv-25kgjn1-1rfrtkc.txt', header=TRUE)$location  # Original
health <- read.csv('RAW_DATA-2iwcznn-2kr2xw0.csv', header=TRUE)  # Additional
N <- 229354  # Base pairs
n <- 296  # Palindromes
```

Scenario 1: Random Scatter

To begin, pursue the point of view that structure in the data is indicated by departures from a uniform scatter of palindromes across the DNA.

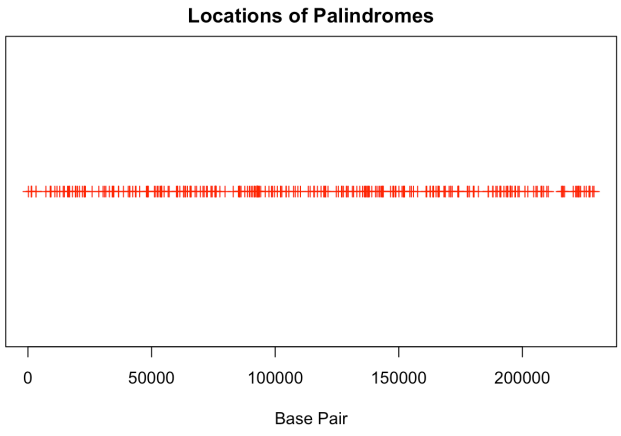
Of course, a random uniform scatter does that mean that palindromes will be equally spaced as milestones on a freeway. There will be some gaps on the DNA where no palindromes occur, and there will be some clumping together of palindromes.

To look for structure examine the locations of the palindromes, the spacing between palindromes, and the counts of palindromes in non overlapping regions of the DNA. One starting place might be to see first how random scatter looks by using a computer to simulate it.

A computer can simulate 296 palindrome sites chosen at random along a DNA sequence of 229,354 bases using a pseudo random number generator. When this is done several times, by making seller sets of simulated palindrome locations, then the real data can be compared to the simulated data.

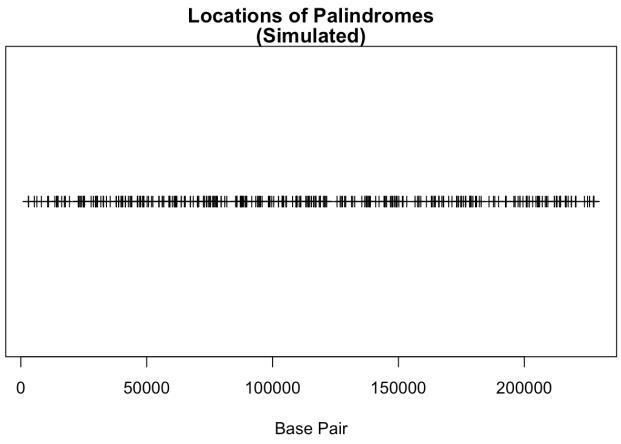
Hide

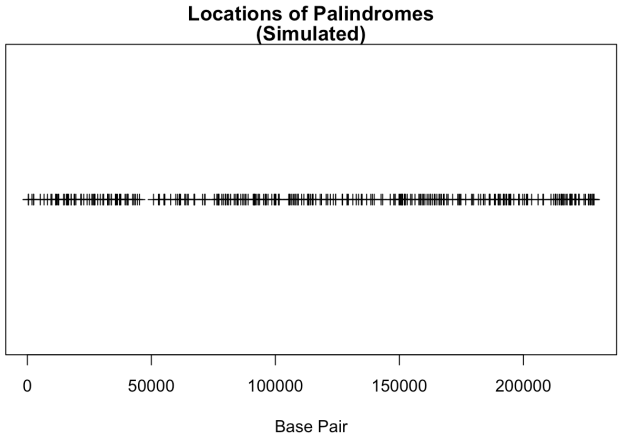
```
set.seed(0)
color <- 'red'
# Generate 3 samples from the uniform distribution with the same size and bounds as o
ur data
uniform.samples=list(sort(runif(n, min=0, max=N)), sort(runif(n, min=0, max=N)), sort
(runif(n, min=0, max=N)))
# Dot plot of locations of palindromes in original data and uniform scatter
title1 <- 'Locations of Palindromes'
title2 <- c(title1,'(Simulated)')
x.axis <- 'Base Pair'
symbol <- 3
stripchart(locations, pch=symbol, col=color, main=title1, xlab=x.axis)
```

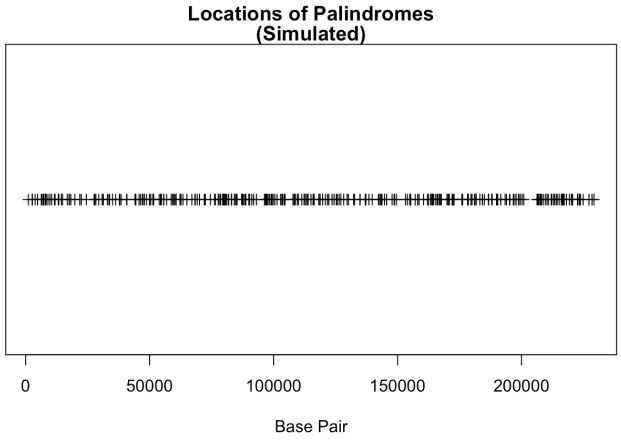


Hide

```
for (sample in uniform.samples) {
  stripchart(sample, pch=symbol, main=title2, xlab=x.axis)
}
```

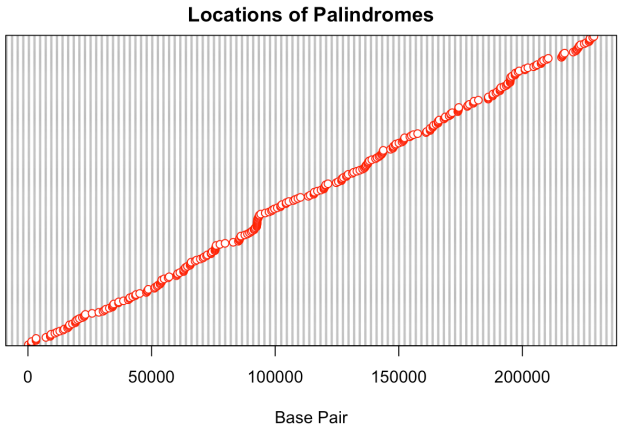






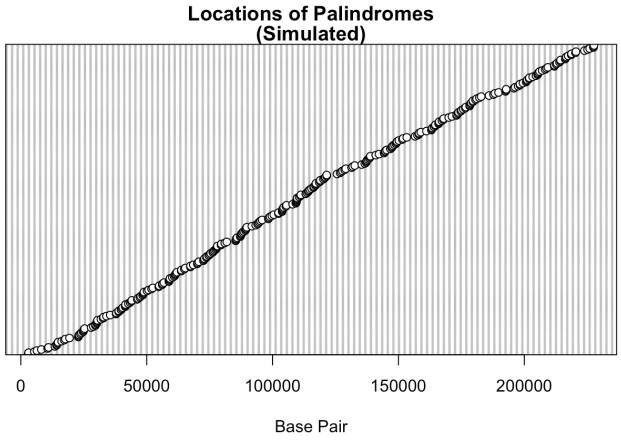
Hide

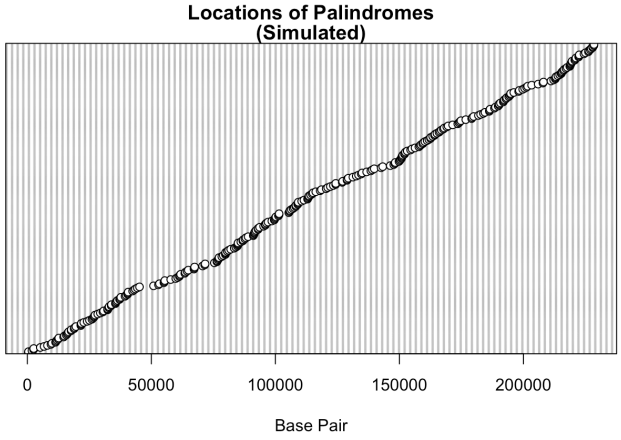
```
# Additional dot plot of locations of palindromes in original data and uniform scatte
r
dotchart(locations, color=color, main=title1, xlab=x.axis)
```

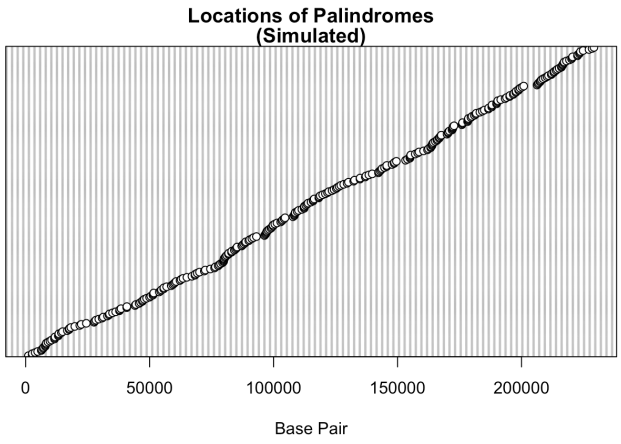


Hide

```
for (sample in uniform.samples) {
  dotchart(sample, main=title2, xlab=x.axis)
}
```

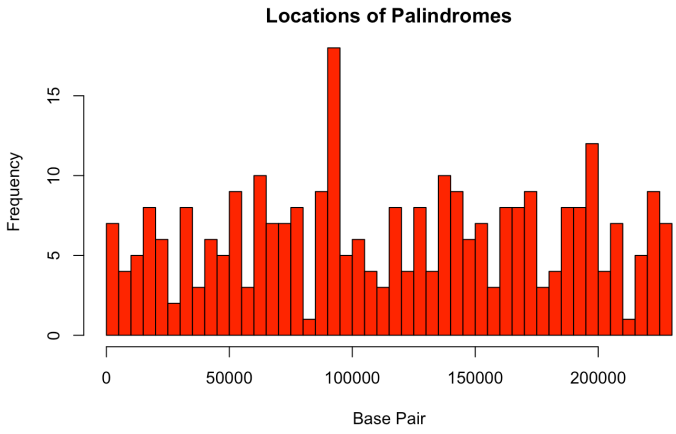






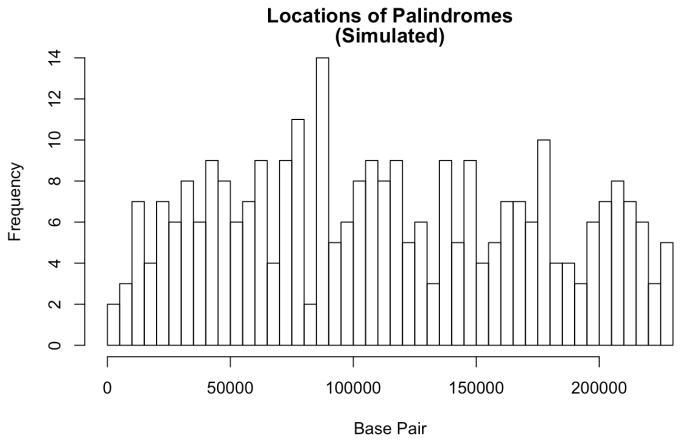
Hide

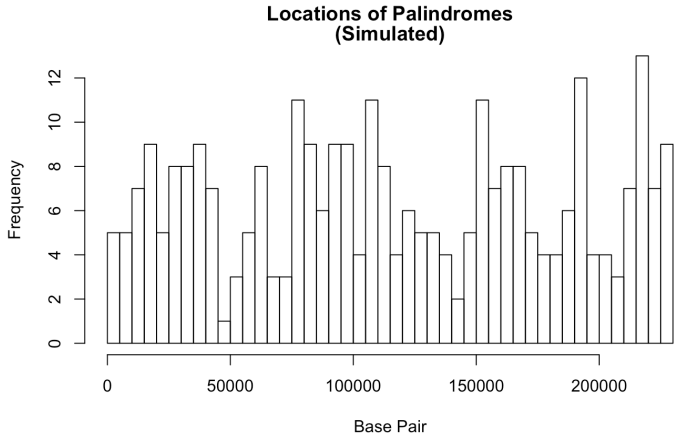
```
# Histogram of locations of palindromes in original data and uniform scatter
bins <- 35
hist(locations, col=color, breaks=bins, main=title1, xlab=x.axis)
```

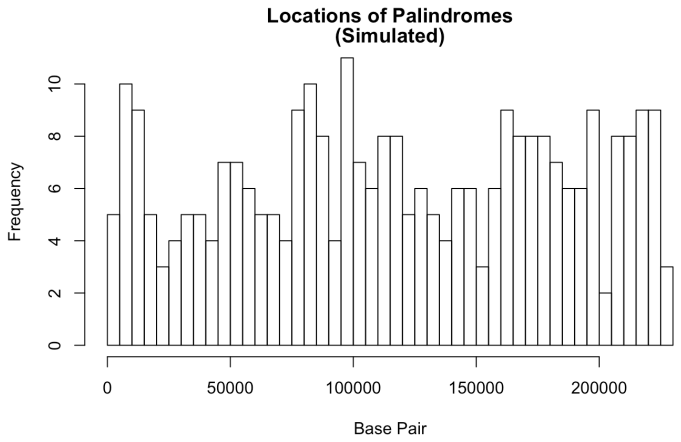


Hide

```
for (sample in uniform.samples) {
  hist(sample, breaks=bins, main=title2, xlab=x.axis)
}
```

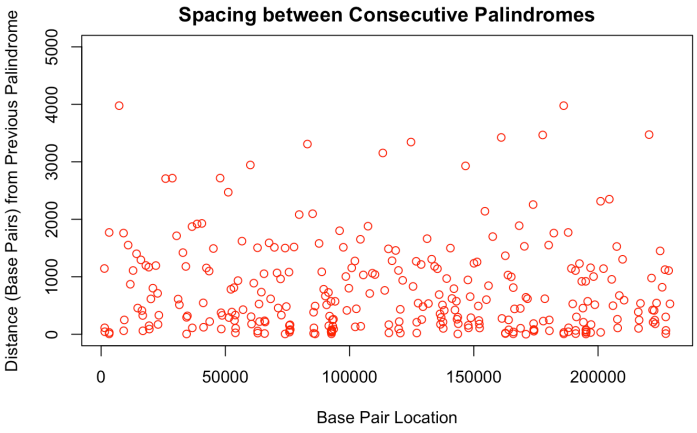






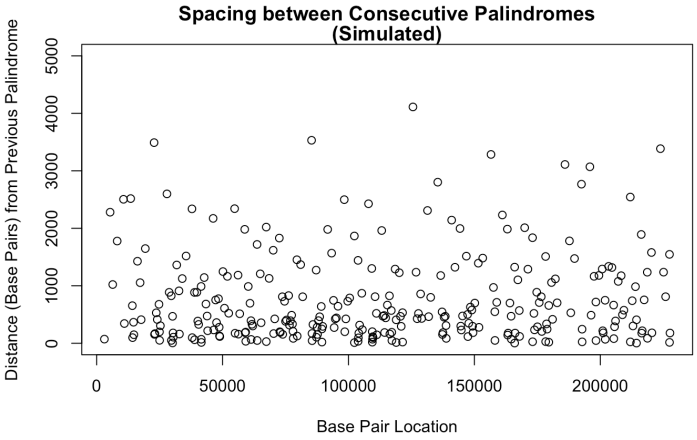
Hide

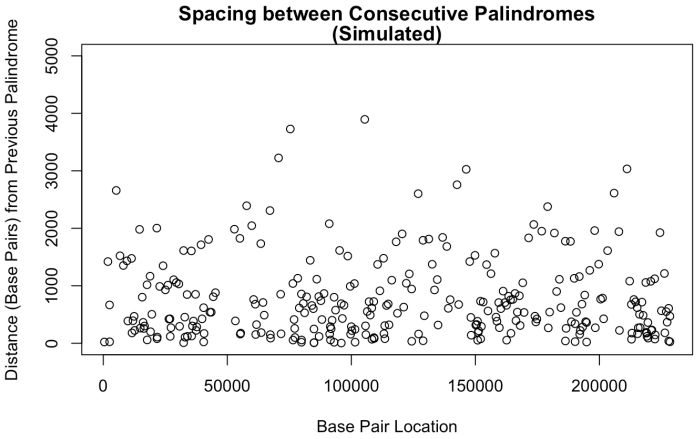
```
# Scatterplot of spacing between consecutive palindromes
title1 <- 'Spacing between Consecutive Palindromes'
title2 <- c(title1,'(Simulated)')
x.axis <- 'Base Pair Location'
y.axis <- 'Distance (Base Pairs) from Previous Palindrome'
y.range <- c(0,5000)
plot(locations[-1], diff(locations), col=color, ylim=y.range, main=title1, xlab=x.axi
s, ylab=y.axis)
```

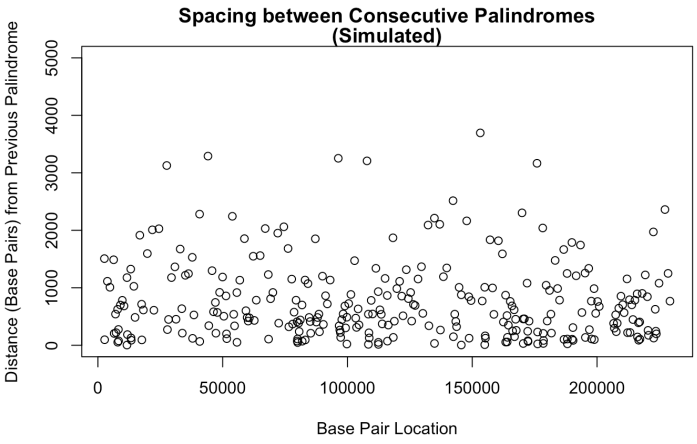


Hide

```
for (sample in uniform.samples) {
  plot(sample[-1], diff(sample), ylim=y.range, main=title2, xlab=x.axis, ylab=y.axis)
}
```

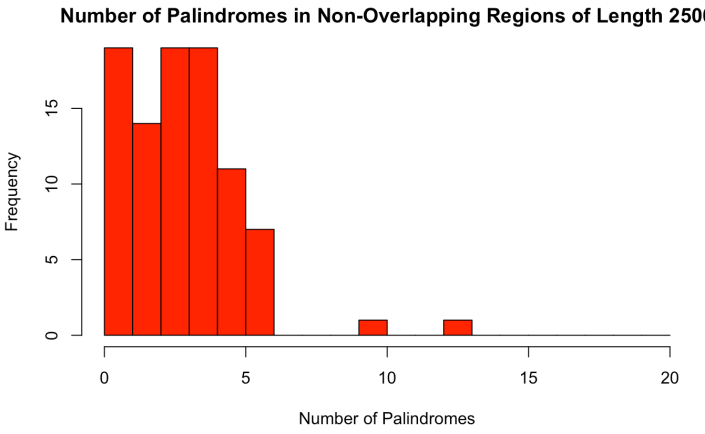






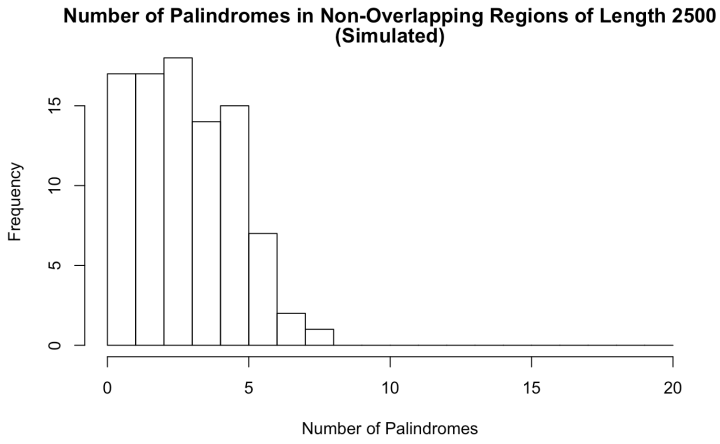
Hide

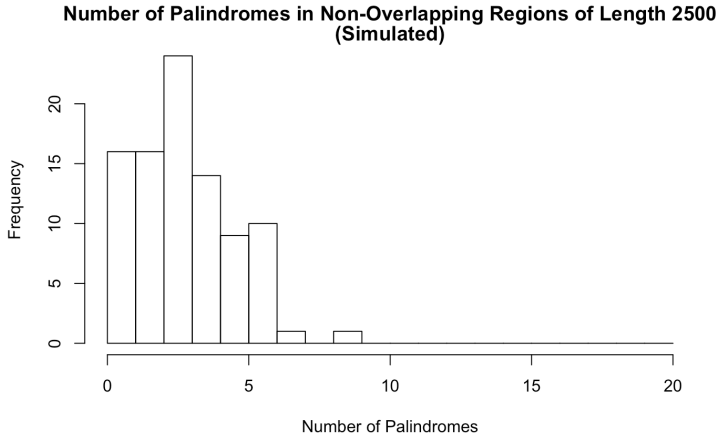
```
# Histogram of counts of palindromes in non-overlapping regions in original data and
uniform scatter
interval.length <- 2500
title1 <- paste('Number of Palindromes in Non-Overlapping Regions of Length', interva
l.length)
title2 <- c(title1,'(Simulated)')
x.axis <- 'Number of Palindromes'
bins <- seq(0,20,1)
hist(as.vector(table(cut(locations, breaks=seq(0,N,interval.length), include.lowest=T
RUE))), breaks=bins, col=color, main=title1, xlab=x.axis)
```

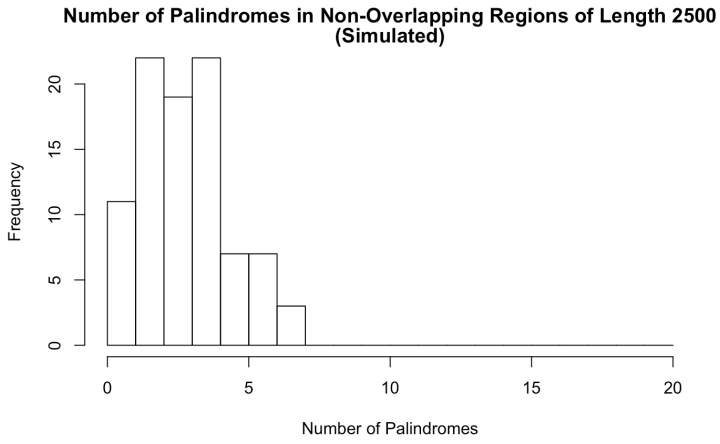


Hide

```
for (sample in uniform.samples) {
  hist(as.vector(table(cut(sample, breaks=seq(0,N,interval.length), include.lowest=TR
UE))), breaks=bins, main=title2, xlab=x.axis)
}
```







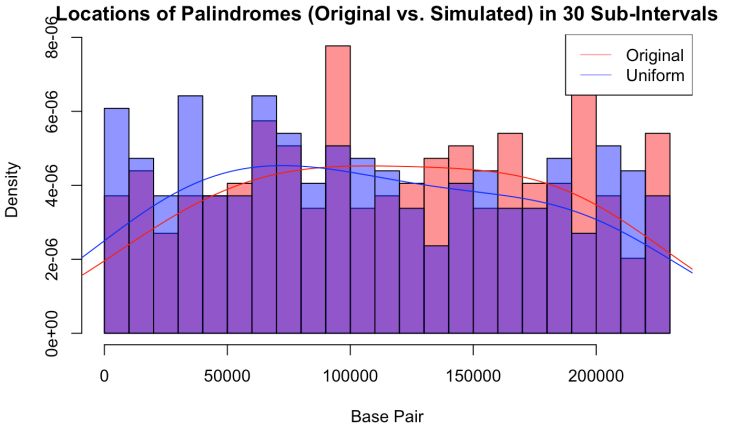
Scenario 2: Locations and Spacings

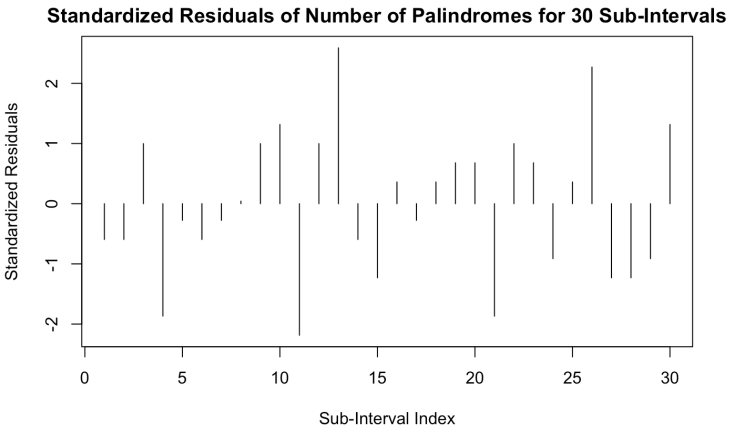
Use graphical methods to examine the spacings between consecutive palindromes and sum of consecutive pairs, triplets, etc, spacings. Compare what you find to what you would expect to find in a random scatter. Also, use graphical methods to compare locations of the palindromes.

Hide

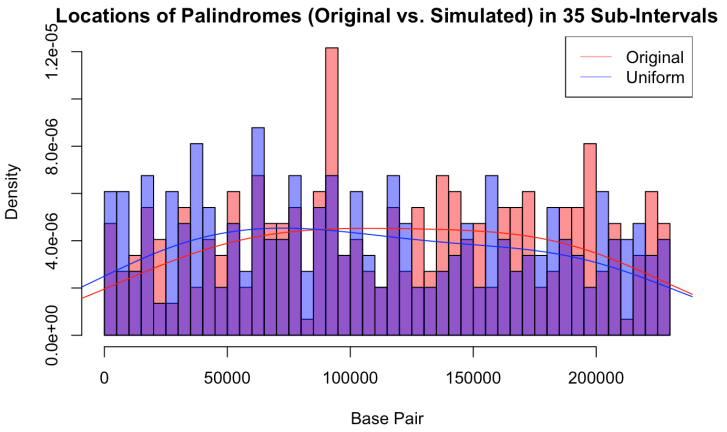
```
seed <- 0
uniform.sample <- runif(n, min=0, max=N)
for (num.regions in c(30,35,45,50,55,59)) {
  expected.counts <- rep(n/num.regions, num.regions)
  observed.counts <- as.vector(table(cut(locations, breaks=seq(0,N,length.out=num.reg
ions+1), include.lowest=TRUE)))

  hist(locations, breaks=num.regions, probability=TRUE, col=rgb(1,0,0,0.5), main=past
e('Locations of Palindromes (Original vs. Simulated) in',num.regions,'Sub-Intervals')
, xlab='Base Pair')
  hist(uniform.sample, breaks=num.regions, probability=TRUE, col=rgb(0,0,1,0.5), add=
TRUE)
  lines(density(locations, adjust=2), col=2)
  lines(density(uniform, adjust=2), col=4)
  legend('topright', legend=c('Original', 'Uniform'), lty=c(1,1), col=c(rgb(1,0,0,0.5
), rgb(0,0,1,0.5)))

  print(num.regions)
  print(chisq.test(observed.counts, p=expected.counts/n))

  residuals <- (observed.counts - expected.counts) / sqrt(expected.counts)
  plot(residuals, type='h', main=paste('Standardized Residuals of Number of Palindrom
es for',num.regions,'Sub-Intervals'), xlab='Sub-Interval Index', ylab='Standardized R
esiduals')
}
```

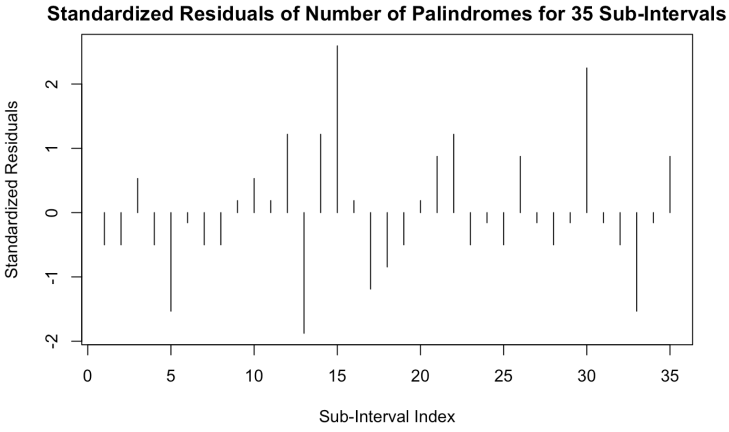
```
[1] 30

    Chi-squared test for given probabilities

data:  observed.counts
X-squared = 40.689, df = 29, p-value = 0.07328
```



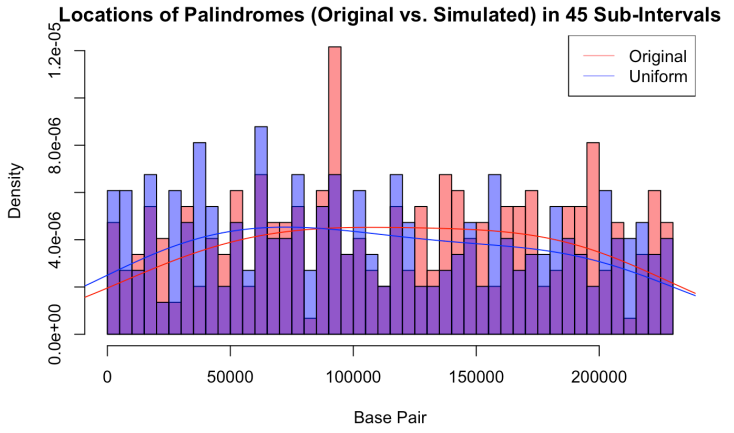


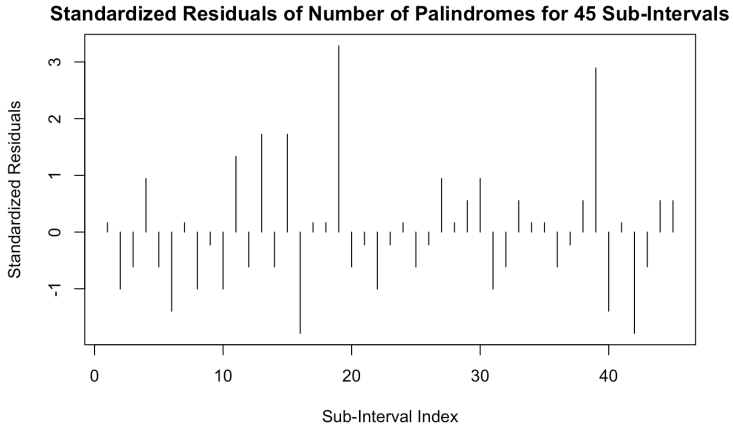
```
[1] 35

    Chi-squared test for given probabilities

data:  observed.counts
X-squared = 32.243, df = 34, p-value = 0.5539
```



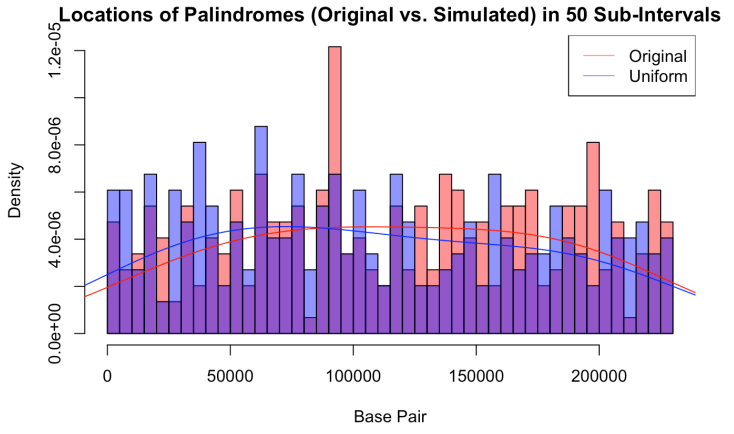


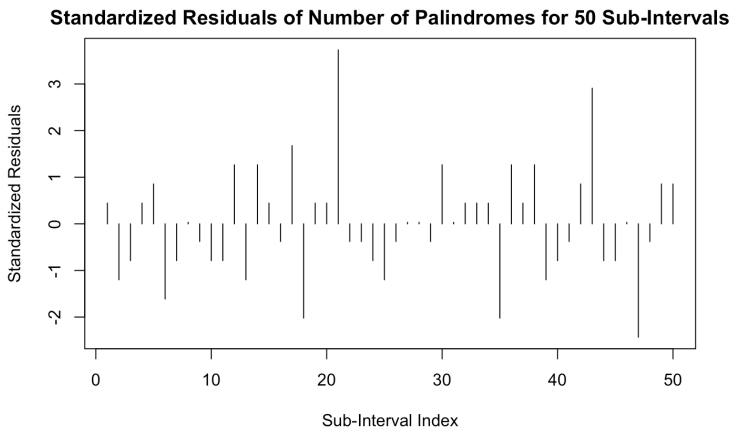

```
[1] 45

    Chi-squared test for given probabilities

data:  observed.counts
X-squared = 50.318, df = 44, p-value = 0.2376
```



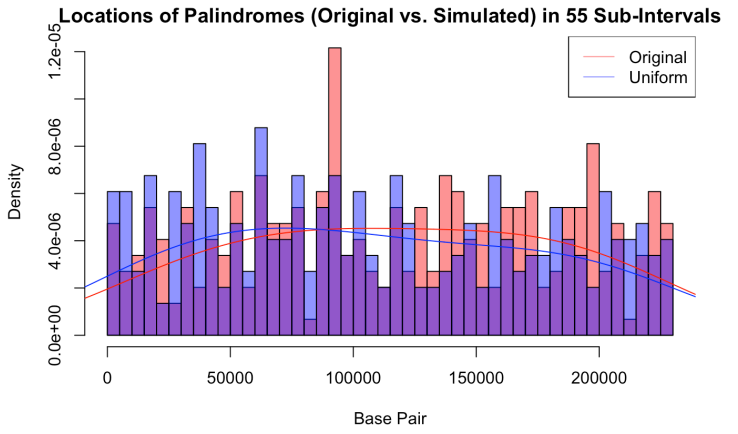


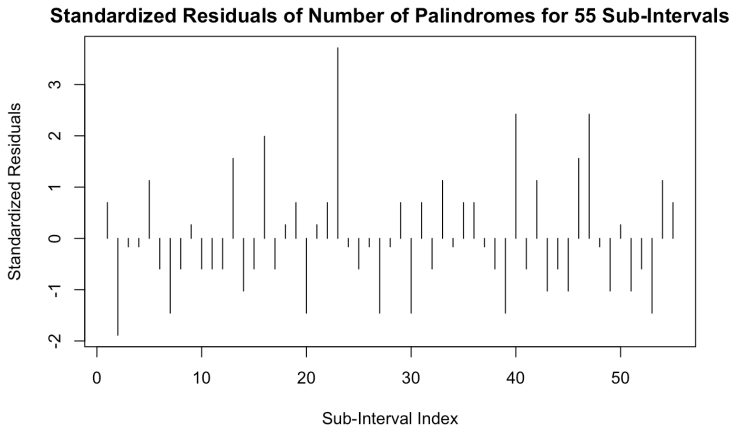

```
[1] 50

    Chi-squared test for given probabilities

data:  observed.counts
X-squared = 66.5, df = 49, p-value = 0.04864
```



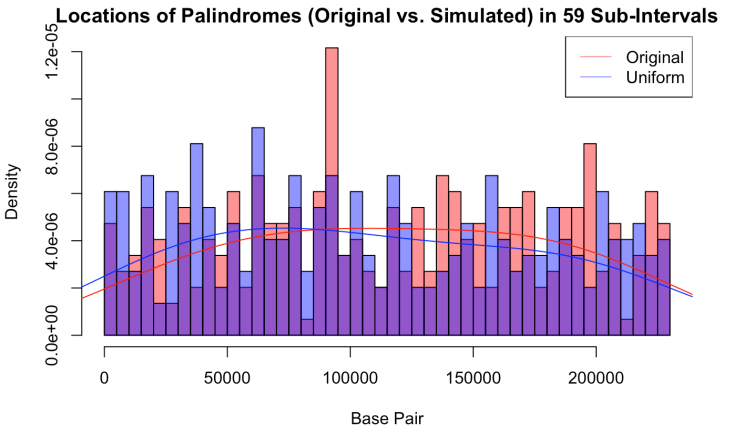


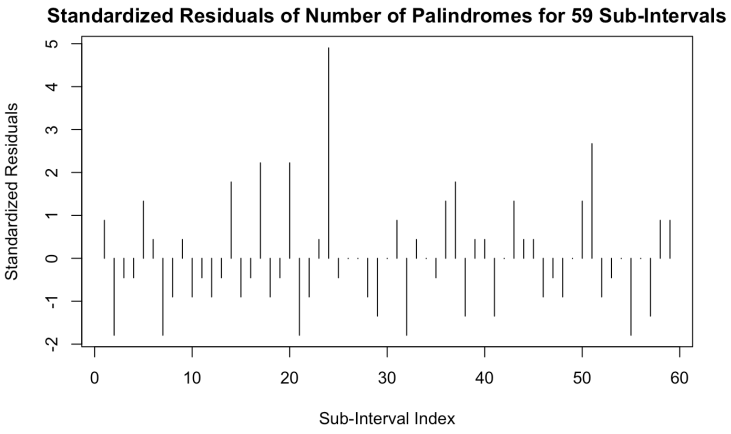

```
[1] 55

    Chi-squared test for given probabilities

data:  observed.counts
X-squared = 70.047, df = 54, p-value = 0.06997
```





```
[1] 59

    Chi-squared test for given probabilities

data:  observed.counts
X-squared = 92.682, df = 58, p-value = 0.002578
```

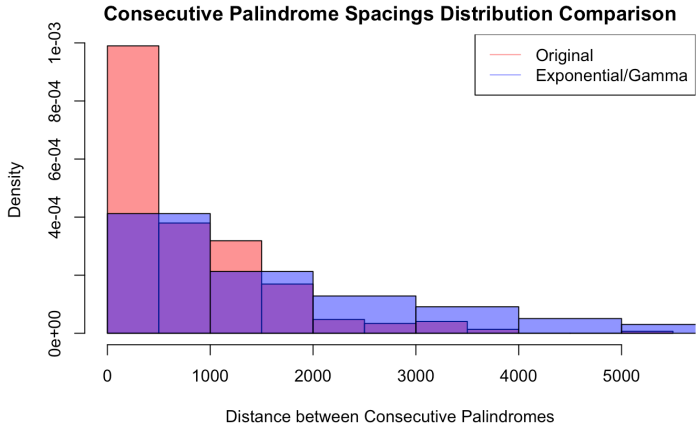




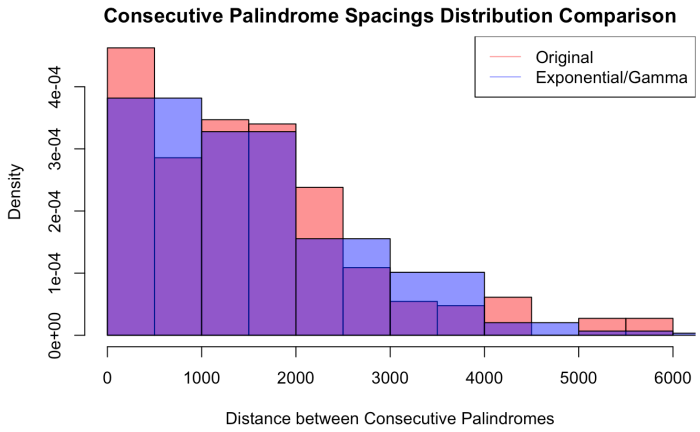
Hide

```
for (shape in 1:3) {
  spacings <- diff(locations, lag=shape)
  hist(probability=TRUE, spacings, col=rgb(1,0,0,0.5), main="Consecutive Palindrome S
pacings Distribution Comparison", xlab="Distance between Consecutive Palindromes")
  hist(probability=TRUE, rgamma(n, shape=shape, rate=lambda), col=rgb(0,0,1,0.5), add
=TRUE)
  legend('topright', legend=c('Original','Exponential/Gamma'), lty=c(1,1), col=c(rgb(
1,0,0,0.5), rgb(0,0,1,0.5)))

  width <- 500 * shape
  bins <- c(0,0.5*width,width,2*width,max(spacings))
  observed.spacings <- as.vector(table(cut(spacings, breaks=bins, include.lowest=TRUE
)))
  lambda <- 1/mean(spacings)
  probabilities <- c(0, pgamma(bins[2], shape=shape, rate=lambda), pgamma(bins[3], sh
ape=shape, rate=lambda), pgamma(bins[4], shape=shape, rate=lambda), 1)
  print(chisq.test(observed.spacings, p=diff(probabilities)))
  diff(probabilities)
}
```

```
    Chi-squared test for given probabilities

data:  observed.spacings
X-squared = 13.949, df = 3, p-value = 0.002975
```

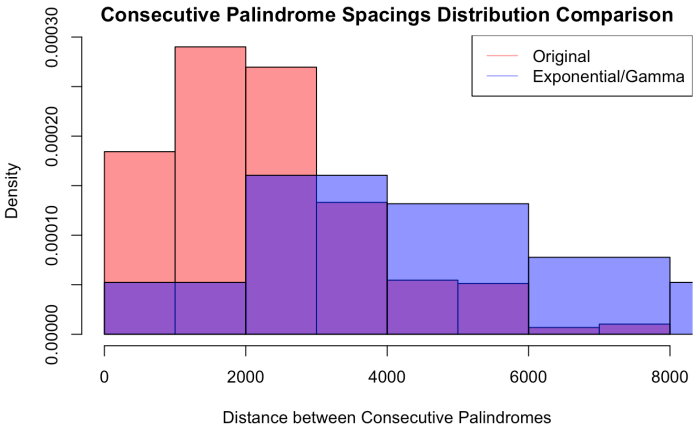



```
    Chi-squared test for given probabilities

data:  observed.spacings
X-squared = 329.54, df = 3, p-value < 2.2e-16
```




```
Chi-squared approximation may be incorrect
```

```
    Chi-squared test for given probabilities

data:  observed.spacings
X-squared = 1752.6, df = 3, p-value < 2.2e-16
```



Scenario 2: Locations and Spacings

Use graphical methods to examine the spacings between consecutive palindromes and sum of consecutive pairs, triplets, etc, spacings. Compare what you find to what you would expect to find in a random scatter. Also, use graphical methods to compare locations of the palindromes.

Hide

```
# Chi-square Goodness of Fit Test
# Case 1: k(number of sub-intervals)=20
k <- 20
locations.expected <- n/k
tab <- table(cut(locations, breaks=seq(0, N, length.out=k+1), include.lowest=TRUE))
locations.observed <- as.vector(tab)
chi_2 <- sum((locations.observed - locations.expected)^2/locations.expected)
chi2_compare <- qchisq(p=0.95, df=19)
p_value <- pchisq(chi_2, df=19, lower.tail=FALSE)
print(cat('\nWhen conducting chi_square Goodness of fit test comparing locations(divi
ded in 20 sub-intervals) against uniform distribution\n'))
```

```

When conducting chi_square Goodness of fit test comparing locations(divided in 20 sub
-intervals) against uniform distribution
NULL
```

Hide

```
print(paste('The value of chi_square statistic is', chi_2))
```

```
[1] "The value of chi_square statistic is 17.9189189189189"
```

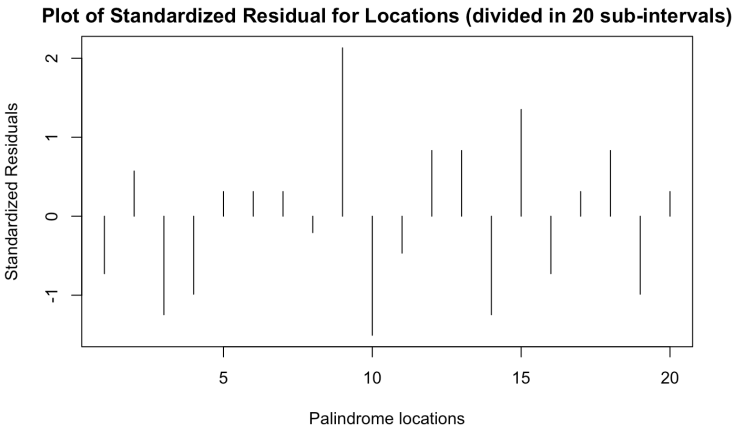
Hide

```
print(paste('The p_value is', p_value))
```

```
[1] "The p_value is 0.527860332119311"
```

Hide

```
## Visualization of the Residual
Residuals <- (locations.observed - locations.expected) / sqrt(locations.expected)
plot(Residuals, type='h', ylab='Standardized Residuals', xlab='Palindrome locations',
main='Plot of Standardized Residual for Locations (divided in 20 sub-intervals)')
```



Hide

```
# Case 2: k(number of sub-intervals)=30
k <- 30
locations.expected <- n/k
tab <- table(cut(locations, breaks=seq(0, N, length.out=k+1), include.lowest=TRUE))
locations.observed <- as.vector(tab)
chi_2 <- sum((locations.observed - locations.expected)^2/locations.expected)
chi2_compare <- qchisq(p=0.95, df=29)
p_value <- pchisq(chi_2, df=29, lower.tail=FALSE)
print(cat('\nWhen conducting chi_square Goodness of fit test comparing locations(divi
ded in 30 sub-intervals) against uniform distribution\n'))
```

```
When conducting chi_square Goodness of fit test comparing locations(divided in 30 sub
-intervals) against uniform distribution
NULL
```

Hide

```
print(paste('The value of chi_square statistic is', chi_2))
```

```
[1] "The value of chi_square statistic is 40.6891891891892"
```

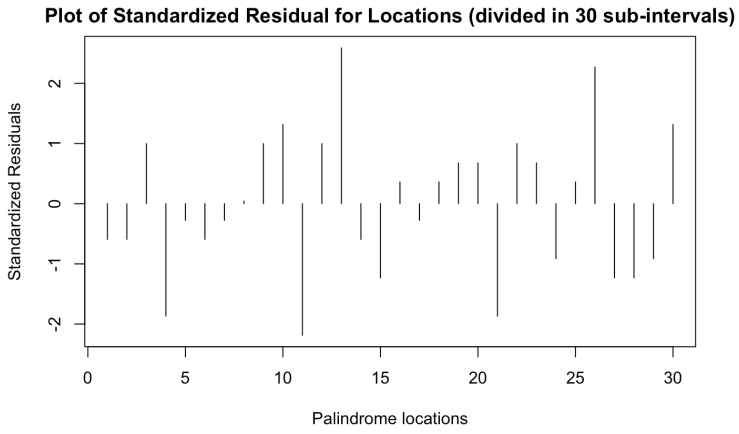
Hide

```
print(paste('The p_value is', p_value))
```

```
[1] "The p_value is 0.0732835870345071"
```

Hide

```
## Visualization of the Residual
Residuals <- (locations.observed - locations.expected) / sqrt(locations.expected)
plot(Residuals, type='h', ylab='Standardized Residuals', xlab='Palindrome locations',
main='Plot of Standardized Residual for Locations (divided in 30 sub-intervals)')
```



Hide

```
# Case 3: k(number of sub-intervals)=60
k <- 60
locations.expected <- n/k
tab <- table(cut(locations, breaks=seq(0, N, length.out=k+1), include.lowest=TRUE))
locations.observed <- as.vector(tab)
chi_2 <- sum((locations.observed - locations.expected)^2/locations.expected)
chi2_compare <- qchisq(p=0.95, df=59)
p_value <- pchisq(chi_2, df=59, lower.tail=FALSE)
print(cat('\nWhen conducting chi_square Goodness of fit test comparing locations(divi
ded in 60 sub-intervals) against uniform distribution\n'))
```

```
When conducting chi_square Goodness of fit test comparing locations(divided in 60 sub
-intervals) against uniform distribution
NULL
```

Hide

```
print(paste('The value of chi_square statistic is', chi_2))
```

```
[1] "The value of chi_square statistic is 79"
```

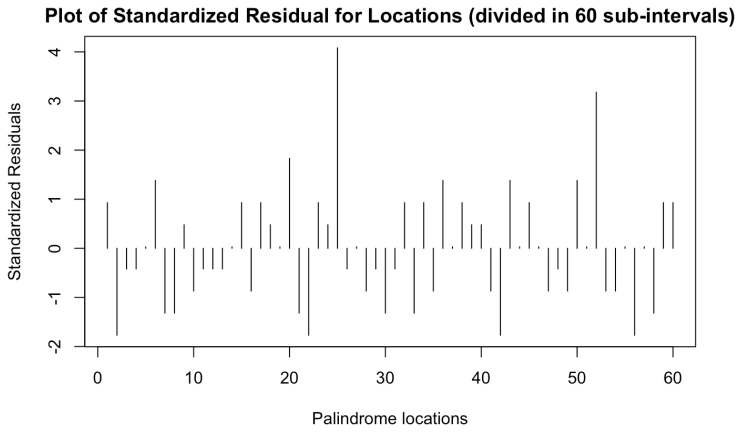
Hide

```
print(paste('The p_value is', p_value))
```

```
[1] "The p_value is 0.0421403871302519"
```

Hide

```
## Visualization of the Residual
Residuals <- (locations.observed - locations.expected) / sqrt(locations.expected)
plot(Residuals, type='h', ylab='Standardized Residuals', xlab='Palindrome locations',
main='Plot of Standardized Residual for Locations (divided in 60 sub-intervals)')
```



Hide

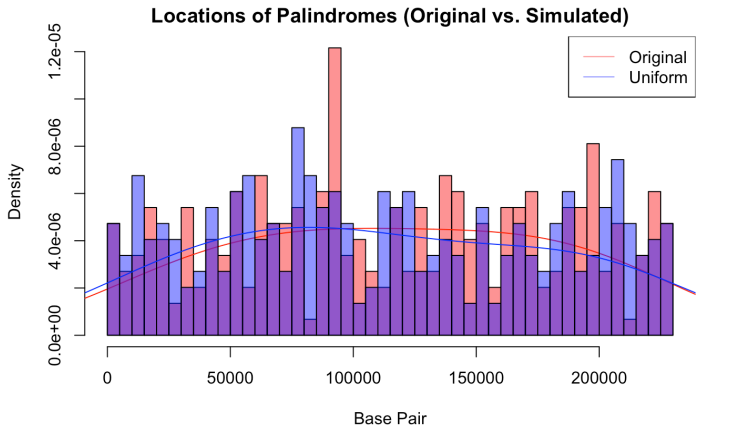
```
# Histogram of locations of palindromes in original data and uniform scatter
sample <- runif(n, min=0, max=N)
title <- 'Locations of Palindromes (Original vs. Simulated)'
x.axis <- 'Base Pair'
bins <- 35
hist(locations, breaks=bins, probability=TRUE, col=rgb(1,0,0,0.5), main=title, xlab=x
.axis)
lines(density(locations, adjust=2), col=2)
```

Hide

```
hist(sample, breaks=bins, probability=TRUE, col=rgb(0,0,1,0.5), add=TRUE)
lines(density(sample, adjust=2), col=4)
```

Hide

```
legend('topright', legend=c('Original', 'Uniform'), lty=c(1,1), col=c(rgb(1,0,0,0.5),
rgb(0,0,1,0.5)))
```



Hide

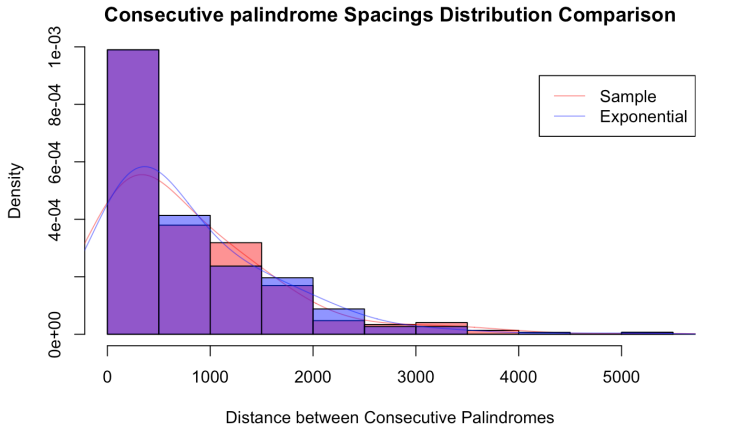
```
# Single Palindrome Spacing
locations.sorted = sort(locations, decreasing = FALSE)
distance.single <- abs(locations.sorted[-1]-locations.sorted[-length(locations.sorted
)])
# Histogram of spacings of palindromes in original data and exponential distribution
hist(distance.single, breaks= 15, col = rgb(1,0,0,0.5), probability = TRUE, main = "C
onsecutive palindrome Spacings Distribution Comparison", xlab = "Distance between Con
secutive Palindromes", ylim = c(0,0.001))
lines(density(distance.single, adjust = 2), col = rgb(1,0,0,0.5))
```

Hide

```
Expo <- rexp(n-1, rate = 1/mean(distance.single))
hist(Expo, breaks = 15, col = rgb(0,0,1,0.5), probability = TRUE, add = TRUE)
lines(density(Expo, adjust = 2), col = rgb(0,0,1,0.5))
```

Hide

```
legend(x = 4200, y = 0.0009, legend = c("Sample", "Exponential"), lty = c(1,1), col =
c(rgb(1,0,0,0.5), rgb(0,0,1,0.5)))
```



Hide

```
# Chi-square Goodness of Fit Test
# Case 1: Divided in 7 intervals
# Construct expected number of intervals
spacings.observed <- sort(distance.single, decreasing = FALSE)
lambda <- 1/mean(distance.single)
spacings.intervals <- as.numeric(quantile(spacings.observed, probs = c(0,0.05, 0.1, 0
.3, 0.5, 0.7, 0.9,1)))
spacings.expected <- (n-1)*(exp(-lambda*spacings.intervals[-length(spacings.intervals
)])-exp(-lambda*spacings.intervals[-1]))
spacings.expected[length(spacings.expected)] <- n-sum(spacings.expected[1:length(spac
ings.expected)-1])
spacings.observed <- as.numeric(table(cut(distance.single, breaks=spacings.intervals,
include.lowest=TRUE)))
contingency_7 <- data.frame(spacings.intervals[-1],spacings.observed,spacings.expecte
d)
contingency_7
```

spacings.intervals..1.	spacings.observed	spacings.expected
<dbl>	<dbl>	<dbl>
18.8	15	6.685271
40.4	15	7.909063
214.4	59	56.279224
512.0	59	71.307030
1037.6	58	75.036667
1760.0	60	46.909310
5333.0	29	31.873435

7 rows

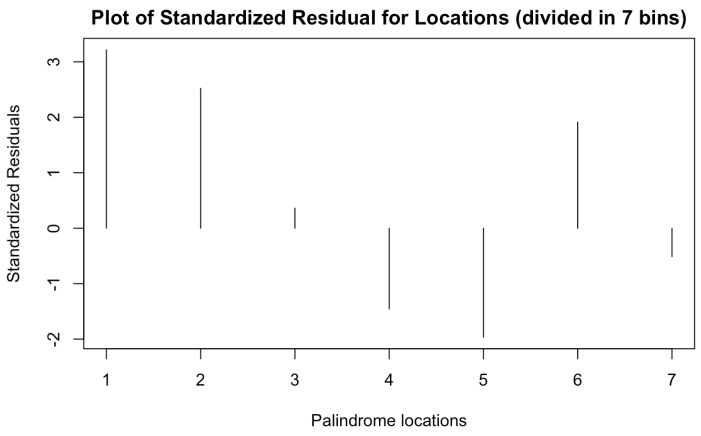
Hide

```
chi_2 <- sum((spacings.observed - spacings.expected)^2/spacings.expected)
chi2_compare <- qchisq(p=0.95, df=5)
p_value <- pchisq(chi_2, df=5, lower.tail=FALSE)
print(paste('The p_value when the distance is splited into 7 sub-intervals is', p_val
ue))
```

```
[1] "The p_value when the distance is splited into 7 sub-intervals is 6.4244787534794
9e-05"
```

Hide

```
## Visualization of the Residual
Residuals <- (spacings.observed - spacings.expected) / sqrt(spacings.expected)
plot(Residuals, type='h', ylab='Standardized Residuals', xlab='Palindrome locations',
main='Plot of Standardized Residual for Locations (divided in 7 bins)')
```



Hide

```
# Case 2: Divided in 10 intervals
# Construct expected number of intervals
spacings.observed <- sort(distance.single, decreasing = FALSE)
spacings.intervals <- as.numeric(quantile(spacings.observed, probs = c(seq(0,1, by =
0.1))))
spacings.expected <- (n-1)*(exp(-lambda*spacings.intervals[-length(spacings.intervals
)])-exp(-lambda*spacings.intervals[-1]))
spacings.expected[length(spacings.expected)] <- n-sum(spacings.expected[1:length(spac
ings.expected)-1])
spacings.observed <- as.numeric(table(cut(distance.single, breaks = spacings.interval
s, include.lowest = TRUE)))
contingency_10 <- data.frame(spacings.intervals[-1],spacings.observed,spacings.expect
ed)
contingency_10
```

spacings.intervals..1. <dbl>	spacings.observed <dbl>	spacings.expected <dbl>
40.4	30	14.59433
108.2	29	23.44190
214.4	30	32.83732
328.2	29	30.53758
512.0	30	40.76945
718.0	29	35.56073
1037.6	29	39.47594
1275.8	30	20.46998
1760.0	30	26.43933
5333.0	29	31.87343

1-10 of 10 rows

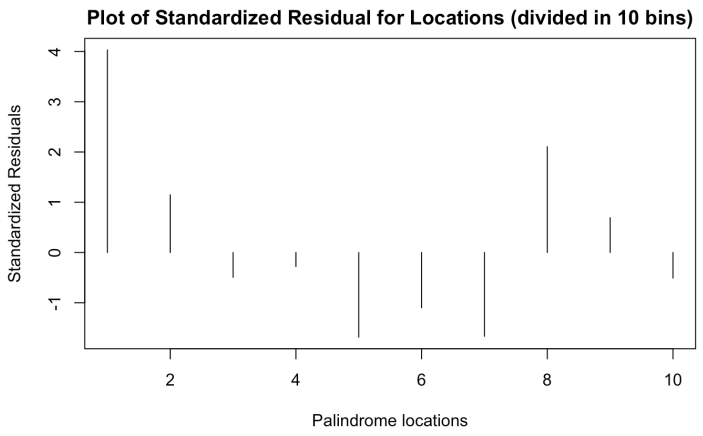
Hide

```
chi_2 <- sum((spacings.observed - spacings.expected)^2/spacings.expected)
chi2_compare <- qchisq(p=0.95, df=8)
p_value <- pchisq(chi_2, df=8, lower.tail=FALSE)
print(paste('The p_value when the distance is splited into 10 sub-intervals is', p_va
lue))
```

```
[1] "The p_value when the distance is splited into 10 sub-intervals is 0.000218981752
365422"
```

Hide

```
## Visualization of the Residual
Residuals <- (spacings.observed - spacings.expected) / sqrt(spacings.expected)
plot(Residuals, type='h', ylab='Standardized Residuals', xlab='Palindrome locations',
main='Plot of Standardized Residual for Locations (divided in 10 bins)')
```



Hide

```
# Case 3: Divided in 20 intervals
# Construct expected number of intervals
spacings.observed <- sort(distance.single, decreasing = FALSE)
spacings.intervals <- as.numeric(quantile(spacings.observed, probs = c(seq(0,1, by =
0.05))))
spacings.expected <- (n-1)*(exp(-lambda*spacings.intervals[-length(spacings.intervals
)])-exp(-lambda*spacings.intervals[-1]))
spacings.expected[length(spacings.expected)] <- n-sum(spacings.expected[1:length(spac
ings.expected)-1])
spacings.observed <- as.numeric(table(cut(distance.single, breaks = spacings.interval
s, include.lowest = TRUE)))
contingency_20 <- data.frame(spacings.intervals[-1],spacings.observed,spacings.expect
ed)
contingency_20
```

spacings.intervals..1. <dbl>	spacings.observed <dbl>	spacings.expected <dbl>
18.8	15	6.685271
40.4	15	7.909063
73.2	15	11.596614
108.2	14	11.845286
160.0	15	16.578556
214.4	15	16.258767
260.5	14	12.912907
328.2	15	17.624677
423.8	15	22.407975
512.0	15	18.361471

1-10 of 20 rows Previous **1** 2 Next

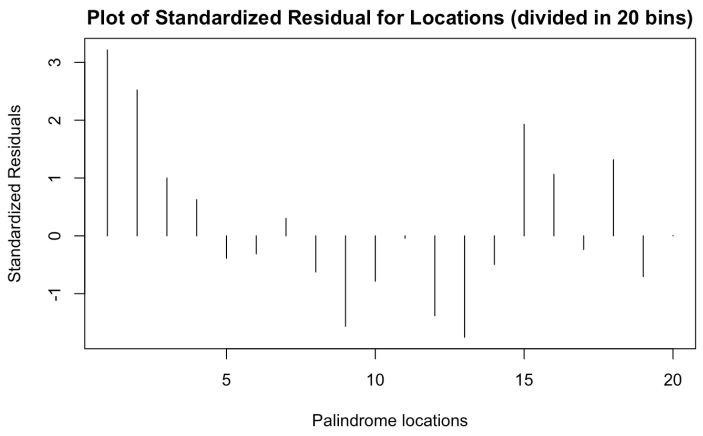
Hide

```
chi_2 <- sum((spacings.observed - spacings.expected)^2/spacings.expected)
chi2_compare <- qchisq(p=0.95, df=18)
p_value <- pchisq(chi_2, df=18, lower.tail=FALSE)
print(paste('The p_value when the distance is splited into 20 sub-intervals is', p_va
lue))
```

```
[1] "The p_value when the distance is splited into 20 sub-intervals is 0.011702069613
4169"
```

Hide

```
## Visualization of the Residual
Residuals <- (spacings.observed - spacings.expected) / sqrt(spacings.expected)
plot(Residuals, type='h', ylab='Standardized Residuals', xlab='Palindrome locations',
main='Plot of Standardized Residual for Locations (divided in 20 bins)')
```



Hide

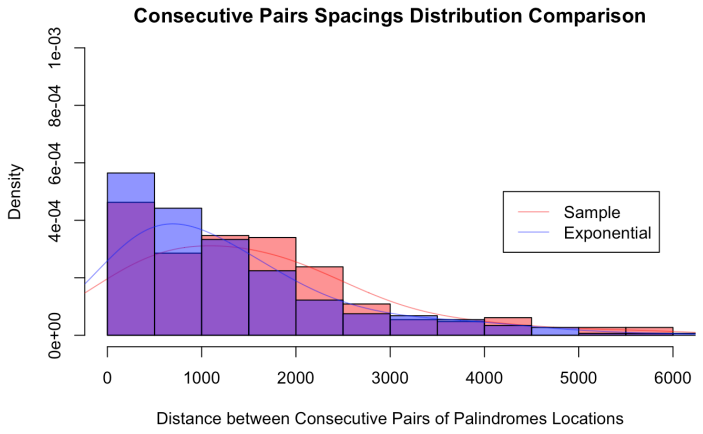
```
# Consecutive Pairs
locations.sorted <-  sort(locations, decreasing = FALSE)
locations.pairs <- locations.sorted[-length(locations.sorted)]
distance.pairs <- abs(locations.sorted[-1][-1]-locations.pairs[-length(locations.pair
s)])
# Histogram of spacings of palindromes in original data and exponential distribution
hist(distance.pairs, breaks= 15, col = rgb(1,0,0,0.5), probability = TRUE, main = "Co
nsecutive Pairs Spacings Distribution Comparison", xlab = "Distance between Consecuti
ve Pairs of Palindromes Locations", ylim = c(0,0.001))
lines(density(distance.pairs, adjust = 2), col = rgb(1,0,0,0.5))
```

Hide

```
Expo <- rexp(n-2, rate = 1/mean(distance.pairs))
hist(Expo, breaks = 15, col = rgb(0,0,1,0.5), probability = TRUE, add = TRUE)
lines(density(Expo, adjust = 2), col = rgb(0,0,1,0.5))
```

Hide

```
legend(x = 4200, y = 0.0005, legend = c("Sample", "Exponential"), lty = c(1,1), col =
c(rgb(1,0,0,0.5), rgb(0,0,1,0.5)))
```



Hide

```
# Chi-square Goodness of Fit Test
# Case 1: Divided in 7 intervals
# Construct expected number of intervals
spacings.observed <- sort(distance.pairs, decreasing = FALSE)
lambda <- 1/mean(distance.pairs)
spacings.intervals <- as.numeric(quantile(spacings.observed, probs = c(0,0.05, 0.1, 0
.3, 0.5, 0.7, 0.9,1)))
spacings.expected <- (n-2)*(exp(-lambda*spacings.intervals[-length(spacings.intervals
)])-exp(-lambda*spacings.intervals[-1]))
spacings.expected[length(spacings.expected)] <- n-sum(spacings.expected[1:length(spac
ings.expected)-1])
spacings.observed <- as.numeric(table(cut(distance.pairs, breaks = spacings.intervals
, include.lowest = TRUE)))
contingency_7 <- data.frame(spacings.intervals[-1],spacings.observed,spacings.expecte
d)
contingency_7
```

spacings.intervals..1.	spacings.observed	spacings.expected
<dbl>	<dbl>	<dbl>
111.65	15	14.23422
231.00	16	20.26702

667.60	57	62.16165
1386.00	59	70.87655
1958.20	59	37.11355
3105.70	58	43.48249
5926.00	30	47.86451

7 rows

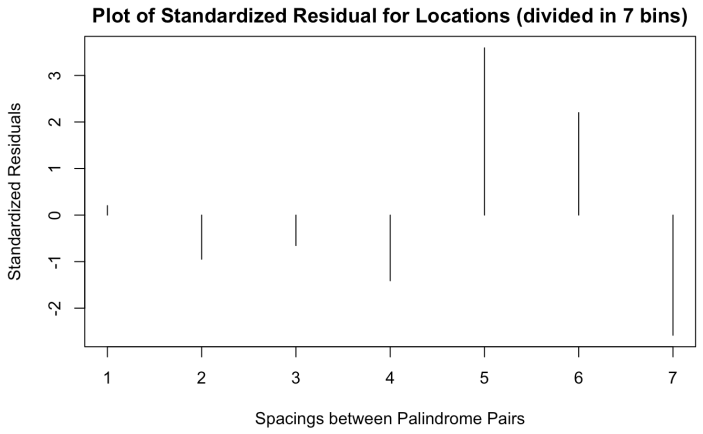
Hide

```
chi_2 <- sum((spacings.observed - spacings.expected)^2/spacings.expected)
chi2_compare <- qchisq(p=0.95, df=5)
p_value <- pchisq(chi_2, df=5, lower.tail=FALSE)
print(paste('The p_value when the distance is splited into 7 sub-intervals is', p_val
ue))
```

```
[1] "The p_value when the distance is splited into 7 sub-intervals is 4.0192806354887
e-05"
```

Hide

```
## Visualization of the Residual
Residuals <- (spacings.observed - spacings.expected) / sqrt(spacings.expected)
plot(Residuals, type='h', ylab='Standardized Residuals', xlab='Spacings between Palin
drome Pairs', main='Plot of Standardized Residual for Locations (divided in 7 bins)')
```



Hide

```
# Case 2: Divided in 10 intervals
# Construct expected number of intervals
spacings.observed <- sort(distance.pairs, decreasing = FALSE)
spacings.intervals <- as.numeric(quantile(spacings.observed, probs = c(seq(0,1, by =
0.1))))
spacings.expected <- (n-2)*(exp(-lambda*spacings.intervals[-length(spacings.intervals
)])-exp(-lambda*spacings.intervals[-1]))
spacings.expected[length(spacings.expected)] <- n-sum(spacings.expected[1:length(spac
ings.expected)-1])
spacings.observed <- as.numeric(table(cut(distance.pairs, breaks = spacings.intervals
, include.lowest = TRUE)))
contingency_10 <- data.frame(spacings.intervals[-1],spacings.observed,spacings.expect
ed)
contingency_10
```

spacings.intervals..1. <dbl>	spacings.observed <dbl>	spacings.expected <dbl>
231.0	31	34.50124
444.2	28	32.54013
667.6	29	29.62152
1080.8	30	44.71402

1386.0	29	26.16253
1699.4	29	22.00917
1958.2	30	15.10439
2353.8	29	18.72519
3105.7	29	24.75731
5926.0	30	47.86451

1-10 of 10 rows

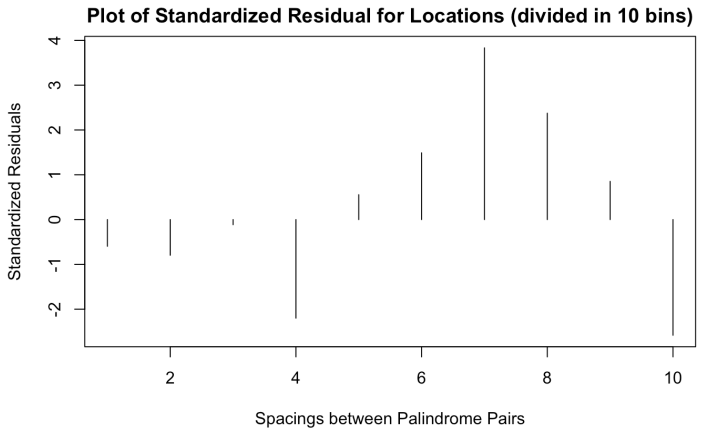
Hide

```
chi_2 <- sum((spacings.observed - spacings.expected)^2/spacings.expected)
chi2_compare <- qchisq(p=0.95, df=8)
p_value <- pchisq(chi_2, df=8, lower.tail=FALSE)
print(paste('The p_value when the distance is splited into 10 sub-intervals is', p_va
lue))
```

```
[1] "The p_value when the distance is splited into 10 sub-intervals is 1.687537955415
38e-05"
```

Hide

```
## Visualization of the Residual
Residuals <- (spacings.observed - spacings.expected) / sqrt(spacings.expected)
plot(Residuals, type='h', ylab='Standardized Residuals', xlab='Spacings between Palin
drome Pairs', main='Plot of Standardized Residual for Locations (divided in 10 bins)'
)
```



Hide

```
# Case 3: Divided in 20 intervals
# Construct expected number of intervals
spacings.observed <- sort(distance.pairs, decreasing = FALSE)
spacings.intervals <- as.numeric(quantile(spacings.observed, probs = c(seq(0,1, by =
0.05))))
spacings.expected <- (n-2)*(exp(-lambda*spacings.intervals[-length(spacings.intervals
)])-exp(-lambda*spacings.intervals[-1]))
spacings.expected[length(spacings.expected)] <- n-sum(spacings.expected[1:length(spac
ings.expected)-1])
spacings.observed <- as.numeric(table(cut(distance.pairs, breaks = spacings.intervals
, include.lowest = TRUE)))
contingency_20 <- data.frame(spacings.intervals[-1],spacings.observed,spacings.expect
ed)
contingency_20
```

spacings.intervals..1. <dbl>	spacings.observed <dbl>	spacings.expected <dbl>
111.65	15	14.234217
231.00	16	20.267025
314.90	13	13.341731
444.20	15	19.198401

559.25	15	15.787312
667.60	14	13.834207
861.85	15	22.506393
1080.80	15	22.207623
1215.85	14	12.213892
1386.00	15	13.948640

1-10 of 20 rows Previous **1** 2 Next

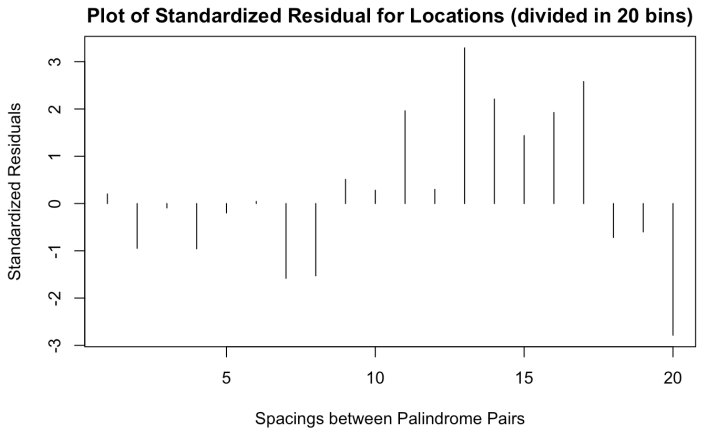
Hide

```
chi_2 <- sum((spacings.observed - spacings.expected)^2/spacings.expected)
chi2_compare <- qchisq(p=0.95, df=18)
p_value <- pchisq(chi_2, df=18, lower.tail=FALSE)
print(paste('The p_value when the distance is splited into 20 sub-intervals is', p_va
lue))
```

```
[1] "The p_value when the distance is splited into 20 sub-intervals is 0.000159225622
786541"
```

Hide

```
## Visualization of the Residual
Residuals <- (spacings.observed - spacings.expected) / sqrt(spacings.expected)
plot(Residuals, type = 'h', ylab = "Standardized Residuals", xlab = "Spacings between
Palindrome Pairs", main = "Plot of Standardized Residual for Locations (divided in 20
bins)")
```



Hide

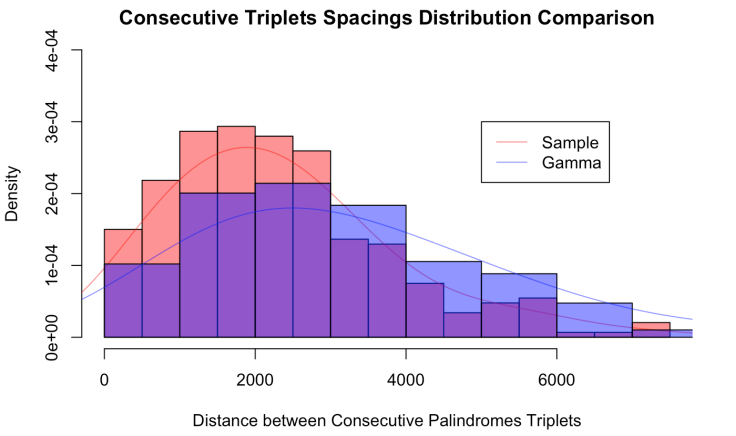
```
# Consecutive Triplets
locations.sorted <-  sort(locations, decreasing = FALSE)
locations.triplets <- locations.sorted[-length(locations.sorted)]
locations.triplets <- locations.triplets[-length(locations.triplets)]
distance.triplets <- abs(locations.sorted[-1][-1][-1]-locations.triplets[-length(loca
tions.triplets)])
# Histogram of spacings of palindromes in original data and exponential distribution
hist(distance.triplets, breaks= 15, col = rgb(1,0,0,0.5), probability = TRUE, main =
"Consecutive Triplets Spacings Distribution Comparison", xlab = "Distance between Con
secutive Palindromes Triplets", ylim = c(0,0.0004))
lines(density(distance.triplets, adjust = 2), col = rgb(1,0,0,0.5))
```

Hide

```
Gam <- rgamma(n-2, 2, rate = 1/mean(distance.pairs))
hist(Gam, breaks = 15, col = rgb(0,0,1,0.5), probability = TRUE, add = TRUE)
lines(density(Gam, adjust = 2), col = rgb(0,0,1,0.5))
```

Hide

```
legend(x = 5000, y = 0.0003, legend = c("Sample", "Gamma"), lty = c(1,1), col = c(rgb
(1,0,0,0.5), rgb(0,0,1,0.5)))
```



Hide

```
# Chi-square Goodness of Fit Test (Need to be changed)
# Case 1: Divided in 7 intervals
# Construct expected number of intervals
spacings.observed <- sort(distance.triplets, decreasing = FALSE)
lambda <- 2/mean(distance.pairs)
spacings.intervals <- as.numeric(quantile(spacings.observed, probs = c(0,0.05, 0.1, 0
.3, 0.5, 0.7, 0.9,1)))
spacings.expected <- (n-3)*(exp(-lambda*spacings.intervals[-length(spacings.intervals
)])-exp(-lambda*spacings.intervals[-1]))
spacings.expected[length(spacings.expected)] <- n-sum(spacings.expected[1:length(spac
ings.expected)-1])
spacings.observed <- as.numeric(table(cut(distance.pairs, breaks = spacings.intervals
, include.lowest = TRUE)))
contingency_7 <- data.frame(spacings.intervals[-1],spacings.observed,spacings.expecte
d)
contingency_7
```

spacings.intervals..1. <dbl>	spacings.observed <dbl>	spacings.expected <dbl>
380.0	41	84.117194
626.4	30	48.864084
1429.8	73	84.272650
2078.0	59	26.255223
2829.0	46	12.460369
4311.8	24	6.497782
7488.0	10	33.532698

7 rows

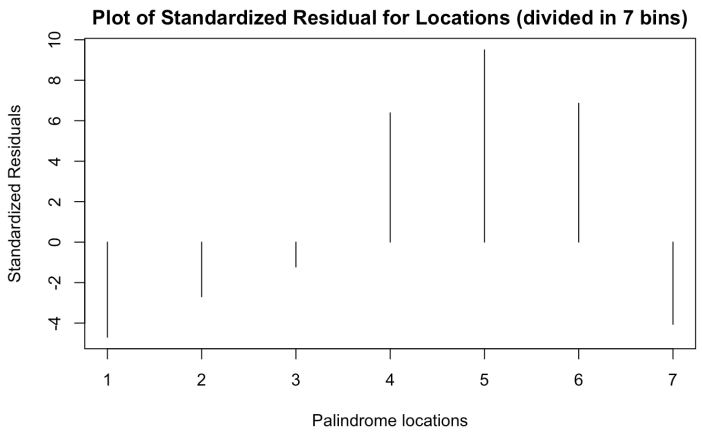
Hide

```
chi_2 <- sum((spacings.observed - spacings.expected)^2/spacings.expected)
chi2_compare <- qchisq(p = 0.95, df = 5)
p_value <- pchisq(chi_2, df = 5, lower.tail = FALSE)
print(paste("The p_value when the distance is splited into 7 sub-intervals is", p_val
ue))
```

```
[1] "The p_value when the distance is splited into 7 sub-intervals is 9.0742150582863
9e-47"
```

Hide

```
## Visualization of the Residual
Residuals <- (spacings.observed - spacings.expected) / sqrt(spacings.expected)
plot(Residuals, type = 'h', ylab = "Standardized Residuals", xlab = "Palindrome locat
ions", main = "Plot of Standardized Residual for Locations (divided in 7 bins)")
```



Hide

```
# Case 2: Divided in 10 intervals
# Construct expected number of intervals
spacings.observed <- sort(distance.pairs, decreasing = FALSE)
spacings.intervals <- as.numeric(quantile(spacings.observed, probs = c(seq(0,1, by =
0.1))))
spacings.expected <- (n-3)*(exp(-lambda*spacings.intervals[-length(spacings.intervals
)])-exp(-lambda*spacings.intervals[-1]))
spacings.expected[length(spacings.expected)] <- n-sum(spacings.expected[1:length(spac
ings.expected)-1])
spacings.observed <- as.numeric(table(cut(distance.pairs, breaks = spacings.intervals
, include.lowest = TRUE)))
contingency_10 <- data.frame(spacings.intervals[-1],spacings.observed,spacings.expect
ed)
contingency_10
```

spacings.intervals..1. <dbl>	spacings.observed <dbl>	spacings.expected <dbl>
231.0	31	63.284741
444.2	28	52.292580
667.6	29	41.360617
1080.8	30	51.167191
1386.0	29	23.652613
1699.4	29	16.303784
1958.2	30	9.288671
2353.8	29	9.368023
3105.7	29	8.736691
5926.0	30	20.545090

1-10 of 10 rows

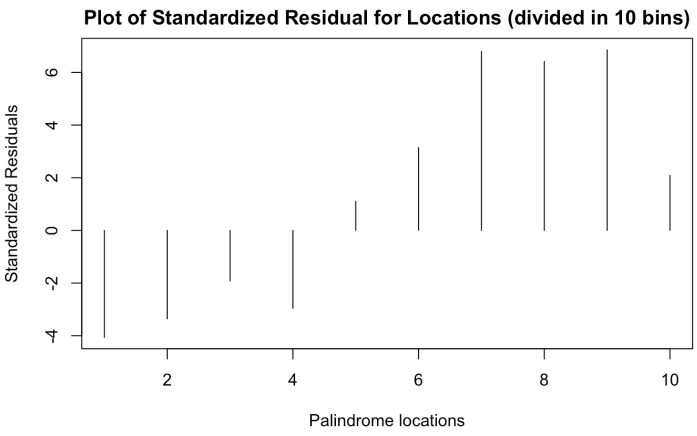
Hide

```
chi_2 <- sum((spacings.observed - spacings.expected)^2/spacings.expected)
chi2_compare <- qchisq(p = 0.95, df = 8)
p_value <- pchisq(chi_2, df = 8, lower.tail = FALSE)
print(paste("The p_value when the distance is splited into 10 sub-intervals is", p_va
lue))
```

```
[1] "The p_value when the distance is splited into 10 sub-intervals is 8.252676182357
e-37"
```

Hide

```
## Visualization of the Residual
Residuals <- (spacings.observed - spacings.expected) / sqrt(spacings.expected)
plot(Residuals, type = 'h', ylab = "Standardized Residuals", xlab = "Palindrome locat
ions", main = "Plot of Standardized Residual for Locations (divided in 10 bins)")
```



Hide

```
# Case 3: Divided in 20 intervals
# Construct expected number of intervals
spacings.observed <- sort(distance.pairs, decreasing = FALSE)
spacings.intervals <- as.numeric(quantile(spacings.observed, probs = c(seq(0,1, by =
0.05))))
spacings.expected <- (n-3)*(exp(-lambda*spacings.intervals[-length(spacings.intervals
)])-exp(-lambda*spacings.intervals[-1]))
spacings.expected[length(spacings.expected)] <- n-sum(spacings.expected[1:length(spac
ings.expected)-1])
spacings.observed <- as.numeric(table(cut(distance.pairs, breaks = spacings.intervals
, include.lowest = TRUE)))
contingency_20 <- data.frame(spacings.intervals[-1],spacings.observed,spacings.expect
ed)
contingency_20
```

spacings.intervals..1. <dbl>	spacings.observed <dbl>	spacings.expected <dbl>
111.65	15	27.087363
231.00	16	36.197378
314.90	13	22.308664
444.20	15	29.983916
559.25	15	22.784219
667.60	14	18.576399
861.85	15	27.448803
1080.80	15	23.718388
1215.85	14	11.619654
1386.00	15	12.032959

1-10 of 20 rows Previous **1** 2 Next

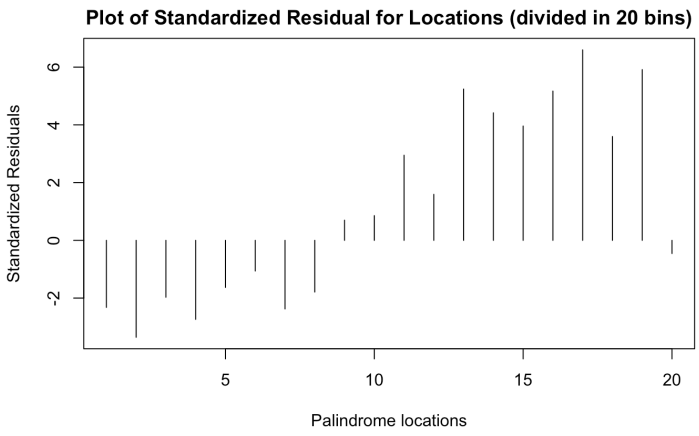
Hide

```
chi_2 <- sum((spacings.observed - spacings.expected)^2/spacings.expected)
chi2_compare <- qchisq(p = 0.95, df = 18)
p_value <- pchisq(chi_2, df = 18, lower.tail = FALSE)
print(paste("The p_value when the distance is splited into 20 sub-intervals is", p_va
lue))
```

```
[1] "The p_value when the distance is splited into 20 sub-intervals is 1.344212750186
16e-39"
```

Hide

```
## Visualization of the Residual
Residuals <- (spacings.observed - spacings.expected) / sqrt(spacings.expected)
plot(Residuals, type = 'h', ylab = "Standardized Residuals", xlab = "Palindrome locat
ions", main = "Plot of Standardized Residual for Locations (divided in 20 bins)")
```



Scenario 3: Counts

Use graphical methods and more formal statistical tests to examine the counts of palindromes in various regions of the DNA. Split the DNA into nonoverlapping regions of equal length to compare the number of palindomres in an interval to the number of that you would expect from uniform random scatter. The counts for shorter regions will be more variable than those for logner regions. Also, consider classifying the regions according to the number of counts.

Hide

```
regionsplit <- function(n.region, gene, site){
  count.int <- table(cut(site, breaks = seq(1, length(gene), length.out=n.region+1),
include.lowest=TRUE))
  count.vector <- as.vector(count.int)
  count.tab <- table(count.vector)
  return (count.tab)
}
```

Hide

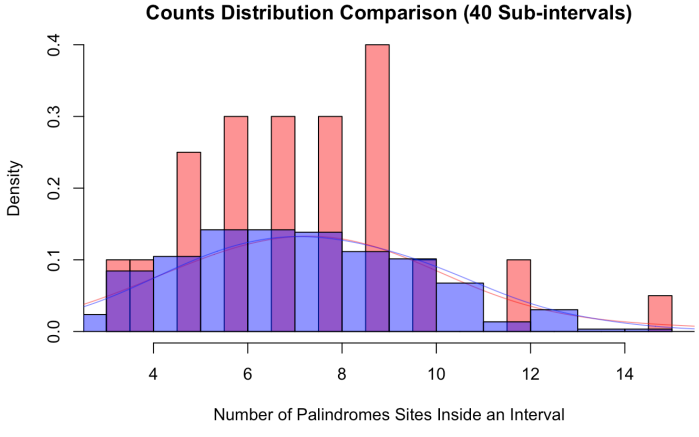
```
# Case 1: divided by 40 intervals
n.region <- 40
gene <- seq(1,N)
observed <- as.numeric(regionsplit(n.region, gene, locations))
interval <- as.numeric(names(regionsplit(n.region, gene, locations)))
lambda <- n/n.region
# Histogram of counts of palindromes in original data and poisson distribution
counts <- as.vector(table(cut(locations, breaks = seq(0, N, length.out = n.region+1),
include.lowest = TRUE)))
hist(counts, breaks = bins, col = rgb(1,0,0,0.5), probability = TRUE, main = "Counts
Distribution Comparison (40 Sub-intervals)", xlab = "Number of Palindromes Sites Insi
de an Interval", ylim = c(0,0.4))
lines(density(counts, adjust = 2), col = rgb(1,0,0,0.5))
```

Hide

```
Pois <- rpois(n, lambda)
hist(Pois, breaks = 15, col = rgb(0,0,1,0.5), probability = TRUE, add = TRUE)
lines(density(Pois, adjust = 2), col = rgb(0,0,1,0.5))
```

Hide

```
legend(x = 18, y = 0.17, legend = c("Sample", "Poisson"), lty = c(1,1), col = c(rgb(1
,0,0,0.5), rgb(0,0,1,0.5)))
```



Hide

Hide

```
# Chi-sqr Goodness of Fit test
expected <- n.region*exp(-lambda)* lambda**(interval)/factorial(interval)
for (i in c(0:2)){
  expect <- n.region*exp(-lambda)* lambda**(i)/factorial(i)
  expected[1] <- expected[1]+ expect
}
expect <- n.region*exp(-lambda)* lambda**(11)/factorial(11)
expected[8] <- expected[8]+ expect
expected[10] <- 0
for (i in c(1:12)){
  expect <- exp(-lambda)* lambda**(i)/factorial(i)
  expected[10] <- expected[10]+ expect
}
expected[10] <- (1-expected[10])*n.region
counts.expected <- c()
counts.interval <- c()
counts.observed <- c()
# Group bins
counts.expected[1] <- sum(expected[1:2])
counts.expected[2] <- sum(expected[3:4])
counts.expected[3] <- sum(expected[5])
counts.expected[4] <- sum(expected[6:7])
counts.expected[5] <- sum(expected[8:10])
counts.observed[1] <- sum(observed[1:2])
counts.observed[2] <- sum(observed[3:4])
counts.observed[3] <- sum(observed[5])
counts.observed[4] <- sum(observed[6:7])
counts.observed[5] <- sum(observed[8:10])
counts.interval[1] <- interval[2]
counts.interval[2] <- interval[4]
counts.interval[3] <- interval[5]
counts.interval[4] <- interval[7]
counts.interval[5] <- interval[7]+1
counts.table40 <- data.frame(counts.interval,counts.observed,counts.expected)
counts.table40
```

counts.interval <dbl>	counts.observed <dbl>	counts.expected <dbl>
4	5	5.581016
6	10	10.097450
7	6	5.894844
9	14	9.936087
10	5	8.515052

5 rows

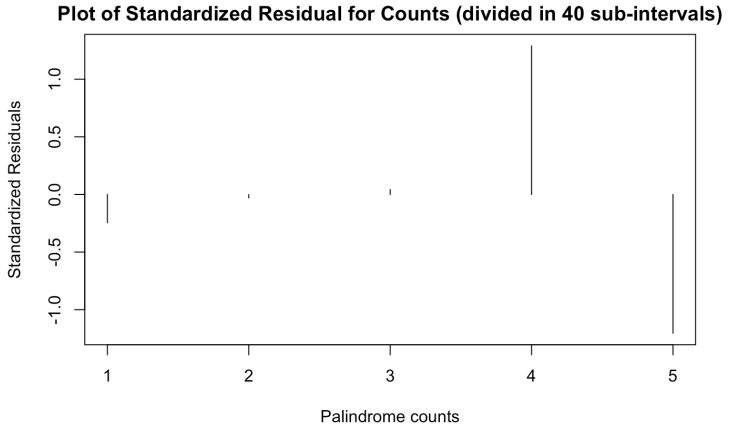
Hide

```
# Chi-square statistic
chi_2 <- sum((counts.observed - counts.expected)^2/counts.expected)
chi2_compare <- qchisq(p = 0.95, df = 3)
p_value <- pchisq(chi_2, df = 3, lower.tail = FALSE)
print(paste("The p_value when the distance is splited into 40 sub-intervals is", p_va
lue))
```

```
[1] "The p_value when the distance is splited into 40 sub-intervals is 0.365205436984
496"
```

Hide

```
## Visualization of the Residual
Residuals <- (counts.observed - counts.expected) / sqrt(counts.expected)
plot(Residuals, type = 'h', ylab = "Standardized Residuals", xlab = "Palindrome count
s", main = "Plot of Standardized Residual for Counts (divided in 40 sub-intervals)")
```



Hide

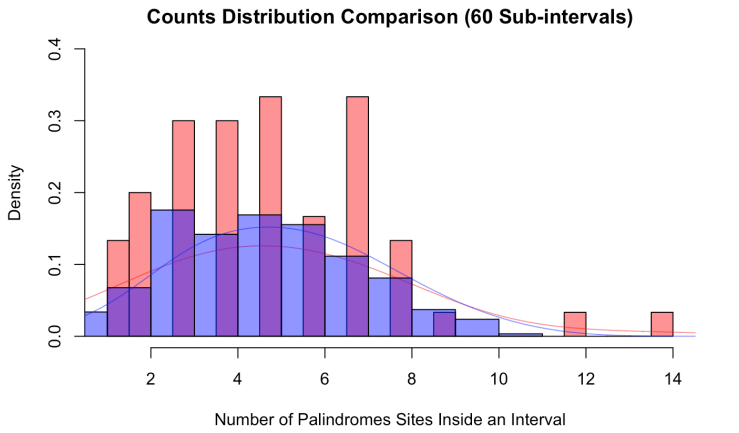
```
# Case 2: divided by 60 intervals
n.region <- 60
gene <- seq(1,N)
observed <- as.numeric(regionsplit(n.region, gene, locations))
interval <- as.numeric(names(regionsplit(n.region, gene, locations)))
lambda <- n/n.region
# Histogram of counts of palindromes in original data and poisson distribution
counts <- as.vector(table(cut(locations, breaks = seq(0, N, length.out = n.region+1),
include.lowest = TRUE)))
hist(counts, breaks = bins, col = rgb(1,0,0,0.5), probability = TRUE, main = "Counts
Distribution Comparison (60 Sub-intervals)", xlab = "Number of Palindromes Sites Insi
de an Interval", ylim = c(0,0.4))
lines(density(counts, adjust = 2), col = rgb(1,0,0,0.5))
```

Hide

```
Pois <- rpois(n, lambda)
hist(Pois, breaks = 15, col = rgb(0,0,1,0.5), probability = TRUE, add = TRUE)
lines(density(Pois, adjust = 2), col = rgb(0,0,1,0.5))
```

Hide

```
legend(x = 18, y = 0.17, legend = c("Sample", "Poisson"), lty = c(1,1), col = c(rgb(1
,0,0,0.5), rgb(0,0,1,0.5)))
```



Hide

```
# Chi-sqr Goodness of Fit test
expected <- n.region*exp(-lambda)* lambda**(interval)/factorial(interval)
expect <- n.region*exp(-lambda)* lambda**(0)/factorial(0)
expected[1] <- expected[1]+ expect
for (i in c(10:11)){
  expect <- n.region*exp(-lambda)* lambda**(i)/factorial(i)
  expected[9] <- expected[9]+ expect
}
expected[11] <- 0
for (i in c(1:12)){
  expect <- exp(-lambda)* lambda**(i)/factorial(i)
  expected[11] <- expected[11]+ expect
}
expected[11] <- (1-expected[11])*n.region
counts.expected <- c()
counts.interval <- c()
counts.observed <- c()
# Group bins
counts.expected[1] <- sum(expected[1:2])
counts.expected[2:6] <- expected[3:7]
counts.expected[7] <- sum(expected[8:11])
counts.observed[1] <- sum(observed[1:2])
counts.observed[2:6] <- observed[3:7]
counts.observed[7] <- sum(observed[8:11])
counts.interval[1] <- interval[2]
counts.interval[2:6] <- interval[3:7]
counts.interval[7] <- interval[7]+1
counts.table60 <- data.frame(counts.interval,counts.observed,counts.expected)
counts.table60
```

counts.interval <dbl>	counts.observed <dbl>	counts.expected <dbl>
2	10	7.822827
3	9	8.647726
4	9	10.665529
5	10	10.523322
6	5	8.652509
7	10	6.097959
8	7	8.022275

7 rows

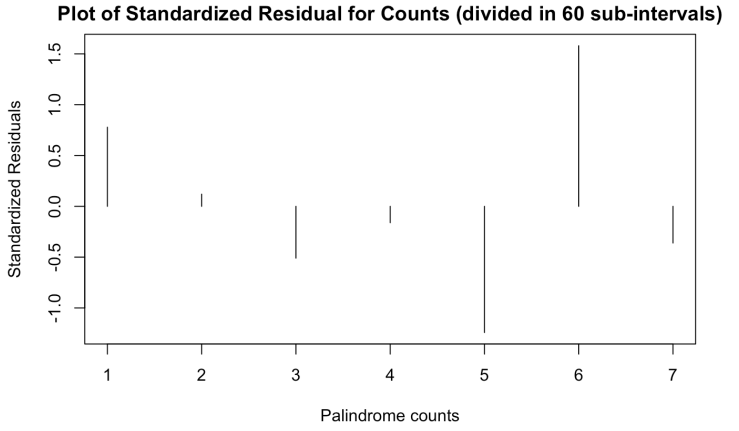
Hide

```
# Chi-square statistic
chi_2 <- sum((counts.observed - counts.expected)^2/counts.expected)
chi2_compare <- qchisq(p = 0.95, df = 5)
p_value <- pchisq(chi_2, df = 5, lower.tail = FALSE)
print(paste("The p_value when the distance is splited into 60 sub-intervals is", p_va
lue))
```

```
[1] "The p_value when the distance is splited into 60 sub-intervals is 0.406748395085
584"
```

Hide

```
## Visualization of the Residual
Residuals <- (counts.observed - counts.expected) / sqrt(counts.expected)
plot(Residuals, type = 'h', ylab = "Standardized Residuals", xlab = "Palindrome count
s", main = "Plot of Standardized Residual for Counts (divided in 60 sub-intervals)")
```



Hide

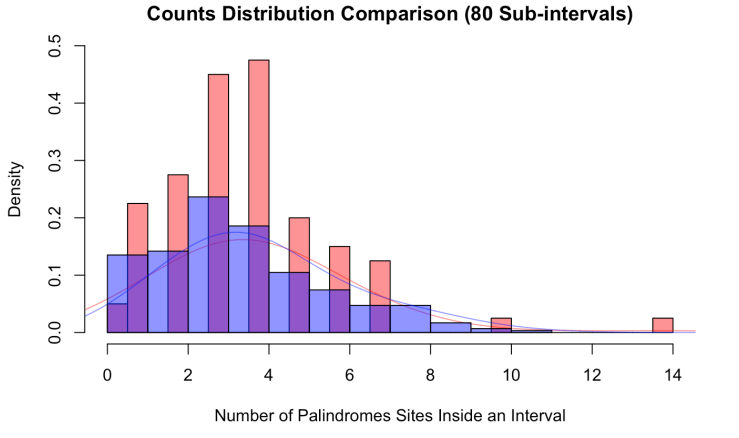
```
# Case 3: divided by 80 intervals
n.region <- 80
gene <- seq(1,N)
observed <- as.numeric(regionsplit(n.region, gene, locations))
interval <- as.numeric(names(regionsplit(n.region, gene, locations)))
lambda <- n/n.region
# Histogram of counts of palindromes in original data and poisson distribution
counts <- as.vector(table(cut(locations, breaks = seq(0, N, length.out = n.region+1),
include.lowest = TRUE)))
hist(counts, breaks = bins, col = rgb(1,0,0,0.5), probability = TRUE, main = "Counts
Distribution Comparison (80 Sub-intervals)", xlab = "Number of Palindromes Sites Insi
de an Interval", ylim = c(0,0.5))
lines(density(counts, adjust = 2), col = rgb(1,0,0,0.5))
```

Hide

```
Pois <- rpois(n, lambda)
hist(Pois, breaks = 15, col = rgb(0,0,1,0.5), probability = TRUE, add = TRUE)
lines(density(Pois, adjust = 2), col = rgb(0,0,1,0.5))
```

Hide

```
legend(x = 18, y = 0.17, legend = c("Sample", "Poisson"), lty = c(1,1), col = c(rgb(1
,0,0,0.5), rgb(0,0,1,0.5)))
```



Hide

Hide

```
# Chi-sqr Goodness of Fit test
expected <- n.region*exp(-lambda)* lambda**(interval)/factorial(interval)
for (i in c(8:9)){
  expect <- n.region*exp(-lambda)* lambda**(i)/factorial(i)
  expected[9] <- expected[9]+ expect
}
expected[10] <- 0
for (i in c(1:10)){
  expect <- exp(-lambda)* lambda**(i)/factorial(i)
  expected[10] <- expected[10]+ expect
}
expected[10] <- (1-expected[10])*n.region
counts.expected <- c()
counts.interval <- c()
counts.observed <- c()
# Group bins
counts.expected[1] <- sum(expected[1:2])
counts.expected[2:6] <- expected[3:7]
counts.expected[7] <- sum(expected[8:10])
counts.observed[1] <- sum(observed[1:2])
counts.observed[2:6] <- observed[3:7]
counts.observed[7] <- sum(observed[8:10])
counts.interval[1] <- interval[2]
counts.interval[2:6] <- interval[3:7]
counts.interval[7] <- interval[7]+1
counts.table60 <- data.frame(counts.interval,counts.observed,counts.expected)
counts.table60
```

counts.interval <dbl>	counts.observed <dbl>	counts.expected <dbl>
1	11	9.296046
2	12	13.538603
3	17	16.697610
4	18	15.445290
5	9	11.429514
6	6	7.048201
7	7	8.522618

7 rows

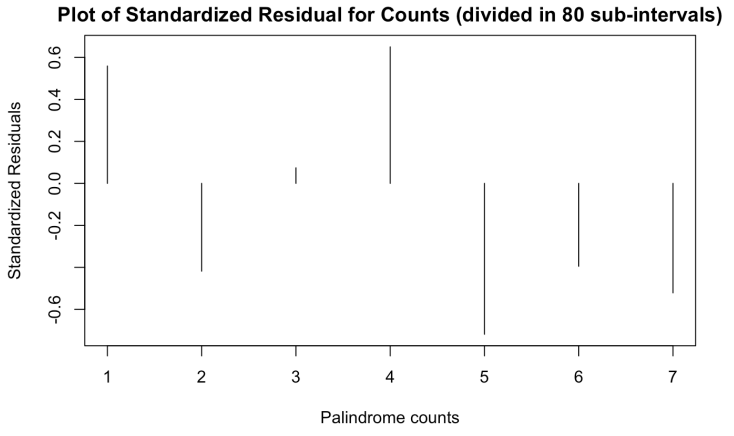
Hide

```
# Chi-square statistic
chi_2 <- sum((counts.observed - counts.expected)^2/counts.expected)
chi2_compare <- qchisq(p = 0.95, df = 5)
p_value <- pchisq(chi_2, df = 5, lower.tail = FALSE)
print(paste("The p_value when the distance is splited into 60 sub-intervals is", p_va
lue))
```

```
[1] "The p_value when the distance is splited into 60 sub-intervals is 0.868215562053
925"
```

Hide

```
## Visualization of the Residual
Residuals <- (counts.observed - counts.expected) / sqrt(counts.expected)
plot(Residuals, type = 'h', ylab = "Standardized Residuals", xlab = "Palindrome count
s", main = "Plot of Standardized Residual for Counts (divided in 80 sub-intervals)")
```



Scenario 4: The Biggest Cluster

Does the interval with the greatest number of palindromes indicate a potential origin of replication? Be careful in making your intervals, for any small, but significant deviations from random scatter, such as a tight cluster of a few palindromes, could easily go undetected if the regions examined are too large. Also, if the regions are

too small, a cluster of palindromes may be split between adjacent interavls and not appear as a high-count interval.

Hide

```
final <- array(dim=c(500,1))
interval_length <- array(dim=c(500,1))
lamda <- array(dim=c(500,1))
for (k in 20:100){
  tab <- table(cut(locations, breaks=seq(0, N, length.out=k+1), include.lowest=TRUE))
  head(tab,10)
  tab<-as.vector(tab)
  lamda[k,] <-sum(tab)/k
  threshold <-max(tab)
  result <- 0
  interval_length[k,] <- N/k
  for (i in 0:(threshold-1)){
    result <- result+((lamda[k]^i)*exp(-lamda[k])/factorial(i))
  }
  final[k,] <- 1-result^k
}
result <- data.frame(lamda,interval_length,final)
# Display Table containing the probability of a Poisson Distribution having e greates
t number of hits at least k for each sub-interval divisions
result[c(40,60,80),]
```

	lamda <dbl>	interval_length <dbl>	final <dbl>
40	7.400000	5733.850	0.308448064
60	4.933333	3822.567	0.036209375
80	3.700000	2866.925	0.002693866

3 rows

Additional Scenario: HIV and Age

TODO Description

Hide

```
# Clean out 'unknown' data and convert factor to numerical values
health <- transform(health, age_yrs=as.numeric(age_yrs),
                            hiv=as.character(hiv))
health.ind <- which(health$hiv != 'unknown')
health <- health[health.ind,]
# Total number of people that have hiv
population=nrow(health)
pop_hiv <- nrow(health[which(health$hiv=='positive'),])
# Split the age into four groups
# 0-20
age_first <- health$age_yrs[which(health$age_yrs<21)]
age_proportion_first <- length(age_first)/population
hiv_proportion_first<- nrow(health[which((health$hiv== 'positive') & (health$age_yrs
<21)),])/pop_hiv
# 21-40
age_second<-health$age_yrs[which(health$age_yrs>20 & health['age_yrs']<41)]
age_proportion_second <- length(age_second)/population
hiv_proportion_second<-nrow(health[which(health$age_yrs>20 &health$age_yrs<41 & healt
h$hiv=='positive'),])/pop_hiv
# 41-60
age_third<-health$age_yrs[which(health['age_yrs']>40 & health['age_yrs']<61)]
age_proportion_third <- length(age_third)/population
hiv_proportion_third<-nrow(health[which(health$age_yrs>40 & health$age_yrs<61
&health$hiv=='positive'),])/pop_hiv
# 61+
age_last<-health$age_yrs[which(health['age_yrs']>60)]
age_proportion_last <- length(age_last)/population
hiv_proportion_last<-nrow(health[which(health$age_yrs>60 & health$hiv=='positive'),])
/pop_hiv
# Expected Data
population_dist <-c(age_proportion_first,age_proportion_second,age_proportion_third,a
ge_proportion_last)
# Observed Data
hiv_dist<-c(hiv_proportion_first,hiv_proportion_second,hiv_proportion_third,hiv_propo
rtion_last)
age_dist <- c("0-20", "21-40", "41-60", "61+")
data.frame(age_dist,population_dist,hiv_dist)
```

age_dist <fctr>	population_dist <dbl>	hiv_dist <dbl>
0-20	0.3917526	0.14
21-40	0.2816307	0.55
41-60	0.2014995	0.27
61+	0.1251172	0.04

4 rows

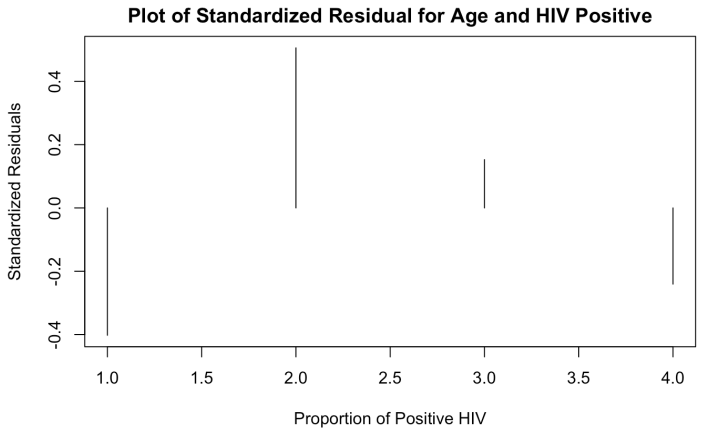
Hide

```
# Goodness-fittest
chi_2 <- sum((hiv_dist - population_dist)^2/population_dist)
chi2_compare <- qchisq(p=0.95, df=3)
p_value <- pchisq(chi_2, df=3, lower.tail=FALSE)
print(paste('The p_value of Goodness of Fit Test is',p_value))
```

```
[1] "The p_value of Goodness of Fit Test is 0.919175049520472"
```

Hide

```
#Visualization
Residuals <- (hiv_dist - population_dist) / sqrt(population_dist)
plot(Residuals, type='h', ylab='Standardized Residuals', xlab='Proportion of Positive
HIV', main='Plot of Standardized Residual for Age and HIV Positive')
```



Null Hypothesis: The proportion of age in the population is unrelated with the proportion of people having hiv. (Age is not an influencing factor for HIV testing positive) Since p-value of this chi-square goodness of fit test is close to 1, we see that deviations as large as ours (or larger) are very likely. In addition, having values of the standardized residual less than 3 suggests that it is a good fit of the age distribution to estimate the people testing positive on hiv. Hence, we reject the null hypothesis and conclude that the distribution of proportion of age matches with the the distribution of people testing positive on HIV.