Final Report in Learning From Genome Data

Sander Boisen, 20114143

Thursday, October 16, 2015

This is the final report for the course "learning from genome data 1". This report is made in R markdown and codechunks as the following will be occuring frequently when deemed necessary and helpful for understanding the work in this report.

1. Import the dataset in R.

The first step will be loading necessary libraries and their dependencies.

```
#Utility
library('dplyr')
library('tidyr')
library('knitr')
#Bootstraps
library('boot')
#Better graphing control
library('ggplot2')
library('cowplot')
library('gridExtra')
```

Next the data is loaded:

```
medicago <- read.table("Dataset_final.txt", sep ="\t", header = TRUE)
#Inspecting the data
summary(medicago)
```

```
##      window          chr             start              LdH_SNPs
##  1_001_1:   1   Min.   :1.000   Min.   :       1   Min.   :   1.0
##  1_002_1:   1   1st Qu.:3.000   1st Qu.: 9000001   1st Qu.: 241.0
##  1_003_1:   1   Median :4.000   Median :18500001   Median : 372.5
##  1_004_1:   1   Mean   :4.473   Mean   :18873996   Mean   : 394.6
##  1_005_1:   1   3rd Qu.:7.000   3rd Qu.:28000001   3rd Qu.: 517.0
##  1_006_1:   1   Max.   :8.000   Max.   :44300001   Max.   :1281.0
##  (Other):2532
##    LdH_start         ldH_win_size       rho_kb            rho_bp
##  Min.   :     553   Min.   :    0   Min.   : 0.0540   Min.   :0.000000
##  1st Qu.: 9001033   1st Qu.:85218   1st Qu.: 0.6378   1st Qu.:0.000640
##  Median :18500014   Median :97919   Median : 1.2830   Median :0.001280
##  Mean   :18880497   Mean   :87213   Mean   : 1.9519   Mean   :0.001951
##  3rd Qu.:28038353   3rd Qu.:99371   3rd Qu.: 2.5288   3rd Qu.:0.002527
##  Max.   :44300831   Max.   :99997   Max.   :32.0440   Max.   :0.032040
##                                     NA's   :2
##    rho_theta          rho_L95           rho_U95            bases
##  Min.   :0.0000   Min.   : 0.0040   Min.   : 0.0570   Min.   :10113
##  1st Qu.:0.1231   1st Qu.: 0.5895   1st Qu.: 0.7017   1st Qu.:40808
##  Median :0.2038   Median : 1.1785   Median : 1.4025   Median :58630
##  Mean   :0.2908   Mean   : 1.7282   Mean   : 2.2176   Mean   :54084
##  3rd Qu.:0.3473   3rd Qu.: 2.2793   3rd Qu.: 2.7812   3rd Qu.:69276
##  Max.   :6.6977   Max.   :25.5830   Max.   :38.9170   Max.   :90367
##                   NA's   :2         NA's   :2
##    mutations           SNPs        singletons        TajimasD
##  Min.   :   9.0   Min.   :   9   Min.   :   4.0   Min.   :-2.462
##  1st Qu.: 868.2   1st Qu.: 854   1st Qu.: 495.2   1st Qu.:-1.593
##  Median :1215.0   Median :1196   Median : 677.5   Median :-1.341
##  Mean   :1222.1   Mean   :1201   Mean   : 668.0   Mean   :-1.331
##  3rd Qu.:1550.5   3rd Qu.:1524   3rd Qu.: 836.0   3rd Qu.:-1.096
##  Max.   :3483.0   Max.   :3366   Max.   :1709.0   Max.   : 0.401
##
##     qp.site            qw.site            dist_cent
##  Min.   :0.000030   Min.   :0.000060   Min.   :-21636558
##  1st Qu.:0.003000   1st Qu.:0.004680   1st Qu.: -4796316
##  Median :0.004045   Median :0.005920   Median :  4700035
##  Mean   :0.004537   Mean   :0.006492   Mean   :  4808929
##  3rd Qu.:0.005710   3rd Qu.:0.007970   3rd Qu.: 13715922
##  Max.   :0.013590   Max.   :0.017590   Max.   : 31516707
##
##    prop_dist        gene_dens
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.2607   1st Qu.:0.2972
##  Median :0.5086   Median :0.3984
##  Mean   :0.5069   Mean   :0.3837
##  3rd Qu.:0.7556   3rd Qu.:0.4828
##  Max.   :1.0000   Max.   :0.7678
##

length(medicago$window)

## [1] 2538
```

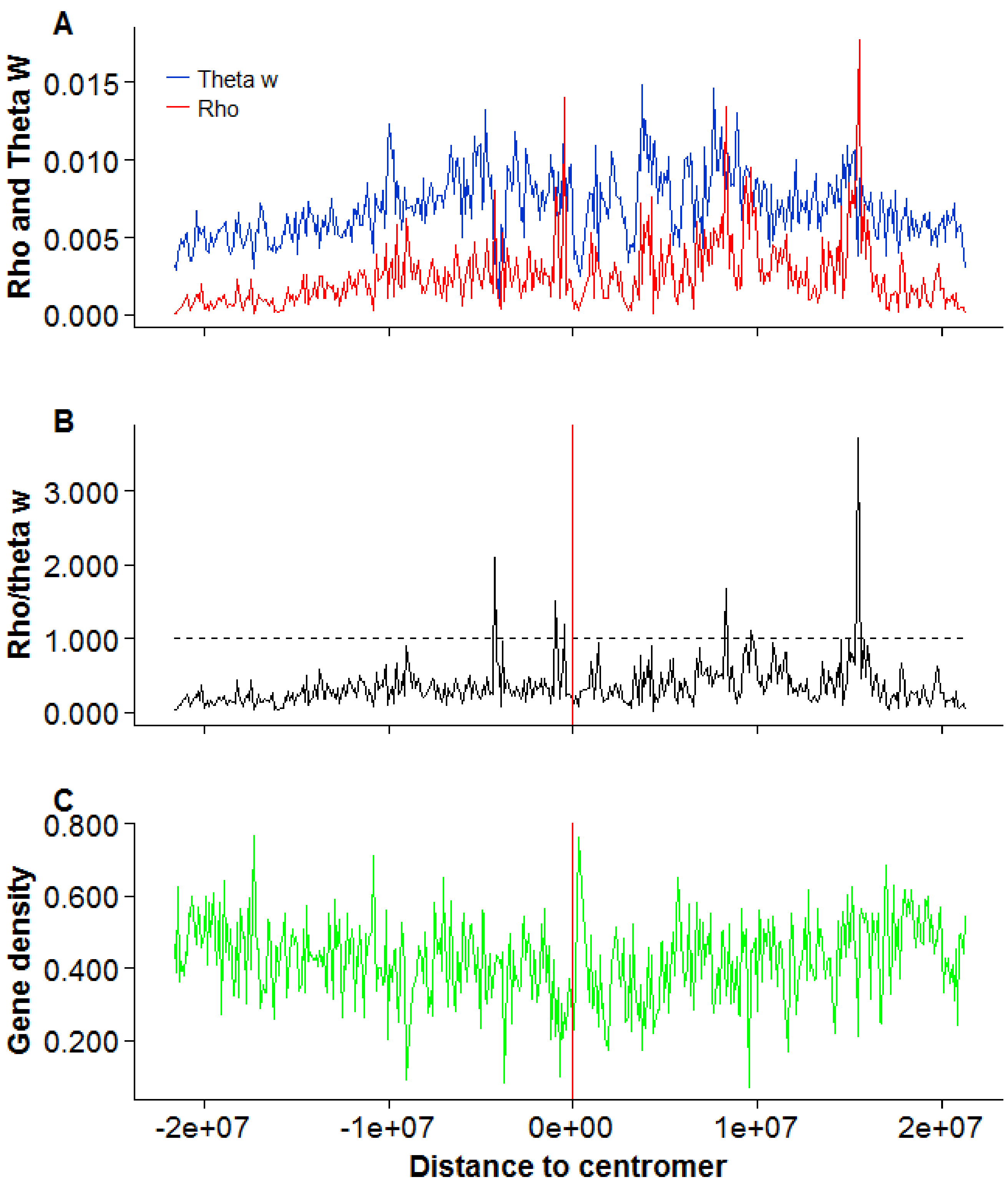
The total number of windows in the data set is 2538. When viewing the dataset it becomes apparent that there are a few missing values("NA") and these will be omitted from much of the analysis through the call na.omit()

Displaying genomic data:

To replicate the three graphs from Branca et al. 2011 three ggplots will be generated, and plotted using the gridExtra library.

```
# Filter the data to get only the data for chr5
medicago %>% filter(chr==5) -> chr5data
# Do the 3 plots
# converts axis values to scientific notation
fmt <- function(x){format(x,nsmall = 3, scientific = F)};
g1 <- ggplot(data = chr5data) + geom_line(aes( x = dist_cent, y = qw.site,
colour = '1 Theta w')) +
  geom_line(aes(x = dist_cent, y = rho_bp, colour = '2 rho'))+
```

```r
  scale_x_continuous("", limits = c(min(chr5data$dist_cent),
                                    max(chr5data$dist_cent))) +
  scale_y_continuous("Rho and Theta W", labels = fmt)+
  scale_colour_manual(values=c("#0033CC", "#ff0000"),labels = c("Theta w",
"Rho"))+
  theme(legend.position = c(0.1,0.8), legend.title =element_blank(),
axis.text.x = element_blank())

g2 <- ggplot(data = chr5data) + geom_line(aes(x = dist_cent, y = rho_theta)) +
  geom_line(aes( x = dist_cent, y = 1), linetype = 2) +
  scale_x_continuous("", limits = c(min(chr5data$dist_cent),
                                    max(chr5data$dist_cent)))+
  geom_vline(xintercept = 0, colour = "red") +
  scale_linetype_discrete(guide = 'none') +
  scale_y_continuous("Rho/theta w", labels = fmt)+
  theme(axis.text.x =element_blank())

g3 <- ggplot(data = chr5data) + geom_line(aes(x = dist_cent, y = gene_dens),
colour = "green") +
  geom_vline(xintercept = 0, colour = "red") +
  scale_x_continuous(name="Distance to centromer", limits =
c(min(chr5data$dist_cent),
                                    max(chr5data$dist_cent)))+
  scale_y_continuous("Gene density", labels = fmt)
#plotting the graphs together
ggdraw()+
  draw_plot(g1,0,2/3, 1, 1/3)+
  draw_plot(g2, 0, 1/3, 1, 1/3)+
  draw_plot(g3, 0, 0,1, 1/3)+
  draw_plot_label(c('A','B','C'), c(0.05,0.05,0.05), c(1,2/3,0.35), size = 15)
```



3. **Select a subset of the dataset by excluding windows that have very few SNPs or a window size that is too small** (ldH_win_size, less than 1000)(LdH_SNPs, less than 200 SNPs)

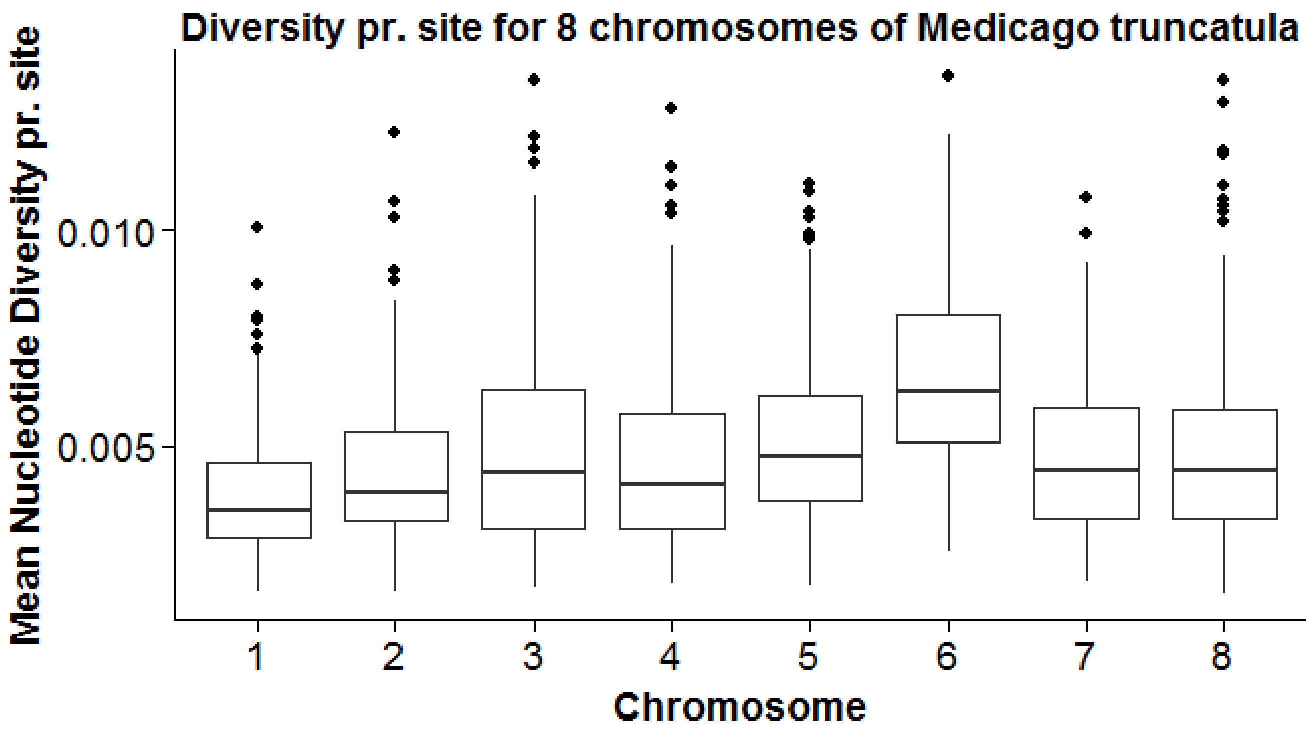
This was done by using the filter function. The pipeline operators "%>%" stems from magrittr which is a dependency library for dplyr. This allows for shorter easier written code. The code follows below:

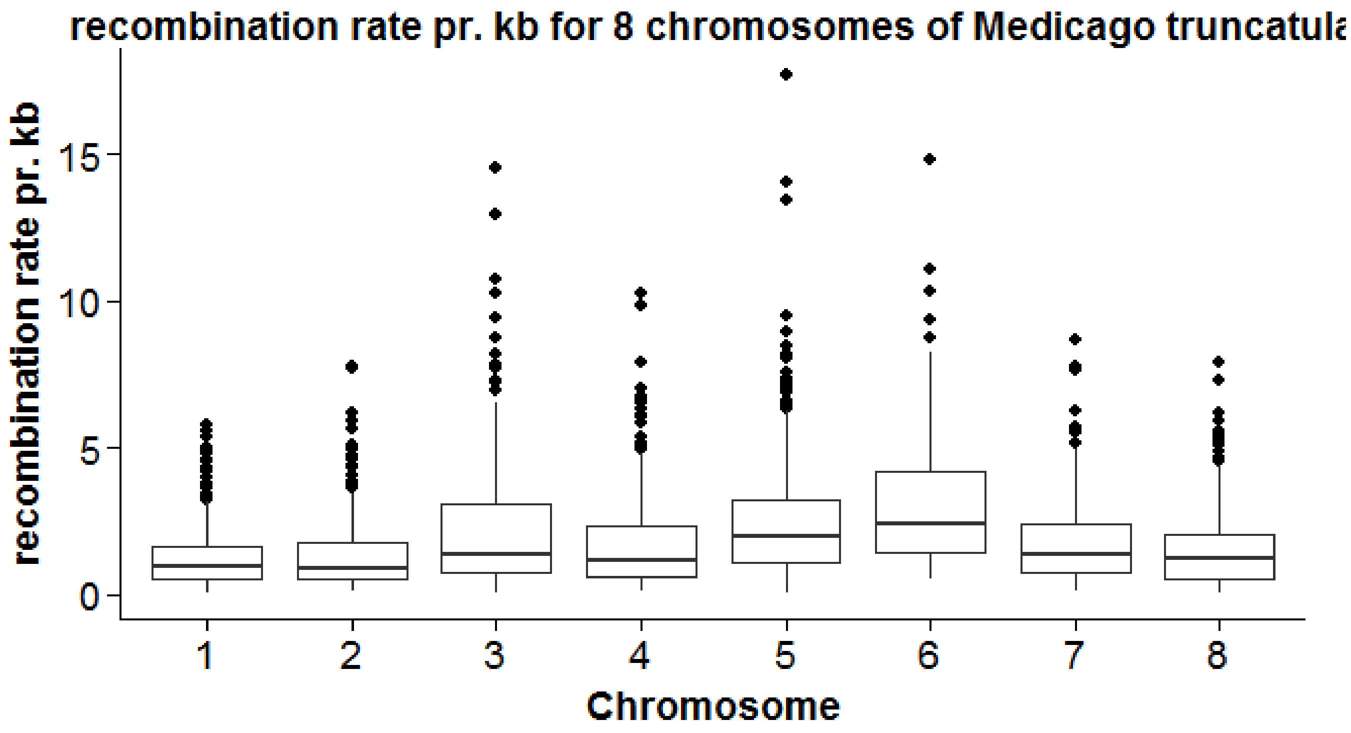
```
#remove data points with NA values
medicago <- na.omit(medicago)

#Filter and save the filtered data
medicago %>% filter(ldH_win_size>=1000) %>% filter(LdH_SNPs >= 200) -> medicago
```

4. **Recombination and polymorphism in M. truncatula:** *Use a boxplot or some other graphical display that contrasts the distribution of diversity (qp.site) and recombination (rho per kb) among the 8 chromosomes of M. truncatula. For each chromosome calculate the median recombination rate and its associated 95%CI. Present these results as a table*

This is quite easily done with ggplot. Boxplots will be used as the graphical display:

```
myPlot1 <- ggplot(data = medicago, aes(x=factor(chr), y=qp.site)) +
geom_boxplot() +
  scale_y_continuous(name = "Mean Nucleotide Diversity pr. site") +
  scale_x_discrete("Chromosome") + ggtitle("Diversity pr. site for 8
chromosomes of Medicago truncatula")

myPlot2 <- ggplot(data = medicago, aes(x=factor(chr), y=rho_kb)) +
geom_boxplot() +
  scale_y_continuous(name = "recombination rate pr. kb") +
  scale_x_discrete("Chromosome") + ggtitle("recombination rate pr. kb for 8
chromosomes of Medicago truncatula")

grid.arrange(myPlot1, myPlot2, nrow = 2)
```





Now for the confidence interval which will be estimated using the bootstrap method. First the median of each chromosome is found and then bootstraps for each chromosome is done and the data is loaded into a table.

```
myTable <- cbind(rep(0, 8), rep(0,8), rep(0,8))
```

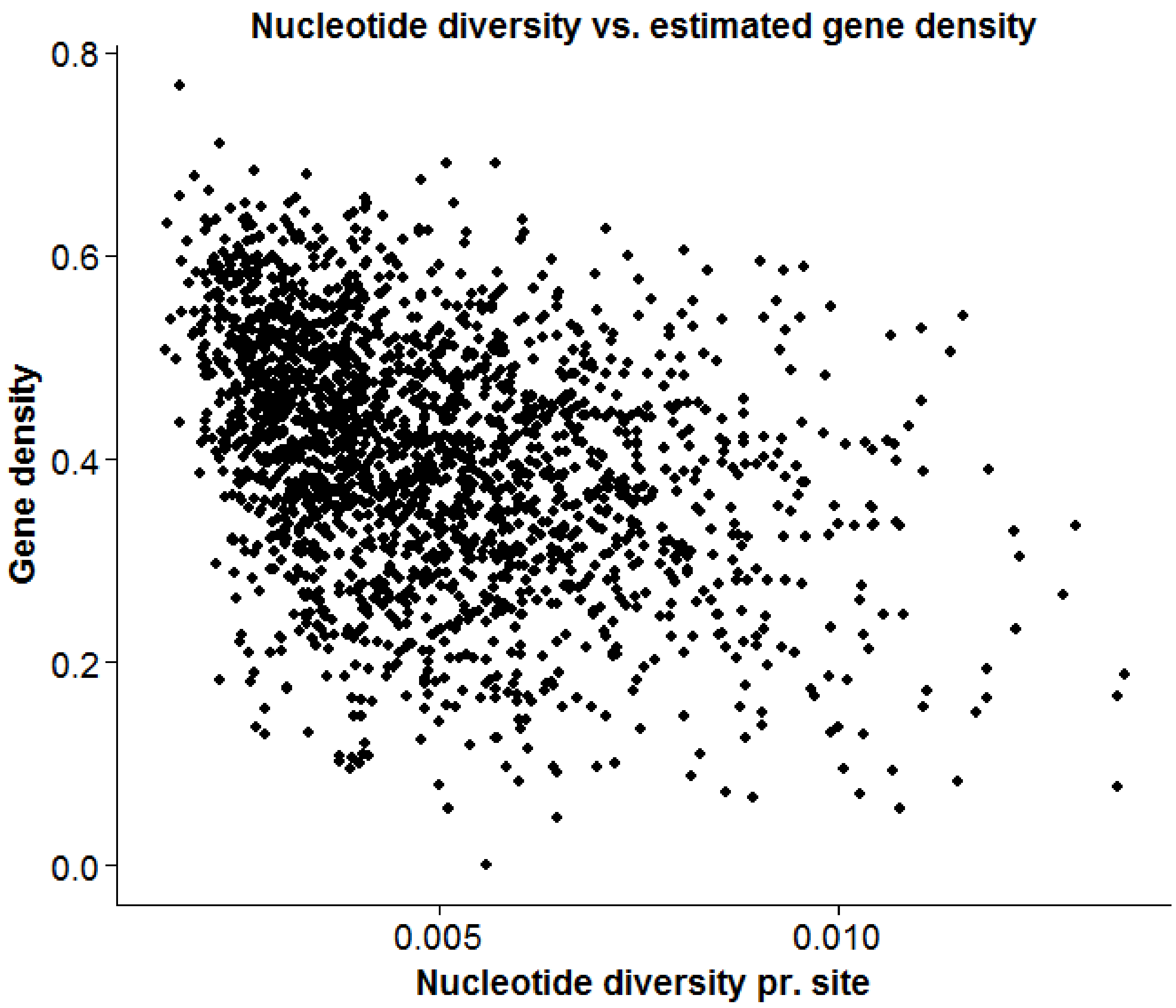
```
#The for-loop is used for ease of control and since it only in a few cases is
faster than the #various apply functions. I am so sorry coding standard, I just
want this to be done. Forgive #me padre.

#Define a good function for boot to work with
bootMedian <- function(x,d){
  return(median(x[d]))
}

for( i in seq(1,8)){
  medicago%>%filter(chr==i)->temp
  myBoot <- boot(data = temp$rho_kb, statistic = bootMedian, R = 1000)

  myTable[i, 1] <- median(temp$rho_kb)
  myTable[i, 2] <- sort(myBoot$t)[25]
  myTable[i, 3] <- sort(myBoot$t)[975]

}
myTable <- as.data.frame(myTable)
colnames(myTable) <- c('Observed Median', 'CI lower bound', 'CI upper bound')
rownames(myTable) <- c('chr 1', 'chr 2','chr 3', 'chr 4', 'chr 5','chr 6', 'chr
7', 'chr 8')
print(myTable)

##       Observed Median CI lower bound CI upper bound
## chr 1          0.9650         0.8340          1.111
## chr 2          0.9170         0.8250          1.040
## chr 3          1.3960         1.2820          1.583
## chr 4          1.1450         1.0270          1.282
## chr 5          1.9715         1.8165          2.185
## chr 6          2.3910         1.8740          2.724
## chr 7          1.3700         1.1840          1.493
## chr 8          1.2335         1.0250          1.431
```

5.

The so called background selection hypothesis states that genomic regions that are experiencing more deleterious mutations may exhibit overall less polymorphism. This hypothesis makes the prediction that everything else being equal very gene-rich regions (more prone to produce deleterious mutation) should be exhibiting comparatively less polymorphism than regions that are gene poor. So far testing this hypothesis was not doable (except for a few model species) because it requires enormous amounts of polymorphism data. Test that hypothesis in Medicago truncatula. To do so start by identifying/calculating a variable that describes how gene rich a
window is and another variable that describes how much polymorphism there is in a window. Then test whether these two continuous variables are associated (correlated).

The *gene_dens* variable will be used as a meassure of gene richness, and *qp.site* will by used as the meassure of polymorphism in the region. First a plot of these variables will be explored, and then a test performed:

```
myPlot3 <- ggplot() + geom_point(data=medicago,aes(x=qp.site, y=gene_dens))
myPlot3 + ggtitle('Nucleotide diversity vs. estimated gene density') +
scale_y_continuous(name = 'Gene density') + scale_x_continuous(name =
'Nucleotide diversity pr. site')
```



From this there indeed seem to be some correlation. To test this the cor.test, using Kendalls permutation based test for correlation So while gene_dens looks a little bit skewed the SNPs variable seems clearly normal distributed. Null hypothesis is zero correlation, and the alternative hypothesis is that the correlation is negative :

```
cor.test(medicago$gene_dens, medicago$qp.site, alternative = "l", method = "k")

##
##  Kendall's rank correlation tau
```

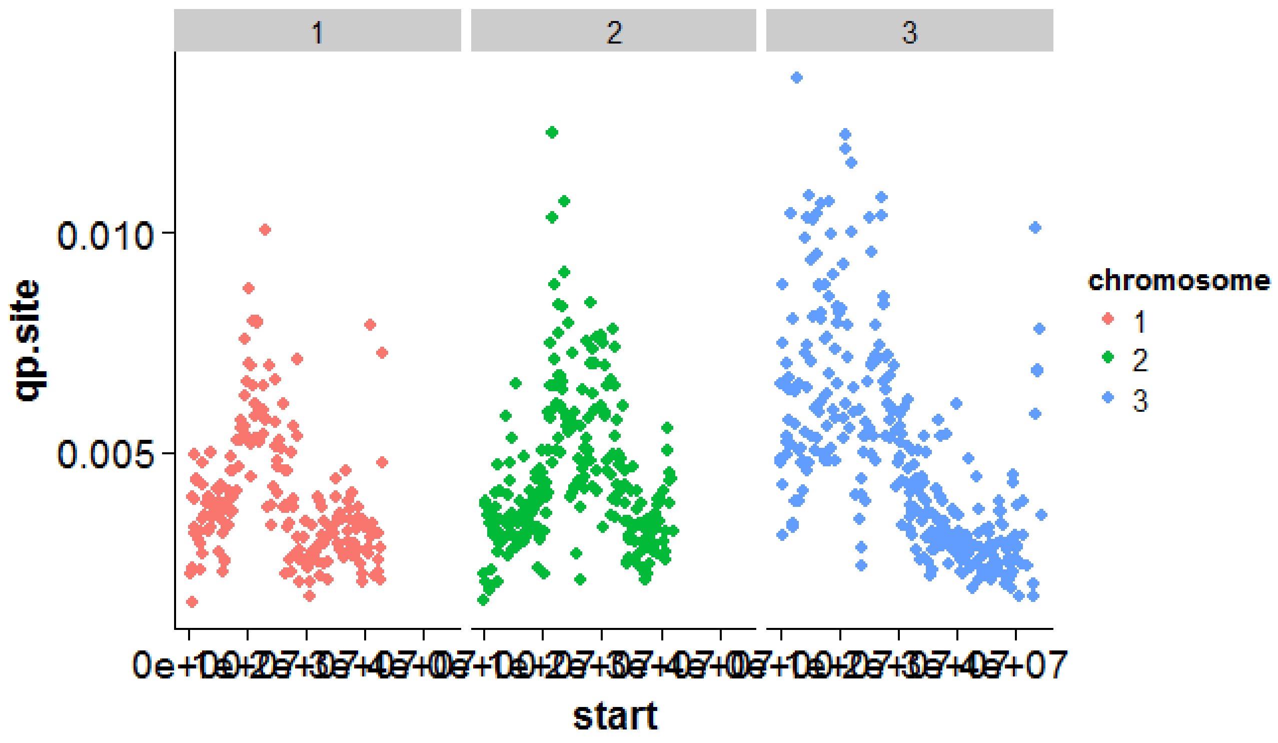
```
##
## data:  medicago$gene_dens and medicago$qp.site
## z = -17.228, p-value < 2.2e-16
## alternative hypothesis: true tau is less than 0
## sample estimates:
##        tau
## -0.2532375
```

So there is a highly significant negative correlation between *qp.site* and *gene.diversity* of -0.2532375.

6. Tracking footprints of recent and intense natural selection in the Medicago truncatula genome.

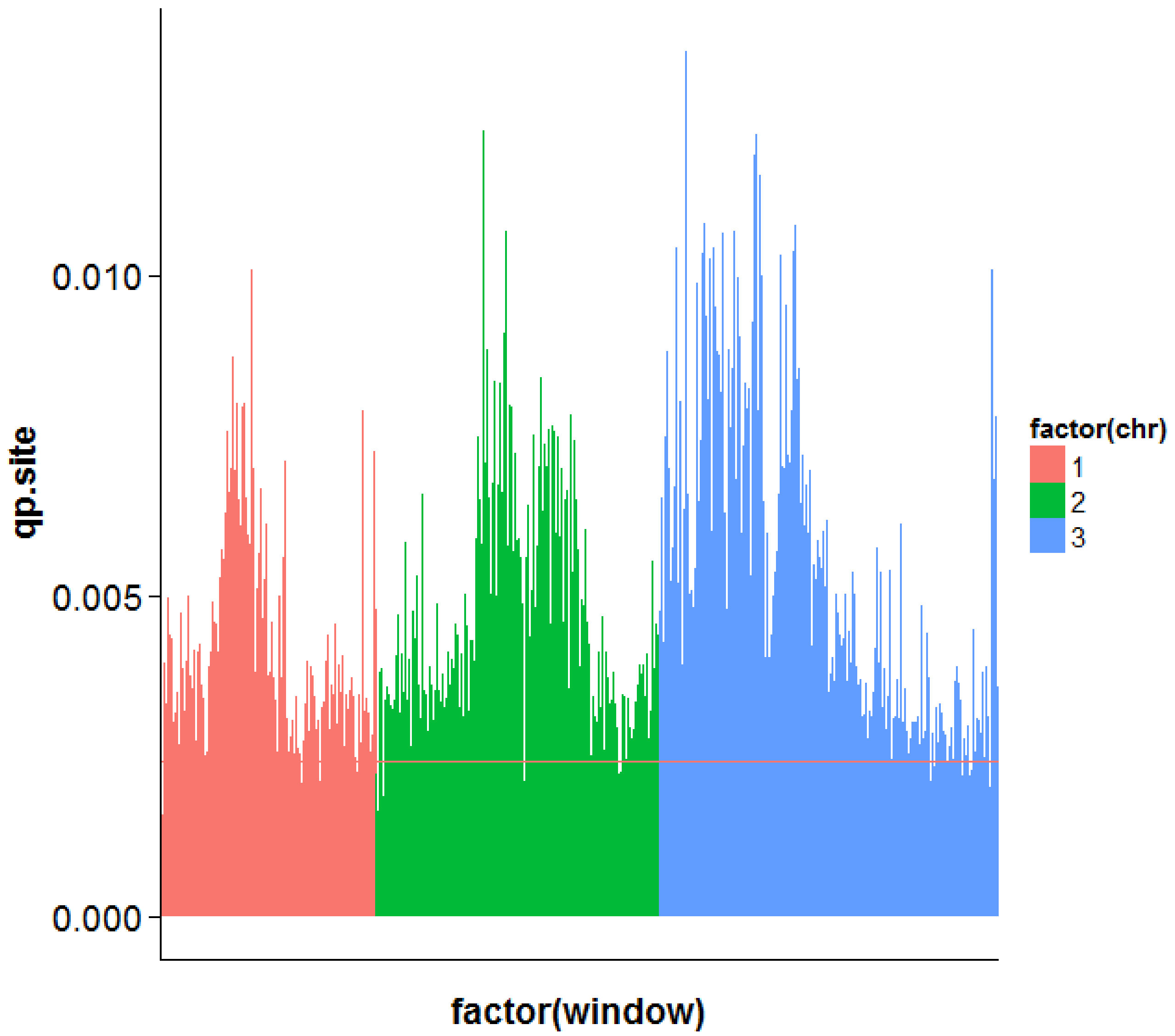
First the data will be filtered again to only look at the first 3 chromosomes and plot *qp.site* against *start* which shows where the 100 kb window begins:

```
medicago %>% filter(chr %in% c(1,2,3)) -> medicago3
myPlot <- ggplot(data = medicago3, aes(x = start,y = qp.site)) +
geom_point(aes(colour = factor(chr)))
myPlot + scale_colour_discrete(name = 'chromosome') + facet_wrap(~chr)
```



So while there seems to be some pattern in these plots with low diversity being located roughly in the same regions, a plot of the regions containing the 5% lowest diversity across the 3 chromosomes.

```
cutoff <- sort(medicago$qp.site)[ceiling(0.05*length(medicago$qp.site))]

myPlot <- ggplot(data = medicago3, aes(y = qp.site, x = factor(window))) +
  geom_bar(aes(fill = factor(chr)), stat = 'identity') +
  geom_hline(aes(yintercept = cutoff,colour ='#FFFF00'))+
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
  scale_colour_discrete(name = 'chromosome')
myPlot
```



From these plots there seem to be a pattern of regions with low diversity especially near the end of chromosome 3 and at the start of chromosome 2.

6.2

Is the location of least polymorphic windows randomly distributed on chromosome 1 and 2, 3?
To make such a test, you can group windows into larger bins comprising maybe 20

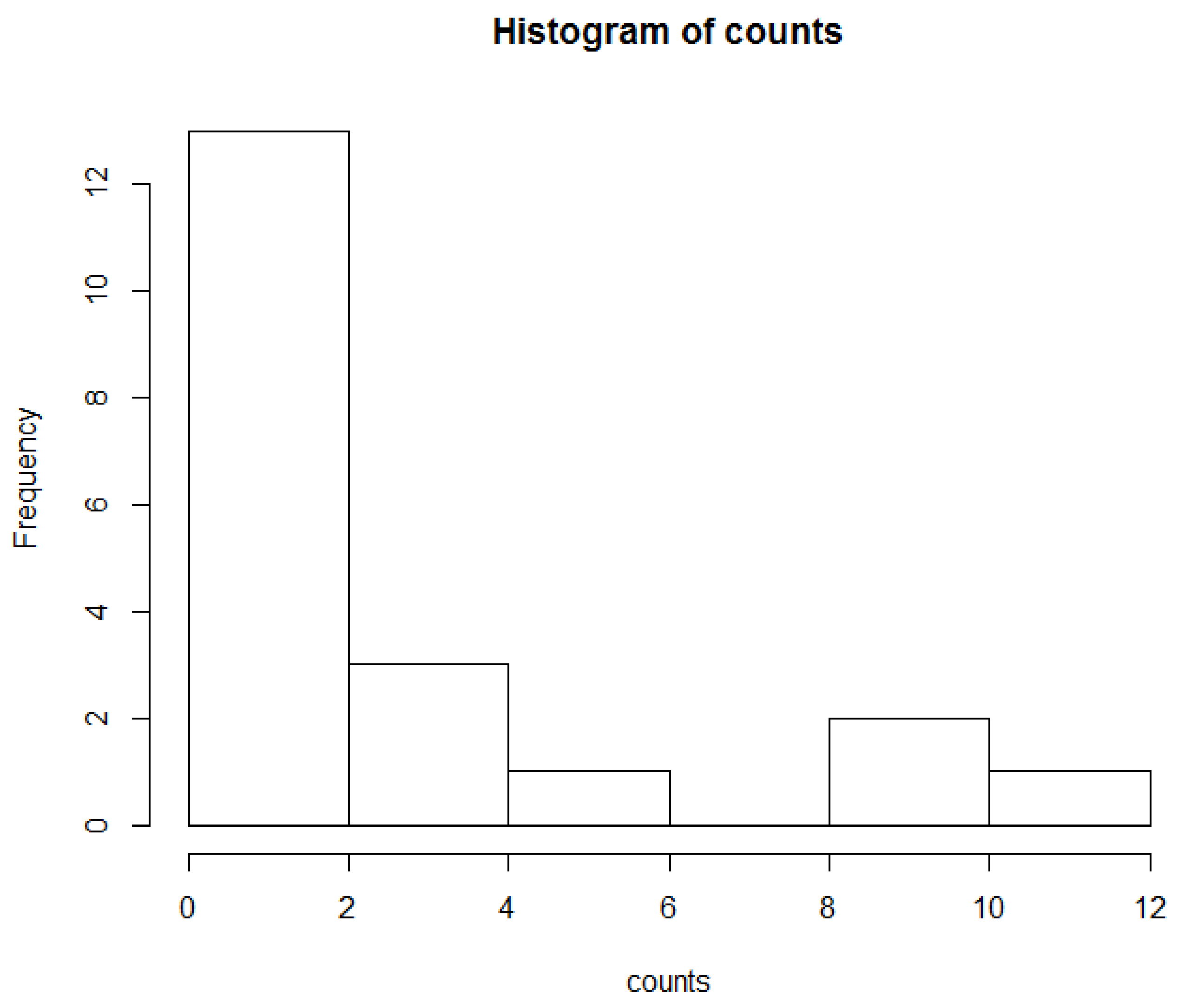
windows and count the number of windows with low diversity per bin. Think about what probability distribution you expect can capture the fact that windows with low level of polymorphisms are occurring ⍰at random⍰ along the chromosome in each bin.
Then decide on a statistic you can use to test the null hypothesis that the location of least polymorphic windows is randomly distributed on the chromosome⍰. Get a null distribution for such test statistic and state whether you reject the null or not for each chromosome by discussing the p-values you obtain.

First the bins are created at a reasonable size and count the number of 5% lowest diversity windows in each bin:

```r
bins <- sort(rep(seq(1,20),ceiling(length(medicago3$qp.site)/20)))
bins <- bins[1:length(medicago3$qp.site)]
medicago3$bins <- bins
#View(medicago3)

#A for loop is used again as the equivalent apply-solution seems cumbersome!
counts <- rep(0, max(medicago3$bins))
for (i in seq(1, max(medicago3$bins))){
  medicago3 %>%
    filter(bins == i )%>%
    filter(qp.site <= cutoff) -> temp
   length(temp$qp.site) -> counts[i]
}
```

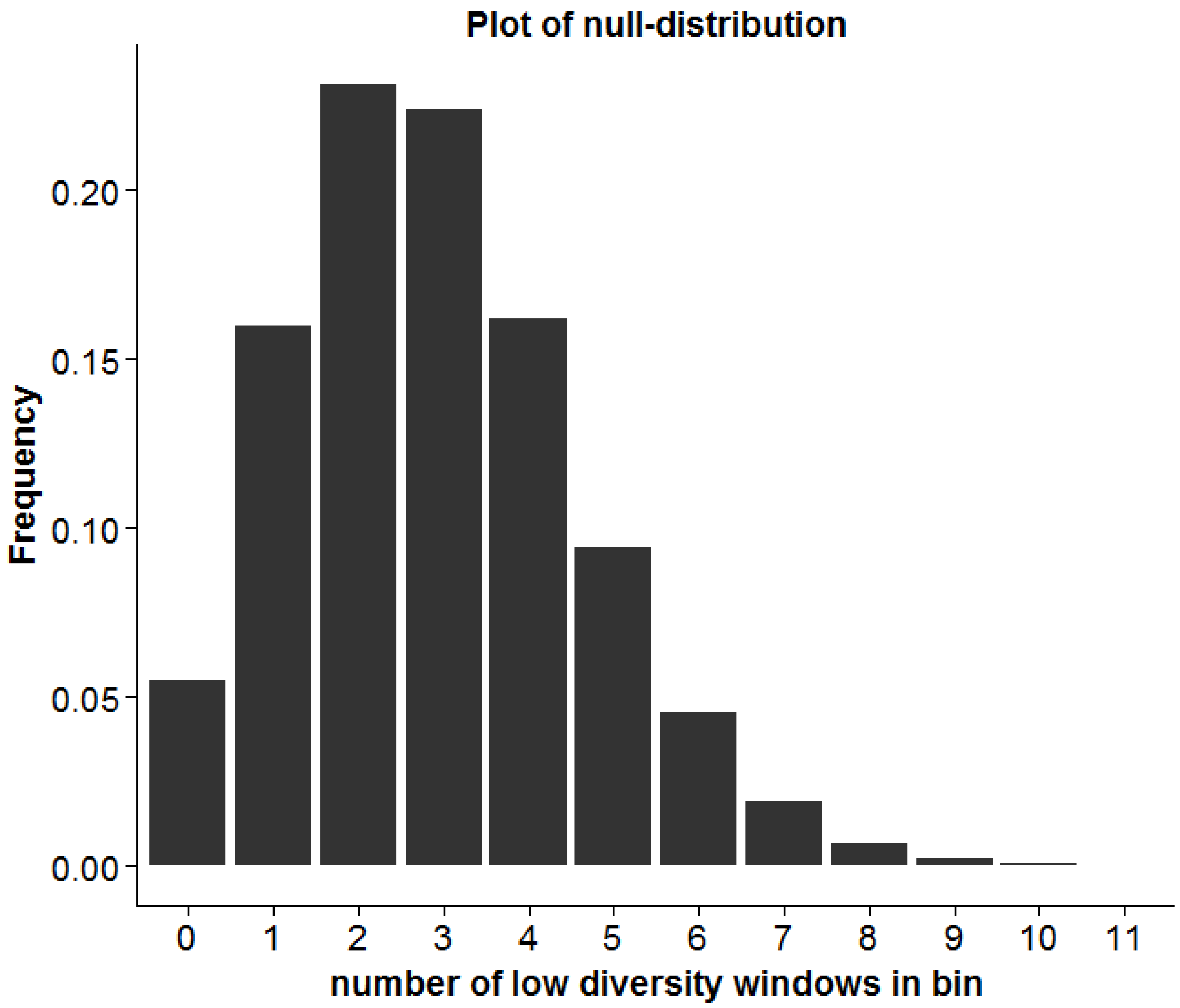
The 'counts' object now contains a count of the number of 5% lowest diversity windows in the data set. This is plottet withthe code below:

```r
hist(counts)
```



So while there are some outlier, this looks roughly like something that could be produced drawing random numbers from a poisson distribution. This also makes perfect sense since our null-hypothesis is that areas of low diversity occur randomly along the space of the chromosomes. A null-distribution will be estimated below:

```
nullDist <- dpois(x=0:(max(counts)), lambda = mean(counts))
myPlot <- ggplot() + geom_bar(aes(x = factor(0:(max(counts))), y = nullDist),
stat = 'identity')
myPlot + scale_x_discrete(name = "number of low diversity windows in bin") +
  scale_y_continuous(name = "Frequency") + ggtitle("Plot of null-distribution")
```

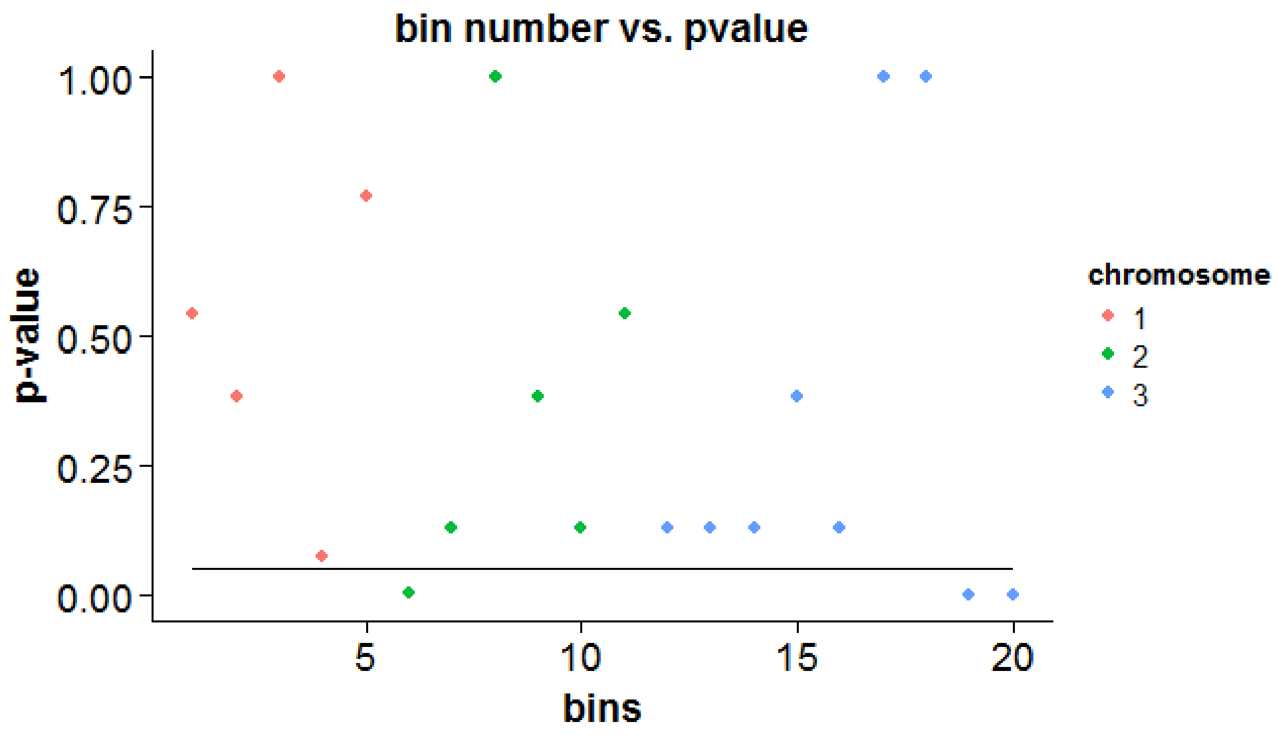


This null-distribution seems quite resonable, and a goodness-of-fit test is now performed

```
#Get the observed values
observed <- lapply(seq(0, max(counts)), FUN = function(x) length(which(counts
== x)))
#A zero is tied on since the nullDist does not sum to zero exactly, and a final
point is
#added to the probability vector later
observed <- as.numeric(observed)
length(observed)

## [1] 12

myChiTest <- chisq.test(x = c(observed,0), p = c(nullDist, 1-sum(nullDist)),
correct = TRUE)

## Warning in chisq.test(x = c(observed, 0), p = c(nullDist, 1 -
## sum(nullDist)), : Chi-squared approximation may be incorrect
```

```
myChiTest

##
##  Chi-squared test for given probabilities
##
## data:  c(observed, 0)
## X-squared = 419.96, df = 12, p-value < 2.2e-16
```

```
pvals <- lapply(counts, FUN = function(x) poisson.test(x, r =
mean(counts))$p.value)
bins<-medicago3$bins
pvals <- pvals[bins]
medicago3$pval <- as.numeric(pvals)
#plot time
ggplot(data = medicago3, aes(x = bins, y = as.numeric(pvals))) +
  geom_point(aes(colour=factor(chr))) +
  geom_line(x = bins, y = 0.05) +
  scale_colour_discrete(name = 'chromosome') +
  ggtitle("bin number vs. pvalue") +
  scale_y_continuous(name = "p-value")
```



The plot shows the p-value from poisson.test for the number of occurences at each bin. This does not yield any convincing pattern. The goodness-of-fit test suggests that the null-distribution is very wrong indeed, and thus we can conclude that the areas of low polymorphism is not distributed according to the null-distribution. This should be taken with some sceptism though, as one of the assumptions for the chi-sq test is violated. A Kolmogorow-Smirnov test was therefore performed:

```
ks.test(observed, y = ppois(0:max(counts), mean(counts)))
```

```
## Warning in ks.test(observed, y = ppois(0:max(counts), mean(counts))):
## cannot compute exact p-value with ties

##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  observed and ppois(0:max(counts), mean(counts))
## D = 0.75, p-value = 0.002342
## alternative hypothesis: two-sided
```

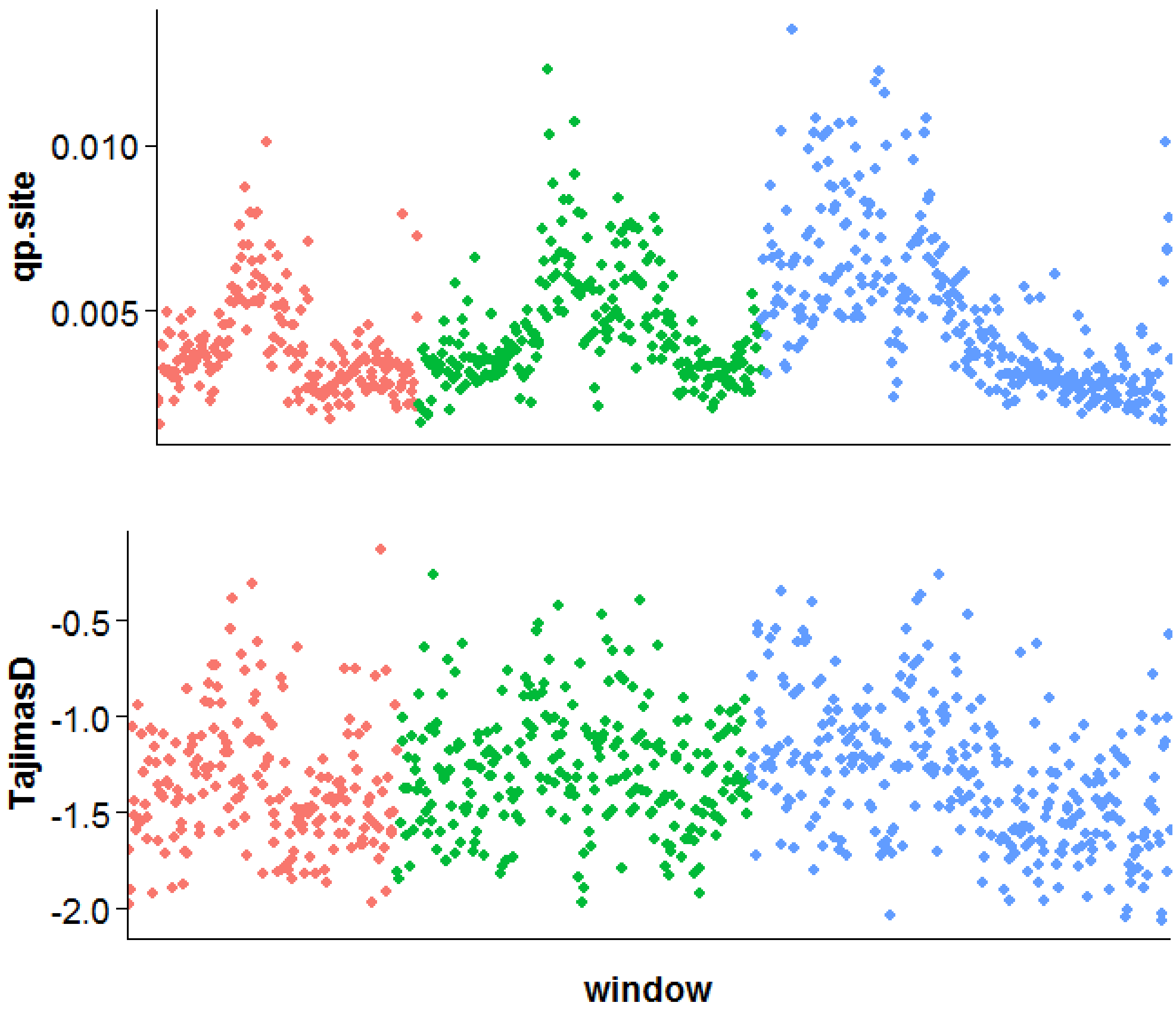
This test also rejects the null-hypothesis. The warning is due to the ppois() function in the range chosen only sum to 0.99994797, and a tie is thus introduced. All in all it can be concluded that the observations do not stem from a poisson distribution with lambda = 2.9.

6.3

Another typical footprint of a selective sweep is the fact that genomic regions that have experienced a recent selective sweep are expected to have an excess of rare variants (i.e. the only mutations that contribute to polymorphism are recent and therefore still in low frequency). The excess of rare variants in a window can be
measured through the summary statistics Tajimas D. Negative Tajimas D values mean an excess of rare variants (SNPs) while positive Tajimas D means an excess of frequent SNPs. Are windows exhibiting abnormally low amounts of polymorphism also displaying more negative Tajimas D values?

Firstly a visual inspection:

```
myPlot1 <- ggplot(data = medicago3) + geom_point(aes(x = window, y = qp.site,
colour = factor(chr))) + theme(axis.text.x  = element_blank(), axis.ticks.x =
element_blank(), axis.title.x = element_blank(), legend.position = 'none')
myPlot2 <- ggplot(data = medicago3) +  geom_point(aes(x=window, y = TajimasD,
colour = factor(chr))) + theme(axis.text.x  = element_blank(), axis.ticks.x =
element_blank(), legend.position = 'none')
grid.arrange(myPlot1, myPlot2, nrow = 2)
```

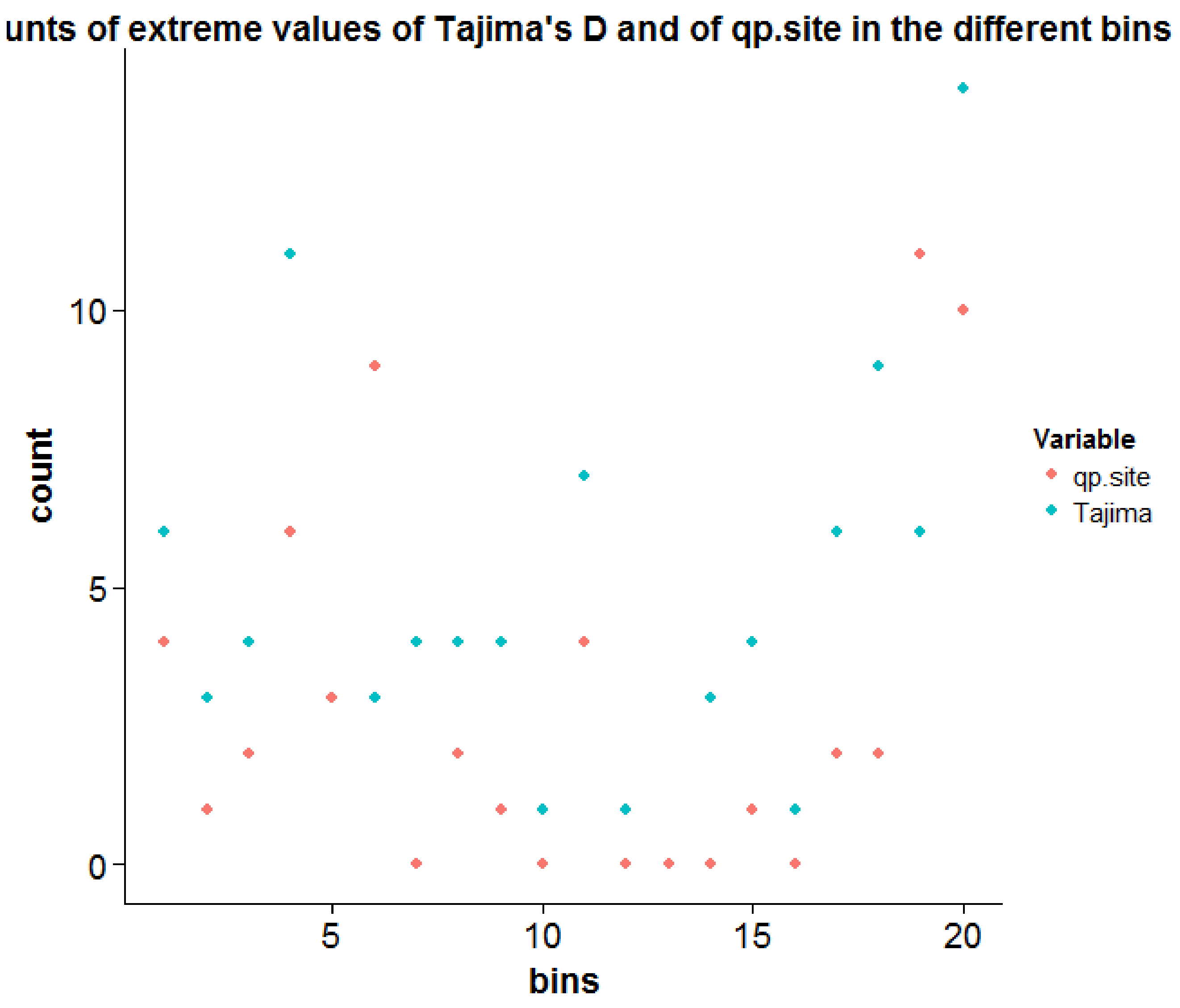


There seems to be a vague pattern when expecting visually, but the interest is on the extreme values. There the number of 5% most negative values of Tajima's D for ecah bin are found and gathered in a variable:

```
cor.test(medicago$TajimasD, medicago$qp.site, method = "k")

##
##  Kendall's rank correlation tau
##
## data:  medicago$TajimasD and medicago$qp.site
## z = 33.23, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.4886439
```

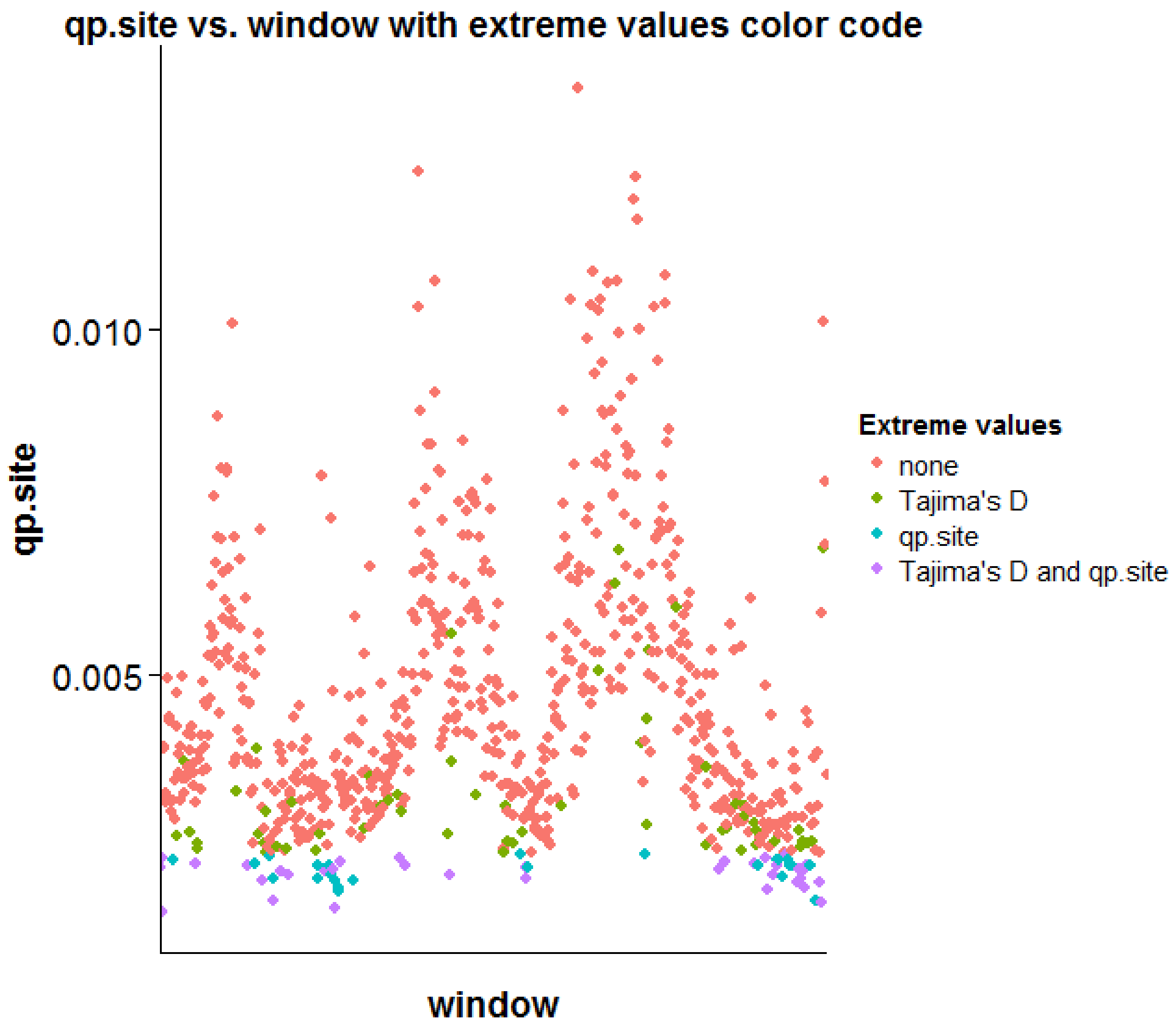
So there is definately a significant correlation between the two variables, but when looking at the 5% most extreme cases of both variables, the pattern is a bit less conclussive:

```r
cutoff_qp.site <- cutoff
cutoff_tajima <-
sort(medicago$TajimasD)[ceiling(0.1*length(medicago$TajimasD))]
counts_qp.site <- counts
counts_tajima <- rep(1, max(medicago3$bins))
remove(counts)
remove(cutoff)
for (i in seq(1, max(medicago3$bins))){
  medicago3 %>%
    filter(bins == i )%>%
    filter(TajimasD <= cutoff_tajima) -> temp
   length(temp$qp.site) -> counts_tajima[i]
}
bins<-medicago3$bins
medicago3$counts_tajima <- counts_tajima[bins]
medicago3$counts_qp.site <- counts_qp.site[bins]
ggplot(data = medicago3) + geom_point(aes(x=bins, y = counts_tajima, colour =
"Tajima"))+
  geom_point(aes(x = bins, y = counts_qp.site, colour = "qp.site")) +
  ggtitle("Counts of extreme values of Tajima's D and of qp.site in the
different bins")+
  scale_y_continuous(name="count") +
  scale_colour_discrete("Variable")
```



In the end it seems that there is some pattern as to the where *qp.site* is low and *Tajima's D* is highly negative, but it is not very distinct or at least not in the way explored here. Since this plot does not show a whole lot as it shows the problem on bin level. A plot will now be produced that shows the extreme values of *qp.site* and *Tajima's D* on vindow level:

```
#first a code for the extreme values are derived 0 is no extreme values, 1 is
Tajima's D
#2 is qp.site and 3 is both:
extremes <- seq(1, length(medicago3$qp.site))
extremes <- lapply(extremes, FUN = function(x) ifelse(medicago3$qp.site[x]<=
cutoff_qp.site, 2, 0) + ifelse(medicago3$TajimasD[x] <= cutoff_tajima,1,0))
extremes <- as.numeric(extremes)
medicago3$extremes <- extremes

myPlot <- ggplot(data = medicago3, aes(x=window, y = qp.site)) +
  geom_point(aes(colour = factor(extremes))) +
  scale_colour_discrete(name = "Extreme values", labels = c("none", "Tajima's
D", "qp.site", "Tajima's D and qp.site"))+
  theme(axis.text.x= element_blank(), axis.ticks.x = element_blank())
myPlot + ggtitle("qp.site vs. window with extreme values color code")
```



From this it becomes more apparent that extreme values of the two variables actually tend to occur together as the number of appearences of only one extreme values is 24 while the number of occurrences where both are extreme is 34. It is thus clear that extreme values of Tajima's D tend to occur in the same windows as those that have extreme values of *qp.site*.