BSG - Homework 3 - Linkage disequilibrium

HENRY QIU LO & MEYSAM ZAMANI

December 08, 2019

2. (1p) Load the FOXP2.dat file into the R environment. How many individuals and how many SNPs are there in the database? What percentage of the data is missing?

First of all we are going to install and load those packages that we are going to use them during this assignment.

```
#install.packages("genetics")
#install.packages("HardyWeinberg")
#install.packages("snpStats")
#install.packages("LDheatmap")
library(genetics)
library(HardyWeinberg)
library(LDheatmap)
```

Next step is to loading data:

```
data <- read.delim("FOXP2/FOXP2.dat", header = TRUE, row.names = 1, sep=" ")
dataset <- read.table("FOXP2/FOXP2.dat", header = T)
dataset <- dataset[2:length(dataset)]
removeSlashes <- function(x) {
  gsub("/", "", x, fixed=TRUE)
}
data <- as.data.frame(apply(data, c(1,2), removeSlashes))

nrows <- nrow(data)
ncols <- ncol(data)
```

There are 104 individuals and 543 SNPs.

```
percentageMissing <- 100*sum(is.na(data))/(nrows*ncols)
percentageMissing
```

```
## [1] 0
```

There is not any missing data in this dataset.

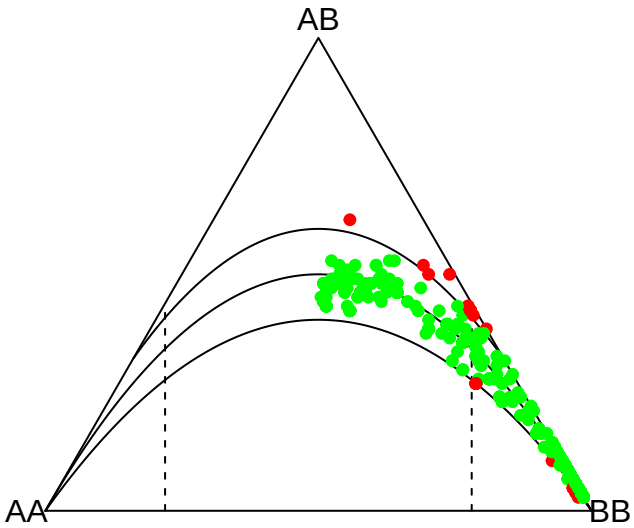
3. (1p) Determine the genotype counts for each SNP, and depict all SNPs simultaeneously in a ternary plot, and comment on your result. For how many variants do you reject Hardy-Weinberg equilibrium using an ordinary chi-square test without continuity correction? (hint: you can read the .bim in R in order to determine the alleles of each SNP, and use function MakeCounts from the HardyWeinberg package to create a matrix of genotype counts).

```
bimData <- read.delim("FOXP2/FOXP2.bim", header = FALSE, sep="\t")
alleles <- c(do.call(paste,c(bimData[c("V5", "V6")],sep="/")))
counts <- MakeCounts(data, alleles)
counts[1:10, 1:3]
```

```
##       AA AB  BB
##  [1,]  3 28  73
##  [2,]  8 28  68
##  [3,]  8 28  68
```

```
##  [4,]  0 15  89
##  [5,] 17 46  41
##  [6,]  8 28  68
##  [7,]  8 28  68
##  [8,]  0  3 101
##  [9,]  8 28  68
## [10,] 12 46  46
```

```
HWTernaryPlot(counts[, 1:3])
```




```
test <- HWChisqStats(counts[, 1:3], pvalues=T)
significant <- test[test < 0.05]

print(paste0("Variants for which we reject equilibrium: ", length(significant)))
```

```
## [1] "Variants for which we reject equilibrium: 33"
```

We are going to reject 33 variants

4. (1p) Using the function LD from the genetics package, compute the LD statistic D for the SNPs rs34684677 and rs2894715 of the database. Is there significant association between the alleles of these two SNPs?

```
datars34684677 <- data$rs34684677
datars2894715 <- data$rs2894715
gtrs34684677 <- genotype(datars34684677,sep="")
gtrs2894715 <- genotype(datars2894715,sep="")
ldQ4 <- LD(gtrs34684677,gtrs2894715)
ldQ4$D
```

```
## [1] -0.05493703
```

```
ldQ4$`D'`
```

```
## [1] 0.9986536
```

```
ldQ4$`P-value`
```

```
## [1] 5.77645e-06
```

The value of the LD statistic D is 0.9986536, which is very high, this means the imbalance between the alleles is big, which means the association between them are quite high. The high value of statistic D' also indicates the strong relation. Lastly, a P-value smaller than 0.05 indicates us the results are reliable.

5. (2p) Also compute the LD statistic D for the SNPs rs34684677 and rs998302 of the database. Is there significant association between these two SNPs? Is there any reason why rs998302 could have stronger or weaker correlation than rs2894715?

```
datars998302 <- data$rs998302
gtrs998302 <- genotype(datars998302,sep="")
ldQ5 <- LD(gtrs34684677,gtrs998302)
ldQ5$D
```

```
## [1] 0.007208888
```

The value of the LD statistic D is 0.007208888, which is very low, this means the balance between the alleles is big, which means the association between them are small because they tends to be inherited randomly. The low value of statistic D' also indicates the weak relation. Lastly, a P-value smaller than 0.05 indicates us the results are reliable. The reason for which rs2894715 could have stronger correlation than rs998302 can be that rs2894715 is closer to rs34684677 in the chromosome.

6. (2p) Given your previous estimate of D for SNPs rs34684677 and rs2894715, infer the haplotype frequencies. Which haplotype is the most common?

```
pA <- summary(gtrs34684677)$allele.freq[,2][1]
pB <- summary(gtrs2894715)$allele.freq[,2][1]
pa <- summary(gtrs34684677)$allele.freq[,2][2]
pb <- summary(gtrs2894715)$allele.freq[,2][2]

halotypeAB <- ldQ4$D + pA * pB
halotypeab <- ldQ4$D + pa * pb
halotypeAB
```

```
##         G
## 0.5000741
```

```
halotypeab
```

```
##            T
## 7.406505e-05
```

To get te result we used the formula D = pAB - pApB or D = pab - papb to know pAB and pab. Where we can see halotype GT' is 0.5000741, where G is and TG' is 7.406505e-05, where G and T are alleles from rs34684677 T' and G' are from gtrs2894715

7. (2p) Compute the LD statistics R2 for all the marker pairs in this data base, using the LD function of the packages genetics. Be prepared that this make take a few minutes. Also compute an alternative estimate of R2 obtained by using the PLINK program.

```
res <- data.frame(genotype(dataset[, 1], sep = "/"))

for(i in 2 : ncol(dataset)) {
  snp <- genotype(dataset[, i], sep = "/")
  res <- cbind(res, snp)
}

r2s <- LD(res)
test.r2 <- r2s$"R^2"

test.r2.2 <- matrix(nrow = ncol(dataset), ncol = ncol(dataset))
for (i in 1:ncol(dataset)) {
  for (j in i:ncol(dataset)) {
    test.r2.2[i, j] = test.r2[i,j]
    test.r2.2[j, i] = test.r2[i,j]
    if (i == j) {
      test.r2.2[i, j] = 1
    }
  }
}

buildMatrix <- function(x){
  m <- matrix(nrow = ncol(x), ncol = ncol(x))
  for(i in 1:nrow(m)) {
    for(j in 1:ncol(m)) {
      gt1 <- genotype(data[,i],sep="")
      gt2 <- genotype(data[,j],sep="")
      m[i,j] <- LD(gt1,gt2)$`D'`
    }
  }
  return(m)
}

vector.r2 <- as.vector(test.r2.2)
vector.r2[1:10]
```

```
##  [1] 1.000000000 0.727480818 0.727480818 0.397153491 0.121838859
##  [6] 0.727480818 0.727480818 0.002686437 0.727480818 0.098850392
```

```
#vector.plink <- as.vector(buildMatrix)
#vector.plink[1:10]

#dataframe.r2 <- data.frame(vector.r2, vector.plink)
#names(dataframe.r2) <- c("R^2 - LD", "R^2 - PLINK")
#dataframe.r2$difference <- dataframe.r2$`R^2 - LD` - dataframe.r2$`R^2 - PLINK`
```

[1] "Correlation coefficient between the R^2 estimators is 0.995"

```
#val <- vector.r2 + vector.plink
#valcol <- (val + abs(min(val)))/max(val + abs(min(val)))
```

The difference is that PLINK is much more faster to calculate, because it provides a lot of options for the execution, what is more, it provides a good way to screen for strong LD. For this reason I prefer PLINK option.

8. (2p) Compute a distance matrix with the distance in base pairs between all possible pairs

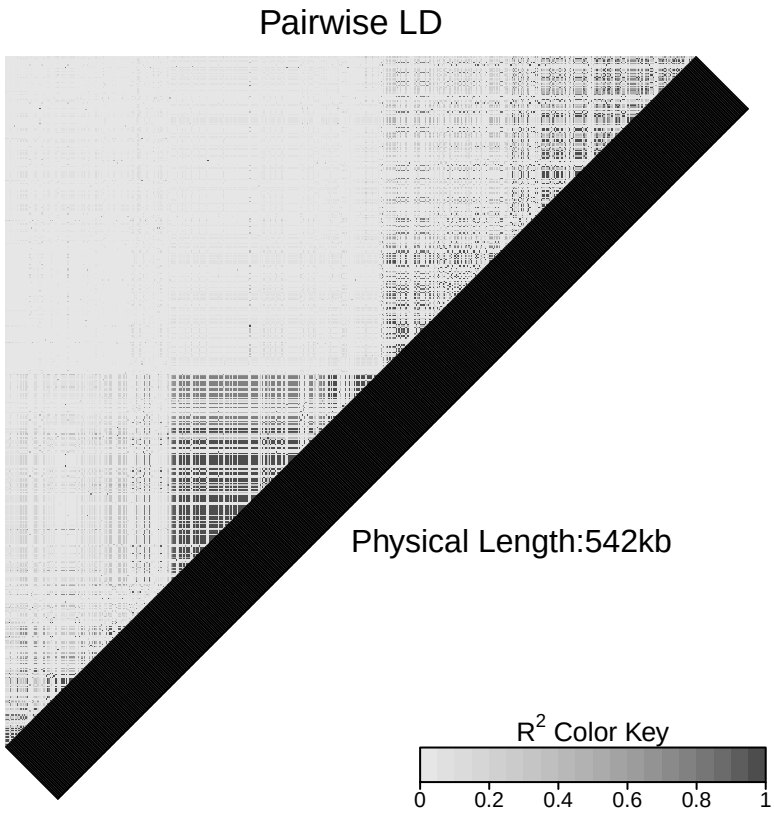
of SNPs, using the basepair position of each SNP given in the .bim file. Make a plot of R's R2 statistics against the distance (expressed as the number of basepairs) between the markers. Comment on your results.

```
#positions <- dataInfo$snp.info[["position"]]
#distances <- dist(positions)

#segments(0, 300000, 1, 0, lwd = 2, lty = 2)
```

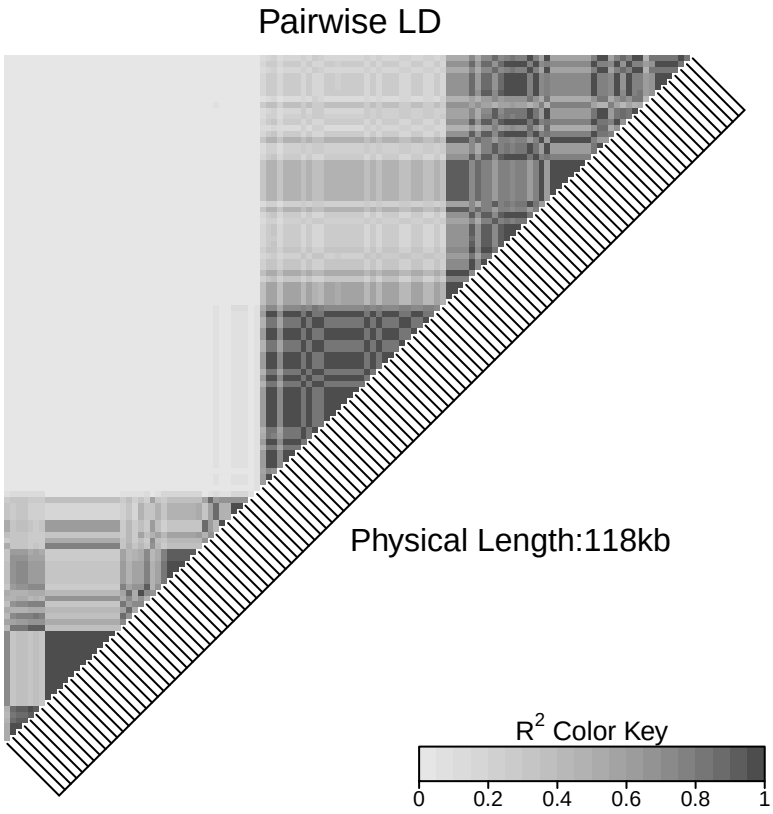
The SNP pairs that have a high R^2 statistic are those that have a small distance between them. Since the R^2 statistic is the squared correlation between these indicators, these shows that the SNP's that are closer to each other have a higher correlation.

9. (2p) Make an LD heatmap of the markers in this database, using the R2 statistic with the LD function. Make another heatmap obtained by filtering out all variants with a MAF below 0.35, and redoing the computations to obtain the R2 statistics in R. Can you explain any differences observed between the two heatmaps?

```
LDheatmap(res, LDmeasure="r")
```




```
mafCount <- function(x) {
  sumInfo <- summary(genotype(x))
  minPercentCount <- min(sumInfo$allele.freq[, 2])
  return(minPercentCount*100)
}
mafVector <- apply(dataset, 2, mafCount)
res.maf <- res[mafVector > 35]
LDheatmap(res.maf, LDmeasure="r")
```



After filtering, the consequence is that the visible blocks of the first plot become more evident. This is due to correlation which becomes substantially high.

10. (1p) Can you distinguish blocks of correlated markers in the area of the FOXP2 gene? How many blocks do you think that at least seem to exist?

Blocks are pretty distinguishable, in fact we distinguish 2 main blocks, each of them is composed by other smaller blocks; in particular one is composed by two and the other one by three strong blocks.

11. (1p) Simulate independent SNPs under the assumption of Hardy-Weinberg equilibrium, using R's sample instruction (sample(c("AA","AB","BB"),n,replace=TRUE,prob=c(pp,2pq,qq))). Simulate as many SNPs as you have in your database, and take care to match each SNP in your database with a simulated SNP that has the same sample size and allele frequency. Make an LD heatmap of the simulated SNPs, using R2 as your statistic. Compare the results with the LD heatmap of the FOXP2 region. What do you observe? State your conclusions.

```
dataset[1:10, 1:10]
```

```
##    rs34684677 rs1839115 rs4727804 rs4727805 rs200888633 rs12534908
## 1         T/G       C/T       G/A       T/G         T/G        G/A
## 2         G/G       T/T       A/A       G/G         T/G        A/A
## 3         G/G       T/T       A/A       G/G         T/G        A/A
## 4         G/G       T/T       A/A       G/G         T/T        A/A
## 5         G/G       T/T       A/A       G/G         T/T        A/A
## 6         T/T       C/C       G/G       T/G         G/G        G/G
## 7         G/G       T/T       A/A       G/G         G/G        A/A
## 8         T/G       C/T       G/A       G/G         G/G        G/A
## 9         T/G       C/T       G/A       G/G         T/G        G/A
## 10        G/G       T/T       A/A       G/G         G/G        A/A
```

```
##    rs12533049 rs77861356 rs6945561 rs2894715
## 1         C/T        T/T       C/T       G/T
## 2         T/T        T/T       T/T       G/T
## 3         T/T        T/T       T/T       G/T
## 4         T/T        T/T       T/T       G/G
## 5         T/T        T/T       T/T       G/G
## 6         C/C        T/T       C/C       T/T
## 7         T/T        T/T       T/T       T/T
## 8         C/T        T/T       C/T       T/T
## 9         C/T        T/T       C/T       G/T
## 10        T/T        A/T       T/T       T/T
```

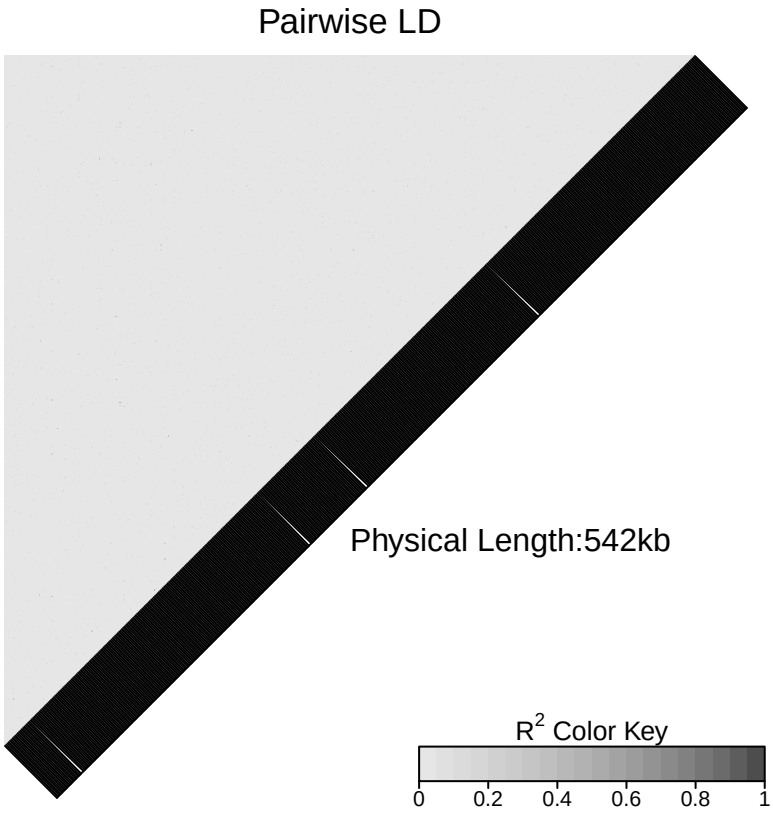
```
nrow(dataset)
```

```
## [1] 104
```

```
dataset.sampled <- data.frame(matrix(ncol = ncol(dataset), nrow = nrow(dataset)))

for (i in 1:ncol(dataset)) {
  column <- dataset[,i]
  column.geno <- summary(genotype(column))
  allele.frequency <- column.geno$allele.freq[,2]
  q <- 0
  p <- allele.frequency[1]
  if (length(allele.frequency) > 1) {
    q <- allele.frequency[2]
  }
  dataset.sampled[,i] <- sample(c("A/A","A/B","B/B"), nrow(dataset), replace = TRUE, prob = c(p*p,2*p*q
}
dataset.sampled[1:10,1:10]
```

```
##     X1  X2  X3  X4  X5  X6  X7  X8  X9 X10
## 1  A/B A/A A/B A/A A/A A/A A/A A/A A/A B/B
## 2  A/B A/B A/A A/A A/A A/B A/A A/A A/A A/B
## 3  A/A A/B A/B A/B A/B A/B A/A A/A A/A A/A
## 4  A/A A/B A/B A/A A/A A/A A/B A/A A/B A/B
## 5  A/B A/A A/A A/A A/B A/A A/A A/A A/A A/B
## 6  A/A A/B A/A A/A A/B A/A A/A A/A A/A A/B
## 7  A/B A/A A/B A/A A/B A/A A/B A/A A/A A/B
## 8  B/B A/A A/A A/A A/A A/B A/A A/A A/A A/A
## 9  A/A A/A A/A A/A A/A A/A A/B A/A A/B A/A
## 10 A/A B/B A/A A/A A/A A/A A/B A/A A/B B/B
```

```
res.sampled <- data.frame(genotype(dataset.sampled[, 1], sep = "/"))

for(i in 2 : ncol(dataset.sampled)) {
  snp <- genotype(dataset.sampled[, i], sep = "/")
  res.sampled <- cbind(res.sampled, snp)
}

LDheatmap(res.sampled, LDmeasure="r")
```



This simulation leads to no correlation between variables, because we are performing independent SNPs behavior, hence, as we expect, there are no distinguishable blocks.