Linkage disequilibrium

Barbara Lugar, Giovanni Laganà

```
library(genetics, quietly = T)
library(HardyWeinberg, quietly = T)
library(LDheatmap, quietly = T)
library(KRIS, quietly = T)
library(data.table, quietly = T)
```

2. Load the data. How many individuals and how many SNPs are there in the database? What percentage of the data is missing?

```
dataset <- read.table("FOXP2/FOXP2.dat", header = T)
dataset <- dataset[2:length(dataset)]
print(paste0("Number of individuals: ", nrow(dataset)))
```

```
## [1] "Number of individuals: 104"
```

```
print(paste0("Number of SNPs: ", ncol(dataset)))
```

```
## [1] "Number of SNPs: 543"
```

```
print(paste0("Percentage of missing data: ", sum(is.na(dataset))/length(dataset), "%"))
```

```
## [1] "Percentage of missing data: 0%"
```

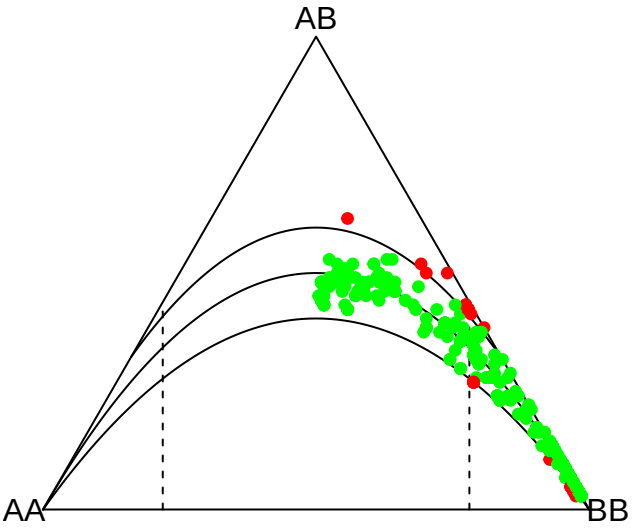
3. Determine the genotype counts for each SNP, and depict all SNPs simultaeneously in a ternary plot, and comment on your result. For how many variants do you reject Hardy-Weinberg equilibrium using an ordinary chi-square test without continuity correction? (hint: you can read the .bim in R in order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

```
dataInfo <- read.bed("FOXP2/FOXP2.bed", "FOXP2/FOXP2.bim", "FOXP2/FOXP2.fam")
snpInfo <- dataInfo$snp.info[, c(2,5,6)]
snpInfo$alleles <- paste0(as.character(snpInfo$allele1), '/', as.character(snpInfo$allele2))
allelesVector <- snpInfo$alleles

temp <- as.data.frame(lapply(dataset, function(y) gsub("/", "", y)))
counts <- MakeCounts(temp, as.vector(allelesVector))
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]
```

With the aim of not confusing letters, we specificy that we labelled the vertices with AA, AB and BB, since we have A, C, G, T variables, and with them we mean the heterozygous (AB) and homozygous cases (AA and BB). In order to have a better idea of the amount of data whose equilibrium is rejected, we display their number and percentage:

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

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

```
print(paste0("Percentage of out of equilibrium data: ", round(length(significant)/length(test)*100, 3),
```

```
## [1] "Percentage of out of equilibrium data: 6.077%"
```

The plot is unbalanced on the right, if we plot a sample of the dataset, we can see that the genotype counts are:

```
#table(dataInfo$snp.info$allele1, dataInfo$snp.info$allele2)

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
```

and since the frequency of BB is way higher than AA, we have this unbalance

```r
print(paste0("p(AA): ", round(sum(counts[, 1])/sum(counts), 3), "%"))
```

```
## [1] "p(AA): 0.066%"
```

```r
print(paste0("p(AB): ", round(sum(counts[, 2])/sum(counts), 3), "%"))
```

```
## [1] "p(AB): 0.274%"
```

```r
print(paste0("p(BB): ", round(sum(counts[, 3])/sum(counts), 3), "%"))
```

```
## [1] "p(BB): 0.66%"
```

```r
#snpInfo$alleles
#dataInfo$snp.info[3, ]
```

We notice that these data are close to the threshold of banana region, consequently we expect the amount of out of equilibrium data to change dramatically as soon as we try with a smaller one, we tried with 4%, 3%, 2% and 1%:

```r
significant <- test[test < 0.04]
print(paste0("Percentage of out of equilibrium data (with alfa = 0.04): ", round(length(significant)/le
```

```
## [1] "Percentage of out of equilibrium data (with alfa = 0.04): 4.236 %"
```

```r
significant <- test[test < 0.03]
print(paste0("Percentage of out of equilibrium data (with alfa = 0.03): ", round(length(significant)/le
```

```
## [1] "Percentage of out of equilibrium data (with alfa = 0.03): 4.052 %"
```

```r
significant <- test[test < 0.02]
print(paste0("Percentage of out of equilibrium data (with alfa = 0.02): ", round(length(significant)/le
```

```
## [1] "Percentage of out of equilibrium data (with alfa = 0.02): 3.683 %"
```

```r
significant <- test[test < 0.01]
print(paste0("Percentage of out of equilibrium data (with alfa = 0.01): ", round(length(significant)/le
```

```
## [1] "Percentage of out of equilibrium data (with alfa = 0.01): 1.473 %"
```

4. 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?

```r
test1 <- LD(genotype(dataset[["rs34684677"]]), genotype(dataset[["rs2894715"]]))
d1 <- test1[2]$D
test1
```

```
##
## Pairwise LD
## -----------
##                       D        D'       Corr
## Estimates: -0.05493703 0.9986536 -0.3144048
##
##               X^2     P-value   N
## LD Test: 20.56088 5.77645e-06 104
```

The D' factor is almost 1, that is to say that the coinheritance of the two variants is really high. p value is very small, and of course smaller than 0.05 so, by the fact that we reject independence between the two

SNPs, we guess there is an association between them. This fact is supplied by the non-negligible correlation value (it could mean there is a negative linear association between the two SNPs).

5. 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?

```
test2 <- LD(genotype(dataset[["rs34684677"]]), genotype(dataset[["rs998302"]]))
d2 <- test2[2]$D
test2
```

```
##
## Pairwise LD
## -----------
##                       D        D'       Corr
## Estimates: 0.007208888 0.1792444 0.09112725
##
##               X^2   P-value   N
## LD Test: 1.727268 0.1887601 104
```

In this case the D' factor is not high anymore. p value is more than 0.05 so we cannot reject independence between this two variants. Moreover, the correlation coefficient approaches to 0, which let us think that there is no association between them.

```
print(paste0("rs34684677 site: ", dataInfo$snp.info$position[dataInfo$snp.info$ID == "rs34684677"]))
```

```
## [1] "rs34684677 site: 114400288"
```

```
print(paste0("rs2894715 site: ", dataInfo$snp.info$position[dataInfo$snp.info$ID == "rs2894715"]))
```

```
## [1] "rs2894715 site: 114402794"
```

```
print(paste0("rs998302 site: ", dataInfo$snp.info$position[dataInfo$snp.info$ID == "rs998302"]))
```

```
## [1] "rs998302 site: 114687416"
```

Since we have rs2894715 closer to rs34684677 than rs998302, a possibile reason is that LD semantics could be the consequence of the physical closeness of the sites, then we can expect cor(rs34684677, rs998302) < cor(rs34684677, rs2894715), but this is not necessarily always true.

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

```
genFreq1 <- summary(genotype(dataset[["rs34684677"]]))
genFreq2 <- summary(genotype(dataset[["rs2894715"]]))

freqA <- max(genFreq1$allele.freq[, 2])
freqa <- min(genFreq1$allele.freq[, 2])

freqB <- max(genFreq2$allele.freq[, 2])
freqb <- min(genFreq2$allele.freq[, 2])

m <- matrix(data = c(freqA*freqB + d1, freqA*freqb - d1, freqa*freqB - d1, freqa*freqb + d1), nrow = 2,

rownames(m) <- c(rownames(genFreq1$allele.freq)[1], rownames(genFreq1$allele.freq)[2])
```

```
colnames(m) <- c(rownames(genFreq2$allele.freq)[1], rownames(genFreq2$allele.freq)[2])
m
```

```
##           T           G
## G 0.5000741 3.364644e-01
## T 0.1633875 7.406505e-05
```

Note: G, T variables of rs34684677 on rows (A, a) T, G variables of rs2894715 on columns (B, b)

Then the most common haplotype is G T (G from rs34684677 and T from rs2894715).

7. 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. Make a scatter plot for R's LD estimates against PLINK's LD estimates. Are they identical or do they at least correlate? What's the difference between these two estimators? Which estimator would your prefer and why?

```
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"
ld <- read.table("FOXP2/FOXP2.ld")

#The squared correlation based on genotypic allele counts is therefore not identical to the r-sq as est

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
    }
  }
}

plink.matrix <- matrix(nrow = ncol(dataset), ncol = ncol(dataset))
for (i in 1:ncol(dataset)) {
  for (j in i:ncol(dataset)) {
    plink.matrix[i, j] = ld[i][j,1]
    plink.matrix[j, i] = ld[i][j,1]
  }
}

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
```

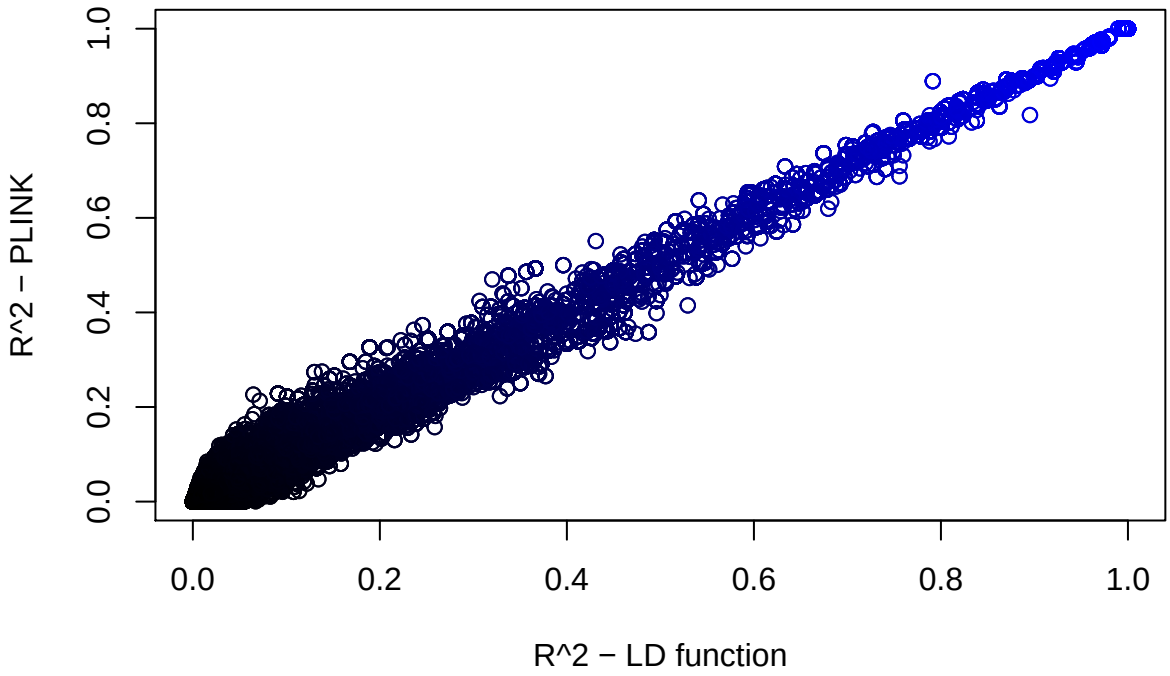
```
vector.plink <- as.vector(plink.matrix)
vector.plink[1:10]
```

```
##  [1] 1.0000000 0.7337180 0.7337180 0.3946210 0.1141820 0.7337180 0.7337180
##  [8] 0.0114303 0.7337180 0.1423540
```

```
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`

print(paste0("Correlation coefficient between the R^2 estimators is ", round(cor(vector.r2, vector.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)))
plot(vector.r2, vector.plink, xlab = "R^2 - LD function", ylab = "R^2 - PLINK", col = rgb(0, 0, valcol)
```

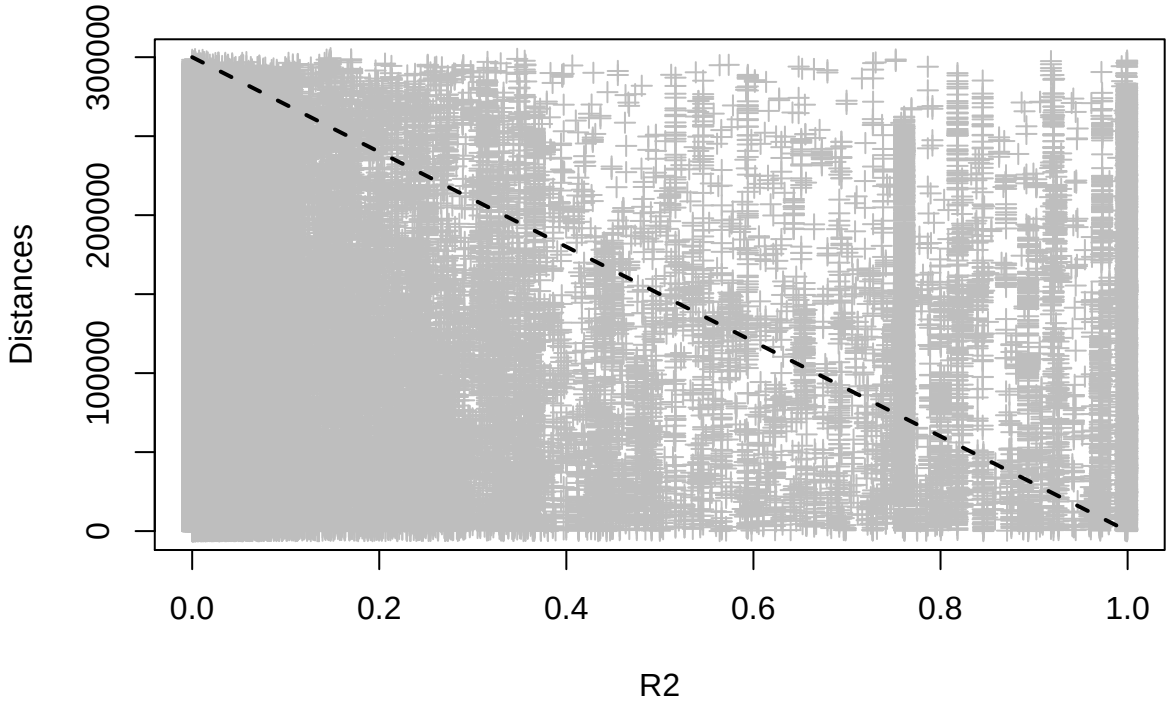


In order to be identical the correlation coefficient should be 1, than no, they are not identical, but they correlate very well (corr = 0.995). Concerning differences, the LD estimator is an unbiased estimator, since the R^2 computation is exact, while Plink command computes an approximation of it, the first one is much slower than the second, we guess that the trade off between an acceptable error and an acceptable time leads us to prefer Plink function.

##8. 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)

plot(test.r2.2[upper.tri(test.r2.2)], distances, pch = 3, col = "grey", xlab = "R2", ylab = "Distances"
```

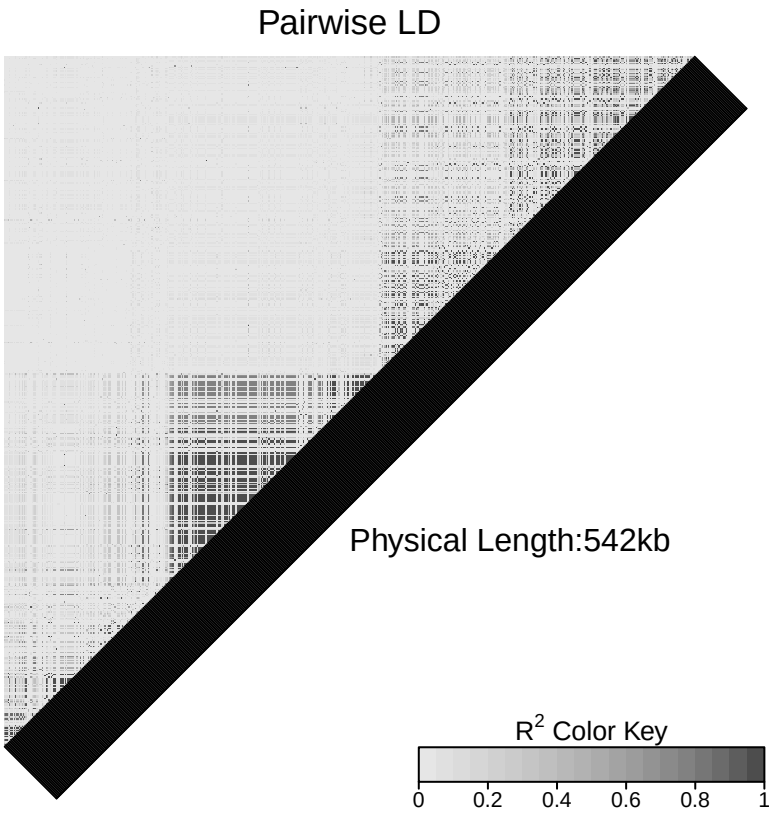
```
segments(0, 300000, 1, 0, lwd = 2, lty = 2)
```



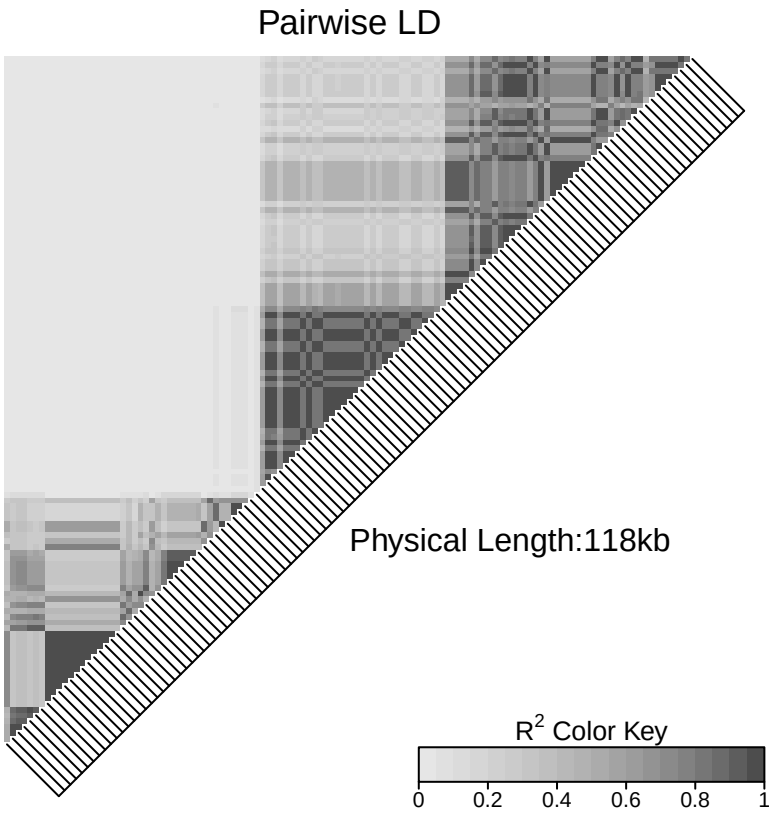
The SNP pairs that have a high R^2 statistic are those that have a small distance between them. Since the R^2 statistic is the squared correlation between these indicators, these shows that the SNP's that are closer to each other have a higher correlation.

9. 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 R 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. 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. 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 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
```

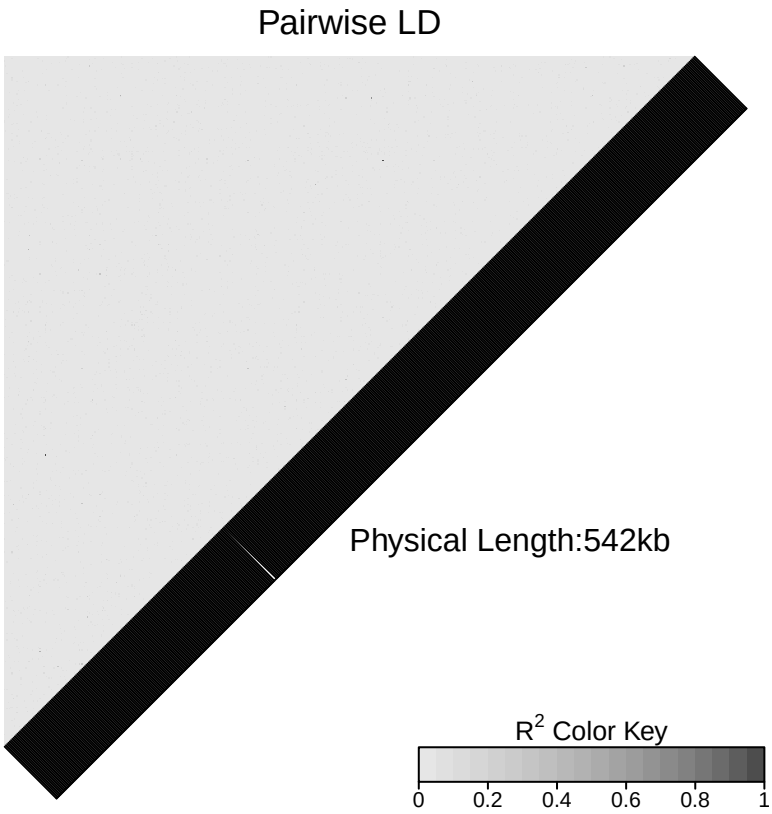
```
nrow(dataset)
```

```
## [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/A A/A A/A A/A A/A A/A A/B A/A A/B A/B
## 2  A/A A/B A/B A/A B/B A/A A/B A/A B/B A/B
## 3  A/B A/A A/A A/A A/B A/A A/A A/A A/A A/B
## 4  A/B A/A A/A A/A A/A A/A A/A A/A A/A A/B
## 5  A/A A/B A/A A/A A/A A/A A/A A/A A/A A/B
## 6  A/A A/B A/A A/B B/B A/B A/A A/A A/B B/B
## 7  A/B A/A A/A A/A A/A A/A A/A A/A A/B A/A
## 8  A/B A/A A/A A/A A/B A/B A/A A/A A/B A/A
## 9  A/A A/A A/A A/A A/B B/B A/A A/A A/A A/A
## 10 A/A A/B A/B A/A A/A A/A A/A A/A A/B A/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.