BSG-MDS practical 3 Statistical Genetics

Eliya Tiram and Ximena Moure

21/11/2023, submission deadline 28/11/2023

Linkage Disequilibrium

1.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?

```
FOXP2_data <- read.table("FOXP2/FOXP2.dat", header = TRUE)

FOXP2_data <- FOXP2_data[,2:ncol(FOXP2_data)]

n_individuals <- nrow(FOXP2_data)
n_SNPs <- ncol(FOXP2_data)
percent_missing <- sum(is.na(FOXP2_data)) / (n_individuals * n_SNPs) * 100

cat("\nNum individuals:",n_individuals, "\n")
```

```
##
## Num individuals: 104
```

```
cat("\nNum SNPs:",n_SNPs, "\n")
```

```
##
## Num SNPs: 543
```

```
cat("\nPercentage missing:",percent_missing, "\n")
```

```
##
## Percentage missing: 0
```

2. 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?

There is a statistically significant association between the alleles of these two SNPs, as indicated by the low p-value and the high D' value (close to 1).

```
rs34684677 <- FOXP2_data[,c("rs34684677")]
rs2894715 <- FOXP2_data[,c("rs2894715")]
rs34684677.gen <- genotype(rs34684677,sep="/")
rs2894715.gen <- genotype(rs2894715,sep="/")
ld <- LD(rs34684677.gen,rs2894715.gen)
ld
```

```
##
## 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
```

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

We can do this using haplo.em or using the previous estimate of D.

Using haplo.em we get that the most common haplotype between rs34684677 and rs2894715 SNPs is "G/T" with freq=0.5.

Using the estimate of D we get that the most common haplotype is GT with a frequency of 0.61.

Even though we get for both that the most common is GT we do get different frequencies, but this might be due to the fact that the EM algorithm uses a probabilistic approach to handle uncertain phase information, which can lead to different haplotype frequency estimates compared to methods that use only allele frequencies and D.

```
Geno <- cbind(substr(FOXP2_data$rs34684677, 1, 1),
              substr(FOXP2_data$rs34684677, 3, 3),
              substr(FOXP2_data$rs2894715, 1, 1),
              substr(FOXP2_data$rs2894715, 3, 3))

snpnames <- c("rs34684677", "rs2894715")
haplo_em <- haplo.em(Geno, locus.label = snpnames)

print(haplo_em)
```

```
## ============================================================================
##                                  Haplotypes
## ============================================================================
##   rs34684677 rs2894715 hap.freq
## 1          G         G  0.33654
## 2          G         T  0.50000
## 3          T         G  0.00000
## 4          T         T  0.16346
## ============================================================================
##                                   Details
## ============================================================================
## lnlike =  -164.8458
## lr stat for no LD =  18.69923 , df =  0 , p-val =  NA
```

```r
D <- ld$D

# Calculate allele frequencies for SNP rs34684677
pG <- (sum(rs34684677.gen == "G/G")*2 + sum(rs34684677.gen == "G/T")) /
  (2 * length(rs34684677.gen))
pT <- 1 - pG

# Calculate allele frequencies for SNP rs2894715
qG <- (sum(rs2894715.gen  == "G/G")*2 + sum(rs2894715.gen  == "T/G")) /
  (2 * length(rs2894715.gen ))
qT <- 1 - qG

# Calculate haplotype frequencies
hGG <- pG * qG + D
hGT <- pG * qT - D
hTG <- pT * qG - D
hTT <- pT * qT + D

haplo_freqs <- c(hGG, hGT, hTG, hTT)
names(haplo_freqs) <- c("hGG", "hGT", "hTG", "hTT")

print(round(haplo_freqs, 3))
```

```
##   hGG   hGT   hTG   hTT
## 0.227 0.610 0.110 0.054
```

```r
most_common_haplo <- names(which.max(haplo_freqs))
cat("The most common haplotype is:", most_common_haplo, "with a frequency of:",
    round(max(haplo_freqs), 3), "\n")
```

```
## The most common haplotype is: hGT with a frequency of: 0.61
```

4. Determine the genotype counts for each SNP. For how many variants do you reject Hardy-Weinberg equilibrium using an ordinary chi-square test without continuity correction? Is this what you would expect by chance? (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).

We reject Hardy-Weinberg equilibrium for 33 variants.

We have 33 SNPs out of equilibrium when 27 are expected so it is not a very large deviation.

In conclusion, it is not what we expect by chance.

```r
bim <- read.table("FOXP2/FOXP2.bim", header=FALSE)
bim_alleles <- paste(bim[,5],bim[,6],sep = "/")
gendata_counts <- MakeCounts(FOXP2_data, bim_alleles, sep = "/")
```

```r
chisq_stats <- HWChisqStats(gendata_counts,pvalues = FALSE)
chisq_pval <- HWChisqStats(gendata_counts,pvalues = TRUE)
chisq_pval_sig <- sum(chisq_pval<0.05) # number of significant SNPs
chisq_pval_sig
```

```
## [1] 33
```

```r
expected_by_chance <- nrow(bim) * 0.05
cat("\nExpected by chance:",expected_by_chance, "(5%)\n")
```

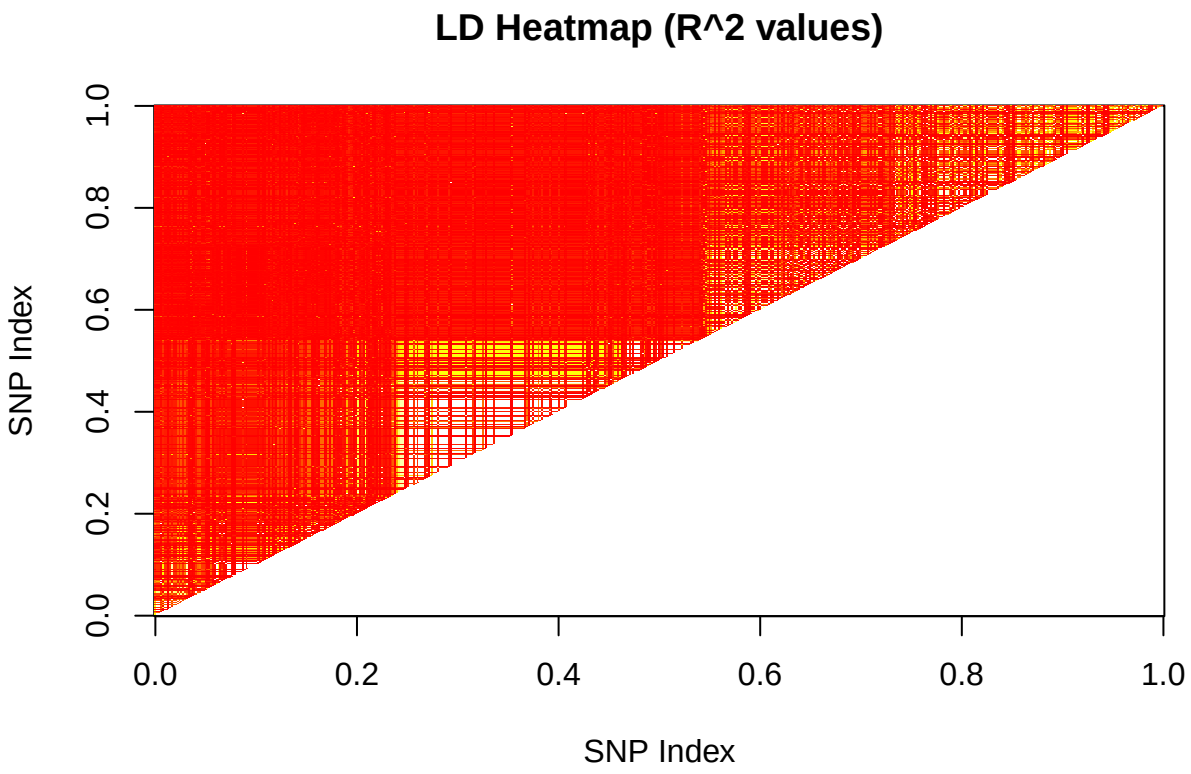
```
##
## Expected by chance: 27.15 (5%)
```

```r
# Total number of SNPs tested
total_SNPs <- nrow(bim)

# Calculate the percentage of significant variants
percentage_significant <- (chisq_pval_sig / total_SNPs) * 100

cat("\nPercentage of significant variants:", percentage_significant, "%\n")
```

```
##
## Percentage of significant variants: 6.077348 %
```

5. Compute the LD 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. Extract the R2 statistics and make an LD heatmap (hint: you can use the command image) using the R2 statistic.

```r
# Extract SNP names and convert to genotypes
snps <- colnames(FOXP2_data)
genotypes <- lapply(FOXP2_data, function(x) genotype(as.character(x)))

# Initialize a matrix to store R^2 values
r2_matrix <- matrix(NA, ncol = length(snps), nrow = length(snps),
                    dimnames = list(snps, snps))

# Compute LD and fill the matrix
for (i in 1:(length(snps) - 1)) {
    for (j in (i + 1):length(snps)) {
        ld_result <- LD(genotypes[[i]], genotypes[[j]])
        r2_matrix[i, j] <- ld_result$`R^2`
    }
}

image(r2_matrix, main = "LD Heatmap (R^2 values)", xlab = "SNP Index",
      ylab = "SNP Index", col = heat.colors(256))
```



6. 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?

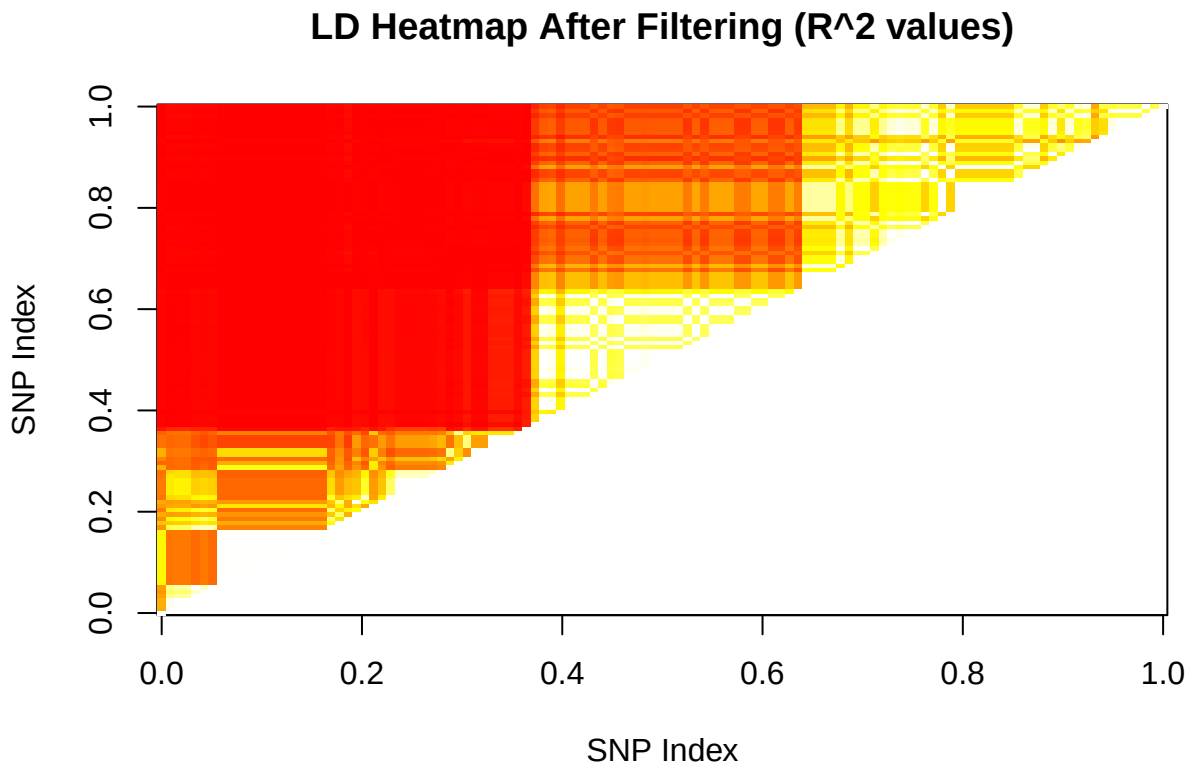
The first heatmap, which includes all variants, shows a higher density of strong linkage disequilibrium (LD) across the entire set of SNPs. This is indicated by the more extensive red areas, suggesting a greater number of SNP pairs with high R2 values.

The second heatmap, created after filtering out SNPs with a minor allele frequency (MAF) below 0.35, shows a reduction in the number of SNP pairs with strong LD. The heatmap appears more yellow than red, suggesting that the R2 values are generally lower.

The SNPs with a MAF below 0.35 played a substantial role in the LD structure of the dataset. By filtering out these SNPs, we could have potentially removed rare variants that may have been in strong LD due to genetic drift or selection effects.

```
calculate_MAF <- function(snp) {
  alleles <- unlist(strsplit(as.character(snp), "/", fixed = TRUE))
  freqs <- table(alleles) / length(alleles)
  return(min(freqs))
}

MAFs <- sapply(FOXP2_data, calculate_MAF)

df_filtered <- FOXP2_data[, MAFs >= 0.35]
```

```
genotypes_filtered <- lapply(df_filtered, function(x) genotype(as.character(x)))

snps_filtered <- colnames(df_filtered)
r2_matrix_filtered <- matrix(NA, ncol = length(snps_filtered),
                             nrow = length(snps_filtered),
                             dimnames = list(snps_filtered, snps_filtered))

for (i in 1:(length(snps_filtered) - 1)) {
    for (j in (i + 1):length(snps_filtered)) {
        ld_result <- LD(genotypes_filtered[[i]], genotypes_filtered[[j]])
        r2_matrix_filtered[i, j] <- ld_result$`R^2`
    }
}
diag(r2_matrix_filtered) <- 1

image(r2_matrix_filtered, main = "LD Heatmap After Filtering (R^2 values)",
      xlab = "SNP Index", ylab = "SNP Index", col = heat.colors(256))
```



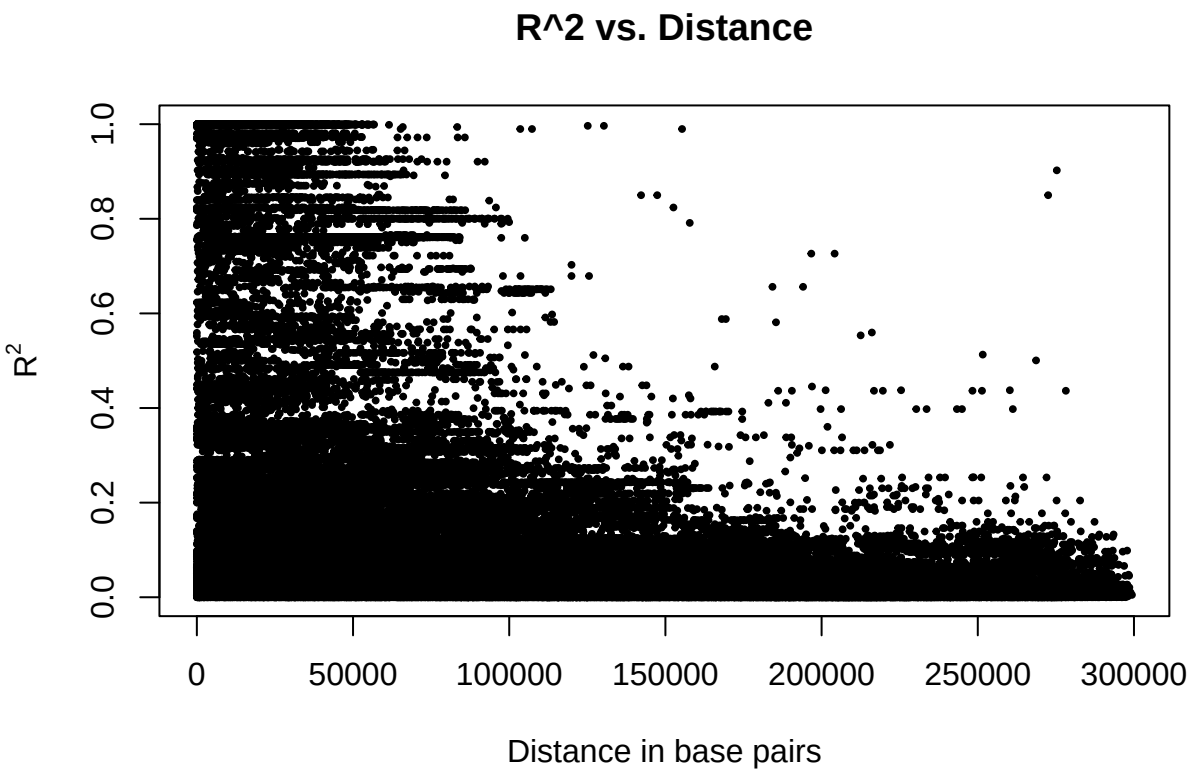
7. Compute a distance matrix with the distance in base pairs between all possible pairs of SNPs, using the basepair position of each SNP given in the .bim file. Make a plot of R's R2 statistics against the distance (expressed as the number of basepairs) between the markers. Comment on your results.

The plot shows that at short distances, the R2 values are generally high, which is expected as SNPs that are closer together are less likely to be separated by recombination.

There's a heavy concentration of points at the lower end of the R2 scale, which indicates that many SNP pairs have low LD.

There are some points with high R2 values at larger distances. These could be SNP pairs that are physically apart but still show strong LD, possibly due to genetic linkage or recent mutations that haven't had time to recombine away.

```
basepair_positions <- bim[, 4]
num_snps <- length(basepair_positions)

# Initialize a matrix to store distances
distance_matrix <- matrix(NA, nrow = num_snps, ncol = num_snps)

# Compute the distances
for (i in 1:(num_snps - 1)) {
  for (j in (i + 1):num_snps) {
    distance_matrix[i, j] <- abs(basepair_positions[i] - basepair_positions[j])
    distance_matrix[j, i] <- distance_matrix[i, j] #Distance matrix is symmetric
  }
}

distances <- as.vector(distance_matrix)
r2_values <- as.vector(r2_matrix)

plot_data <- data.frame(Distance = distances, R2 = r2_values)

plot(plot_data$Distance, plot_data$R2,
       xlab = "Distance in base pairs",
       ylab = expression(R^2),
       main = "R^2 vs. Distance",
       pch = 20,
       cex = 0.6)
```



Haplotype estimation

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

```
APOE_data <- read.table("APOE/APOE.dat", header = TRUE)

APOE_data <- APOE_data[,2:ncol(APOE_data)]

n_individuals <- nrow(APOE_data)
n_SNPs <- ncol(APOE_data)

percent_missing <- sum(is.na(APOE_data)) / (n_individuals * n_SNPs) * 100

cat("\nNum individuals:",n_individuals, "\n")
```

```
##
## Num individuals: 107
```

```
cat("\nNum SNPs:",n_SNPs, "\n")
```

```
##
## Num SNPs: 162
```

```r
cat("\nPercentage missing:",percent_missing, "\n")
```

```
##
## Percentage missing: 0
```

2. Assuming that all SNPs are biallelic, how many haplotypes can theoretically be found for this data set?

```r
theoretical_haplotypes <- 2^n_SNPs
cat("\nTheoretically there can be :",theoretical_haplotypes,"haplotypes\n")
```

```
##
## Theoretically there can be : 5.846007e+48 haplotypes
```

3. Estimate haplotype frequencies using the haplo.em function that you will find in the haplo.stats package. How many haplotypes do you find? List the estimated probabilities in decreasing order. Which haplotype number is the most common?

```r
geno <- data.frame(row.names = row.names(APOE_data))
for(i in 1:ncol(APOE_data)){
  geno <- cbind(geno,substr(APOE_data[,i],1,1),substr(APOE_data[,i],3,3))
}
geno_matrix <- as.matrix(geno)

snpts <- colnames(APOE_data)
haplo_em <- haplo.em(geno_matrix,locus.label = snpts)

haplotypes <- length(haplo_em$hap.prob)

cat("\nHaplotypes :",haplotypes,"\n")
```

```
##
## Haplotypes : 34
```

```r
probabilities <- sort(haplo_em$hap.prob,decreasing = T)
cat("\nProbabilities in decreasing order :",probabilities,"\n")
```

```
##
## Probabilities in decreasing order : 0.4027266 0.1308411 0.07167119 0.06821167 0.05020965 0.04548184 0
```

```r
most_common <-  which.max(haplo_em$hap.prob)
cat("\nMost common :", most_common,"\n")
```

```
##
## Most common : 29
```

4.Remove all genetic variants that have a minor allele frequency below 0.10 from the database, and re-run haplo.em. How does this affect the number of haplotypes? Comment on your results.

The SNPs with a MAF below 0.10 are less common variants in the population. By removing these SNPs, we are excluding less frequent genetic variations from the analysis. This reduction in genetic diversity naturally leads to a decrease in the number of unique haplotypes. Essentially, we're focusing on more common genetic patterns shared among individuals.

```
bim_he_data <- read.table("APOE/APOE.bim", header = FALSE,
                          col.names = c("chromosome", "SNP", "genetic_dist",
                                        "position", "allele1", "allele2"))
allele_pairs <- paste(bim_he_data$allele1, bim_he_data$allele2, sep = "/")

df <- APOE_data
df[] <- lapply(df, function(x) gsub("/", "", x))
df_counts <- MakeCounts(df, allele_pairs)
df_counts[,c(1,3)] <- df_counts[,c(3,1)]
pb <- numeric(nrow(df_counts))
for(i in 1:nrow(df_counts)) {
  pb[i] <- (df_counts[i,3] + 0.5 * df_counts[i,2]) / sum(df_counts[i,])
}

to_remove <- which(pb<0.1);

df_filtered <- APOE_data[,-to_remove]

geno <- data.frame(row.names = row.names(df_filtered))

for(i in 1:ncol(df_filtered)){
  geno <- cbind(geno,substr(df_filtered[,i],1,1),substr(df_filtered[,i],3,3))
}
geno <- as.matrix(geno)
snpnames <- colnames(df_filtered)
HaploEM <- haplo.em(geno, locus.label = snpnames)
prob <- sort(HaploEM$hap.prob, decreasing = T)
cat("Number of haplotypes:", length(prob), "\n")
```

```
## Number of haplotypes: 9
```