BSG-MDS practical 6 Statistical Genetics

Eliya Tiram and Ximena Moure

12/12/2023, submission deadline 19/12/2023

1.Load the YRI06.raw file into the R environment. How many individuals and how many SNPs are there in the database? What percentage of the data is missing?

```
# Load data
yri6 <- fread("YRI6.raw")

yri6_snps <- yri6[, 7:ncol(yri6)]

yri6_snps <- as.data.frame(yri6_snps)


# Number of individuals and SNPs
num_individuals <- nrow(yri6_snps)
num_snps <- ncol(yri6) - 6  # Subtracting non-SNP columns

# Calculate missing data percentage
missing_data_percentage <- sum(is.na(yri6[, 7:ncol(yri6)])) /
  (num_individuals * num_snps) * 100

# Output
cat("Number of individuals:", num_individuals, "\n")
```

```
## Number of individuals: 84
```

```
cat("Number of SNPs:", num_snps, "\n")
```

```
## Number of SNPs: 56574
```

```
cat("Percentage of missing data:", missing_data_percentage, "%\n")
```

```
## Percentage of missing data: 0 %
```

2. Compute, for each pair of individuals (and report the first 5), the mean m of the number of alleles shared and the standard deviation s of the number of alleles shared.

```r
# Function to calculate shared alleles
shared_alleles <- function(a, b) {
  2 - abs(a - b)
}

num_individuals <- nrow(yri6)
num_combinations <- choose(num_individuals, 2)
results <- matrix(nrow = num_combinations, ncol = 3)
colnames(results) <- c("Pair", "m", "s")

# Generate combinations of individuals
combos <- combn(num_individuals, 2)
counter <- 1

# Loop through each combination of individuals
for (i in 1:ncol(combos)) {
  ind1 <- combos[1, i]
  ind2 <- combos[2, i]

  # Extract SNP data for the pair
  snps1 <- as.integer(yri6[ind1, 6:ncol(yri6)])
  snps2 <- as.integer(yri6[ind2, 6:ncol(yri6)])

  shared <- shared_alleles(snps1, snps2)


  m <- mean(shared, na.rm = TRUE)
  s <- sd(shared, na.rm = TRUE)

  results[counter, ] <- c(paste("Ind", ind1, "- Ind", ind2), m, s)
  counter <- counter + 1
}

results_df <- as.data.frame(results)
results_df$m <- as.numeric(as.character(results_df$m))
results_df$s <- as.numeric(as.character(results_df$s))

head(results_df, 5)
```

```
##            Pair        m         s
## 1 Ind 1 - Ind 2 1.248714 0.6618949
## 2 Ind 1 - Ind 3 1.245126 0.6616178
## 3 Ind 1 - Ind 4 1.247212 0.6632975
## 4 Ind 1 - Ind 5 1.246681 0.6610929
## 5 Ind 1 - Ind 6 1.245621 0.6606320
```

3.Compute, for each pair of individuals (and report the first 5), the fraction of variants for which the individuals share 0 alleles (p0), and the fraction of variants for which the individuals share 2 alleles (p2). Check if m = 1 − p0 + p2 holds.

```
results_part3 <- data.frame(
  Pair = character(),
  p0 = numeric(),
  p2 = numeric(),
  m = numeric(),
  m_check = logical()
)

for (i in 1:(nrow(yri6) - 1)) {
  for (j in (i + 1):nrow(yri6)) {
    snp_data_i <- unlist(yri6[i, 6:ncol(yri6)])
    snp_data_j <- unlist(yri6[j, 6:ncol(yri6)])

    # Shared alleles calculation
    shared <- 2 - abs(snp_data_i - snp_data_j)
    #shared <- mapply(share, snp_data_i, snp_data_j)


    # Calculate p0, p2, and check if m = 1 - p0 + p2
    p0 <- mean(shared == 0, na.rm = TRUE)
    p2 <- mean(shared == 2, na.rm = TRUE)
    m <- mean(shared, na.rm = TRUE)
    m_check <- abs(m - (1 - p0 + p2)) < 0.0001

    # Add the results to the data frame
    results_part3 <- rbind(results_part3, data.frame(
      Pair = paste("Ind", i, "- Ind", j),
      p0 = p0,
      p2 = p2,
      m= m,
      m_check = m_check
    ))
  }
}


str(results_part3)
```

```
## 'data.frame':    3486 obs. of  5 variables:
##  $ Pair   : chr  "Ind 1 - Ind 2" "Ind 1 - Ind 3" "Ind 1 - Ind 4" "Ind 1 - Ind 5" ...
##  $ p0     : num  0.126 0.126 0.127 0.126 0.126 ...
##  $ p2     : num  0.374 0.371 0.374 0.372 0.371 ...
##  $ m      : num  1.25 1.25 1.25 1.25 1.25 ...
##  $ m_check: logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
```

```
head(results_part3, 5)
```

```
##            Pair        p0        p2        m m_check
```

```
## 1 Ind 1 - Ind 2 0.1256209 0.3743350 1.248714    TRUE
## 2 Ind 1 - Ind 3 0.1263456 0.3714715 1.245126    TRUE
## 3 Ind 1 - Ind 4 0.1269289 0.3741405 1.247212    TRUE
## 4 Ind 1 - Ind 5 0.1256032 0.3722846 1.246681    TRUE
## 5 Ind 1 - Ind 6 0.1255678 0.3711887 1.245621    TRUE
```

4. Plot m against s and plot p0 against p2. Comment on the results.

Mean vs. Standard Deviation of Shared Alleles Plot:

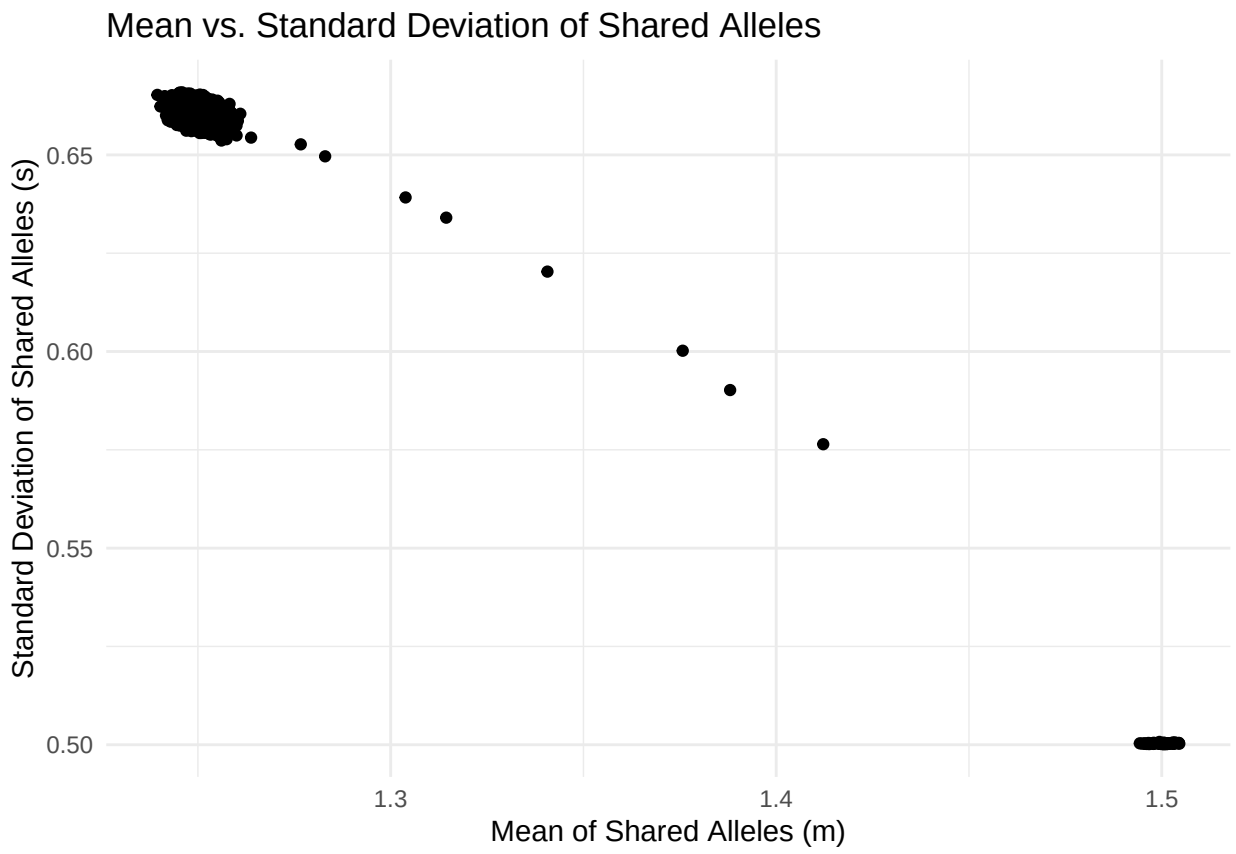
- The majority of data points are clustered in a region that indicates a relatively high mean of shared alleles (around 1.3) and a low standard deviation (around 0.55 to 0.65). This clustering suggests that for most individual pairs, there is a consistent level of allele sharing.
- There is an outlier group with a higher mean (around 1.5) and very low standard deviation (close to 0.5). This could indicate a subset of pairs with a higher degree of relatedness, such as parent-offspring pairs, where we would expect them to share more alleles on average.

Fraction of Variants Shared 0 Alleles vs. 2 Alleles Plot:

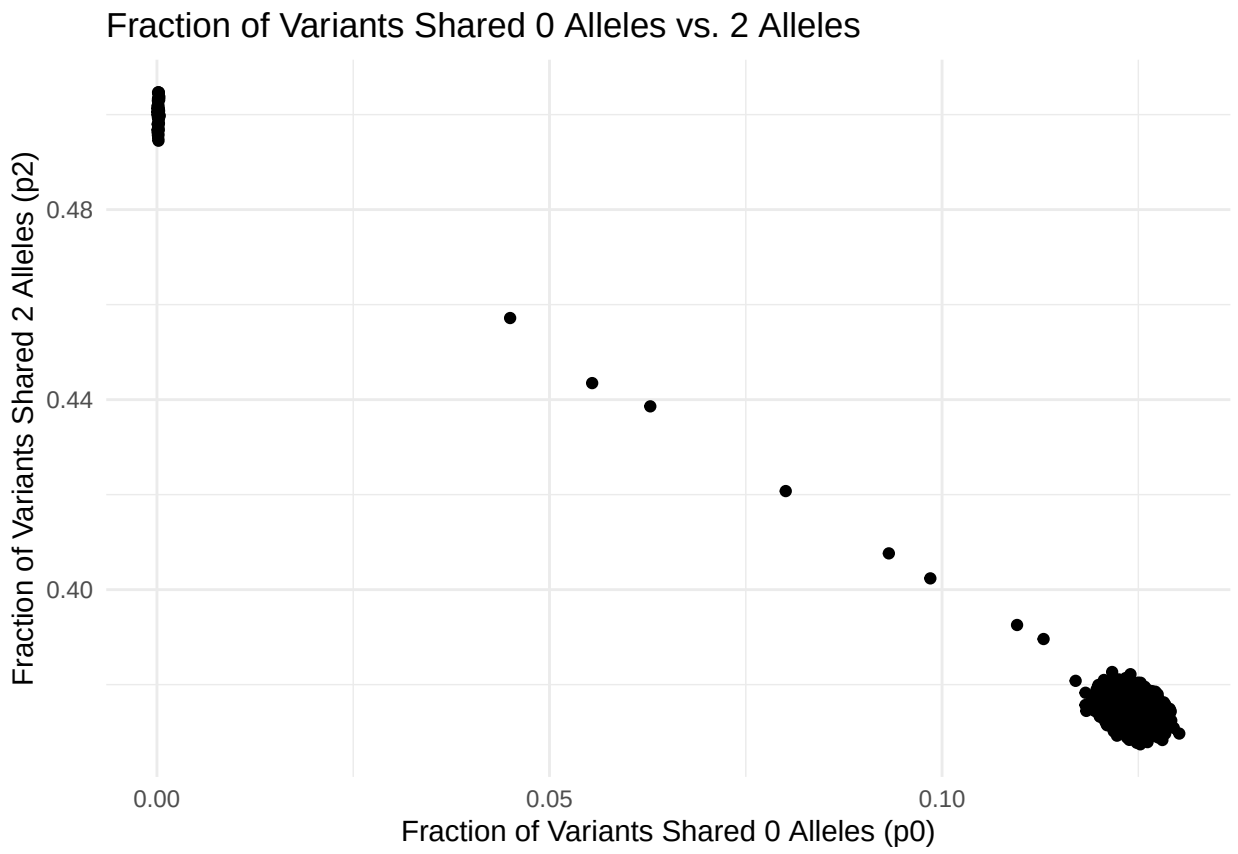
- Most points are clustered towards the lower end of the x-axis (p0) and the higher end of the y-axis (p2), suggesting that for the majority of individual pairs, there is a low fraction of variants with no shared alleles and a higher fraction with both alleles shared. This pattern might indicate some degree of relatedness among the individuals.
- There is an outlier group with a high fraction of variants shared 0 alleles (p0 around 0.1) and a low fraction of variants shared 2 alleles (p2 around 0.48). This could represent pairs of individuals that are less related or unrelated.

In summary, the data seems to show a general trend of shared genetic similarity with some distinct groups that could represent different degrees of relatedness.

```r
# Plot m against s
plot_m_vs_s <- ggplot(results_df, aes(x = m, y = s)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Mean vs. Standard Deviation of Shared Alleles",
       x = "Mean of Shared Alleles (m)",
       y = "Standard Deviation of Shared Alleles (s)")

print(plot_m_vs_s)
```




```r
results_part3$p0 <- as.numeric(as.character(results_part3$p0))
results_part3$p2 <- as.numeric(as.character(results_part3$p2))

# Plot p0 against p2
plot_p0_vs_p2 <- ggplot(results_part3, aes(x = p0, y = p2)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Fraction of Variants Shared 0 Alleles vs. 2 Alleles",
       x = "Fraction of Variants Shared 0 Alleles (p0)",
       y = "Fraction of Variants Shared 2 Alleles (p2)")

print(plot_p0_vs_p2)
```



5.Plot m against s and use the pedigree information of the YRI06.raw file to label the data points in the scatterplot. Recall that column 3 and 4 from the YRI06.raw contain information about the family relationship of the participants. Create two labels: one for individuals that have a parent-offspring relationship and another one for unrelated individuals. Comment on the results.

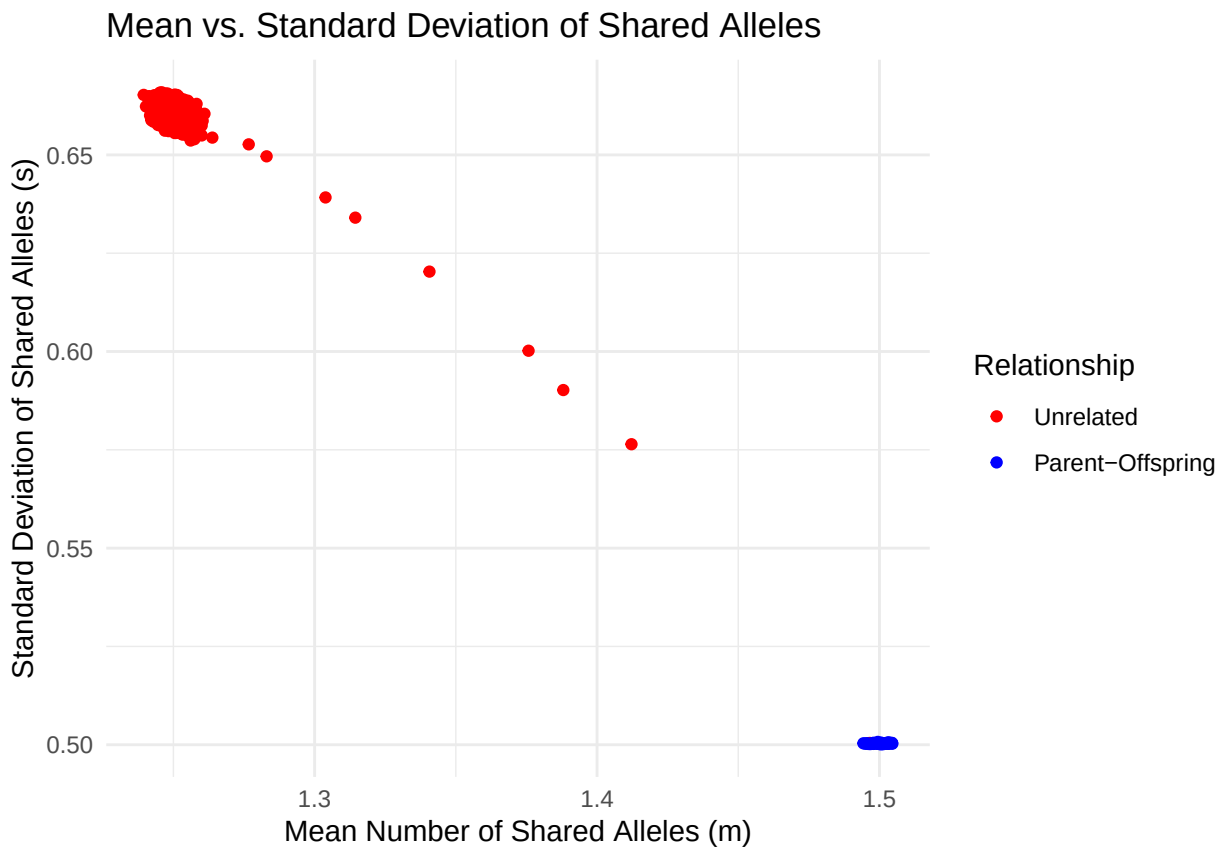
There is a distinct point (or potentially a cluster of points) on the far right, which has a very high mean number of shared alleles (around 1.5) and a very low standard deviation. They are labeled as a "Parent-Offspring" relationship, which is consistent with the expectation that a parent and offspring would share approximately 50% of their alleles, leading to a high mean of shared alleles.

The remaining points, labeled as "Unrelated," have lower means (approximately between 1.3 and 1.4) and higher standard deviations (approximately between 0.55 and 0.65). These points' positions suggest that there is more variability in the number of shared alleles among unrelated individuals, as indicated by the higher standard deviation.

Overall, the clear separation between the "Parent-Offspring" point and the "Unrelated" points indicates that the genetic data effectively differentiates between these two types of relationships.

```
is_parent_offspring <- function(ind1, ind2, data) {
  parents_ind1 <- data[IID == ind1, .(PAT, MAT)]
  parents_ind2 <- data[IID == ind2, .(PAT, MAT)]

  return((ind1 %in% c(parents_ind2$PAT, parents_ind2$MAT)) |
         (ind2 %in% c(parents_ind1$PAT, parents_ind1$MAT)))
```

```r
}

label_matrix <- matrix(NA, nrow = nrow(yri6), ncol = nrow(yri6))
rownames(label_matrix) <- yri6$IID
colnames(label_matrix) <- yri6$IID

for (i in 1:(nrow(yri6) - 1)) {
  for (j in (i + 1):nrow(yri6)) {
    # Check if the pair is parent-offspring
    if (is_parent_offspring(yri6$IID[i], yri6$IID[j], yri6)) {
      label_matrix[i, j] <- "Parent-Offspring"
      label_matrix[j, i] <- "Parent-Offspring"  # Symmetry
    } else {
      label_matrix[i, j] <- "Unrelated"
      label_matrix[j, i] <- "Unrelated"  # Symmetry
    }
  }
}

# Applying the function to label each pair as parent-offspring or unrelated
results_df$Label <- mapply(is_parent_offspring,
                           as.character(yri6$IID[combos[1,]]),
                           as.character(yri6$IID[combos[2,]]),
                           MoreArgs = list(data = yri6))

# Correcting the labels from TRUE/FALSE to Parent-Offspring/Unrelated
results_df$Label <- factor(results_df$Label,
                           labels = c("Unrelated", "Parent-Offspring"))

# Create the scatter plot
plot_m_vs_s_labeled <- ggplot(results_df, aes(x = m, y = s, color = Label)) +
  geom_point() +
  theme_minimal() +
  labs(title = "Mean vs. Standard Deviation of Shared Alleles",
       x = "Mean Number of Shared Alleles (m)",
       y = "Standard Deviation of Shared Alleles (s)",
       color = "Relationship") +
  scale_color_manual(values = c("Unrelated" = 'red',
                                "Parent-Offspring" = 'blue'))

# Print the labeled plot
print(plot_m_vs_s_labeled)
```



6. Use the package SNPRelate to estimate the IBD probabilities, and plot the probabilities of sharing 0 and 1 IBD alleles (k0 and k1) for all pairs of individuals. Use the pedigree information of the YRI06.raw file to label the data points in the scatterplot (same as before, one colour for parent-offspring relationship and another colour for unrelated individuals).

```
yri6_snps_matrix <- as.matrix(yri6_snps)
sample_ids <- yri6$IID

# Create the GDS file
gds_file <- "genotype_data.gds"
snpgdsCreateGeno(gds_file, yri6_snps_matrix, sample.id = sample_ids,
                 snp.id = colnames(yri6_snps_matrix), snpfirstdim = FALSE)

genofile <- snpgdsOpen(gds_file)

# Perform LD pruning
snpset <- snpgdsLDpruning(genofile, ld.threshold = 0.2)
```

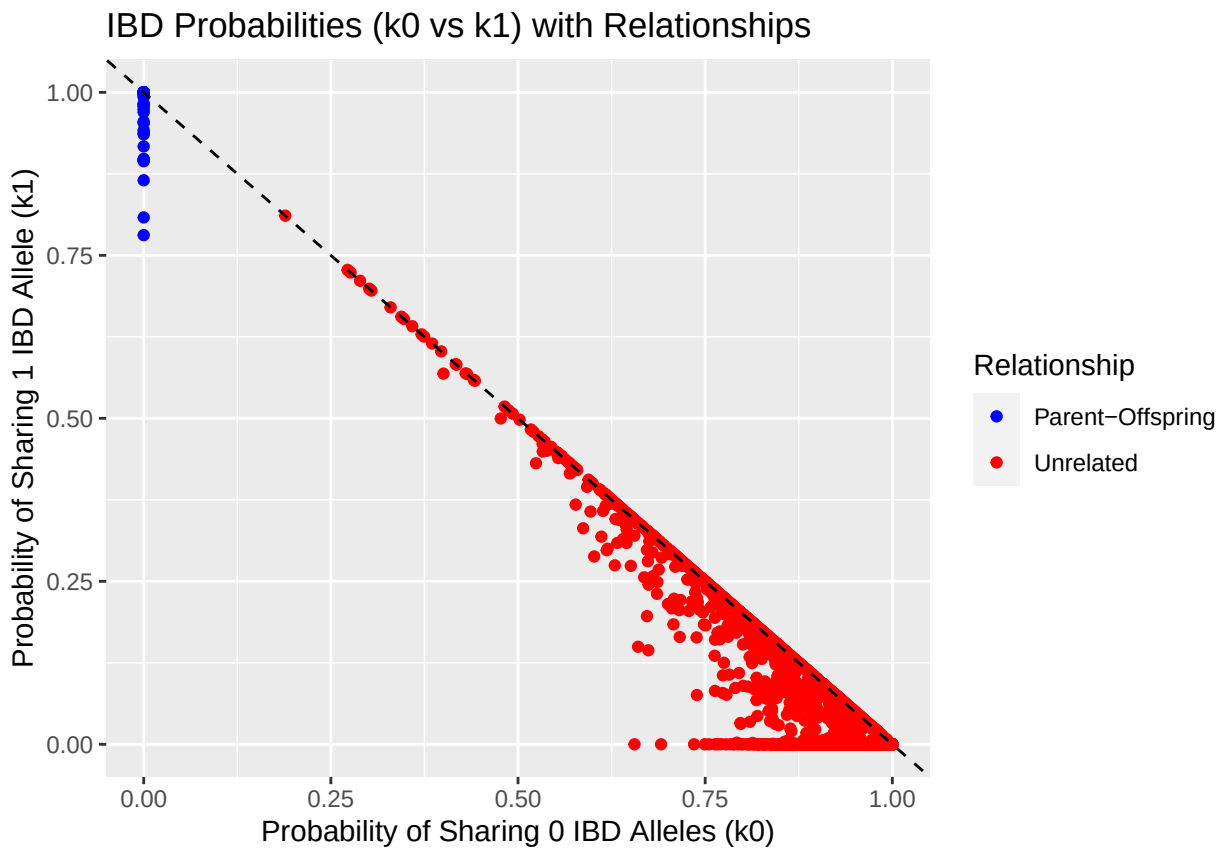
```
## SNP pruning based on LD:
## Excluding 0 SNP on non-autosomes
## Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
##     # of samples: 84
```

```
##     # of SNPs: 56,574
##     using 1 thread
##     sliding window: 500,000 basepairs, Inf SNPs
##     |LD| threshold: 0.2
##     method: composite
## Chromosome 1: 0.12%, 66/56,574
## 66 markers are selected in total.
```

```
snpset.id <- unlist(unname(snpset))

# Estimate IBD probabilities
ibd <- snpgdsIBDMLE(genofile, maf = 0.05, missing.rate = 0.05,
                    snp.id = snpset.id)
```

```
## Identity-By-Descent analysis (MLE) on genotypes:
## Excluding 56,508 SNPs (non-autosomes or non-selection)
## Excluding 0 SNP (monomorphic: TRUE, MAF: 0.05, missing rate: 0.05)
##     # of samples: 84
##     # of SNPs: 66
##     using 1 thread
## MLE IBD:    the sum of all selected genotypes (0,1,2) = 5113
## MLE IBD: Mon Dec 18 19:56:05 2023    0%
## MLE IBD: Mon Dec 18 19:56:06 2023    100%
```

```
# Select IBD estimates using the 'ibd' method
ibd_selection <- snpgdsIBDSelection(ibd)


snpgdsClose(genofile)

# Assign labels from the labels_matrix to the IBD selection data frame
ibd_selection$label <- mapply(function(id1, id2) label_matrix[id1, id2],
                              ibd_selection$ID1, ibd_selection$ID2)

# Plot using ggplot2
library(ggplot2)
ggplot(ibd_selection, aes(x = k0, y = k1, color = label)) +
  geom_point() +
  geom_abline(intercept = 1, slope = -1, linetype = "dashed", color = "black") +
  scale_color_manual(values =
                       c("Unrelated" = 'red', "Parent-Offspring" = 'blue')) +
  labs(title = "IBD Probabilities (k0 vs k1) with Relationships",
       x = "Probability of Sharing 0 IBD Alleles (k0)",
       y = "Probability of Sharing 1 IBD Allele (k1)",
       color = "Relationship")
```



7.Do you think the family relationships between all individuals were correctly specified?

From the IBD plot, we observe that:

- The blue points, representing parent-offspring pairs, are closely clustered at high values of sharing 1 IBD allele (k1) and low values of sharing 0 IBD alleles (k0). This cluster suggests a correct specification of parent-offspring relationships because such pairs are expected to share a large proportion of their genomes identically by descent.
- The red points, representing unrelated pairs, are spread across the plot with a trend that when the probability of sharing 0 IBD alleles (k0) increases, the probability of sharing 1 IBD allele (k1) decreases. This is a typical pattern for unrelated individuals in a population.

From Part 5, the distinct separation between the "Parent-Offspring" pair and the "Unrelated" pairs in the mean vs. standard deviation of shared alleles plot suggests that the genetic data is generally consistent with the expected relationships. Parent-offspring pairs show a higher mean and lower standard deviation, which is consistent with the inheritance patterns.

For Part 4, if the fraction of variants shared 0 alleles vs. 2 alleles plot showed a similar consistency and no anomalies, it would further support the correct specification of relationships.

Considering the clustering and separation observed in the IBD plot, along with consistent results in the mean vs. standard deviation and the fraction of variants shared plots, there is a strong indication that family relationships have been correctly specified for the most part. The genetic data appears to align well with the expected patterns of allele sharing for both parent-offspring and unrelated pairs.