BSG-MDS practical: Relatedness analysis

Sara Montese Anton Ickler

December 19, 2023

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

```
import numpy as np
import pandas as pd

data = pd.read_csv("yri.csv")

np_dat = data.iloc[:, 5:].values
n_ind = np_dat.shape[0]
n_snps = np_dat.shape[1]
```

n_ind Out[5]: 84
n_snps Out[6]: 56575

```
np.sum(np.sum(data.isna()))
```

Out[8]: 0

So we have 84 individuals, 56575 snps and 0% of missing data.

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.*

```
np_dat = data.iloc[:, 6:].values

mean_dist = np.zeros((n_ind, n_ind))
var_dist = np.zeros((n_ind, n_ind))

from tqdm import tqdm

for i in tqdm(range(n_ind)):
    for j in range(i+1, n_ind):
        shared = 2 - abs(np_dat[i,:] - np_dat[j,:])
        mean_dist[i, j] = shared.mean()
        mean_dist[j, i] = shared.mean()
        var_dist[i, j] = shared.std()
        var_dist[j, i] = shared.std()


#if you want the rownames, we would have to map them from data
print(mean_dist[:5, :5])
print(var_dist[:5, :5])
```

See results in Tables 1 and 2. The check between equal points was just set to 0, because it holds no information.

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.*

Table 1: Mean

0	1	2	3	4
0.000000	1.248714	1.245126	1.247212	1.246681
1.248714	0.000000	1.246293	1.250782	1.247353
1.245126	1.246293	0.000000	1.500557	1.252779
1.247212	1.250782	1.500557	0.000000	1.255042
1.246681	1.247353	1.252779	1.255042	0.000000

Table 2: Mean

0	1	2	3	4
0.000000	0.661889	0.661612	0.663292	0.661087
0.661889	0.000000	0.663580	0.658765	0.665607
0.661612	0.663580	0.000000	0.500282	0.662298
0.663292	0.658765	0.500282	0.000000	0.660735
0.661087	0.665607	0.662298	0.660735	0.000000

Table 3: Fraction of zero shared alleles

0	1	2	3	4
0.000000	0.125621	0.126346	0.126929	0.125603
0.125621	0.000000	0.127353	0.123040	0.128431
0.126346	0.127353	0.000000	0.000141	0.124878
0.126929	0.123040	0.000141	0.000000	0.123288
0.125603	0.128431	0.124878	0.123288	0.000000

Table 4: Fraction of two shared alleles

0	1	2	3	4
0.000000	0.374335	0.371471	0.374141	0.372285
0.374335	0.000000	0.373646	0.373822	0.375784
0.371471	0.373646	0.000000	0.500698	0.377658
0.374141	0.373822	0.500698	0.000000	0.378330
0.372285	0.375784	0.377658	0.378330	0.000000

```
fract_of_zeros = np.zeros((n_ind, n_ind))
fract_of_sharing_both = np.zeros((n_ind, n_ind))

for i in tqdm(range(n_ind)):
    for j in range(i+1, n_ind):
        fract_of_sharing_both[i,j] = np.sum((np_dat[i,:]
                                             - np_dat[j,:]) == 0) / n_snps
        fract_of_sharing_both[j, i] = fract_of_sharing_both[i, j]
        fract_of_zeros[i, j] = np.sum(abs((np_dat[i, :] -
                                           np_dat[j, :])) == 2) / n_snps
        fract_of_zeros[j, i] = fract_of_zeros[i, j]

print(pd.DataFrame(fract_of_zeros[:5, :5]).to_latex(
                                        index=False, caption="Fraction of zero
                                         shared alleles", label="tab:mean"))
print(pd.DataFrame(fract_of_sharing_both[:5, :5]).to_latex
                                        (index=False, caption="Fraction of two
                                         shared alleles", label="tab:mean"))
```

See results in Tables 3 and 4. The check between equal points was just set to 0, because it holds no information.

```
m = mean_dist
s = var_dist
p_0 = fract_of_zeros
p_2 = fract_of_sharing_both
check_m = m - (1 - p_0 + p_2)
np.fill_diagonal(check_m, 0)
```

4. *Plot m against s and plot p0 against p2. Comment on the results.* For the plot 1, we can see two clusters of data points that are more densely populated, one around (1.25, 0.65) and another around (1.50, 0.50). We notice that as the

mean of shared alleles increases, their standard deviation tends to decrease, and this indicates a negative correlation between the two variables. This may suggest that as the average number of shared alleles increases, the variability in the number of shared alleles decreases. Also plot 2 reveals clusters that correspond to the different family relationships. The individuals in the red cluster are sharing two alleles indicating that they have similar genetic information. The individuals in the blue cluster share 2 alleles less frequently, indicating less similar genetic information.

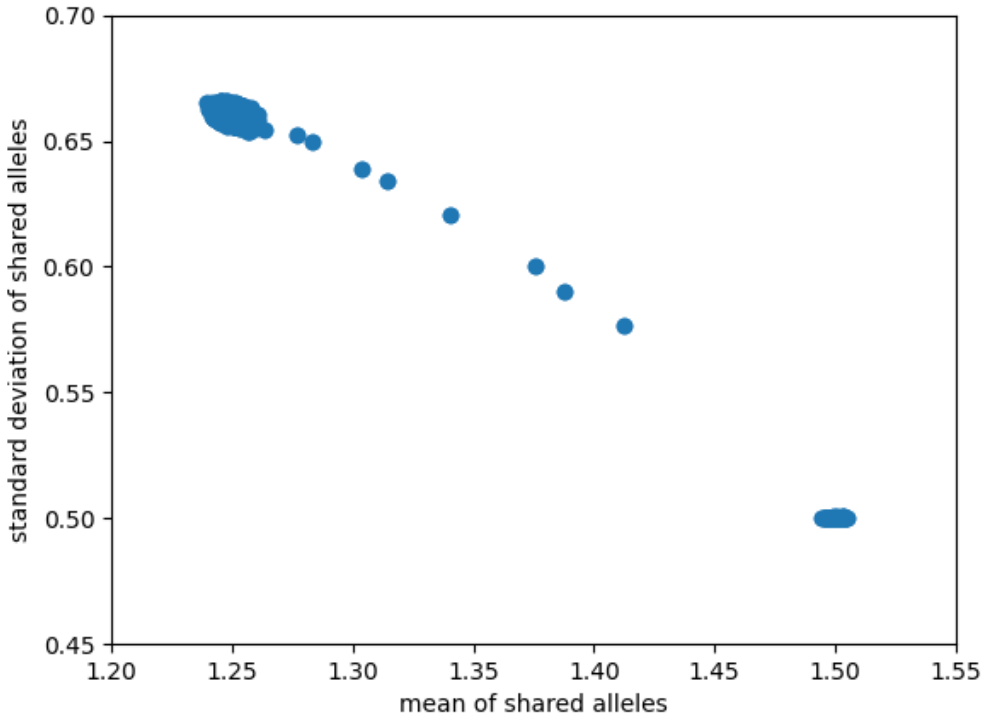

Figure 1:

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.*

From the plot 3, as expected we can see that there are two distinct clusters of data points: related individual (in red) and unrelated individuals (in blue). Parent-offspring related individuals show higher mean of shared alleles and lower standard deviation. Unrelated individuals show, on the contrary, higher standard deviation and lower mean. This confirms that unrelated individuals can share alleles, but the extent of allele sharing is generally lower compared to closely related individuals.

```
parent_offspring = np.zeros((n_ind, n_ind))

data_indexed = data.copy()
data_indexed["indexer"] = data_indexed.index
data_indexed.set_index("IID", inplace=True)

for idx, row in data.iterrows():
    if row["PAT"] != '0':
        pat = row["PAT"]
        try:
            idx_pat = data_indexed.loc[pat, "indexer"]
            parent_offspring[idx, idx_pat] = 1
            parent_offspring[idx_pat, idx] = 1
        except:
            pass
```

```
    if row["MAT"] != '0':
        pat = row["MAT"]
        try:
            idx_pat = data_indexed.loc[pat, "indexer"]
            parent_offspring[idx, idx_pat] = 1
            parent_offspring[idx_pat, idx] = 1
        except:
            pass

#%%

import matplotlib.pyplot as plt
plt.scatter(m, s, c=parent_offspring, cmap='bwr')
plt.xlabel("mean of shared alleles")
plt.ylabel("standard deviation of shared alleles")

plt.show()
plt.scatter(p_0, p_2, c=parent_offspring, cmap='bwr')
plt.xlabel("fract of 0 shared alleles")
plt.ylabel("fract of 2 shared alleles")
plt.show()
```

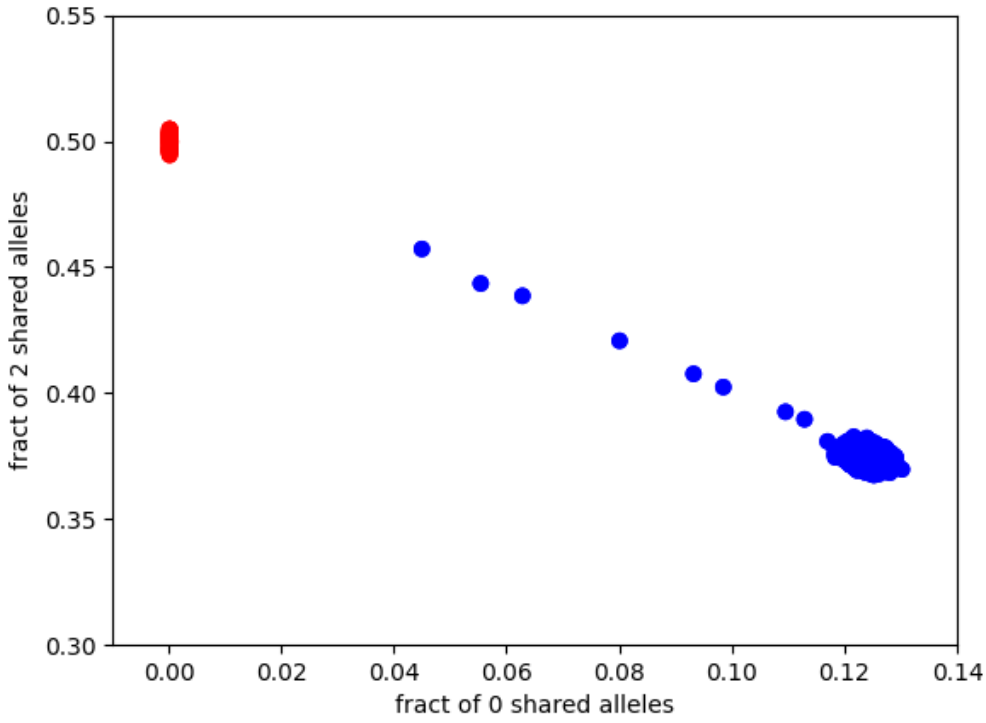

Figure 2:

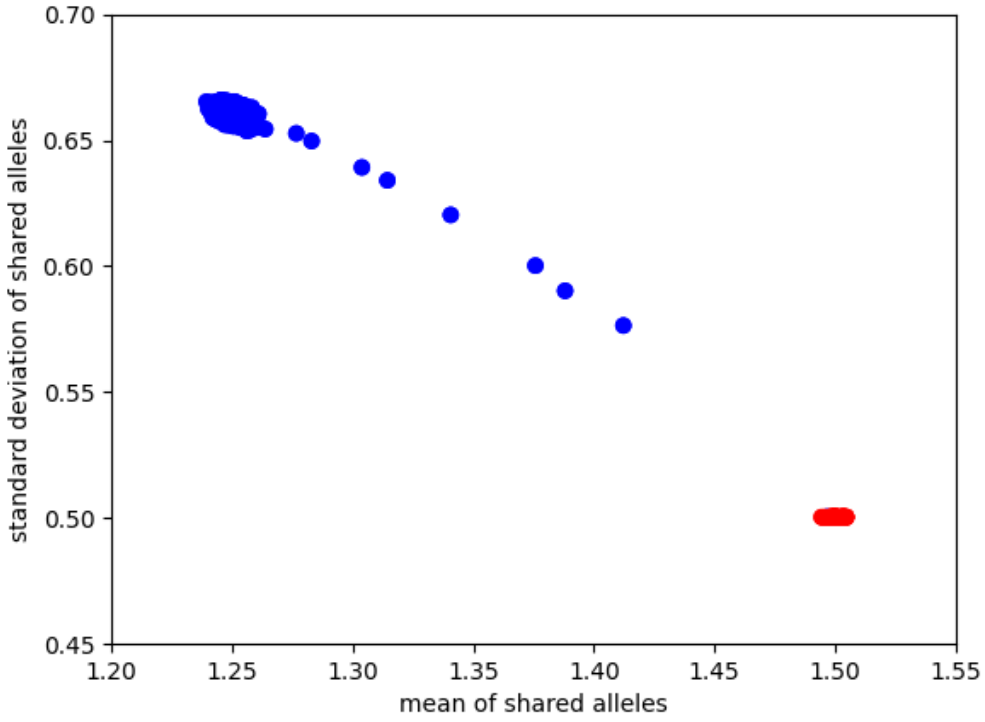

Figure 3:

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).*

```
# Create a gds file
snpgdsCreateGeno("YRI06.gds", genmat = as.matrix
                                        (raw_data_df[, 7:ncol(raw_data_df)]),
sample.id = raw_data_df$IID, snp.id = colnames(raw_data_df[, 7:ncol
                                        (raw_data_df)]), snpfirstdim=FALSE)
# Open the GDS file
genofile <- snpgdsOpen("YRI06.gds")

# Estimate IBD coefficients
ibd <- snpgdsIBDMLE(genofile,maf=0.05, missing.rate=0.05)


ibd.coeff <- snpgdsIBDSelection(ibd)

#extract IBD probabilities
k0 <- ibd.coeff$k0
k1 <- ibd.coeff$k1


is_related2<- function(individual1, individual2) {
  # Function to check if two individuals have a parent offspring relationship.
  ind1_data <- data[data$IID == individual1, c("PAT", "MAT")]
  ind2_data <- data[data$IID == individual2, c("PAT", "MAT")]
  if (ind1_data$PAT == individual2 | ind1_data$MAT == individual2
                                        | ind2_data$PAT == individual1
                                        | ind2_data$MAT == individual1) {

    return (TRUE)
 } else {
```

```
    return (FALSE)
  }
}

ibd.coeff$related <- apply(ibd.coeff[, c("ID1", "ID2")], 1, function
                                          (row) is_related2(row[1], row[2]))



# Plot IBD probabilities
point_colors <- ifelse(ibd.coeff$related, "blue", "green")

# Create a scatterplot with colors
plot(k0, k1, xlim = c(0, 1), ylim = c(0, 1), xlab = "k0",
                                          ylab = "k1", main = "IBD Probabilities
                                           - k0 vs k1 with Family Relationship
                                           Labels", col = point_colors)

# Draw a red dashed line for reference
lines(c(0, 1), c(1, 0), col = "red", lty = 2)

# Add legend
legend("topright", legend = c("Related", "Unrelated
                                          "), col = c("blue", "green"), pch = 1,
       title = "Relationship Type")
```

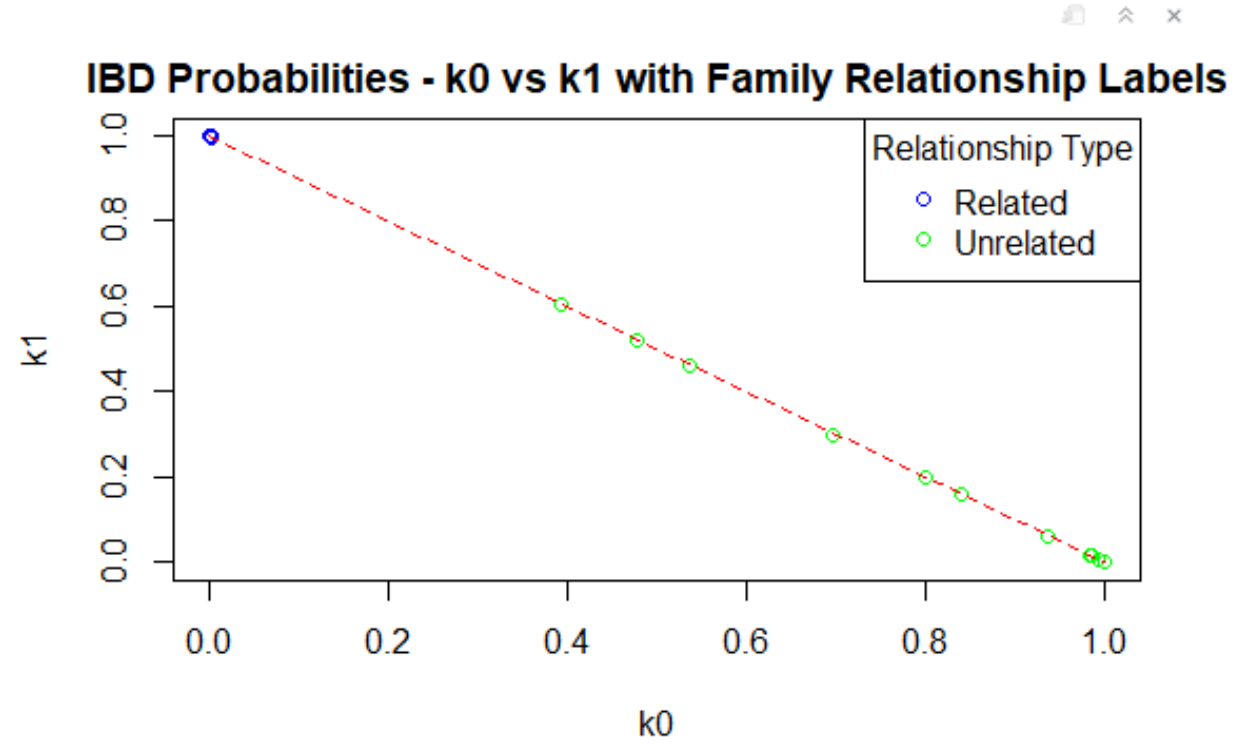

Figure 4:

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

By observing the plot, we notice that individuals related through a parent-off spring relation (blue points) are clustered at k0=0 and k1=1, correctly indicating a parent-offspring relationship. This aligns with the expectation that related individuals inherit at least one IBD allele from a common parent. On the other hand, in the case of unrelated individuals (green points) we see slightly more scatter of points, but in general all points cluster in the lower right corner which indicates that for unrelated individuals in general k0=1, meaning that in almost all cases of comparisons, there is a high probability of sharing no alleles inherited from a recent common ancestor. The green dots in the middle of the plot are interesting as they don't align with the general trend of the other green dots. These points represent unrelated individuals who have a higher than usual k1, indicating a higher probability of sharing one allele inherited from a recent common ancestor. This could be due to several reasons like errors in the data, misidentified relationships (misidentified as unrelated when they are actually related) or also could also be a sign of population substructure, where individuals that seem unrelated actually share a common ancestor.