GWAS Project

Keenan Elliott

2023-10-10

Introduction

Single nucleotide polymorphisms (SNPs) represent differences between individuals in a single position or nucleotide in the genome. These SNPs are commonly studied through Genome Wide Association Studies (GWAS), which report SNPs that statistically correlate with a specific phenotype or disease state [1]. However, given the high dimensionality associated with SNP data, it can be difficult to determine true positives from false positives or control for the type I error. Here, we utilized the 1000 genomes project data to visualize the false positive rate among uncorrelated case-control data. The 1000 genomes project is a database of the genotypes of over 2300 individuals from 5 continents and 19 populations. The data are publicly available and were produced on the Illumina Omni2.5 platform. Here, we utilized a cleaned subset of this dataset that had undergone quality control (QC) processing by Roslin et al. The QC dataset was obtained from https://tcag.ca/tools/1000genomes.html, and contained 1,989,184 SNPs with 1756 individuals. It should be noted that the report from Roslin et al. states that the final dataset should contain 1736 quasi-unrelated individuals with consistent ethnicity and well-inferred sex. Some additional data processing steps were required to remedy this discrepancy; the details will be outlined below. In this report, I outline the steps taken to perform a GWAS of the data discussed above. To do this, individuals were randomized to either case or control and logistic regression was performed using PLINK on the subsequent phenotypic groups. In addition, an attempt was made to replicate the PCA performed in the study by Roslin et al.

Methods

Data Acquisition

Data was acquired from The Centre for Applied Genomics at Sick Kids at https://tcag.ca/tools/1000genome s.html. The data downloaded had passed quality control and consisted of 1756 individuals and 1,989,184 SNPs produced on the Illumina Omni2.5 platform.

Software

The analysis was performed using Plink (V 1.90) for macOS and was accessed through the command line. The steps used were modified from the document obtained from https://github.com/sugolov/GWAS-Workshop/blob/master/GWAS_Manual_PLINK_1.9.pdf. Data cleaning and visualization were performed using R (V. 4.3.1) with R Studio.

Data Cleaning and Visualizations

Data was cleaned in R prior to analysis using Plink to exclude individuals lacking sex data or having sex data opposite to those reported in the dataset and individuals not matching their reported ancestry. Phenotype data was randomly generated using the random binary data generator rbinom and assigned to individuals. Visualizations were created in R, using the ggplot2 and qqman libraries.

Results

Prepping and Cleaning the Dataset for Plink

After loading the dataset into R Studio, I noted more individuals were present in the .fam file than was specified in the report. The downloaded .fam file consisted of 1756 individuals instead of the 1736 reported by Roslin et al. To remedy this issue, I referred to the report and identified four individuals that did not cluster well with their continental groups (HG01241, HG01242, HG01108, NA20314) as well as one individual in the dataset with a genotype consistent with the opposite sex they had been reported to (NA21310). Moreover, there were an additional 15 individuals with no reported sex that were removed from the dataset. With these 20 individuals removed, the total number of individuals in the dataset were 1736, which matched the reported number by Roslin et al. The code to read the data and perform the removal steps is included below. The final 1736 individuals were copied to a new data table and written to a new file titled phenotypes.txt. This file contained the randomly generated assignment to either case or control for each of the 1736 individuals in the final dataset.

```
bim <- read.table("indep.bim",
header = FALSE,
sep = "",
fill = TRUE,
quote = "",
check.names = FALSE)

fam= read.table("indep.fam",
header = FALSE,
sep = "",
fill = TRUE,
quote = "",
check.names = FALSE)
```

```
set.seed(7)

individuals=unique(fam$V2)
numInd=length(individuals)
print(numInd)
```

```
## [1] 1756
```

```
CaseControlVector=(rbinom(n=numInd, size=1, prob=0.5)+1)

fam$V6=CaseControlVector
```

```
library(data.table)
phenotable=data.table(v1=fam$V1, v2=fam$V2, v6=fam$V6)

SexUnknown=which(fam$V5==0)

phenotable=phenotable[-c(SexUnknown),]


##drop IDs with non maching
loc1=which(phenotable$v2=='HG01241')
phenotable=phenotable[-loc1,]

loc2=which(phenotable$v2=='HG01242')
phenotable=phenotable[-loc2,]
```

```
loc3=which(phenotable$v2=='HG01108')
phenotable=phenotable[-loc3,]

loc4=which(phenotable$v2=='NA20314')
phenotable=phenotable[-loc4,]


##DROP individuals with sex incorrectly labelled
loc5=which(phenotable$v2=='NA21310')
phenotable=phenotable[-loc5,]


write.table(phenotable, "phenotypes.txt", col.names = F, row.names = F, quote = F)
```

Running a GWAS: Plink Commands

Plink was used to complete the GWAS. First, the phenotype.txt file was used to filter the dataset and produce .bed files for the 1736 individuals to be included in the analysis. The following code was used:

plink –bfile indep –pheno phenotypes.txt –prune –make-bed –out MSG_GWAS

Next, we filtered the data prior to analysis. SNPs with minor alleles having a frequency of less than 5% of the total allele pool were excluded. Moreover, we limited our analysis to the autosomes (chromosomes 1-22).

plink –bfile MSG_GWAS –chr 1-22 –maf 0.05 –make-bed –out MSG_GWAS_clean

Finally, the association test was completed using logistic regression. Logistic regression was chosen as the phenotype data are discrete.

plink –bfile MSG_GWAS_clean –logistic sex hide-covar –out MSG_GWAS_sex

The .assoc.logistic files were produced and uploaded into R Studio for visualization.

Visualizing the GWAS: Histogram, QQ plot, and Manhattan Plot

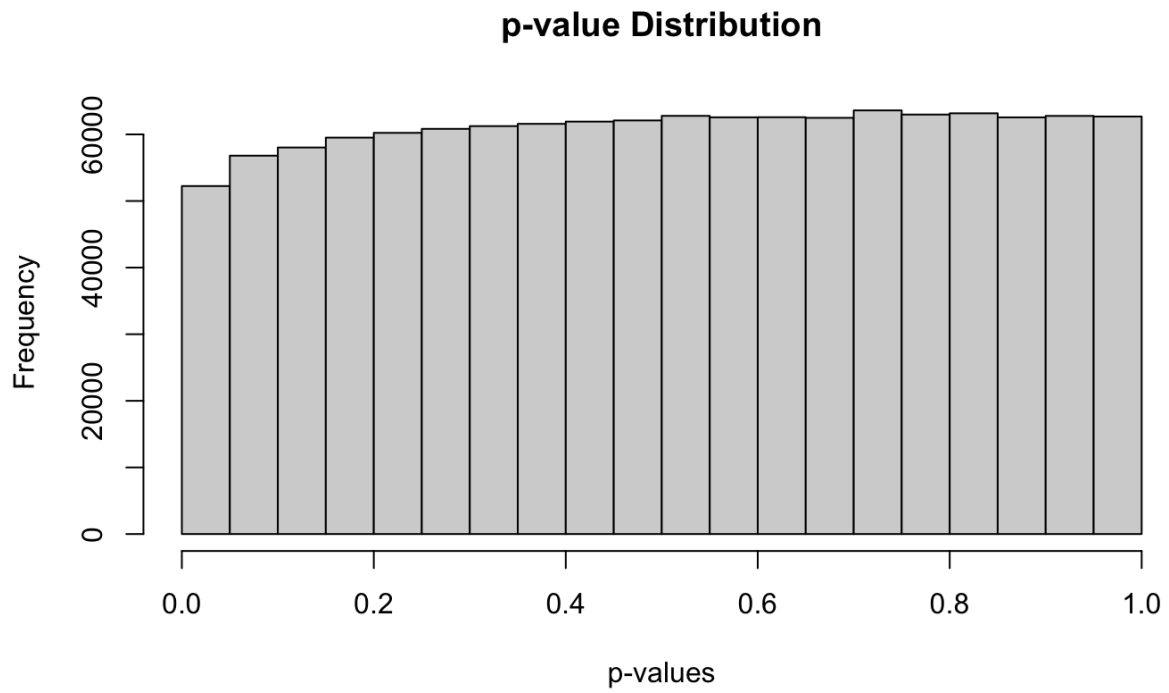
Using the assoc.logistic files produced by plink, and code from the NA_removal.R file, the results of the logistic association test were filtered to remove null values and sort the data for the top 50 SNPs identified by the association test. The code for these operations is below:

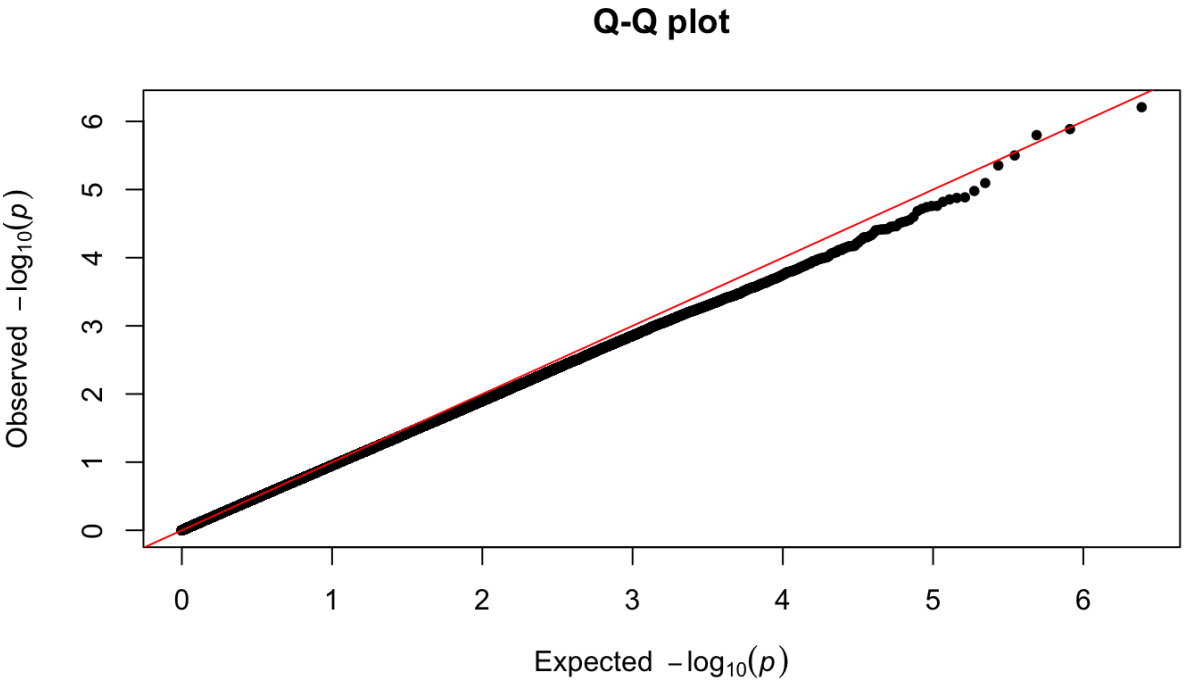
```
results <- read.table("MSG_GWAS_sex.assoc.logistic",
                      header = TRUE,
                      sep = "",
                      fill = TRUE,
                      quote = "",
                      check.names = FALSE)


NA_removed <- as.data.frame( na.omit(results))


write.table(NA_removed,
            "GWAS_results_na_removed.assoc.logistic",
            append=FALSE,
            sep='\t',
            row.names=FALSE,
            col.names=TRUE,
```

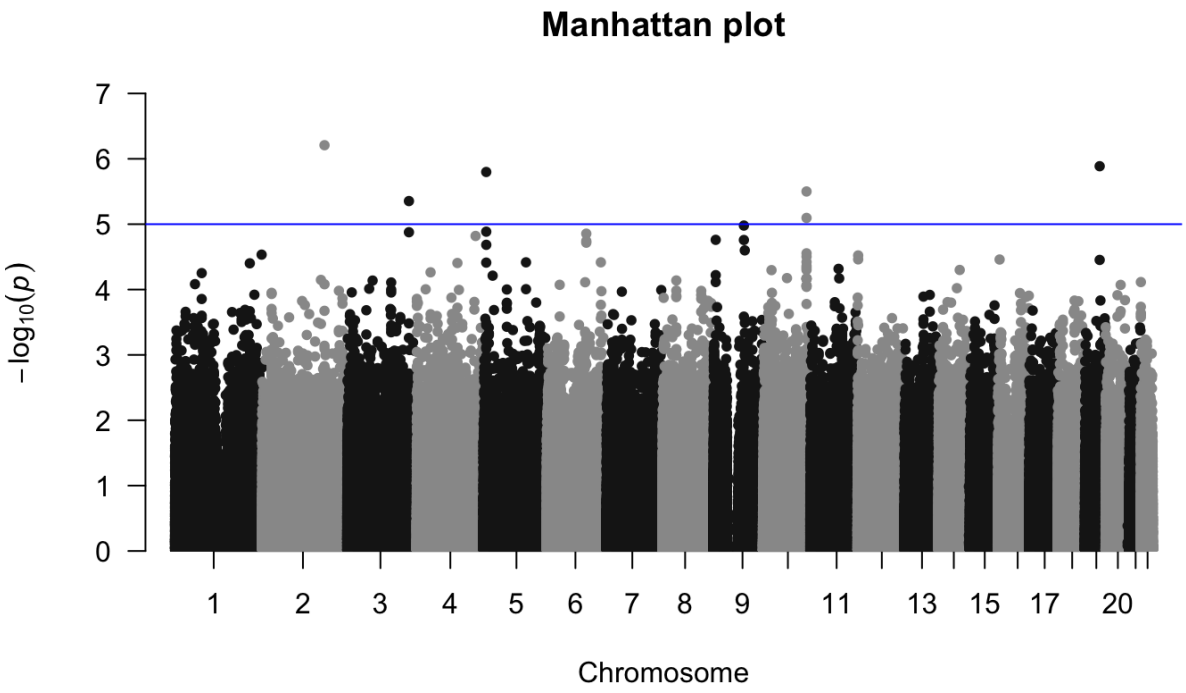
```
            quote=FALSE)


sorted <- NA_removed[order(NA_removed$"P"),]

#assigns the first 50 rows of the sorted data frame to the table top50
#             - this is done by subseting the table with the indices 1,...,50
top50 <- sorted[1:50,]

write.table(NA_removed,
            "GWAS_results_top50_SNPs.assoc.logistic",
            append=FALSE,
            sep='\t',
            row.names=FALSE,
            col.names=TRUE,
            quote=FALSE)
```

Next, the data with removed NA values was used to produce the histogram, QQ plot, and Manhattan plot. The code and resulting plots can be seen below:

```
data <- read.table("GWAS_results_na_removed.assoc.logistic", header=TRUE, sep = "",
fill = TRUE, quote = "")
hist(as.numeric(data$P),
main="p-value Distribution",
xlab="p-values",
ylab="Frequency",
breaks = seq(0,1,0.05),
freq = TRUE, cex = 3)

library("qqman")
qq(data$P,
main = "Q-Q plot"
)

library("qqman")
manhattan(data,
chr="CHR",
bp="BP",
p="P",
snp="SNP",
main = "Manhattan plot"
)
```


p-value Distribution
Frequency
0
20000
40000
60000
0.0
0.2
0.4
0.6
0.8
1.0
p-values


Q-Q plot
Observed $-\log_{10}(p)$
0
1
2
3
4
5
6
Expected $-\log_{10}(p)$



Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique often used to represent high-dimensional data in a reduced number of dimensions or vectors. This technique can be helpful in visualizing similarities in the genotypes present in your dataset, as was demonstrated by Roslin et al. In their report, they identified the four individuals that did not cluster well with their continental groups. We attempted to recreate the PCA performed by Roslin et al. using the Plink built-in functionality. To do this, the following command was applied to the 1756 individuals from the indep dataset:

```
plink --bfile indep --chr 1-22 --pca --out plinkPCA
```

The output file was a collection of eigenvectors and eigenvalues that can be used to visualize the PCA results. The results in the report by Roslin et al. grouped the samples by continental groups, and population codes provided in Appendix 2 of the report were no longer available. I found population codes reported on the 1000 genome project webpage for the dataset. However, not all individuals in the eigenvector PCA results were present in the Population code database and thus, after filtering, the table was reduced to 1662 individuals with associated population codes. Nonetheless, I was able to visualize the PCA plot for these samples, coded for their continental location. The results of this PCA are visualized below, with the code to produce the visualizations. The PCA plot is consistent with the plot produced by Roslin et al. despite using different algorithms to produce the plot (plink PCA vs SmartPCA) and having a reduced number of samples visualized (1662 vs 1736).

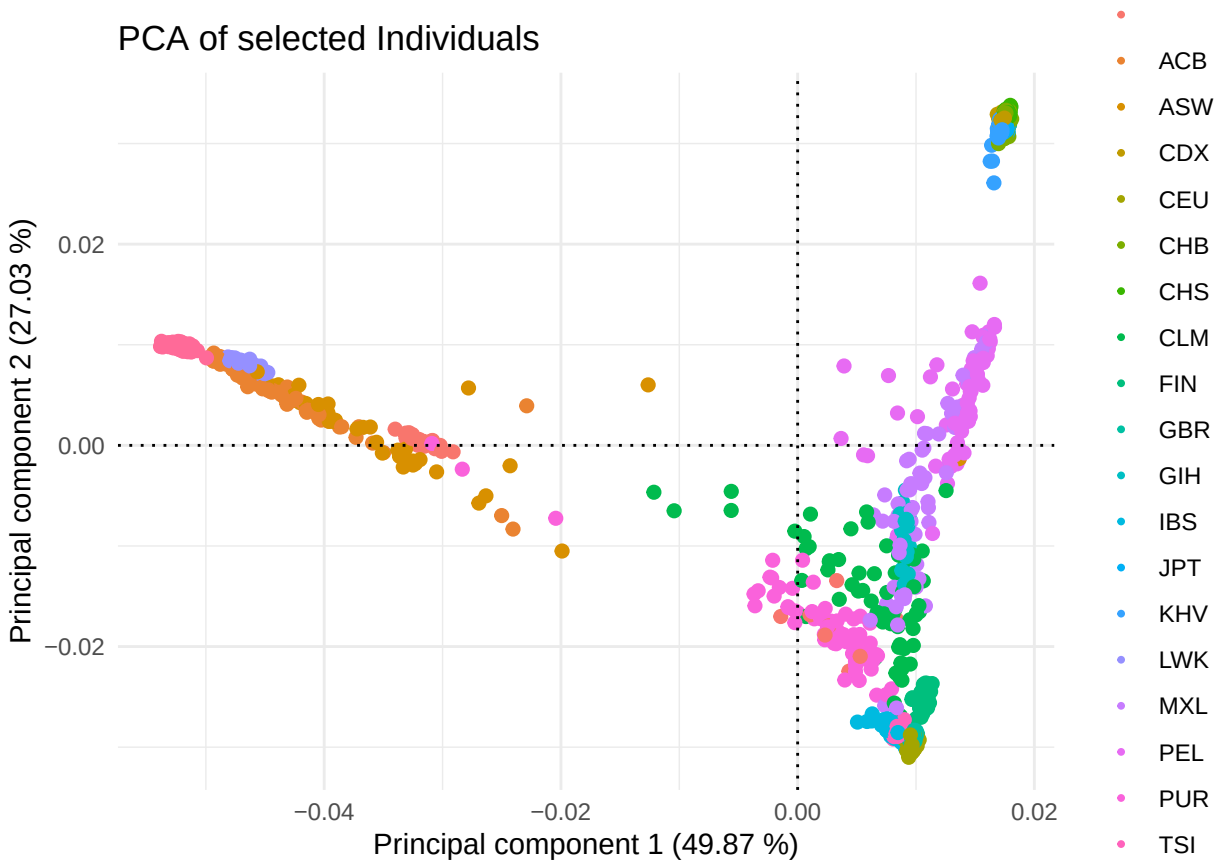
```
eigenValues= read.delim("plinkPCA.eigenval", sep =" ", header = FALSE)
eigenVectors= read.delim("plinkPCA.eigenvec", sep= " ", header=F)

eigen_percent =round((eigenValues / (sum(eigenValues))*100), 2)

populationresults =read.table("omni25.2141.sample.panel",
                      header = FALSE,
                      sep = "",
```

```
                     fill = TRUE,
                     quote = "",
                     check.names = FALSE)

library(data.table)

# set as data.table
lapply(list(eigenVectors, populationresults), \(i) setDT(i))
```

```
## [[1]]
##         V1      V2          V3          V4           V5          V6          V7
##    1: 1328 NA06984  0.00976161 -0.02870420 -0.010530500 -0.01177500 -0.00219466
##    2: 1328 NA06989  0.00985391 -0.02863220 -0.012178200 -0.01020190 -0.00197341
##    3: 1330 NA12340  0.00947436 -0.02974860 -0.011326000 -0.01070790 -0.00271370
##    4: 1330 NA12341  0.00905447 -0.02900420 -0.010802600 -0.00930393 -0.00341569
##    5: 1330 NA12342  0.00958784 -0.02951870 -0.011040100 -0.01152040 -0.00293812
##   ---
## 1752: YRI2 NA18861 -0.05278190  0.01026390  0.000650399 -0.00386392 -0.04522080
## 1753: YRI2 NA19105 -0.05135250  0.00933728  0.000399485 -0.00513286 -0.04652870
## 1754: YRI3 NA19152 -0.05175900  0.01008550 -0.000279202 -0.00372092 -0.04300550
## 1755: YRI4 NA19185 -0.05166870  0.00944444  0.000435657 -0.00505504 -0.04087540
## 1756: YRI5 NA19214 -0.05150500  0.00990394  0.001446750 -0.00448327 -0.03795240
##                V8          V9          V10          V11          V12
##    1: -0.00260094  0.00726213 -0.002703880 -0.000846399  0.000224808
##    2: -0.00399835  0.01058310 -0.003905990 -0.000877397  0.002048030
##    3: -0.00213027  0.00768288 -0.001872090  0.000067050 -0.001589040
##    4: -0.00454672  0.00999080 -0.000431531  0.001036500 -0.000945263
##    5: -0.00254387  0.00862743 -0.002606170  0.000210186  0.000569692
##   ---
## 1752: -0.00348606 -0.00662782  0.006208160  0.037783800  0.094933700
## 1753:  0.00682390 -0.00991794  0.025625700 -0.029975100 -0.002836960
## 1754:  0.00249722 -0.00854821  0.022426500  0.024534600  0.027346100
## 1755:  0.00130929 -0.00503207  0.017040500 -0.011339900 -0.001941290
## 1756:  0.00577916 -0.00225196  0.012545700 -0.010940800 -0.010343000
##               V13          V14          V15          V16          V17
##    1:  0.000782929  0.001475550  1.31469e-03 -0.000230645  0.002181980
##    2:  0.000436091 -0.000521910 -2.01087e-03 -0.000200809 -0.000347159
##    3:  0.000547930 -0.000349555  7.26164e-04 -0.000284128  0.000959965
##    4: -0.000274206  0.000342096 -6.99245e-04  0.001816800  0.000166060
##    5:  0.002778780 -0.000202866  1.84780e-03 -0.000461322 -0.002134400
##   ---
## 1752:  0.116023000  0.052396900 -6.13299e-02 -0.069096300  0.184079000
## 1753:  0.016743000  0.007044400  2.05294e-05  0.028126600  0.017302400
## 1754: -0.048956200  0.040551000 -5.23323e-02  0.002128230  0.034643300
## 1755: -0.027996500 -0.001557850  1.30928e-02 -0.010229400  0.025422400
## 1756:  0.017162100  0.004228070  2.33113e-02  0.010395900  0.000961696
##               V18          V19          V20          V21          V22
##    1: -0.000152698 -1.58820e-05  2.37001e-03  0.000406577  0.001263940
##    2: -0.001904950 -2.99152e-05 -9.53875e-04  0.001120560  0.002988190
##    3: -0.001414730 -6.46747e-05  7.67512e-04 -0.000856716 -0.001070260
##    4:  0.000503090  1.86172e-03  7.19223e-04 -0.000311034 -0.000119828
##    5:  0.000778264  4.35695e-04  7.24317e-05  0.002106100 -0.001837470
##   ---
## 1752:  0.258885000 -2.00284e-01 -4.70386e-01  0.504968000  0.222515000
```

```
## 1753:  0.005657180  9.87326e-03 -2.97493e-02 -0.006057640 -0.016524400
## 1754:  0.047750700  6.23509e-02 -1.75914e-02 -0.046207100  0.029409700
## 1755: -0.006575180 -2.01170e-02 -1.34055e-02  0.014037800  0.000565859
## 1756:  0.003940610 -6.29318e-04  1.02183e-02  0.033402000 -0.001658630
##
## [[2]]
##            V1  V2
##    1: HG02291 PEL
##    2: NA20289 ASW
##    3: HG00694 CHS
##    4: HG00635 CHS
##    5: HG00135 GBR
##   ---
## 2137: HG01084 PUR
## 2138: HG02139 KHV
## 2139: NA19681 MXL
## 2140: NA21110 GIH
## 2141: HG00706 CHS
```

```
names(eigenVectors)[names(eigenVectors)=="V2"]="ID"
names(populationresults)[names(populationresults)=="V1"]="ID"


# inner join
vectors=eigenVectors[populationresults, on=.(ID), nomatch=0]


library(ggplot2)


#plot_ly(x=vectors$V3, y=vectors$V4, z=vectors$V5, color = vectors$V2, type="scatter3d", mode="markers"


ggplot(data = vectors) +
  geom_point(mapping = aes(x = V3, y = V4, color=V2), size = 2, show.legend = TRUE) +
  geom_hline(yintercept = 0, linetype="dotted") +
  geom_vline(xintercept = 0, linetype="dotted") +
  labs(title = "PCA of selected Individuals",
      x = paste0("Principal component 1 (",eigen_percent[1,1]," %)"),
      y = paste0("Principal component 2 (",eigen_percent[2,1]," %)"),
      colour = "", shape = "") +
  guides(color = guide_legend(override.aes = list(size = 0.7)))+
  theme_minimal()
```



Discussion

We performed a GWAS analysis on the samples identified by Roslin et al. to be of high quality and consisting only of unrelated individuals. The sample dataset provided consisted of 1756 individuals and 1 989 184 SNPs. This dataset was filtered to remove individuals with missing/incorrect sex data and a genotypic profile inconsistent with their continental location, leaving a total of 1736 individuals for association analysis.

Logistic regression was performed to determine SNPs associated with either cases or controls. However, case and control data were randomly assigned to highlight the limitation of statistical inference, with nearly 2 million parameters being tested on 1736 samples. Given this experimental setup, we expect to see an approximately equal distribution of p-values across all values from 0 to 1 in a histogram of p-values. Indeed, observing the histogram, we see almost an equal frequency of all possible p-values, with lower p-values occurring with slightly less regularity. The slight deviation away from higher p-values may be explained by random chance. A random case-control vector was created during the analysis while setting a seed for replicability. When repeating the analysis with other classmates, it was noted that depending on your choice of seed, you may produce a histogram of p-values with slight deviations from an even distribution of all p-values. The Q-Q plot similarly shows that most of the data follows the expected p-value distribution we would expect, with some deviation in points with smaller p-values.

The Manhattan plot displays that most p-values do not cross the threshold to significance, and there is no strong association between case and control with any specific region of the genome. However, there are still points that achieve statistical significance with our test; this is intriguing given that the case and control individuals are randomly assigned, and there should be no association between specific SNPs and the cases when compared to the controls. This aptly displays the multiple-testing problem, whereby if numerous statistical tests are performed, there is a high likelihood of identifying significant associations by chance alone.

Overall, I was able to perform a GWAS of data that had been previously subject to quality control processing.

Additionally, I was able to recapitulate a PCA that agreed with the one reported by Roslin et al. Throughout this project, I was able to familiarize myself with the Plink documentation, became more familiar with file types used in GWAS analysis, and became more comfortable with command-line analysis tools, all indispensable skills for a future in genetic research.

References

1. Uffelmann E, Huang QQ, Munung NS, de Vries J, Okada Y, Martin AR, et al. Genome-wide association studies. Nature Reviews Methods Primers. 2021;1(1). doi:10.1038/s43586-021-00056-9
2. Roslin NM, Weili L, Paterson AD, Strug LJ. Quality Control Analysis of the 1000 genomes project OMNI25 genotypes. 2016; doi:10.1101/078600
3. Chang CC, Chow CC, Tellier LC, Vattikuti S, Purcell SM, Lee JJ. Second-generation PLINK: Rising to the challenge of larger and richer datasets. GigaScience. 2015;4(1). doi:10.1186/s13742-015-0047-8
4. Sugolov A, Emmenegger E, Paterson AD, Sun L. Statistical learning of large-scale genetic data: How to run a genome-wide association study of gene-expression data using the 1000 Genomes Project Data. Statistics in Biosciences. 2023; doi:10.1007/s12561-023-09375-9
5. Quality Control for 1000 genomes [Internet]. [cited 2023 Oct 10]. Available from: https://tcag.ca/tools/1000genomes.html