STAT115 Homework 5: Population Structure, GWAS

Shawn Pan

2017-04-17

"This is a very interesting homework".

Part I– Population Structure

Principal component analysis (PCA) is a widely used tool in genomics to infer the population structure (such as race/ancestry) from genome-wide data such as single nucleotide polymorphisms (SNPs).

1. Conduct principal component analysis (PCA) for genomic variant data of 1252 people from the 1000 genomes project[1].

Please perform PCA on 1252 individuals from 1000 genomes project containing 234,148 SNPs on chromosome 8 and plot all individuals by the first two principal components (color code each individual according to their race/ancestry). The SNP data and R code can be found in /n/stat115/hws/5/population/

Hint: R code - PopulationStructure.R; File - 1000genomes.SNP.parse.result

```
#!/bin/bash
#SBATCH -p seas_iacs
#SBATCH -J h5p1
#SBATCH -n 2
#SBATCH -N 1
#SBATCH -t 0-4:00
#SBATCH --mem 16000
#SBATCH -o h5p1%j_%N.out
#SBATCH -e h5p1%j_%N.err

#Echo Commands
set -x

#Load Modules
source new-modules.sh
module load R/3.3.3-fasrc01

#Run script
R CMD BATCH prob1.R
```

```
# Question 1
library(Matrix)

file.snp = "/n/stat115/hws/5/population/1000genomes.SNP.parse.result"
file.pop = "/n/stat115/hws/5/population/Popinfo.1000.Genomes.txt"
file.lung.snp = "/n/stat115/hws/5/population/lungcancer.SNP.parse.result"
file.lung.pop = "/n/stat115/hws/5/population/Popinfo.lung.cancer.txt"
file.lung.pheno = "/n/stat115/hws/5/population/lungcancer.pheno"
```

```r
#Read Chr8 SNP info
x = read.table(file.snp, colClasses=c("integer","integer"), fill=TRUE, row.names=NULL)

# Convert to a sparse matrix of people (rows) x variant (columns)
chr8 = sparseMatrix(i=x[,2], j=x[,1], x=1.0)

# Inspect the dimensions of this matrix - people (rows) x variant (columns)
print(dim(chr8))

# install.packages("irlba")
library(irlba)
cm = colMeans(chr8)
p = irlba(chr8, nv=3, nu=3, tol=0.1, du=rep(1,nrow(chr8)), ds=1, dv=cm)

# Read race information for 1000 Genomes
popinfo = read.table(file.pop, sep="\t",header=TRUE,colClasses=c("character","factor"))

# Plot with colors corresponding to super populations
N = length(levels(popinfo$Population))


#Plot all individuals by first two principal components (color each individual according to their race/
png(filename="pc2race.png")
plot(p$u[,1],p$u[,2],col=rainbow(N)[popinfo$Population],xlab="Component 1", ylab="Component 2")
legend("topright",levels(popinfo$Population),col=rainbow(N),pch = 1)
dev.off()

# Question 2
# Read lung cancer samples
y = read.table(file.lung.snp, colClasses=c("integer","integer"), fill=TRUE, row.names=NULL)

# Convert to a sparse matrix of people (rows) x variant (columns)
chr8_ = sparseMatrix(i=y[,2], j=y[,1], x=1.0)

# Inspect the dimensions of this matrix - people (rows) x variant (columns)
# In order to make you can finish the HW in time. We provide the GWAS file have the same dimensions as
print(dim(chr8_))

# Read race information for sample in 1000 Genomes
popinfo_ = read.table(file.lung.pop,sep="\t",header=TRUE,colClasses=c("character","factor"))

# Plot with colors corresponding to super populations
N = length(levels(popinfo_$Population))

#Use the eigenvector from PCA in question 1 as project directions lung-cancer files
#################You code here#############################
pheno = read.table(file.lung.pheno,sep="\t",header=FALSE,colClasses=c("character","character","factor"))

alpha1 = chr8_%*%p$v[,1]
alpha2 = chr8_%*%p$v[,2]

#Plot all individuals by first two principal components (color each individual according to their race/
png(filename="pc2lungrace.png")
```

```
plot(alpha1,alpha2,col=rainbow(N)[popinfo_$Population],xlab="Component 1", ylab="Component 2")
legend("topright",levels(popinfo$Population),col=rainbow(N),pch = 1)
dev.off()

#Plot all individuals by first two principal components (color each individual according to their pheno
png(filename="pc2lungpheno.png")
plot(alpha1,alpha2,col=rainbow(N)[pheno[,3]],xlab="Component 1", ylab="Component 2")
legend("topright",c("control", "case"),col=rainbow(N),pch = 1)
dev.off()

#Write PC1 and PC2 to file, for part
pc = pheno[,1:2]
pc[,3] = alpha1
pc[,4] = alpha2
print(dim(pc))
write.table(pc, file="PC.txt", sep="\t", row.names=FALSE, col.names=FALSE, quote=FALSE)
```

2. Use the eigenvectors derived from question 1 as project directions for the following lung-cancer GWAS data.

Data for this analysis is from a published lung-cancer GWAS study with 1252 individuals, including 567 cases (smokers) and 685 controls (non-smokers). The SNP data is "lungcancer.SNP.parse.result", containing 234148 SNPs in chromosome 8 for all individuals.

Hint: File - lungcancer.SNP.parse.result

```
rclogin01 /n/stat115/hws/5/population 2030 $ cut -f3 lungcancer.pheno | grep 1 | wc -l
685
rclogin01 /n/stat115/hws/5/population 2031 $ cut -f3 lungcancer.pheno | grep 2 | wc -l
567
```

By the counts, we can tell that that phenotype 1 corresponds to control and 2 corresponds to case. The code for plotting is combined with the code for question 1 above.

3. (Graduate students) Compare the PCA plot in question 2 with question 1, and answer the following questions:

a. Do individuals with similar population structure (such as race/ancestry) cluster together?

Yes, individuals with the same ancestry tend to cluster together well, although AMR is more spread out than other backgrounds.

b. Do the case (disease) and control samples cluster together?

The case and control samples are partially clustered together on the PCA plot, but all samples of one type are not in a single cluster. Samples on the left contains almost all controls, while the samples on the right are a mixture of case and control. Comparing with the race PCA plot, it appears that AFR and SAS ancestry samples have almost all controls, while EUR samples have almost all cases. Thus, the proportions of case and control are different for various population strata.

c. How would you deal with it if you found the presence of confounding covariate in the data?

One way of correct for population stratification is to use EIGENSTRAT. We perform linear regression against the confounding principle component eigenvectors to subtract out the variation caused by race, and then we

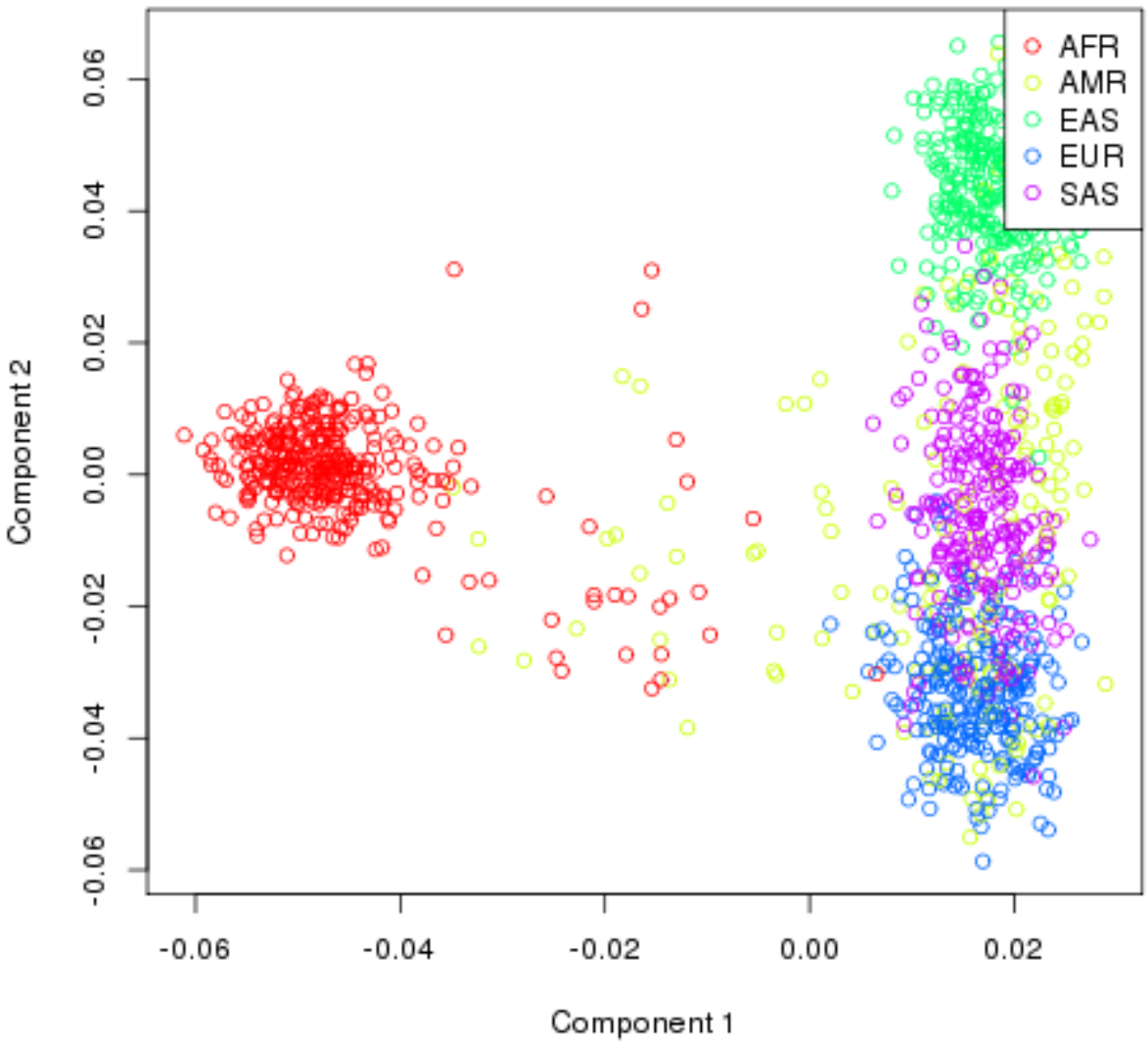


Figure 1: Q1 First two PCs of 1000 genomes project

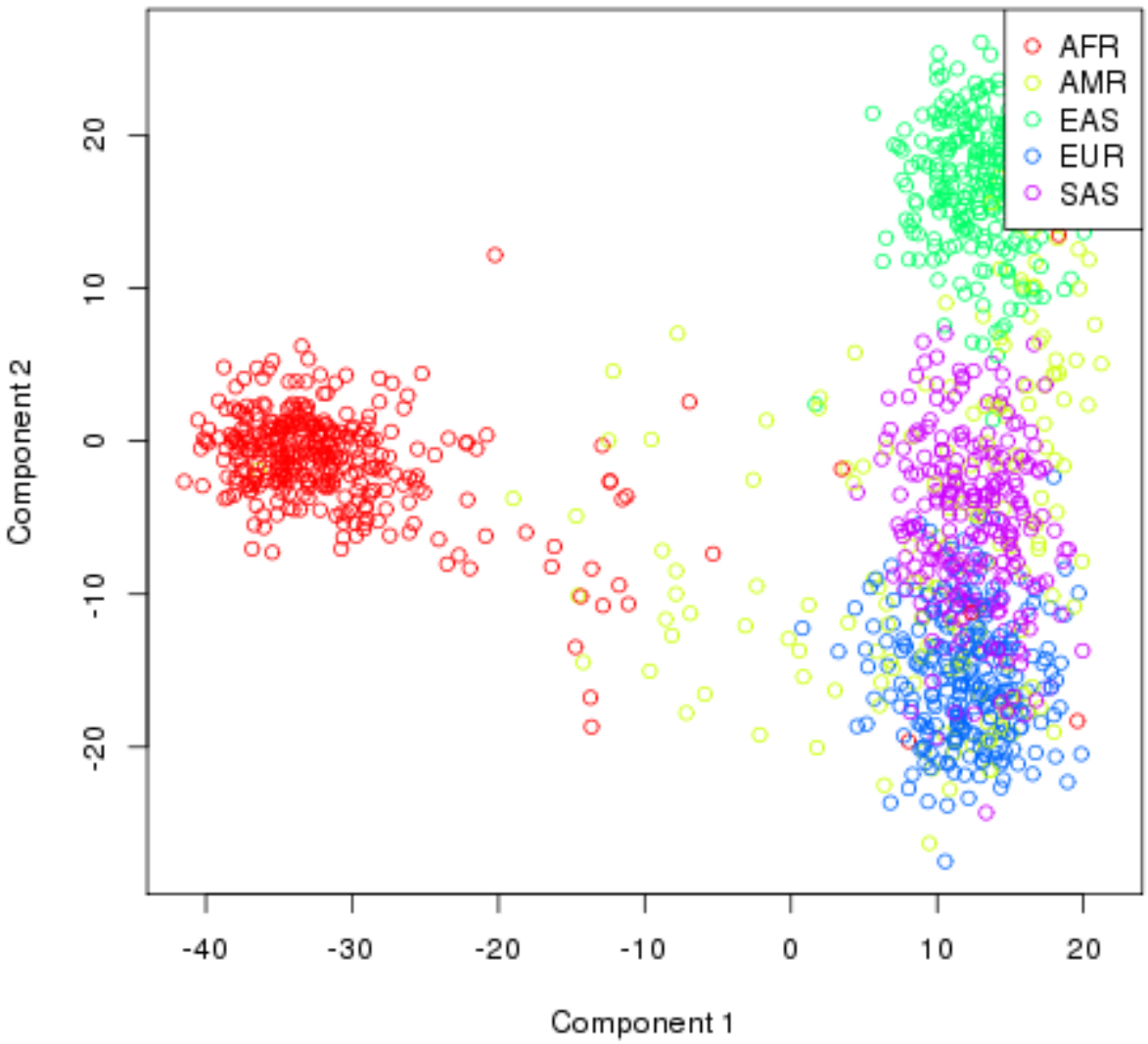


Figure 2: Q2 First two PCs of lung cancer by race

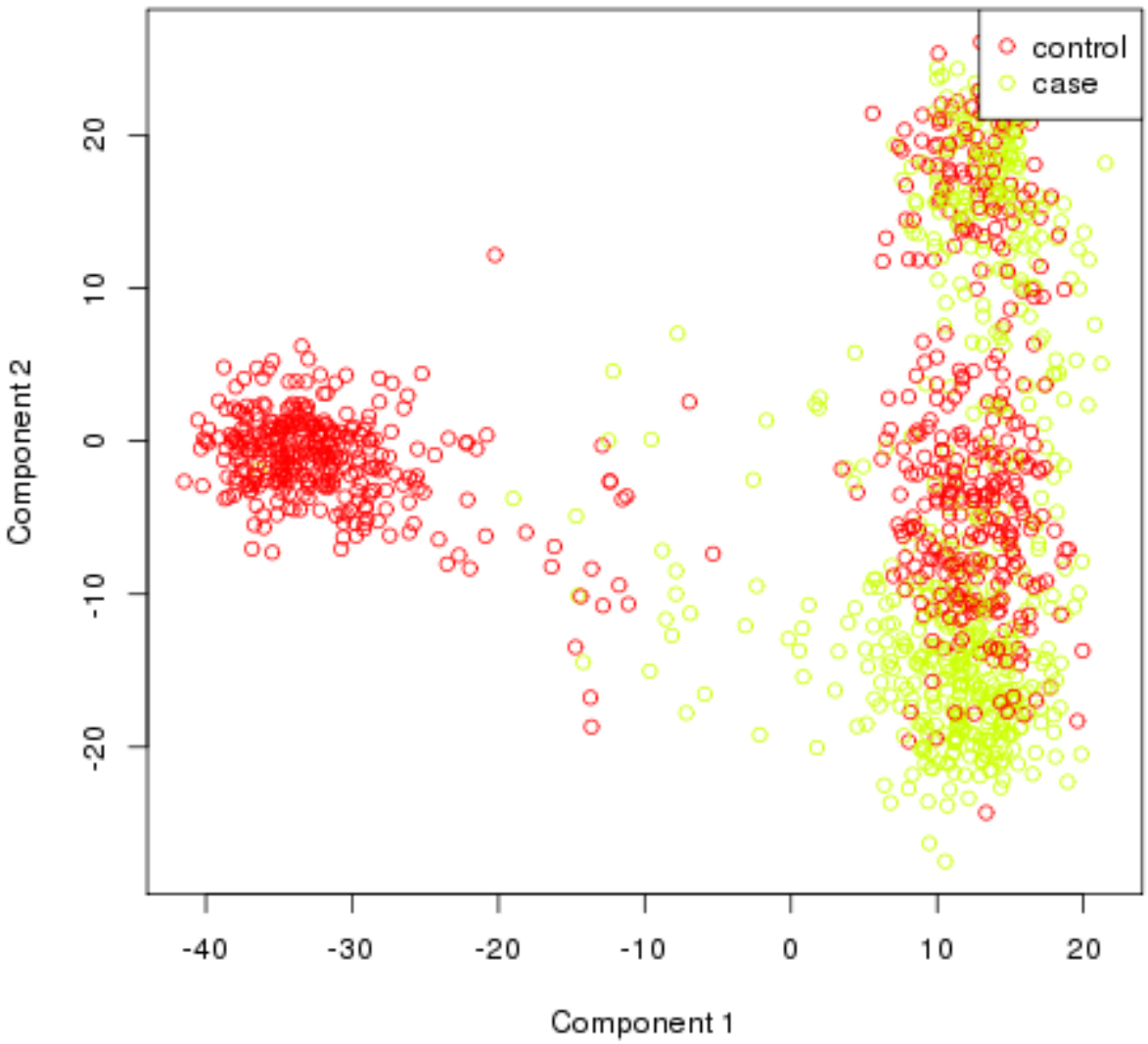


Figure 3: Q2 First two PCs of lung cancer by phenotype

perform GWAS tests on the residuals. Alternatively, we can perform IBS clustering and use the clusters as covariates.

Hint: Color code each individual in the PCA plot using information regarding race/ancestry or phenotype.

Part II. GWAS

This part will walk you through GWAS quality control and analysis using PLINK[2].

The data for this part can be found in the following folders: /n/stat115/hws/5/GWAS/ binary_formats – phenotype and genotype information in binary formats. map_ped_formats – phenotype and genotype information in map and ped formats.

We recommend you to use the binary format files since plink is optimized for binary formats.

Hint: installation of PLINK

1. wget http://pngu.mgh.harvard.edu/~purcell/plink/dist/plink-1.07-x86_64.zip
2. unzip plink-1.07-x86_64.zip
3. export PATH=$PATH: [install.path]/plink-1.07-x86_64/plink
4. Type "plink" to test

4. Using Plink for quality control (QC). GWAS analysis starts from well quality controlled, correctly filtered GWAS data set.

How would you run Plink to perform a good quality control for GWAS dataset?

Please explain which commands would be appropriate, and why did you decide such thresholds to filter out some SNPs and individuals. It's not necessary to actually run the command.

Hint: Plink website and –mind –geno –maf –hwe, etc.

The following Plink flags can be used to filter out low quality data.

The –mind flag filters out individual samples with a missing call rate below the cutoff. Because we only have 1,252 samples, we set the cutoff to 0.1 to not filter out too much of our data.

The –geno flag filters out varients with a missing call rate above the cutoff. With 234,148 SNPs, we can be a little more agressive than for –mind and set the cutoff to 0.01.

The –maf flag filters out varients with a minor allele frequency below the cutoff. We use the typical cutoff of 0.05, because it is difficult to get enough statistical power otherwise with only 1,252 samples.

The –hwe flag filters out varients with that do not satisfy Hardy-Weinberg equilibrium. We set the p-value cutoff to 0.001, because the documentation recommends being conservative with this filter. We don't want to filter out too much and actual experimental mistakes tend to generate extreme Hardy-Weinberg equilibrium deviations. We also include the mid p-value adjustment as recommended.

```
plink --bfile chr8 --mind 0.1 --maf 0.05 --geno 0.01 --hwe 0.001 midp --out chr8filtered --make-bed
```

5. Association tests using Plink.

Use Plink to run the following association tests between the SNPs and the disease with default parameters.

Hint: plink –bfile chr8 –model –allow-no-sex –noweb

a. Allelic Association Test

b. Cochran-Armitage trend test

c. Genotypic Association test

d. Dominant gene action test

e. Recessive gene action test

```
#!/bin/bash
#SBATCH -p seas_iacs
#SBATCH -J plink
#SBATCH -n 2
#SBATCH -N 1
#SBATCH -t 0-4:00
#SBATCH --mem 16000
#SBATCH -o plink%j_%N.out
#SBATCH -e plink%j_%N.err

#Echo Commands
set -x

#Load Modules
source new-modules.sh
module load plink/1.90-fasrc01

#Run plink
FILENAME=/n/stat115/hws/5/GWAS/binary_formats/chr8
#Question 5
plink --bfile $FILENAME --model --allow-no-sex --noweb
#Question 7
plink --bfile $FILENAME --logistic --allow-no-sex --out logis
plink --bfile $FILENAME --logistic --allow-no-sex --covar ../part1/PC.txt --out logispc
```

6. Read the results in R, do multiple testing corrections for the p-values. Then for each test (GENO, TREND, ALLELIC, DOM A=ND REC), do a Manhattan Plot in chromosome 8.

Hint: R package qqman, function manhattan()

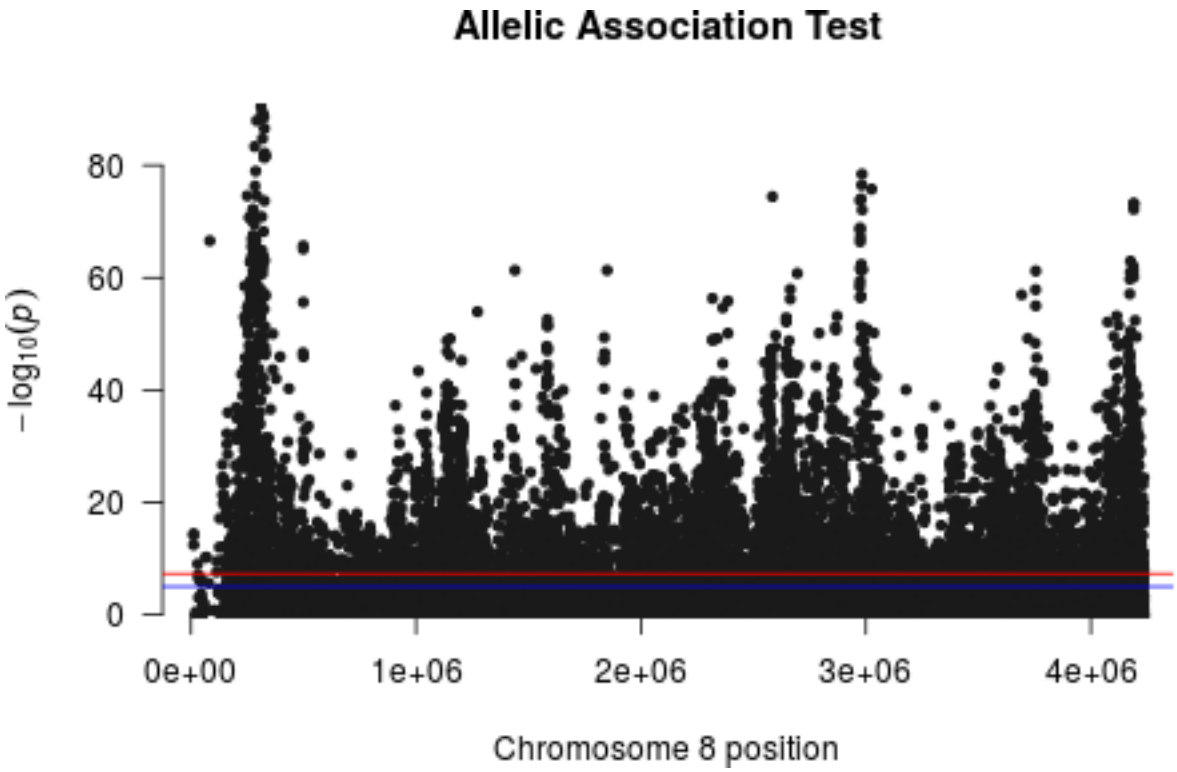
```
library(qqman)
```

```
##

## For example usage please run: vignette('qqman')

##

## Citation appreciated but not required:

## Turner, S.D. qqman: an R package for visualizing GWAS results using Q-Q and manhattan plots. biorXiv

##
```

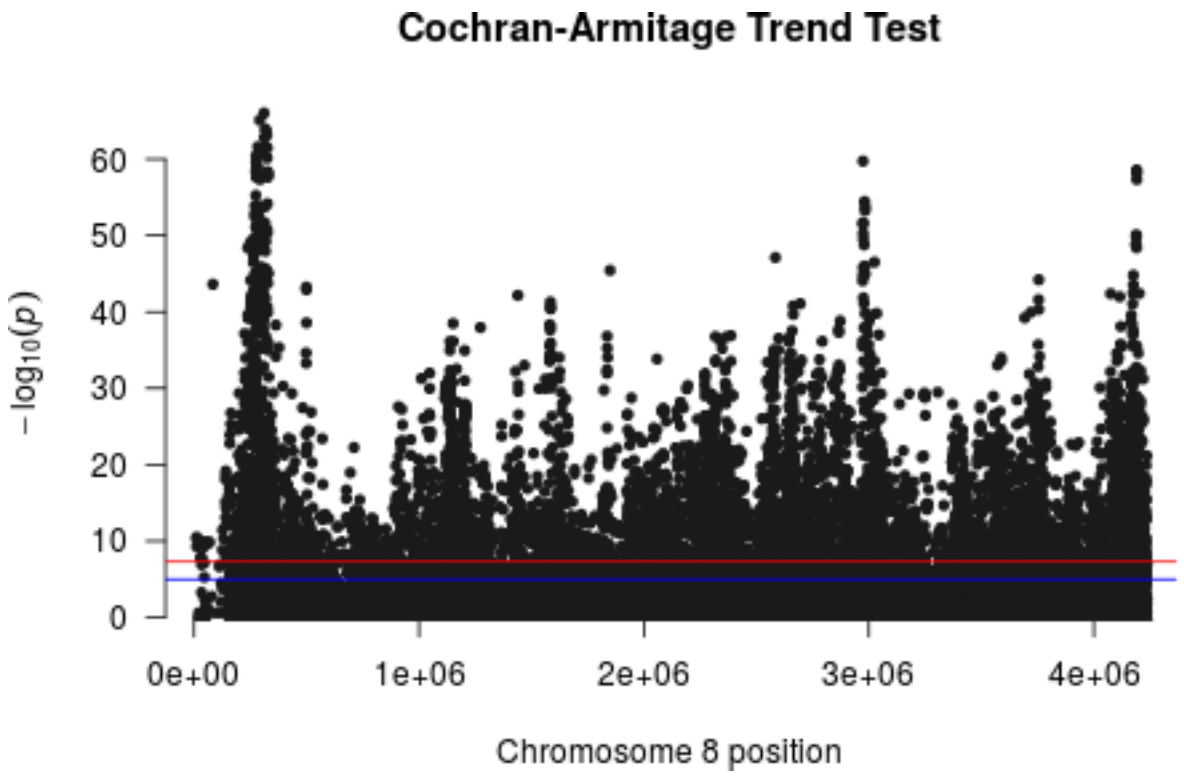
```
#load and filter models
results = read.table("odyssey/part2/plink.model", header=TRUE)
bpmap = read.table("odyssey/part2/chr8.map", col.names = c("CHR", "SNP", "DIST", "BP"))
#filter out NA
```

```
models = na.omit(results)
#merge with BP
bpmap$CHR = NULL
bpmap$DIST = NULL
models = merge(models, bpmap, by.x="SNP", by.y="SNP")
#bonferroni correction on p-values
models$P = models$P * dim(models)[1]
```

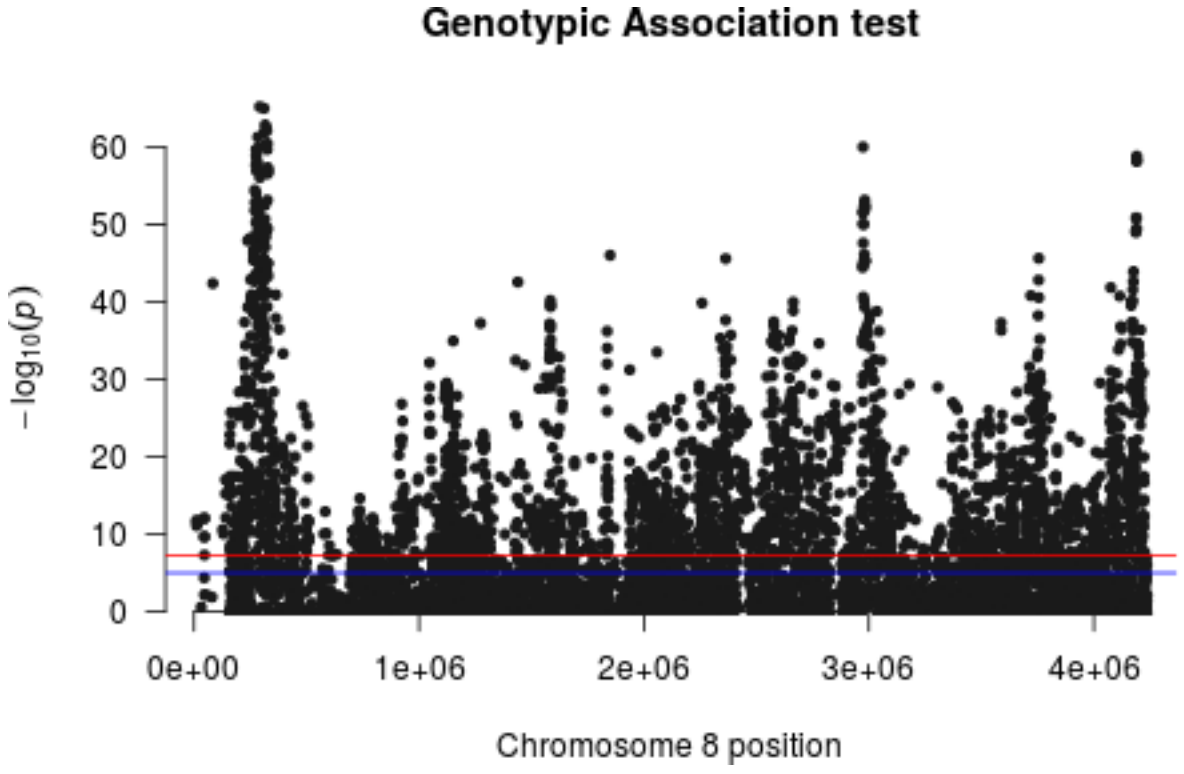
```
models.test = models[models[,"TEST"]=="ALLELIC",]
manhattan(models.test, main="Allelic Association Test")
```



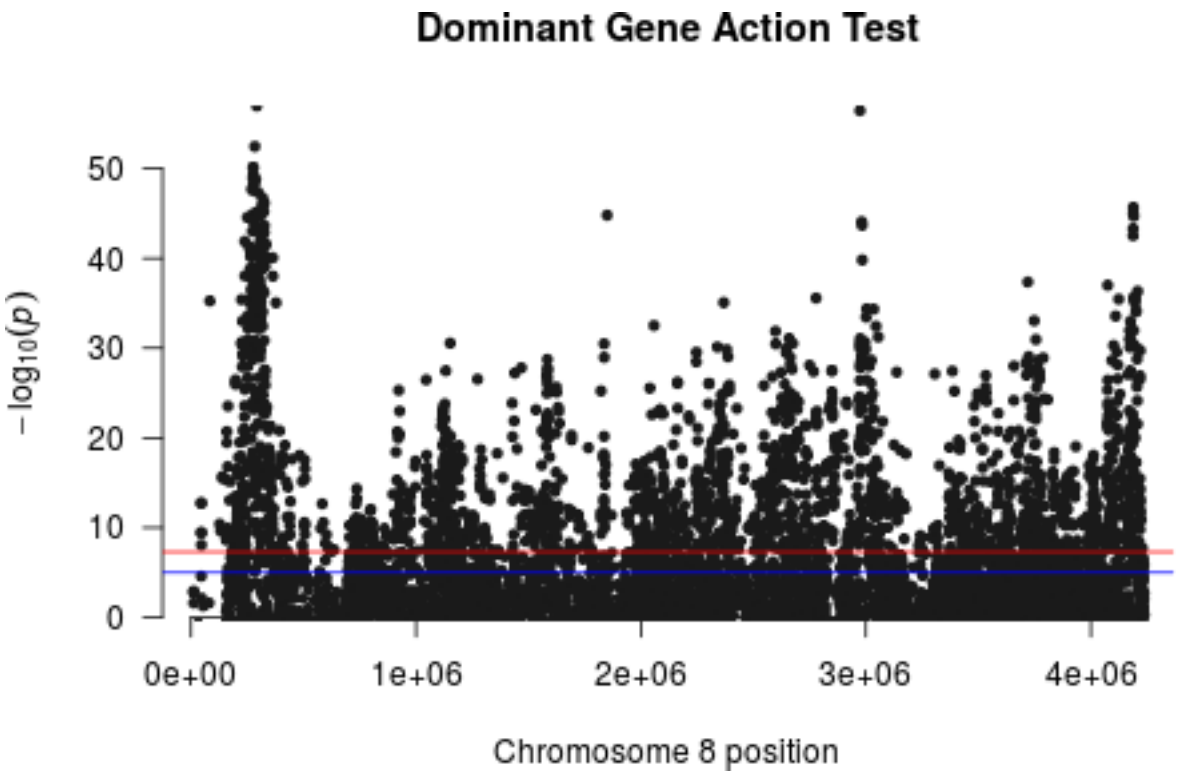

```
models.test = models[models[,"TEST"]=="TREND",]
manhattan(models.test, main="Cochran-Armitage Trend Test")
```



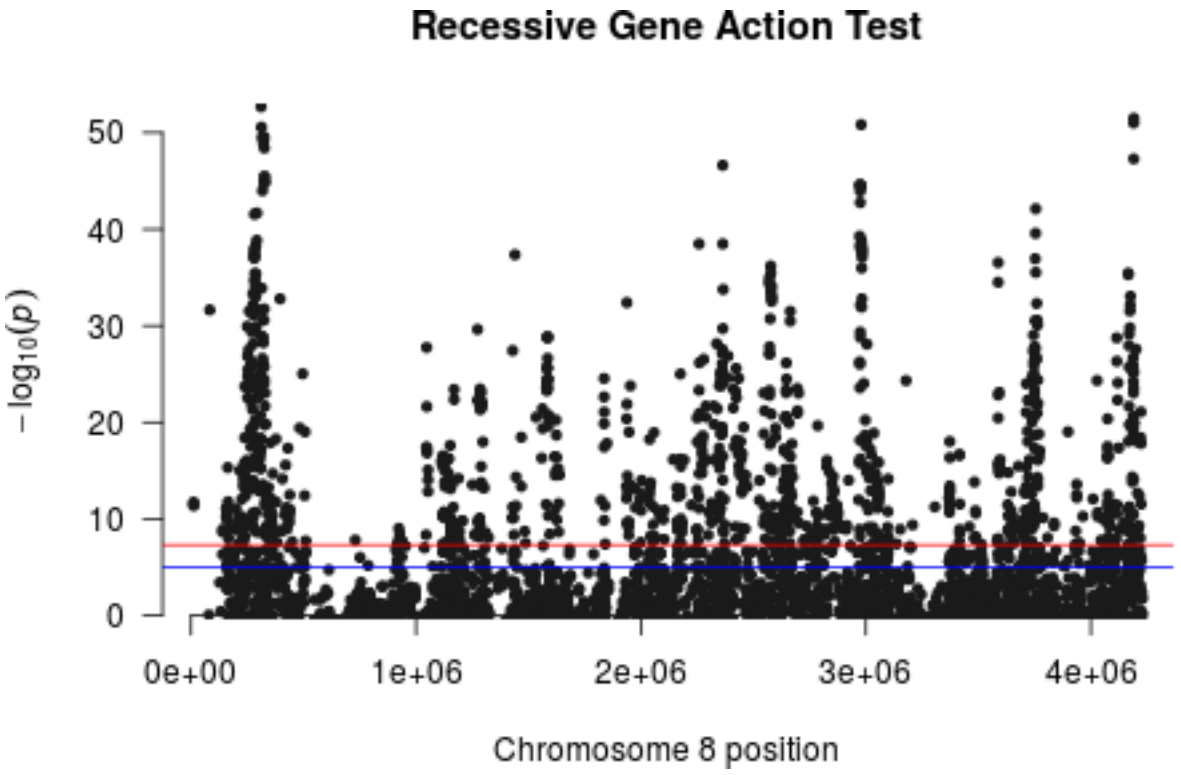

```
models.test = models[models[,"TEST"]=="GENO",]
manhattan(models.test, main="Genotypic Association test")
```




```
models.test = models[models[,"TEST"]=="DOM",]
manhattan(models.test, main="Dominant Gene Action Test")
```




```
models.test = models[models[,"TEST"]=="REC",]
manhattan(models.test, main="Recessive Gene Action Test")
```



Even after correcting p-values with Bonferroni, we see a surprising number of significant results, which suggests an issue with our data (e.g. confounding covariates).

7. (Graduate students) Run logistic regression and test association correcting for population structure.

We recommend you to read the following review paper: Price et al (2010) New approaches to population stratification in genome-wide association studies.

Try to do the following tests:

a. Logistic regression of the disease status and the genotype (no additional covariates).

b. Logistic regression of the disease status and the genotype (with PCA1 and PCA2 as the covariates).

Hint: plink –bfile chr8 –logistic –allow-no-sex –covar PC.txt

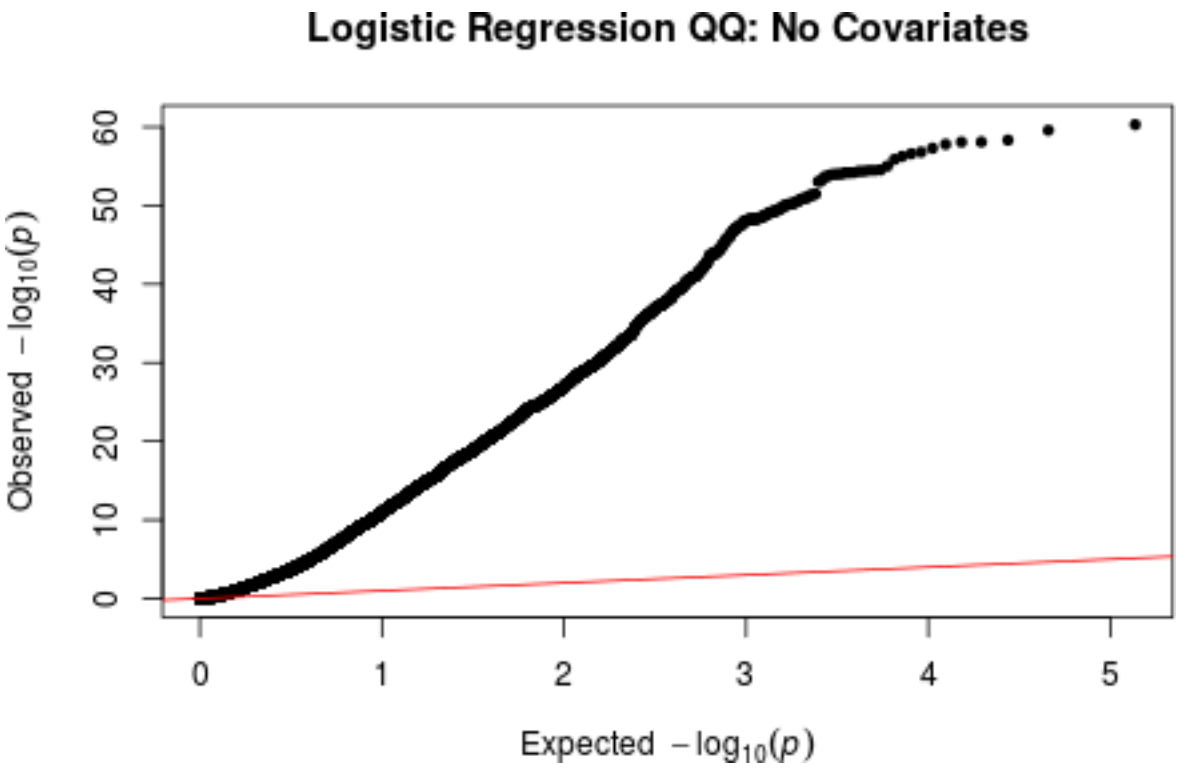
See script in question 5 for running Plink logistic regression. For part b, see bottom of script for question 1 for writing PCA1 and PCA2 to the file PC.txt.

8.(Graduate students) Compare the QQ-plot before and after correcting the population structure (race/ancestry). Briefly describe what you observe.

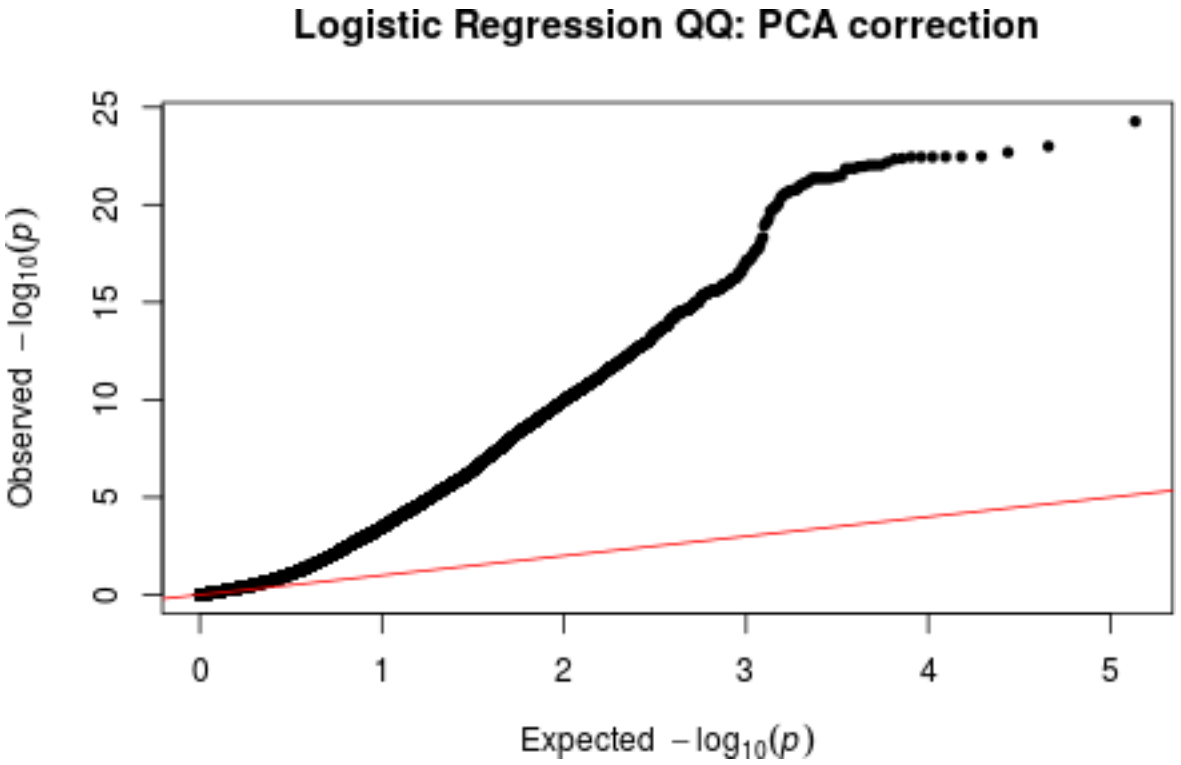
```
library(qqman)
```

```
#read results
logis = read.table("odyssey/part2/logis.assoc.logistic", header=TRUE)
logispc = read.table("odyssey/part2/logispc.assoc.logistic", header=TRUE)
#extract result with covariates
logispc = logispc[logispc[,"TEST"] == "ADD",]
#filter out na
logis = na.omit(logis)
logispc = na.omit(logispc)
```

```
qq(logis$P, main="Logistic Regression QQ: No Covariates")
```




```
qq(logispc$P, main="Logistic Regression QQ: PCA correction")
```



For the QQ-plots, we expect the points to lie along the red diagonal (baseline) and curve up a bit at the top end (actually significant SNPs). However, for logistic regression without covariates, we notice that the slope is much higher than expected. After including PCA1 and PCA2 as covariates, we note that the slope of the QQ-plot is lower and closer to the diagonal line. Although better, the resulting plot is still far from the

diagonal and suggests additional covariates exists beyond just PCA1 and PCA2.

9. From a GWAS study on Lung Cancer patient, the top three SNPs are rs2736100, rs9357152, rs2981579. What pathway analysis for each SNP would you consider? After your pathway analysis, which SNP would be more interesting for further investigation?

Hint: https://www.ebi.ac.uk/gwas/home

Source used: https://www.ebi.ac.uk/gwas/home and https://www.ncbi.nlm.nih.gov/snp

rs2736100 is associated with the TERT gene (telomerase reverse transcriptase) and an EGFR positive lung cancer study. TERT maintains the telomere ends and allows cells to continue dividing. rs9357152 is associated with the HLA-DQB1 and HPV studies. HLA-DQB1 is an MHC immune gene. rs2981579 is associated with the FGFR2 gene (fibroblast growth factor receptor 2) and various breast cancer studies. FGFR2 is a signaling protein that controls differentiation.

The most promising SNP for further analysis is rs2736100, because it is associated with an function that suppresses apoptosis and previously associated in another lung cancer study.

Part III Genome-wide Complex Trait Analysis

For most human complex diseases and traits, SNPs identified by genome-wide association studies (GWAS) explain only a small fraction of the heritability. A new quantitative genetic method, Genome-wide Complex Trait Analysis (GCTA) for estimating genetic influence using DNA, has become a popular method to the armamentarium of quantitative genetics.

GCTA allows estimations of heritability due to common SNPs using relatively small sample sizes (e.g., a few thousand genotype-phenotype pairs). The method is independent of, but delivers results consistent with, "classical" methods such as twin and adoption studies.

We highly recommend you to read the following paper: Yang et al (2010) Nature Genetics Common SNPs explain a large proportion of the heritability for human height. (Cited by 1600+).

10 Apply the GCTA analysis on the same lung-cancer GWAS data in Part II and estimate the genetic relationship matrix (GRM) between pairs of individuals.

Hint: gcta64 –bfile hw_chr8 –autosome –make-grm –out chr8

```
#!/bin/bash
#SBATCH -p seas_iacs
#SBATCH -J gtca
#SBATCH -n 2
#SBATCH -N 1
#SBATCH -t 0-4:00
#SBATCH --mem 16000
#SBATCH -o gtca%j_%N.out
#SBATCH -e gtca%j_%N.err

#Echo Commands
set -x

#Run GTCA
```

```
FILENAME=/n/stat115/hws/5/GWAS/binary_formats/chr8
PHENOFILE=/n/stat115/hws/5/population/lungcancer.pheno
#Question 10
../gcta/gcta64 --bfile $FILENAME --autosome --make-grm --out part10chr8
#Question 11
../gcta/gcta64 --reml --grm part10chr8 --pheno $PHENOFILE --grm-adj 0 --grm-cutoff 0.05 --out part11chr
```

As output, we get a binary file part10chr8.grm.bin containing the matrix and a text file part10chr8.grm.id containing the ids.

11 Perform REML (restricted maximum likelihood) analysis to estimate the phenotypic variance explained by the SNPs

Hint: gcta64 –reml –grm chr8 –pheno chr8.phen –grm-adj 0 –grm-cutoff 0.05 –out hw_chr8

With the commands shown in question 10, we get the following hsq output.

```
Source  Variance    SE
V(G)    0.347812    0.053988
V(e)    0.000000    0.033627
Vp  0.347813    0.029193
V(G)/Vp 0.999999    0.096682
logL    264.111
logL0   115.115
LRT 297.991
df  1
Pval    0
n   554
```

The phenotypic variance is 0.347812 and the ratio of genotypic to phenotypic variance is 0.999999. This result suggests that near all of the observed phenotypic variance can be explained by genetics. (Note that this result does not factor in the population stratification covariates that we found in previous parts.)

Part IV (Graduate students) Perform dynamic programming using python:

Given a list of finite integer numbers: e.g. -2, 1, 7, -4, 5, 2, -3, -6, 4, 3, -8, -1, 6, -7, -9, -5, ... Write a python script to maximize the Z where Z is the sum of the numbers from location X to location Y on this list. Be aware, your algorithm should look at each number ONLY ONCE from left to right.

Hint: You can use dynamic programming to solve this problem with <20 lines of codes.

```
def max_sum(arr):
  x = 0 #start index
  y = 0 #end index
  z = 0 #current sum
  best_x = 0 #best start index
  best_y = 0 #best stop index
  best_z = 0 #best sum
  for y, value in enumerate(arr):
    z += value
    if z > best_z: #update best
      best_x = x
```

```
      best_y = y
      best_z = z
    if z < 0: #discard negative subsequence, better to leave out
      x = y + 1
      z = 0
  return (best_x, best_y, best_z)

def run_test(arr):
  print("Input Array")
  print(arr)
  print("X (start), Y (end), Z (sum)")
  print(max_sum(arr))

#tests
run_test([1, 2, 3, 4, 5])
run_test([1, -5, 3])
run_test([3, -2, 3])
run_test([-2, 1, 7, -4, 5, 2, -3, -6, 4, 3, -8, -1, 6, -7, -9, -5])
```

```
## Input Array
## [1, 2, 3, 4, 5]
## X (start), Y (end), Z (sum)
## (0, 4, 15)
## Input Array
## [1, -5, 3]
## X (start), Y (end), Z (sum)
## (2, 2, 3)
## Input Array
## [3, -2, 3]
## X (start), Y (end), Z (sum)
## (0, 2, 4)
## Input Array
## [-2, 1, 7, -4, 5, 2, -3, -6, 4, 3, -8, -1, 6, -7, -9, -5]
## X (start), Y (end), Z (sum)
## (1, 5, 11)
```

Note that my X and Y positions are 0-indexed and includes both endpoints.

Submission

Please submit your solution directly on the canvas website. Please provide your code (.Rmd) and a pdf file for your final write-up. Please pay attention to the clarity and cleanness of your homework. Page numbers and figure or table numbers are highly recommended for easier reference.

The teaching fellows will grade your homework and give the grades with feedback through canvas within one week after the due date. Some of the questions might not have a unique or optimal solution. TFs will grade those according to your creativity and effort on exploration, especially in the graduate-level questions.