Genome-Wide Association Exercise Association Analysis Controlling for Population Substructure

1. Population Stratification and Association Testing

The dataset from part I of this exercise which you performed data quality control (QC) on was obtained from HapMap Phase III data. It contains CEU founders (Caucasians from Utah), MEX founders (Mexicans from Los Angeles) and TSI (Tuscans from Italy). The CEU pedigree identifiers begin with only numbers e.g., 1347, the MEX pedigree identifies all start with M e.g., M017 and the TSI pedigree identifiers all start with NA e.g., NA0217. Before we start testing for association, we want to know if there are outliers. Even after removing the outliers when association analysis is performed population substructure and admixture may need to be controlled. If not, we risk observing an association, which is due to a difference in genotype frequencies in cases and controls, because of population substructure/admixture and not because of linkage disequilibrium (LD) between tagSNP(s) and the functional variant(s). We are going to use multidimensional scaling (MDS) and principal components analysis (PCA) within the PLINK software to generate 10 components.

Disclaimer: You usually should not analyze data from European-Americans, Mexican- Americans and Italians together even if you control for population stratification. They can be analyzed separately, and the data combined using meta-analysis.

Note: For a GWAS study instead of this toy study, you will have a denser set of markers of which some will be in LD. You should first prune your SNPs to obtain a subset in linkage equilibrium/weak LD (R2<0.1) prior to performing MDS or PCA analysis on the data. Although for association analysis is performed on the entire data set will be analyzed only this a subset of SNPs which are not in LD will be used to construct PCA and MDS components. For more information on how to do this in PLINK see https://www.cog-genomics.org/plink/1.9/ld.

A note on running PLINK from this notebook

This notebook runs in a single R kernel from start to finish — no more switching between a Bash kernel and an R kernel. PLINK is a command-line program, not an R package, so to run it from R we use the built-in `system()` function. `system()` hands its argument (a string) to your computer's shell exactly as if you had typed it into a terminal yourself,

waits for it to finish, and lets any files it writes (`.log` , `.mds` , `.eigenvec` , etc.) show up in your working directory as usual.

For example, the R code

`system("plink --file GWAS_clean4 --genome --cluster --mds-plot 10")`
does exactly the same thing as opening a terminal, `cd` -ing into the data directory, and typing:

`plink --file GWAS_clean4 --genome --cluster --mds-plot 10`
Every `system()` call below is preceded by a comment showing that plain command-line form, so you can always see exactly what is being run — you could copy that line straight into a terminal and get the same result.

Set Kernel to R

In [ ]:
```
# Move into the working directory containing the PLINK input files.
# Command-line equivalent:
#   cd ./data
setwd("./data/")
```

In [ ]:
```
# Command-line equivalent:
#   plink --file GWAS_clean4 --genome --cluster --mds-plot 10
plink_out <- system("plink --file GWAS_clean4 --genome --cluster --mds-plot
cat(plink_out, sep = "\n")
```

This command outputs the file `plink.mds` that contains the subject IDs and values for the 10 components we just generated. There is another file in your folder called `mds_components.txt` . This file is identical to your `plink.mds` file with the exception that a group column which codes CEU individuals as 1, MEX individuals as 2 and TSI individuals as 3. This is done so when we plot the MDS components in R you can see which group the points belong to and judge how well does the data cluster, e.g., are there outliers. The following commands will generate a jpeg image file containing the mds `plot (filename=mds.jpeg)` in your current working directory. Open R and use the following command:

In [ ]:
```
mydata = read.table("mds_components.txt", header=T)
mydata$pch[mydata$Group==1 ] <-15
mydata$pch[mydata$Group==2 ] <-16
mydata$pch[mydata$Group==3 ] <-2
#jpeg("mds.jpeg", height=500, width=500)
plot(mydata$C1, mydata$C2 ,pch=mydata$pch)
dev.off()
```

Visualizing population structure using MDS is useful for identifying subpopulations, population stratification and systematic genotyping or sequencing errors, and can also be used to detect individual outliers that may need to be removed, e.g. European-Americans included in a study of African-Americans. MDS coordinates help with

visualizing genetic distances and population substructure. PLINK also offers another dimension reduction, --pca, for PCA, the PC components which can also be used for visualizing data to detect outliers in the same manner which was performed using MDS. Additionally, covariates either from either MDS or PCA can be used in a regression model to aid in correcting for population substructure and admixture.

We will now continue performing the analysis using PLINK but will use PCA instead of MDS. We will generate PCs and determine how many PC covariates should be included in the regression model. When SNPs are tested for an association with a trait analysis can be performed, first by including no PC components, then one PC component and then two PC components and so on. Please note that as each PC component is added all the SNPs are analyzed, e.g. a complete GWAS is performed. Examining λ can aid in determining how many PC components should be included in the analysis. If there is no population stratification or other biases, then λ should equal 1 or ~1. We will use λ to determine how many PC components from our analysis will be added to the logistic regression model. First, estimate λ without adjusting for any PC components:

In [ ]:
```
# Command-line equivalent:
#   plink --file GWAS_clean4 --pheno pheno.txt --pheno-name Aff --logistic -
plink_out <- system("plink --file GWAS_clean4 --pheno pheno.txt --pheno-name
cat(plink_out, sep = "\n")
```

Generated the first 10 PCA values:

In [ ]:
```
# Command-line equivalent:
#   plink --file GWAS_clean4 --genome --cluster --pca 10 header
plink_out <- system("plink --file GWAS_clean4 --genome --cluster --pca 10 he
cat(plink_out, sep = "\n")
```

Eigenvectors are written to plink.eigenvec, and top eigenvalues are written to plink.eigenval. The 'header' modifier adds a header line to the .eigenvec file(s). And then find out what λ is when we adjust for the first component:

In [ ]:
```
# Command-line equivalent:
#   plink --file GWAS_clean4 --pheno pheno.txt --pheno-name Aff --covar plin
plink_out <- system("plink --file GWAS_clean4 --pheno pheno.txt --pheno-name
cat(plink_out, sep = "\n")
```

And the first and second components:

In [ ]:
```
# Command-line equivalent:
#   plink --file GWAS_clean4 --pheno pheno.txt --pheno-name Aff --covar plin
plink_out <- system("plink --file GWAS_clean4 --pheno pheno.txt --pheno-name
cat(plink_out, sep = "\n")
```

and so forth for all 10 components in the .log file completing the table:

Table 1	Un-adjusted	PC 1	PC 1-2	PC 1-3	PC 1-4	PC 1-5	PC 1-6	PC 1-7	PC 1-8	PC 1-9	PC 1-10
λ											

The number closest to 1.0, with the least number of PC components, would be the best for adjusting without overfitting and introducing unnecessary noise. You can check your table against the one provided in the answers section.

Go to the **`assoc.logistic` file that corresponds to that number of components** and make a note of how you named the `.assoc.logistic` file for it and when you did not adjust for any components. Then go back to the R program to load the results and create a jpeg image file containing QQ plots for the adjusted and unadjusted results (using a modified script from http://www.broad.mit.edu/node/555) as follows:

In [ ]:
```
broadqq <-function(pvals, title) {
    observed <- sort(pvals)
    lobs <- -(log10(observed))
    expected <- c(1:length(observed))
    lexp <- -(log10(expected / (length(expected)+1)))
    plot(c(0,7), c(0,7), col="red", lwd=3, type="l", xlab="Expected (-logP)"
         xlim=c(0,max(lobs)), ylim=c(0,max(lobs)), las=1, xaxs="i", yaxs="i"
    points(lexp, lobs, pch=23, cex=.4, bg="black")
}
#jpeg("qqplot_compare.jpeg", height=1000, width=500)
par(mfrow=c(2,1))
aff_unadj<-read.table("unadj.assoc.logistic", header=TRUE)
aff_unadj.add.p<-aff_unadj[aff_unadj$TEST==c("ADD"),]$P
broadqq(aff_unadj.add.p,"Some Trait Unadjusted")
aff_C1C2<-read.table("PC1-PC2.assoc.logistic", header=TRUE)
aff_C1C2.add.p<-aff_C1C2[aff_C1C2$TEST==c("ADD"),]$P
broadqq(aff_C1C2.add.p, "Some Trait Adjusted for PC1 and PC2")
dev.off()
```

Now look for SNPs with genome-wide significance using the following R connamds:

In [ ]:
```
gws_unadj = aff_unadj[which(aff_unadj$P < 0.0000001),]
gws_unadj
gws_adjusted = aff_C1C2[which(aff_C1C2$P < 0.0000001),]
gws_adjusted
```

Note: These are the uncorrected p-values for multiple testing. The p-values which have been corrected using various multiple testing methods can be found in the .adjusted file.

A common question when you have a finding with genome-wide significance in a GWAS is “Is the SNP in a known gene?” One way to look this information up is annotate variants in batch (please look at the annotating exercise for more information). You can do this using the Ensembl Variant Predictor. Go to the website:

http://grch37.ensembl.org/Homo_sapiens/Tools/VEP (GRCh37 version)

Type the rs number(s) of the SNP(s) with genome-wide significance in “Either paste data”, leave all options default and press run. In a few minutes you can view the results of your query.

2. Questions

Question 1: Did this study have a finding with genome-wide significance after adjusting for population substructure? Did you notice any difference in the p-values before and after adjustment for substructure? How many PC components should you include in the regression model?

Table 2. SNPs with genome-wide significance unadjusted for substructure:

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P

Table 3. SNPs with genome-wide significance adjusted for components 1 and 2:

CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P

Question 2: Why would you not want to include in your analysis individuals from different ethnic backgrounds even if you control for population substructure?

Question 3: Are any SNPs with genome-wide significance in known genes?