



Lab 6: Genomics Data Analysis, using population structure with PCA, Clustering, and Regression (80 Points)

Objective

To provide students with hands-on experience in genomics data analysis techniques, including data preprocessing, Principal Component Analysis (PCA), clustering, and association analysis with phenotypes (including population substructure) using linear and logistic regression. The tasks are to be done in R and PLINK software.

Requirements

- You are obligated to attempt all tasks
- You **MUST** use R Markdown to submit your report
- You **MUST** submit the screenshots of PLINK logs
- State your name and ID at the first cell in your markdown report
- For all the tasks, you should use the [Dataset of 156 Qataris](#).

Part 1: Quality Control and linkage disequilibrium (LD) with PLINK (10 Points)

Task 1.1: QC, LD, and PCA (10 Points)

- Run the PLINK command that filters out variants below the minor allele frequency threshold, excludes samples with low genotyping rates, and removes SNPs not in Hardy-Weinberg equilibrium using the following thresholds (hwe: 0.00001, maf: 0.05, geno: 0.001). The output is called **cleaned data**.

- On the **cleaned data**, run the following command to perform linkage disequilibrium pruning. The output is called **pruned data**

```
plink --bfile [cleaned_data] --indep-pairwise 100 5 0.1 --out [pruned_data]
```

- Experiment with the thresholds 100, 5, and 0.1 and comment on the effect of increasing or decreasing the threshold.
- Run PCA on the **pruned data** using the PLINK and --pca flag

Part 2: Identify SNPs associated with population structure (40 Points)

Task 2.1: Identify SNPs associated with genomic PCs using linear Regression Analysis (20 Points)

Conduct linear regression analysis for all SNPs in the **cleaned data (NOT the pruned data)**, with the model specified below

Perform the following steps:

- Use Plink to recode your data to the 0, 1, 2 format with the --recode A option. This will generate a .raw file
plink --bfile cleaned_data --recode A --out recoded_data
- Read the .raw file into R (**use the function read.table and specify header=T and sep=""**)
- Isolate the columns that contain the SNP data, which usually follows the first six columns (**use the select function from tidyverse**). Keep the IDs to match those with the IDs of the PCs
- Read the PCA eigenvalues and eigenvectors into R (**use the function read.table**)
- Create a nested loop to run linear regressions for the 3 PCs with **all SNPs**. Run the following for each PC with a single SNP, while correcting the model for the other two PCs.

$lm(PC_i \sim SNP_x + PC_j + PC_k, data = dataset); x: 1..40411; i, j, k: 1..3, i \neq j \neq k$

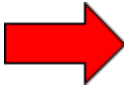
- Determine all associations that pass a **Bonferonni threshold of significance**. What is this threshold?

Task 2.2: Identify SNPs that associate with the population subgroups (clusters) using logistic regression (20 Points)

Implement logistic regression models to cluster data using 3 highest variance PCA components assuming the number of clusters = 3 (i.e. $k = 3$).

Perform the following steps:

- Set your pseudorandom seed before running the clustering algorithm. (**use `set.seed`**)
- Run the k-means clustering algorithm using the first 3 PCs. Plot the 3D scatterplot with different colors for each cluster. (**use `scatterplot3d`**)
- Create a new dataframe of the 3 PCs and the one-hot encoding of cluster labels like the following illustration. (**use `dummyVars` from the `caret` package to do so**). You should have 6 columns in this dataframe.



	category
1	3
2	1
3	1
4	3
5	3
6	3
7	1
8	1
9	3
10	3
11	1
12	1
13	2

	category.1	category.2	category.3
0	1	0	0
1	0	0	1
2	1	0	0
3	1	0	0
4	0	0	1
5	0	0	1
6	0	0	1
7	1	0	0
8	1	0	0
9	0	0	1
10	0	0	1
11	1	0	0
12	1	0	0
13	0	1	0

- Create a nested loop to run logistic regressions for each of the 3 clusters with ALL SNPs including PCs as covariates. Store the summary of each regression in a list for subsequent analysis and comparison.

$glm(Cluster_i \sim SNP_x + PC_1 + PC_2 + PC_3, family = binomial, data = dataset); i:1..3, x:1..40411$

- Identify significant SNPs that are associated with each of the 3 clusters using a **Bonferroni threshold**

Task 3: Manhattan, Annotation, and Discussion (25 points)

- Plot a **Manhattan plot** using the ggman package in R for the results you got (you should use all association results you have for ALL SNPs, but you can select those that pass the threshold of 0.05 if your data is too huge for the memory). This will need preparing a file with SNP chromosome and position (get from the map file) and matching those with the p-values obtained from regression. Do this once for the results from Task 2.1 and Task 2.2 separately.
- Use dbSNP from NCBI website to check the information on the 10 most significant SNPs, by doing the following steps for both results obtained from Task 2.1 and Task 2.2 separately:
 - Report a table of the 10 most significant SNPs that includes
 - Chromosome (CHR) of the SNP
 - Position (BP) of the SNP
 - Statistics from regression results (beta coefficient, standard error and p-value)
 - From dbSNP, the gene that contains the SNP
 - From dbSNP, the allele frequency in different databases

An example would be (not necessarily one of the top 10 but just for illustration)

SNP	CH R	BP	Beta	P-Val	SE	Gene	Freq	Max Freq	Min Freq
rs2840528	1	2352457	3.57e-2	1.04e-9	0.00589	MORN1	A=0.465458 (123202/264690, TOPMED) G=0.429532 (97435/226840, ALFA) A=0.473731 (66346/140050, GnomAD)	0.429532	0.473731

- Comment on the significant SNPs identified in relation to the PCs and clusters. Were the SNPs different for each PC and for each cluster? Did the analysis using PCs lead to different SNPs from those used with clustering? What genes contain those SNPs (or at least the 10 top significant SNPs)?

Reporting (5 points)

- You are required to write a report including the commands, and file formats and include any screenshots of the output of such commands

References

- [Dataset of 156 Qataris](#)
- [PLINK file formats](#)
- [NCBI - dbSNP](#)
- [scatterplot3d](#)
- [ggman](#)
- [kmeans](#)
- [Markdown Cheatsheet](#)