Lab 6

Maria Bassem_20011141

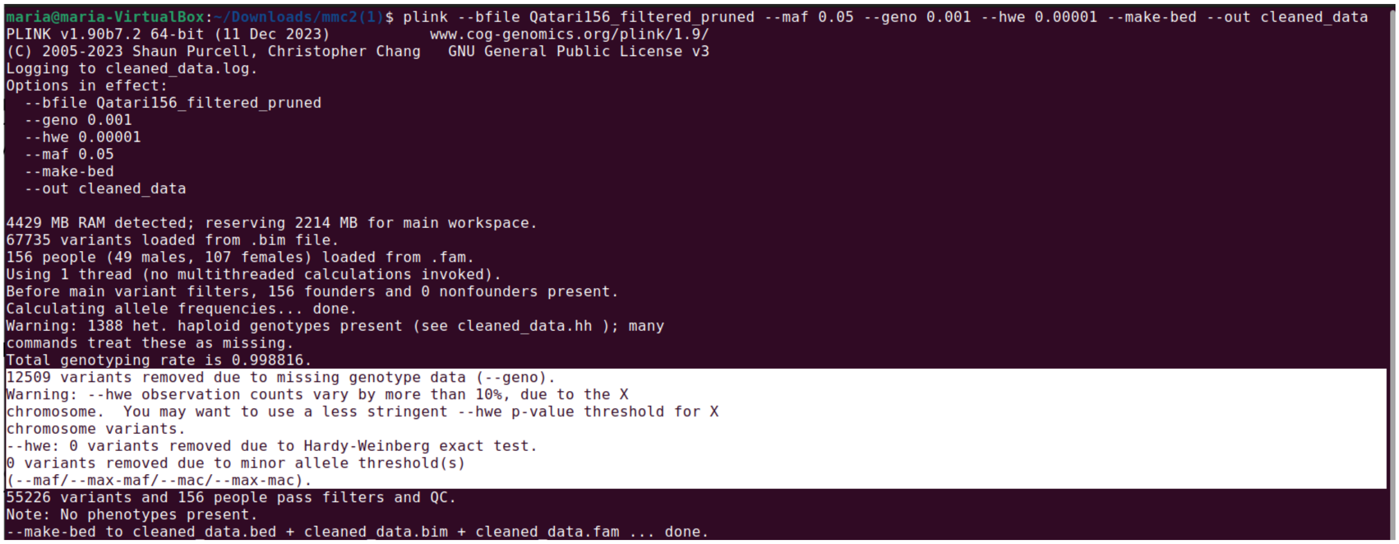
2024-04-15

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.

```
plink --bfile Qatari156_filtered_pruned --maf 0.05 --geno 0.001 --hwe 0.00001 --make-bed --out cleaned_data
```



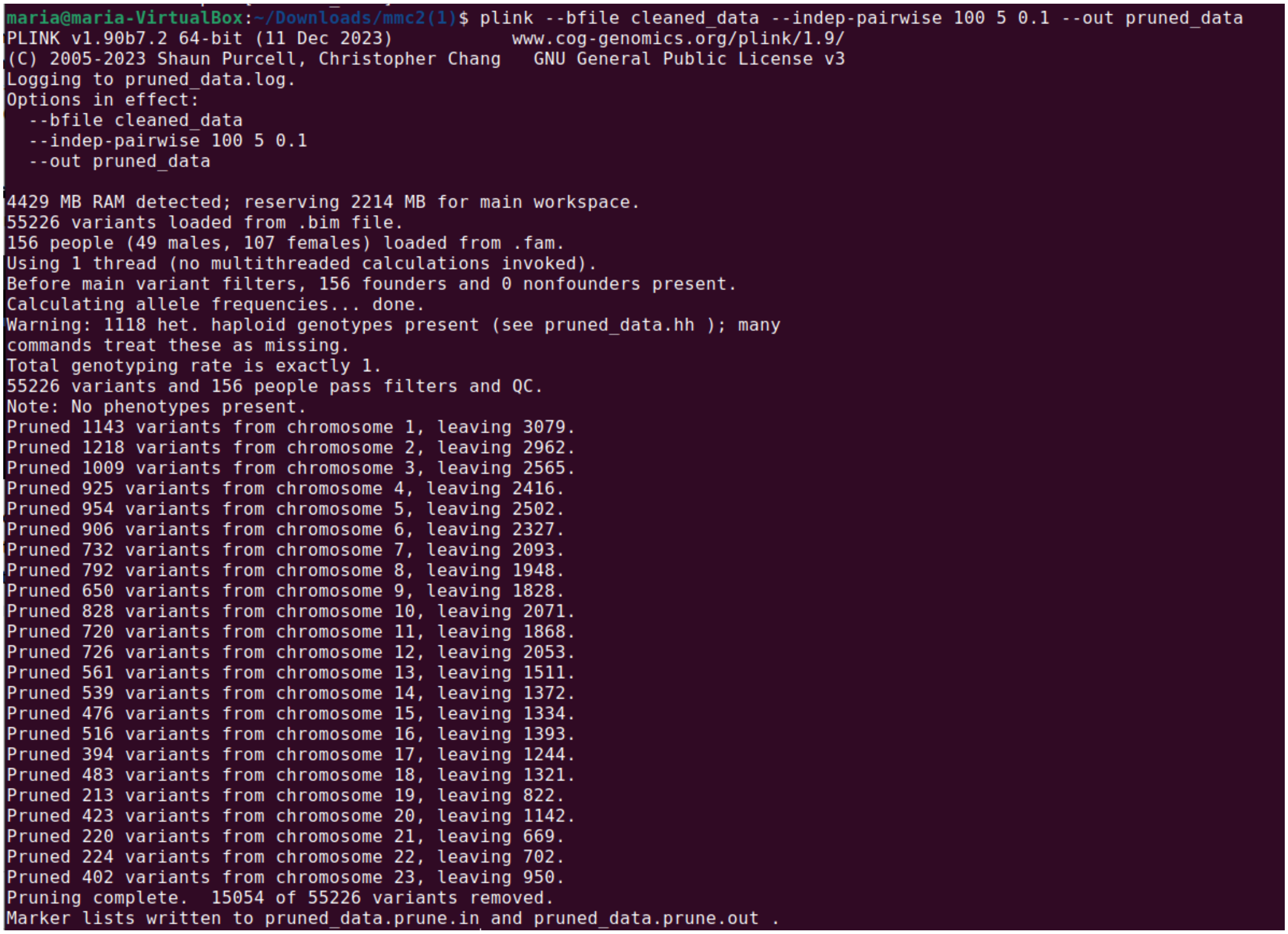
```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --maf 0.05 --geno 0.001 --hwe 0.00001 --make-bed --out cleaned_data
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to cleaned_data.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --geno 0.001
  --hwe 0.00001
  --maf 0.05
  --make-bed
  --out cleaned_data

4429 MB RAM detected; reserving 2214 MB for main workspace.
67735 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1388 het. haploid genotypes present (see cleaned_data.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
12509 variants removed due to missing genotype data (--geno).
Warning: --hwe observation counts vary by more than 10%, due to the X
chromosome.  You may want to use a less stringent --hwe p-value threshold for X
chromosome variants.
--hwe: 0 variants removed due to Hardy-Weinberg exact test.
0 variants removed due to minor allele threshold(s)
(--maf/--max-maf/--mac/--max-mac).
55226 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--make-bed to cleaned_data.bed + cleaned_data.bim + cleaned_data.fam ... done.
```

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
```

`--indep-pairwise 100 5 0.1` : This flag instructs PLINK to perform LD pruning. The three parameters control the pruning process:

- **100**: Specifies the window size (number of variants) for sliding windows across the genome. Variants within this window are considered for LD calculation.
- **5**: Sets the number of variants to shift the window at each step.
- **0.1**: Sets the threshold for LD (Linkage Disequilibrium). Pairs of variants with an r^2 value above this threshold are considered to be in LD and are candidates for removal.



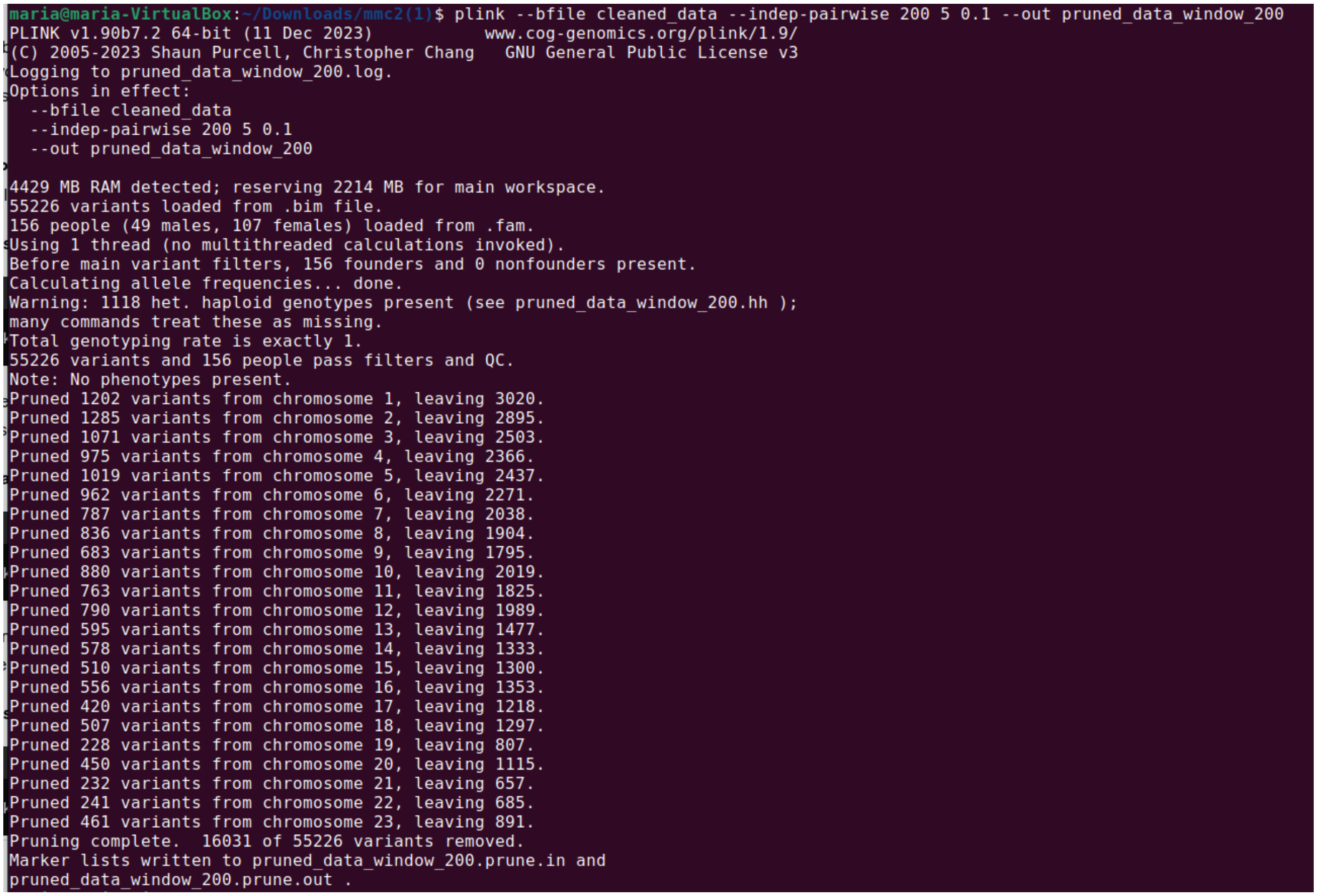
```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile cleaned_data --indep-pairwise 100 5 0.1 --out pruned_data
PLINK v1.90b7.2 64-bit (11 Dec 2023)          www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to pruned_data.log.
Options in effect:
  --bfile cleaned_data
  --indep-pairwise 100 5 0.1
  --out pruned_data

4429 MB RAM detected; reserving 2214 MB for main workspace.
55226 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1118 het. haploid genotypes present (see pruned_data.hh ); many
commands treat these as missing.
Total genotyping rate is exactly 1.
55226 variants and 156 people pass filters and QC.
Note: No phenotypes present.
Pruned 1143 variants from chromosome 1, leaving 3079.
Pruned 1218 variants from chromosome 2, leaving 2962.
Pruned 1009 variants from chromosome 3, leaving 2565.
Pruned 925 variants from chromosome 4, leaving 2416.
Pruned 954 variants from chromosome 5, leaving 2502.
Pruned 906 variants from chromosome 6, leaving 2327.
Pruned 732 variants from chromosome 7, leaving 2093.
Pruned 792 variants from chromosome 8, leaving 1948.
Pruned 650 variants from chromosome 9, leaving 1828.
Pruned 828 variants from chromosome 10, leaving 2071.
Pruned 720 variants from chromosome 11, leaving 1868.
Pruned 726 variants from chromosome 12, leaving 2053.
Pruned 561 variants from chromosome 13, leaving 1511.
Pruned 539 variants from chromosome 14, leaving 1372.
Pruned 476 variants from chromosome 15, leaving 1334.
Pruned 516 variants from chromosome 16, leaving 1393.
Pruned 394 variants from chromosome 17, leaving 1244.
Pruned 483 variants from chromosome 18, leaving 1321.
Pruned 213 variants from chromosome 19, leaving 822.
Pruned 423 variants from chromosome 20, leaving 1142.
Pruned 220 variants from chromosome 21, leaving 669.
Pruned 224 variants from chromosome 22, leaving 702.
Pruned 402 variants from chromosome 23, leaving 950.
Pruning complete.  15054 of 55226 variants removed.
Marker lists written to pruned_data.prune.in and pruned_data.prune.out .
```

Experiment with the thresholds 100, 5, and 0.1 and comment on the effect of increasing or decreasing the threshold.

Window size = 200

```
plink --bfile cleaned_data --indep-pairwise 200 5 0.1 --out pruned_data_window_200
```

By increasing the window size, PLINK considers a larger region for LD pruning. This leads to pruning more variants from chromosomes.

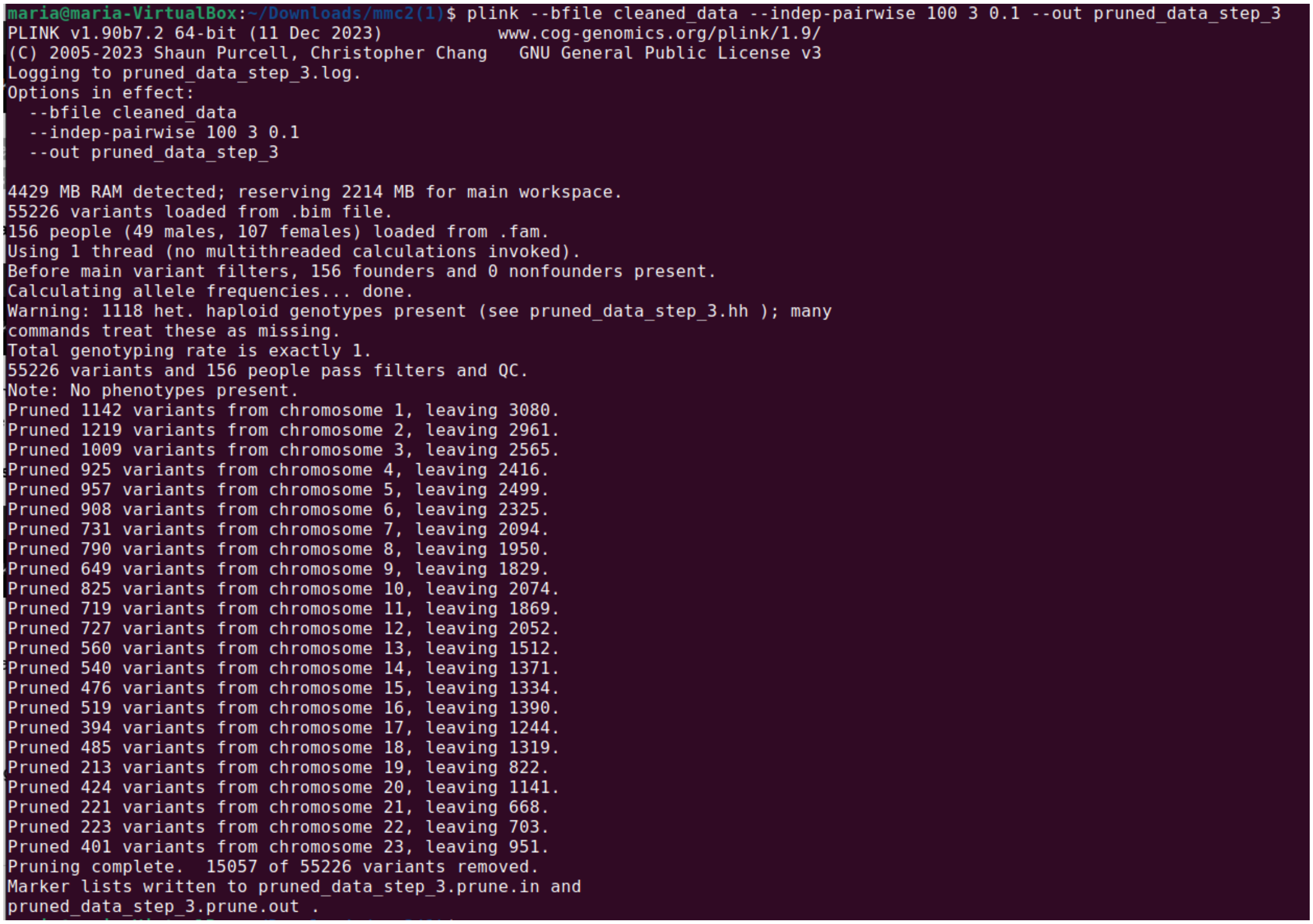


```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile cleaned_data --indep-pairwise 200 5 0.1 --out pruned_data_window_200
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to pruned_data_window_200.log.
Options in effect:
  --bfile cleaned_data
  --indep-pairwise 200 5 0.1
  --out pruned_data_window_200

4429 MB RAM detected; reserving 2214 MB for main workspace.
55226 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1118 het. haploid genotypes present (see pruned_data_window_200.hh );
many commands treat these as missing.
Total genotyping rate is exactly 1.
55226 variants and 156 people pass filters and QC.
Note: No phenotypes present.
Pruned 1202 variants from chromosome 1, leaving 3020.
Pruned 1285 variants from chromosome 2, leaving 2895.
Pruned 1071 variants from chromosome 3, leaving 2503.
Pruned 975 variants from chromosome 4, leaving 2366.
Pruned 1019 variants from chromosome 5, leaving 2437.
Pruned 962 variants from chromosome 6, leaving 2271.
Pruned 787 variants from chromosome 7, leaving 2038.
Pruned 836 variants from chromosome 8, leaving 1904.
Pruned 683 variants from chromosome 9, leaving 1795.
Pruned 880 variants from chromosome 10, leaving 2019.
Pruned 763 variants from chromosome 11, leaving 1825.
Pruned 790 variants from chromosome 12, leaving 1989.
Pruned 595 variants from chromosome 13, leaving 1477.
Pruned 578 variants from chromosome 14, leaving 1333.
Pruned 510 variants from chromosome 15, leaving 1300.
Pruned 556 variants from chromosome 16, leaving 1353.
Pruned 420 variants from chromosome 17, leaving 1218.
Pruned 507 variants from chromosome 18, leaving 1297.
Pruned 228 variants from chromosome 19, leaving 807.
Pruned 450 variants from chromosome 20, leaving 1115.
Pruned 232 variants from chromosome 21, leaving 657.
Pruned 241 variants from chromosome 22, leaving 685.
Pruned 461 variants from chromosome 23, leaving 891.
Pruning complete.  16031 of 55226 variants removed.
Marker lists written to pruned_data_window_200.prune.in and
pruned_data_window_200.prune.out .
```

Step = 3

```
plink --bfile cleaned_data --indep-pairwise 100 3 0.1 --out pruned_data_step_3
```

Reducing the step size allows for capturing more accurate LD (correlations) structures but could increase computational time as a drawback.

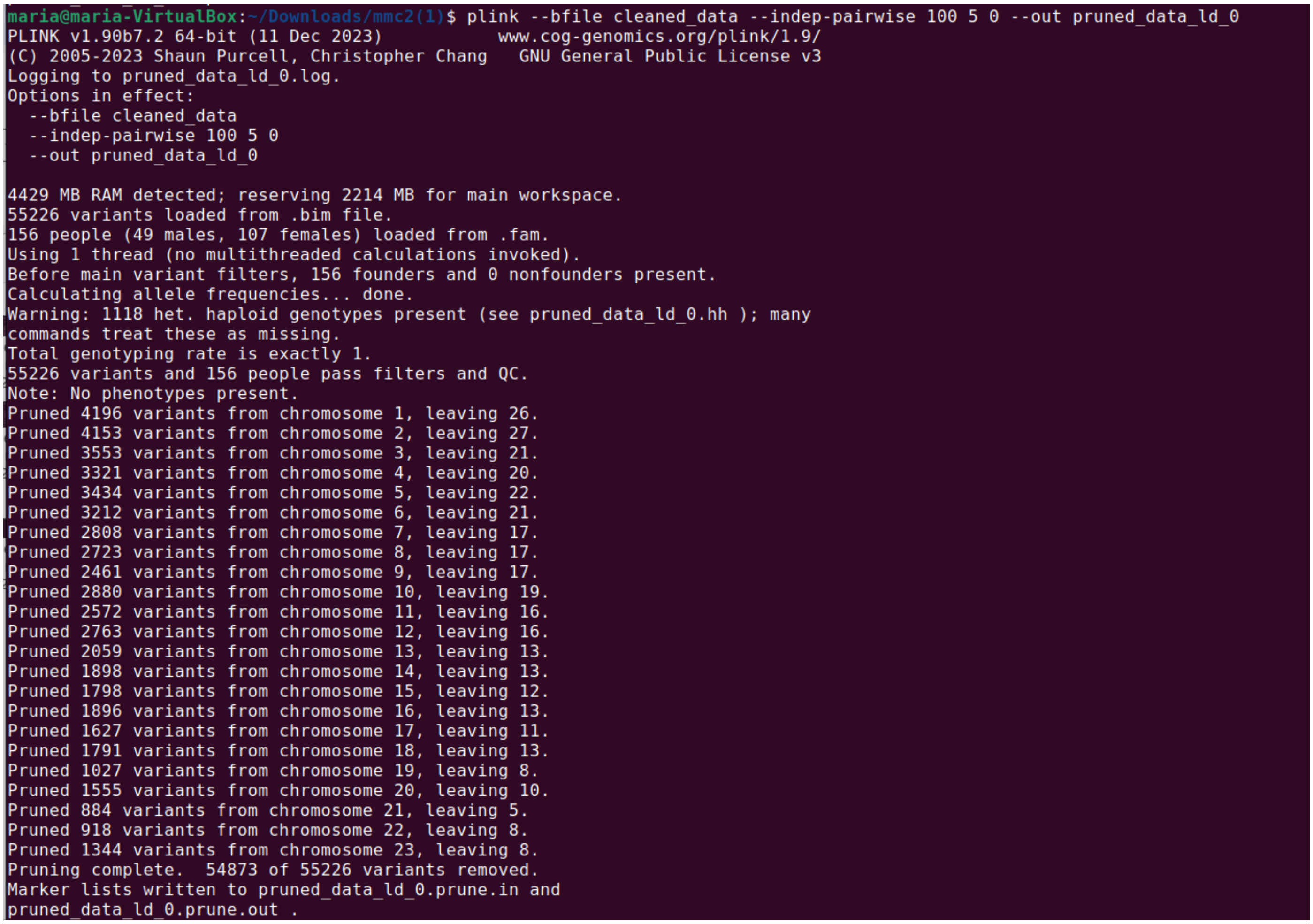


```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile cleaned_data --indep-pairwise 100 3 0.1 --out pruned_data_step_3
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to pruned_data_step_3.log.
Options in effect:
  --bfile cleaned_data
  --indep-pairwise 100 3 0.1
  --out pruned_data_step_3

4429 MB RAM detected; reserving 2214 MB for main workspace.
55226 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1118 het. haploid genotypes present (see pruned_data_step_3.hh ); many
commands treat these as missing.
Total genotyping rate is exactly 1.
55226 variants and 156 people pass filters and QC.
Note: No phenotypes present.
Pruned 1142 variants from chromosome 1, leaving 3080.
Pruned 1219 variants from chromosome 2, leaving 2961.
Pruned 1009 variants from chromosome 3, leaving 2565.
Pruned 925 variants from chromosome 4, leaving 2416.
Pruned 957 variants from chromosome 5, leaving 2499.
Pruned 908 variants from chromosome 6, leaving 2325.
Pruned 731 variants from chromosome 7, leaving 2094.
Pruned 790 variants from chromosome 8, leaving 1950.
Pruned 649 variants from chromosome 9, leaving 1829.
Pruned 825 variants from chromosome 10, leaving 2074.
Pruned 719 variants from chromosome 11, leaving 1869.
Pruned 727 variants from chromosome 12, leaving 2052.
Pruned 560 variants from chromosome 13, leaving 1512.
Pruned 540 variants from chromosome 14, leaving 1371.
Pruned 476 variants from chromosome 15, leaving 1334.
Pruned 519 variants from chromosome 16, leaving 1390.
Pruned 394 variants from chromosome 17, leaving 1244.
Pruned 485 variants from chromosome 18, leaving 1319.
Pruned 213 variants from chromosome 19, leaving 822.
Pruned 424 variants from chromosome 20, leaving 1141.
Pruned 221 variants from chromosome 21, leaving 668.
Pruned 223 variants from chromosome 22, leaving 703.
Pruned 401 variants from chromosome 23, leaving 951.
Pruning complete.  15057 of 55226 variants removed.
Marker lists written to pruned_data_step_3.prune.in and
pruned_data_step_3.prune.out .
```

r^2 value = 0

```
plink --bfile cleaned_data --indep-pairwise 100 5 0 --out pruned_data_ld_0
```

The r^2 value equals 0, this means that we want to prune all the combinations of the SNPs (we want the independent or correlated). In other words, PLINK will retain variants that are not in LD with each other. In other words, variants with r^2 value less than or equal to the 0 will be retained, while variants with r^2 value above 0 will be pruned or removed.

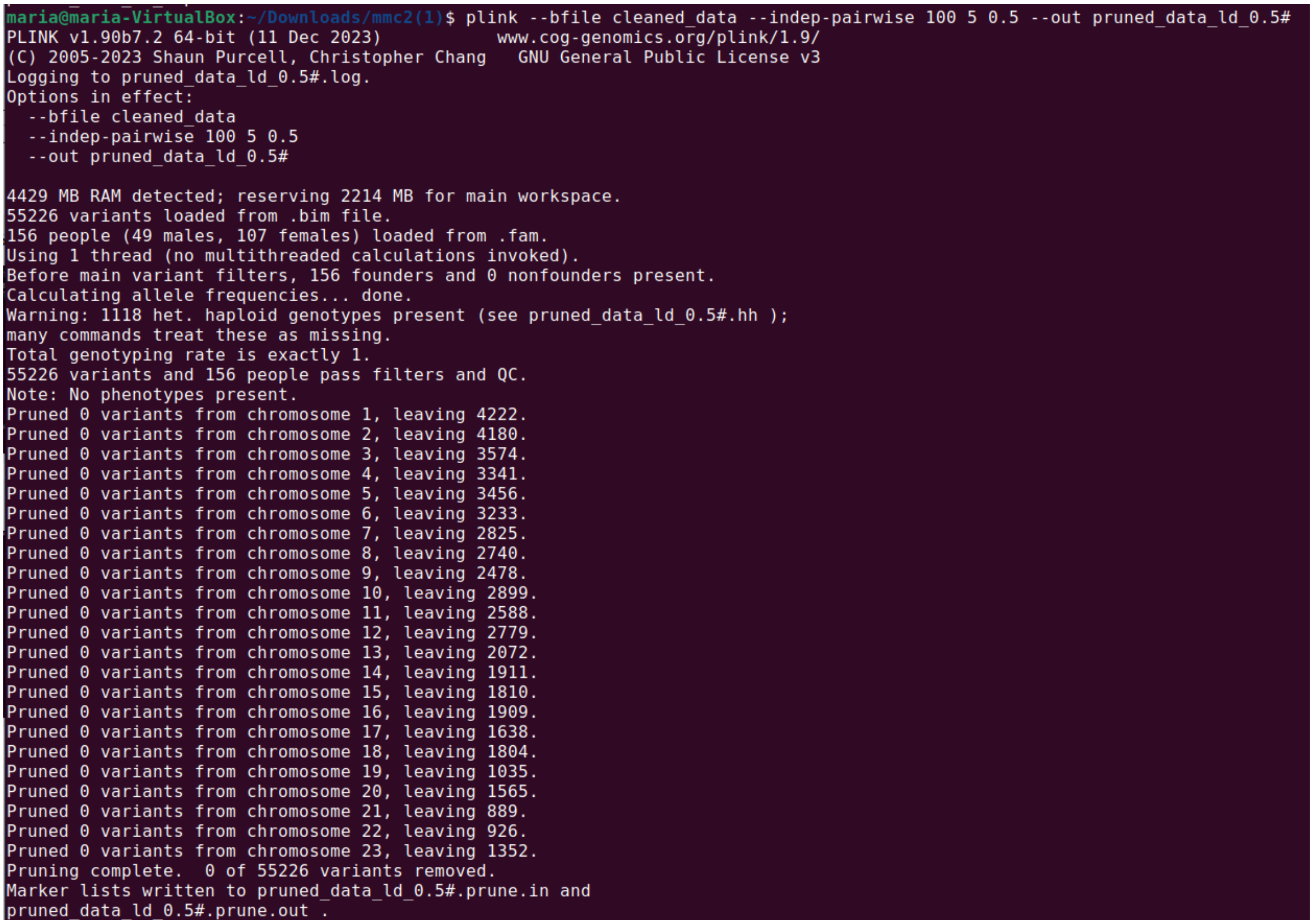


```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile cleaned_data --indep-pairwise 100 5 0 --out pruned_data_ld_0
PLINK v1.90b7.2 64-bit (11 Dec 2023)          www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to pruned_data_ld_0.log.
Options in effect:
  --bfile cleaned_data
  --indep-pairwise 100 5 0
  --out pruned_data_ld_0

4429 MB RAM detected; reserving 2214 MB for main workspace.
55226 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1118 het. haploid genotypes present (see pruned_data_ld_0.hh ); many
commands treat these as missing.
Total genotyping rate is exactly 1.
55226 variants and 156 people pass filters and QC.
Note: No phenotypes present.
Pruned 4196 variants from chromosome 1, leaving 26.
Pruned 4153 variants from chromosome 2, leaving 27.
Pruned 3553 variants from chromosome 3, leaving 21.
Pruned 3321 variants from chromosome 4, leaving 20.
Pruned 3434 variants from chromosome 5, leaving 22.
Pruned 3212 variants from chromosome 6, leaving 21.
Pruned 2808 variants from chromosome 7, leaving 17.
Pruned 2723 variants from chromosome 8, leaving 17.
Pruned 2461 variants from chromosome 9, leaving 17.
Pruned 2880 variants from chromosome 10, leaving 19.
Pruned 2572 variants from chromosome 11, leaving 16.
Pruned 2763 variants from chromosome 12, leaving 16.
Pruned 2059 variants from chromosome 13, leaving 13.
Pruned 1898 variants from chromosome 14, leaving 13.
Pruned 1798 variants from chromosome 15, leaving 12.
Pruned 1896 variants from chromosome 16, leaving 13.
Pruned 1627 variants from chromosome 17, leaving 11.
Pruned 1791 variants from chromosome 18, leaving 13.
Pruned 1027 variants from chromosome 19, leaving 8.
Pruned 1555 variants from chromosome 20, leaving 10.
Pruned 884 variants from chromosome 21, leaving 5.
Pruned 918 variants from chromosome 22, leaving 8.
Pruned 1344 variants from chromosome 23, leaving 8.
Pruning complete.  54873 of 55226 variants removed.
Marker lists written to pruned_data_ld_0.prune.in and
pruned_data_ld_0.prune.out .
```

value = 0.5

```
plink --bfile cleaned_data --indep-pairwise 100 5 0.5 --out pruned_data_ld_0.5
```

As the r^2 value increases, this means that we remove the variants with high LD correlation, we leave the variants with r^2 value >= 0.5. It seams that our data is not highly correlated.



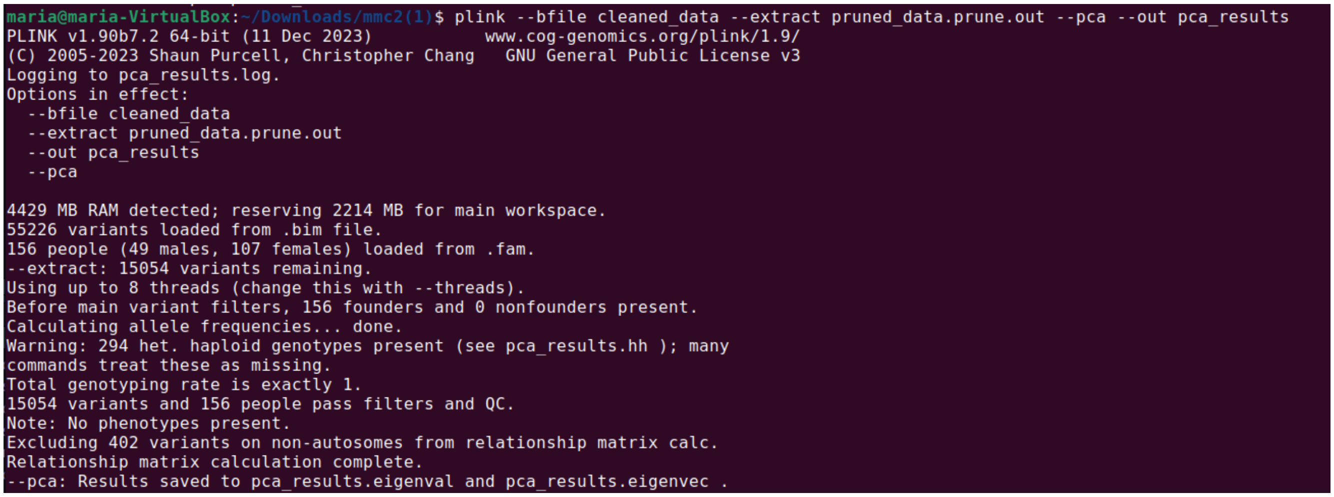
```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile cleaned_data --indep-pairwise 100 5 0.5 --out pruned_data_ld_0.5#
PLINK v1.90b7.2 64-bit (11 Dec 2023)          www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to pruned_data_ld_0.5#.log.
Options in effect:
  --bfile cleaned_data
  --indep-pairwise 100 5 0.5
  --out pruned_data_ld_0.5#

4429 MB RAM detected; reserving 2214 MB for main workspace.
55226 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1118 het. haploid genotypes present (see pruned_data_ld_0.5#.hh );
many commands treat these as missing.
Total genotyping rate is exactly 1.
55226 variants and 156 people pass filters and QC.
Note: No phenotypes present.
Pruned 0 variants from chromosome 1, leaving 4222.
Pruned 0 variants from chromosome 2, leaving 4180.
Pruned 0 variants from chromosome 3, leaving 3574.
Pruned 0 variants from chromosome 4, leaving 3341.
Pruned 0 variants from chromosome 5, leaving 3456.
Pruned 0 variants from chromosome 6, leaving 3233.
Pruned 0 variants from chromosome 7, leaving 2825.
Pruned 0 variants from chromosome 8, leaving 2740.
Pruned 0 variants from chromosome 9, leaving 2478.
Pruned 0 variants from chromosome 10, leaving 2899.
Pruned 0 variants from chromosome 11, leaving 2588.
Pruned 0 variants from chromosome 12, leaving 2779.
Pruned 0 variants from chromosome 13, leaving 2072.
Pruned 0 variants from chromosome 14, leaving 1911.
Pruned 0 variants from chromosome 15, leaving 1810.
Pruned 0 variants from chromosome 16, leaving 1909.
Pruned 0 variants from chromosome 17, leaving 1638.
Pruned 0 variants from chromosome 18, leaving 1804.
Pruned 0 variants from chromosome 19, leaving 1035.
Pruned 0 variants from chromosome 20, leaving 1565.
Pruned 0 variants from chromosome 21, leaving 889.
Pruned 0 variants from chromosome 22, leaving 926.
Pruned 0 variants from chromosome 23, leaving 1352.
Pruning complete.  0 of 55226 variants removed.
Marker lists written to pruned_data_ld_0.5#.prune.in and
pruned_data_ld_0.5#.prune.out .
```

Run PCA on the pruned data using the PLINK and –pca flag

```
plink --bfile cleaned_data --extract pruned_data.prune.out --pca --out pca_results
```

Explanation:

- `--bfile cleaned_data` : Specifies the original input dataset in PLINK binary format (.bed, .bim, .fam) named “cleaned_data”.
- `--extract pruned_data.prune.out` : Specifies the list of SNPs to keep, which is generated by the LD pruning step and saved in the file “pruned_data.prune.out”.
- `--out pca_results` : Specifies the name for the output files.



```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile cleaned_data --extract pruned_data.prune.out --pca --out pca_results
PLINK v1.90b7.2 64-bit (11 Dec 2023)          www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to pca_results.log.
Options in effect:
  --bfile cleaned_data
  --extract pruned_data.prune.out
  --out pca_results
  --pca

4429 MB RAM detected; reserving 2214 MB for main workspace.
55226 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
--extract: 15054 variants remaining.
Using up to 8 threads (change this with --threads).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 294 het. haploid genotypes present (see pca_results.hh ); many
commands treat these as missing.
Total genotyping rate is exactly 1.
15054 variants and 156 people pass filters and QC.
Note: No phenotypes present.
Excluding 402 variants on non-autosomes from relationship matrix calc.
Relationship matrix calculation complete.
--pca: Results saved to pca_results.eigenval and pca_results.eigenvec .
```

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)

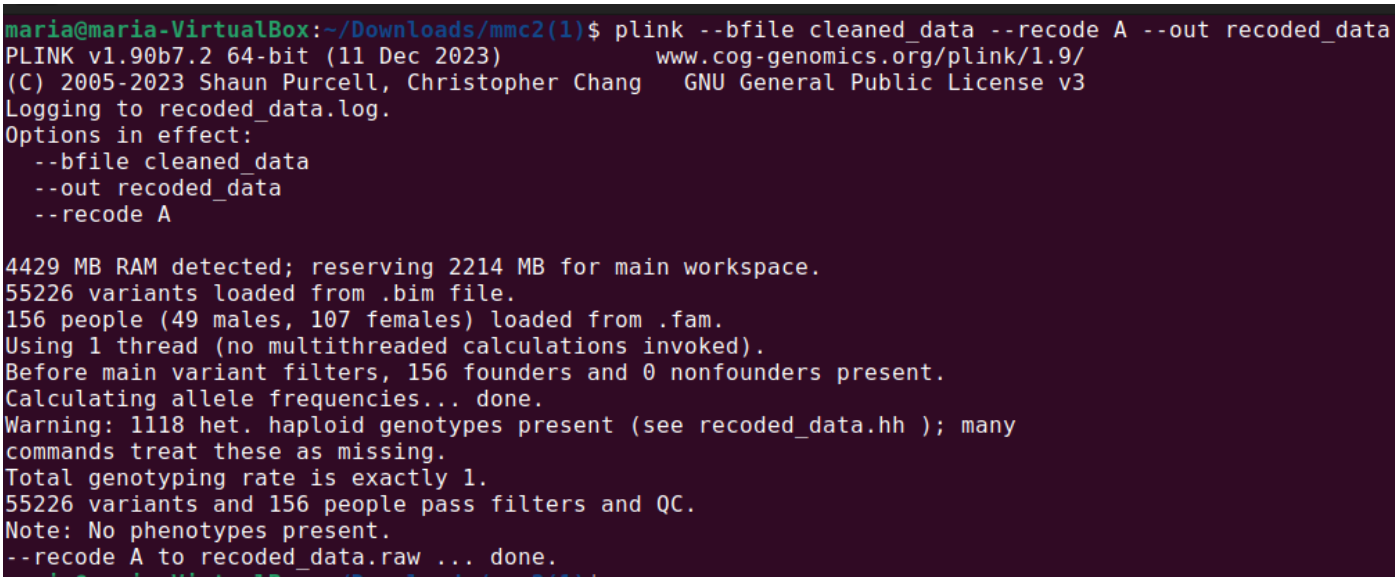
Conduct linear regression analysis for all SNPs in the cleaned data (NOT the pruned data), with the model specified below

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
```

The –recode A option in PLINK generates a text file in the "allele format", which includes an additive component (0/1/2) representing the number of alternate alleles present, without considering dominance. By default, the alleles are counted based on the A1 allele (minor allele). So, in summary, –recode A generates an allele file with an additive (0/1/2) component.



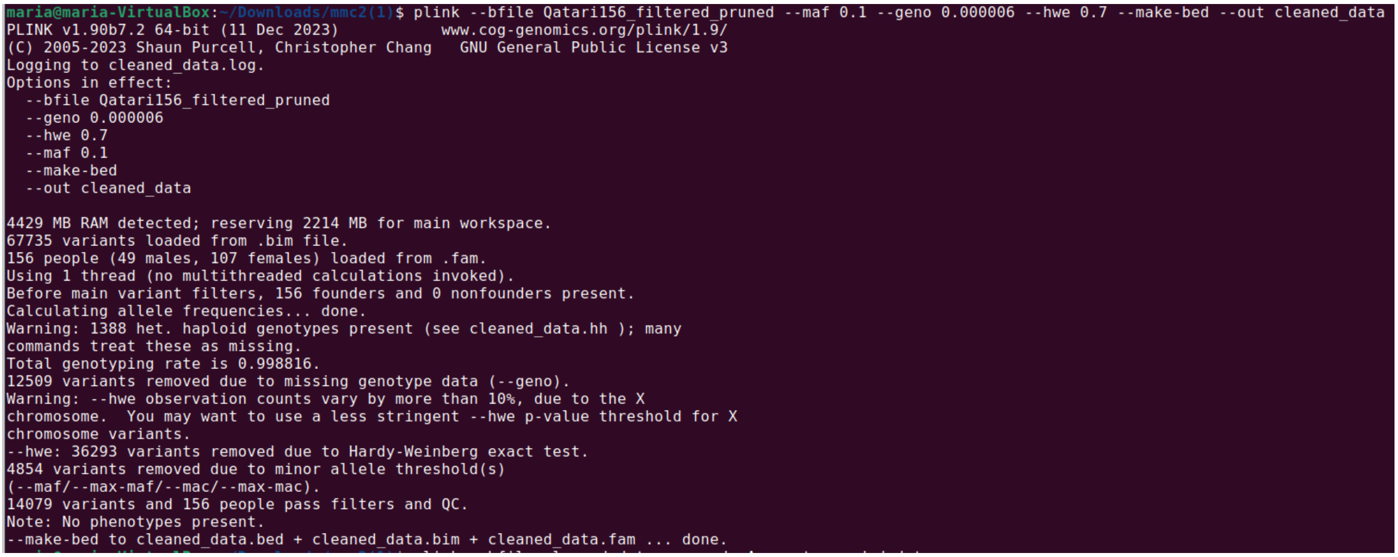
```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile cleaned_data --recode A --out recoded_data
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to recoded_data.log.
Options in effect:
  --bfile cleaned_data
  --out recoded_data
  --recode A

4429 MB RAM detected; reserving 2214 MB for main workspace.
55226 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1118 het. haploid genotypes present (see recoded_data.hh ); many
commands treat these as missing.
Total genotyping rate is exactly 1.
55226 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--recode A to recoded_data.raw ... done.
```

Note:

I cleaned the data further by changing the qc parameters so that the lm and glm can run on my device

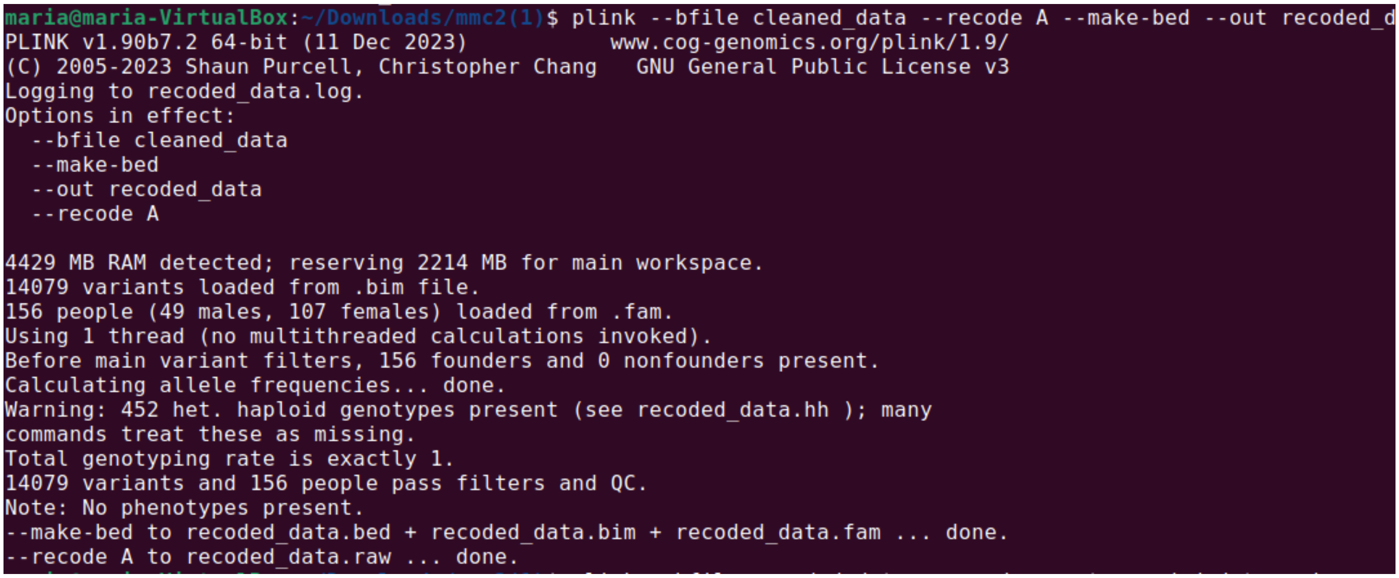
The data is now 14,079 variants.

New cleaned data:



```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile Qatari156_filtered_pruned --maf 0.1 --geno 0.000006 --hwe 0.7 --make-bed --out cleaned_data
PLINK v1.90b7.2 64-bit (11 Dec 2023)           www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to cleaned_data.log.
Options in effect:
  --bfile Qatari156_filtered_pruned
  --geno 0.000006
  --hwe 0.7
  --maf 0.1
  --make-bed
  --out cleaned_data

4429 MB RAM detected; reserving 2214 MB for main workspace.
67735 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 1388 het. haploid genotypes present (see cleaned_data.hh ); many
commands treat these as missing.
Total genotyping rate is 0.998816.
12509 variants removed due to missing genotype data (--geno).
Warning: --hwe observation counts vary by more than 10%, due to the X
chromosome.  You may want to use a less stringent --hwe p-value threshold for X
chromosome variants.
--hwe: 36293 variants removed due to Hardy-Weinberg exact test.
4854 variants removed due to minor allele threshold(s)
(--maf/--max-maf/--mac/--max-mac).
14079 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--make-bed to cleaned_data.bed + cleaned_data.bim + cleaned_data.fam ... done.
```

recoded data from the new cleaned data:



```
maria@maria-VirtualBox:~/Downloads/mmc2(1)$ plink --bfile cleaned_data --recode A --make-bed --out recoded_d
PLINK v1.90b7.2 64-bit (11 Dec 2023)          www.cog-genomics.org/plink/1.9/
(C) 2005-2023 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to recoded_data.log.
Options in effect:
  --bfile cleaned_data
  --make-bed
  --out recoded_data
  --recode A

4429 MB RAM detected; reserving 2214 MB for main workspace.
14079 variants loaded from .bim file.
156 people (49 males, 107 females) loaded from .fam.
Using 1 thread (no multithreaded calculations invoked).
Before main variant filters, 156 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Warning: 452 het. haploid genotypes present (see recoded_data.hh ); many
commands treat these as missing.
Total genotyping rate is exactly 1.
14079 variants and 156 people pass filters and QC.
Note: No phenotypes present.
--make-bed to recoded_data.bed + recoded_data.bim + recoded_data.fam ... done.
--recode A to recoded_data.raw ... done.
```

- Read the .raw file into R (use the function read.table and specify header=T and sep="")

```
# Set the file path
file_path <- "D:\\Third Year Computer\\Term 2\\Bio\\Labs\\Lab 6\\mmc2\\recoded_data.raw"

# Read the .raw file into R
data <- read.table(file_path, header = TRUE, sep = "")

head(data[, 1:6])
```

```
##        FID      IID PAT MAT SEX PHENOTYPE
## 1  QBC-092  QBC-092   0   0   2        -9
## 2  QBC-256  QBC-256   0   0   2        -9
## 3  QBC-107  QBC-107   0   0   1        -9
## 4  QBC-171  QBC-171   0   0   2        -9
## 5 QPRC-110 QPRC-110   0   0   1        -9
## 6  QBC-240  QBC-240   0   0   2        -9
```

- 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

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
snp_data <- data %>%
  select(1:2, 7:ncol(data))
```

- Read the PCA eigenvalues and eigenvectors into R (use the function read.table)

```
pca_eigenval <- read.table("D:\\Third Year Computer\\Term 2\\Bio\\Labs\\Lab 6\\mmc2\\pca_results.eigenval", header =
FALSE, sep = "")


pca_eigenvec <- read.table("D:\\Third Year Computer\\Term 2\\Bio\\Labs\\Lab 6\\mmc2\\pca_results.eigenvec", header =
FALSE, sep = "")
colnames(pca_eigenvec) <- c("Family ID", "Sample ID", paste0("PC", 1:(ncol(pca_eigenvec) - 2)))

head(pca_eigenval)
```

```
##        V1
## 1 7.66770
## 2 2.53658
## 3 1.71423
## 4 1.70743
## 5 1.55992
## 6 1.48598
```

```
head(pca_eigenvec)
```

```
##   Family ID Sample ID        PC1        PC2        PC3         PC4         PC5
## 1   QBC-092   QBC-092  0.0188404  0.0411612 -0.0178717  0.00909137 -0.00500099
## 2   QBC-256   QBC-256 -0.0370551 -0.0150276  0.0072131 -0.06500340 -0.01878410
## 3   QBC-107   QBC-107 -0.0320635 -0.0204935 -0.0243070 -0.07399310 -0.00268545
## 4   QBC-171   QBC-171 -0.0218557  0.0558842  0.0216805  0.13059100 -0.01196000
## 5  QPRC-110  QPRC-110 -0.0309817  0.1118350  0.0273144  0.00858629 -0.01806050
## 6   QBC-240   QBC-240  0.0413564  0.0933106  0.0118433  0.08434680 -0.05299150
##          PC6        PC7          PC8         PC9        PC10       PC11
## 1 -0.0327082  0.0383056 -0.047521000 -0.06906100 -0.07820240  0.0775551
## 2 -0.0322392  0.0161855  0.006354480  0.02609540  0.03317180 -0.0124379
## 3 -0.0133878  0.0155483 -0.000737125  0.01557380  0.00921206 -0.0338062
## 4 -0.0214334 -0.0377511 -0.025017500  0.00449141  0.01462260 -0.0479673
## 5 -0.0106281  0.0288726  0.006823710 -0.02585070 -0.05990910  0.0498506
## 6 -0.0321668 -0.1128870 -0.025731800  0.10665200  0.06310830 -0.0714728
##           PC12        PC13       PC14        PC15        PC16        PC17
## 1 -0.025070300 -0.05279200  0.0623865 -0.00121441 -0.02427060  0.00257891
## 2 -0.009533210  0.05900730  0.0285143 -0.02355800  0.02985420 -0.01710890
## 3  0.022178300  0.06315780  0.0237664  0.03923320 -0.00099609  0.01228950
## 4  0.000125998  0.00362886  0.0235296 -0.00299675  0.00920752 -0.00649560
## 5 -0.040026700 -0.03375490 -0.0365536  0.01994480  0.01591700  0.00604240
## 6  0.019416500  0.11996400  0.1417390 -0.08521060 -0.10247600 -0.07579640
##         PC18         PC19       PC20
## 1  0.0254193 -0.000150478  0.0278649
## 2 -0.0162090  0.004360950 -0.0300080
## 3  0.0233020 -0.029529200 -0.0311493
## 4 -0.0774288  0.015926100 -0.0682076
## 5  0.0214078 -0.002661630  0.0220326
## 6 -0.0542272 -0.133373000 -0.1085900
```

- 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.

```
# initialize dataset
data_set <- cbind(pca_eigenvec[, c("PC1", "PC2", "PC3")], snp_data[,3:ncol(snp_data)])
head(data_set[, 1:4])
```

```
##          PC1        PC2        PC3 rs7513222_A
## 1  0.0188404  0.0411612 -0.0178717           1
## 2 -0.0370551 -0.0150276  0.0072131           1
## 3 -0.0320635 -0.0204935 -0.0243070           0
## 4 -0.0218557  0.0558842  0.0216805           0
## 5 -0.0309817  0.1118350  0.0273144           0
## 6  0.0413564  0.0933106  0.0118433           2
```

```
# Initialize an empty data frame to store coefficients
sig_assoc_lm <- data.frame(pc = character(), rsid = character(), beta = numeric(), std_error = numeric(), p_value =
numeric())

# Initialize a data frame to store p-values and SNP columns
pvalues_snp_df_lm <- data.frame(pc = character(), rsid = character(), p_value = numeric())

# Bonferroni correction = 0.05/#tests
bonferroni_threshold <- 0.05 / (ncol(snp_data) - 2) # ((nrow(snp_data) - 2) * 3)

for(pc in paste0("PC", 1:3)){
  for(snp_x in names(snp_data)[-(1:2)]){
    formula <- paste(pc, "~", snp_x, "+", paste(setdiff(c("PC1", "PC2", "PC3"), pc), collapse = " + "))
    lm_result <- lm(as.formula(formula), data = data_set)
    p_value <- summary(lm_result)$coefficients[2, "Pr(>|t|)"]
    # to remove the _T for example
    snp_removed <- gsub("_.*", "", snp_x)
    pvalues_snp_df_lm <- rbind(pvalues_snp_df_lm, data.frame(pc = pc, rsid = snp_removed, p_value = p_value))

    if(p_value < bonferroni_threshold){
      sig_assoc_lm <- rbind(sig_assoc_lm, data.frame(pc = pc, rsid = snp_removed, beta = summary(lm_result)$coeffici
ents[2, "Estimate"], std_error = summary(lm_result)$coefficients[2, "Std. Error"], p_value = p_value))
      # cat(pc, ", SNP = ", snp_x, ", p_value = ", p_value, "\n")
    }
  }
}
```

```
head(pvalues_snp_df_lm)
```

```
##    pc       rsid    p_value
## 1 PC1  rs7513222 0.23072483
## 2 PC1  rs6424068 0.22673306
## 3 PC1  rs2493277 0.89845707
## 4 PC1 rs17379833 0.06198946
## 5 PC1  rs4400585 0.25209245
## 6 PC1  rs9426494 0.40971578
```

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

This is a method to correct the significance threshold for multiple hypothesis testing.

```
bonferroni_threshold <- 0.05 / ((ncol(snp_data) - 2) * 3)
```

```
top_10_pcs_sig_snps <- sig_assoc_lm %>%
  group_by(pc) %>%
  slice_min(order_by = p_value, n = 10)
head(top_10_pcs_sig_snps)
```

```
## # A tibble: 6 × 5
## # Groups:   pc [1]
##   pc    rsid         beta std_error  p_value
##   <chr> <chr>       <dbl>     <dbl>    <dbl>
## 1 PC1   rs12613530 0.120     0.0119 9.01e-19
## 2 PC1   rs213037   0.0954    0.0103 1.73e-16
## 3 PC1   rs6850796  0.0893    0.0114 6.92e-13
## 4 PC1   rs359428   0.0916    0.0120 2.44e-12
## 5 PC1   rs3181360  0.0936    0.0123 3.12e-12
## 6 PC1   rs235679   0.0892    0.0118 3.17e-12
```

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:

```
#install.packages("caret")
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
## Loading required package: lattice
```

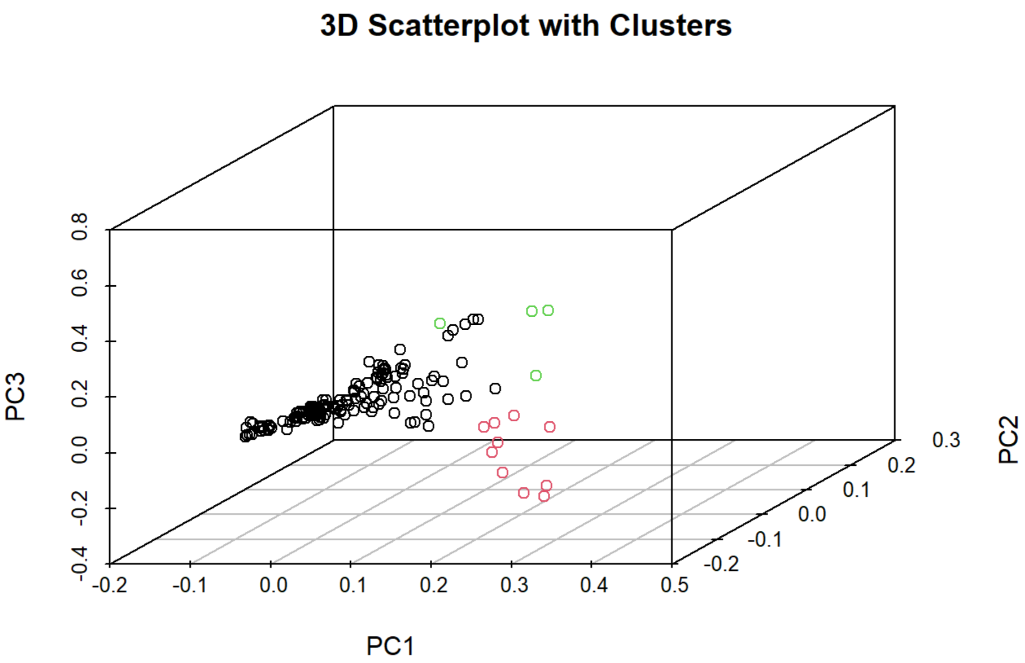
```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(scatterplot3d)
```

- 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)

```
set.seed(1)  # Set seed for reproducibility
k <- 3  # Number of clusters
nstart <- 25  # Set nstart

cluster_result <- kmeans(pca_eigenvec[, c("PC1", "PC2", "PC3")], centers = k, nstart = nstart)

# Plot 3D scatterplot with cluster colors
scatterplot3d(pca_eigenvec$PC1, pca_eigenvec$PC2, pca_eigenvec$PC3, color = cluster_result$cluster, xlab = "PC1", yl
ab = "PC2", zlab = "PC3", main = "3D Scatterplot with Clusters")
```



- 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.

```
# factor() converts the vector into a factor variable. The function assigns a unique numeric value to each distinct level or category present in the vector, and these numeric values are used internally to represent the categories. It also shows the levels {1, 2, 3}
cluster_labels <- factor(cluster_result$cluster)
# dummyVars() --> separate each factor into its own column.
cluster_dummy <- predict(dummyVars(~., data = data.frame(cluster = cluster_labels)), newdata = data.frame(cluster = cluster_labels))

cluster_df <- cbind(pca_eigenvec[, c("PC1", "PC2", "PC3")], cluster_dummy)
head(cluster_df)
```

```
##          PC1        PC2        PC3 cluster.1 cluster.2 cluster.3
## 1  0.0188404  0.0411612 -0.0178717         1         0         0
## 2 -0.0370551 -0.0150276  0.0072131         1         0         0
## 3 -0.0320635 -0.0204935 -0.0243070         1         0         0
## 4 -0.0218557  0.0558842  0.0216805         1         0         0
## 5 -0.0309817  0.1118350  0.0273144         1         0         0
## 6  0.0413564  0.0933106  0.0118433         1         0         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.

```
# Initialize an empty data frame to store coefficients
sig_assoc_glm <- data.frame(cluster = character(), rsid = character(), beta = numeric(), std_error = numeric(), p_va
lue = numeric())

pvalues_snp_df_glm <- data.frame(cluster = character(), rsid = character(), p_value = numeric())
# Bonferroni threshold
bonferroni_threshold <- 0.05 / (nrow(snp_data) - 2) # ((nrow(snp_data) - 2) * 3)
# Initialize the data called cluster_data
cluster_data <- cbind(cluster_df, snp_data)

for(cl in 1:k){
  for(snp_column in names(snp_data)[-(1:2)]){
    formula <- paste(paste0("cluster.", cl), "~", snp_column)

    logistic_model <- glm(as.formula(formula), family = binomial, data = cluster_data)
    p_value <- summary(logistic_model)$coefficients[2, "Pr(>|z|)"]

    snp_removed <- gsub("_.*", "", snp_column)
    pvalues_snp_df_glm <- rbind(pvalues_snp_df_glm, data.frame(cluster = paste0("Cluster.", cl), rsid = snp_removed,
p_value = p_value))

    if(p_value < bonferroni_threshold){
      sig_assoc_glm <- rbind(sig_assoc_glm, data.frame(cluster = cl, rsid = snp_removed, beta = summary(logistic_mod
el)$coefficients[2, "Estimate"], std_error = summary(logistic_model)$coefficients[2, "Std. Error"], p_value = p_valu
e))
      # cat("cluster.", cl, ", SNP = ", snp_column, ", p_value = ", p_value, "\n")
    }
  }
}
```

```
head(pvalues_snp_df_glm)
```

```
##     cluster       rsid   p_value
## 1 Cluster.1  rs7513222 0.1389951
## 2 Cluster.1  rs6424068 0.1367824
## 3 Cluster.1  rs2493277 0.2008642
## 4 Cluster.1 rs17379833 0.4917345
## 5 Cluster.1  rs4400585 0.3042808
## 6 Cluster.1  rs9426494 0.9418573
```

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

```
top_10_clusters_sig_snps <- sig_assoc_glm %>%
  group_by(cluster) %>%
  slice_min(order_by = p_value, n = 10)
head(top_10_clusters_sig_snps)
```

```
## # A tibble: 6 × 5
## # Groups:   cluster [1]
##   cluster rsid       beta std_error    p_value
##     <int> <chr>     <dbl>     <dbl>      <dbl>
## 1       1 rs1595361 -3.38     0.707 0.00000173
## 2       1 rs5956063 -2.19     0.465 0.00000255
## 3       1 rs3787397 -2.97     0.647 0.00000429
## 4       1 rs7525142 -2.81     0.620 0.00000607
## 5       1 rs6845502 -3.07     0.685 0.00000771
## 6       1 rs230103  -3.07     0.685 0.00000771
```

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

Plot a Manhattan plot using the qqman 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.

```
library(tidyverse)

# Set the file path
map_file_path <- "D:\\Third Year Computer\\Term 2\\Bio\\Labs\\Lab 6\\mmc2\\recoded_data_ped_map.map"

# Read the .raw file into R
map_data <- read.table(map_file_path, header = FALSE)
colnames(map_data) <- c("CHR", "rsid", "GD", "BP")

# Select only the required columns
map_data <- map_data[, c("rsid", "CHR", "BP")]

# Merge p-values data frame from pcs with map data based on rsid
merged_data_lm <- merge(map_data, pvalues_snp_df_lm, by = "rsid", all.x = TRUE)
# Display the first few rows of the merged data
head(merged_data_lm)
```

```
##         rsid CHR        BP   pc      p_value
## 1 rs10000748   4 129301941 PC2 0.0370419188
## 2 rs10000748   4 129301941 PC3 0.6748886541
## 3 rs10000748   4 129301941 PC1 0.0002509993
## 4 rs10000864   4 187936417 PC3 0.4907317627
## 5 rs10000864   4 187936417 PC2 0.4277846089
## 6 rs10000864   4 187936417 PC1 0.4146908446
```

```
# Merge p-values data frame from clusters with map data based on rsid
merged_data_glm <- merge(map_data, pvalues_snp_df_glm, by = "rsid", all.x = TRUE)
# Display the first few rows of the merged data
head(merged_data_glm)
```

```
##         rsid CHR        BP   cluster   p_value
## 1 rs10000748   4 129301941 Cluster.2 0.9903799
## 2 rs10000748   4 129301941 Cluster.3 0.9941761
## 3 rs10000748   4 129301941 Cluster.1 0.9901424
## 4 rs10000864   4 187936417 Cluster.3 0.2954940
## 5 rs10000864   4 187936417 Cluster.2 0.8972137
## 6 rs10000864   4 187936417 Cluster.1 0.4825363
```

Manhattan For PCs

```
# install.packages("qqman")
library(qqman)
```

```
## Warning: package 'qqman' was built under R version 4.3.3
```

```
##
```

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

```
##
```

```
## Citation appreciated but not required:
```

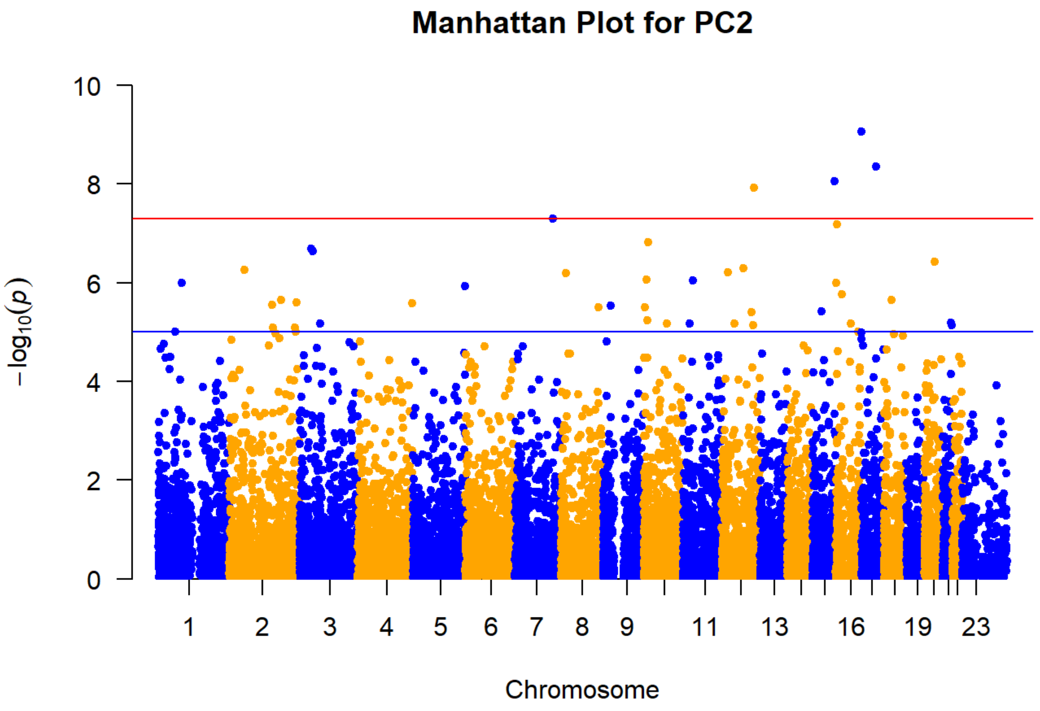
```
## Turner, (2018). qqman: an R package for visualizing GWAS results using Q-Q and manhattan plots. Journal of Open S
ource Software, 3(25), 731, https://doi.org/10.21105/joss.00731.
```

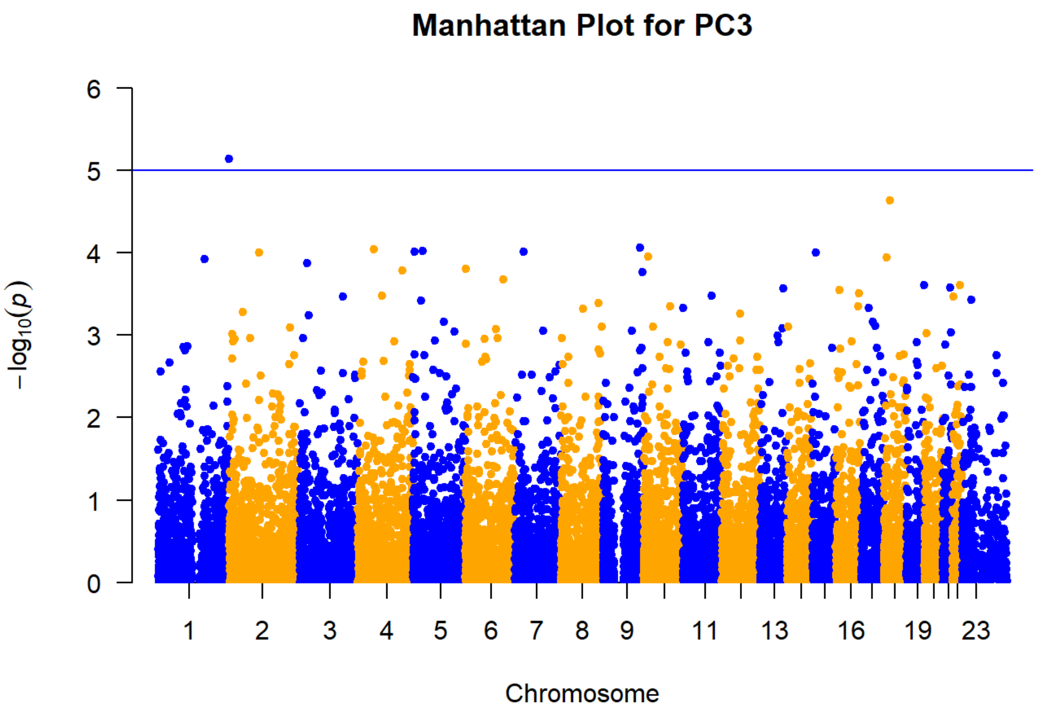
```
##
```

```
##
## Attaching package: 'qqman'
```

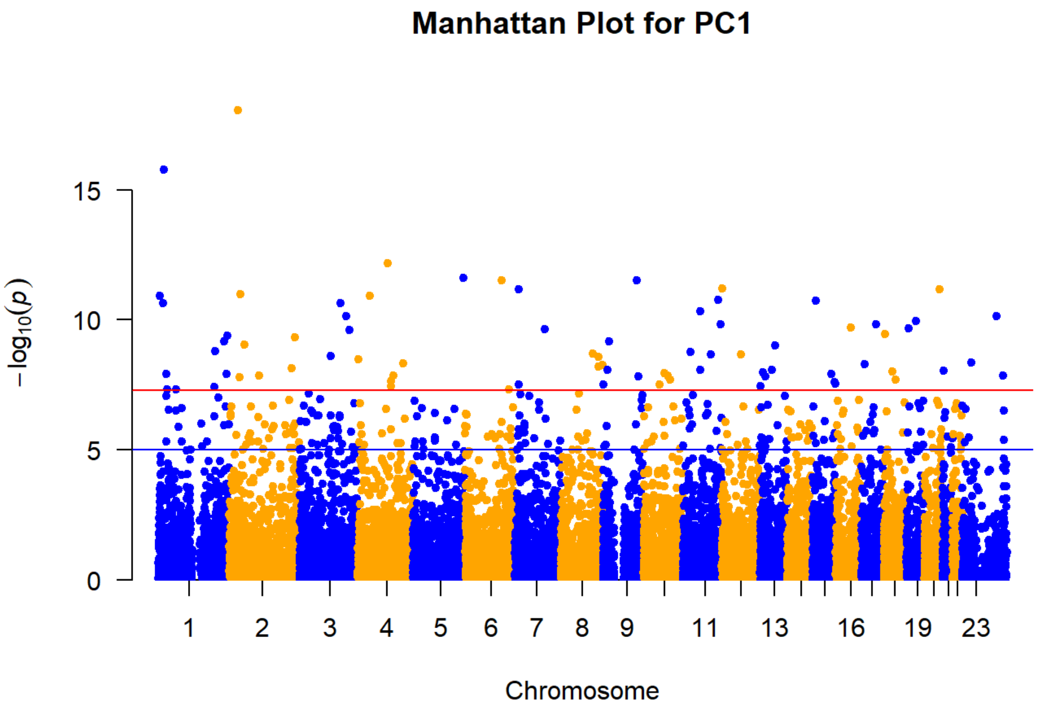
```
## The following object is masked from 'package:lattice':
##
##     qq
```

```
# Unique PCs
unique_pcs <- unique(merged_data_lm$pc)

# Plot Manhattan plot for each PC
for(pc in unique_pcs) {
  # Subset data for the current PC
  pc_data <- merged_data_lm[merged_data_lm$pc == pc, ]

  main_title <- paste("Manhattan Plot for", pc)
  # Plot the Manhattan plot
  manhattan(pc_data, chr="CHR", bp="BP", snp="rsid", p="p_value", col = c("blue", "orange"), main = main_title)
}
```





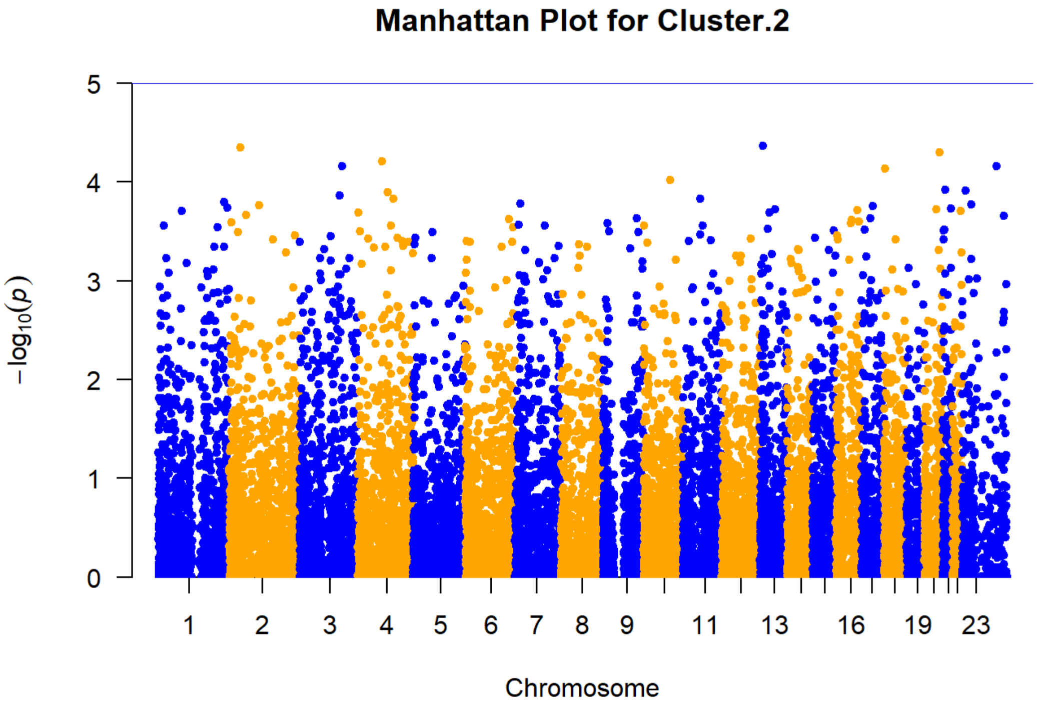
Manhattan Plot for PC3

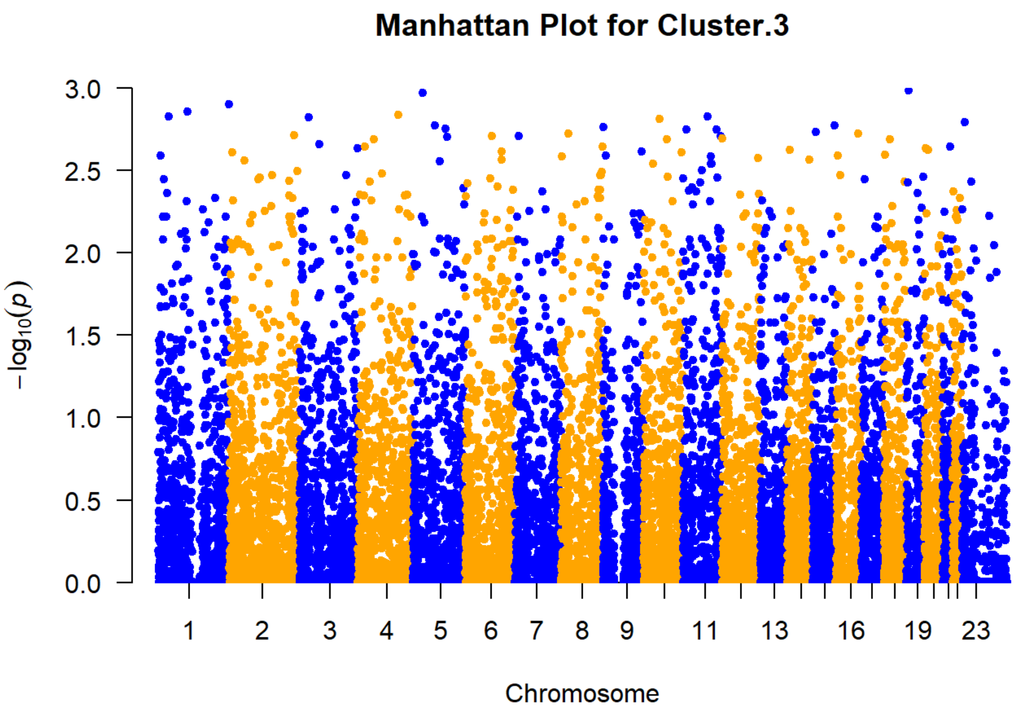


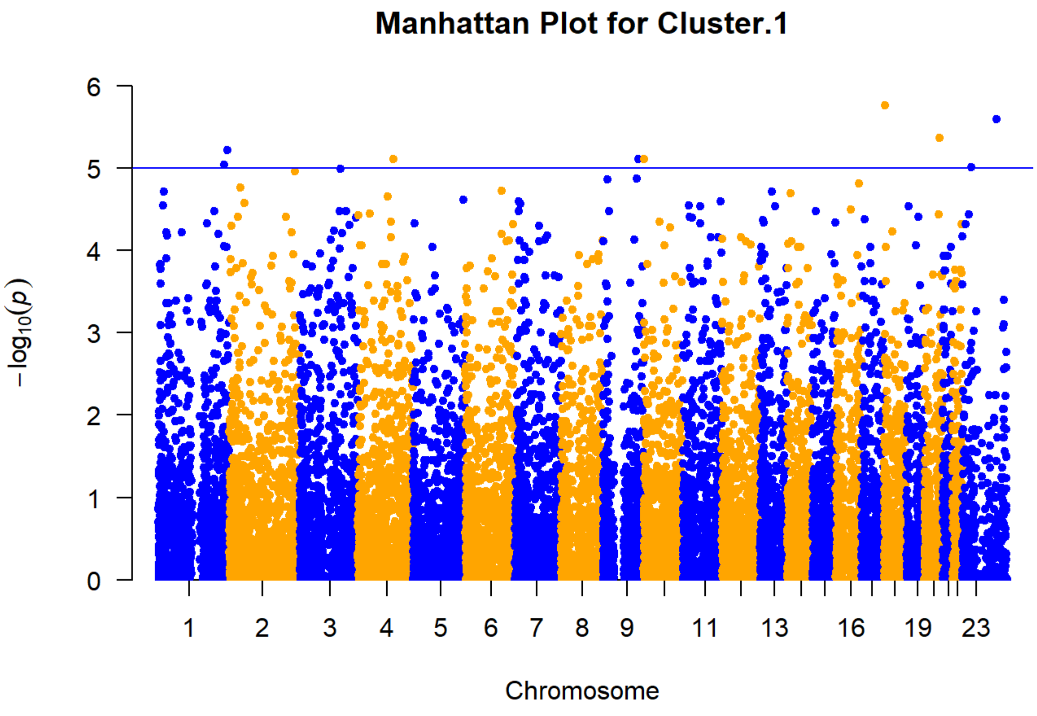
Manhattan Plot for PC1

Manhattan For Clusters

```r
# Unique clusters
unique_clusters <- unique(merged_data_glm$cluster)
# Plot Manhattan plot for each PC
for(cl in unique_clusters) {
  # Subset data for the current PC
  cluster_data <- merged_data_glm[merged_data_glm$cluster == cl, ]

  main_title <- paste("Manhattan Plot for", cl)
  # Plot the Manhattan plot
  manhattan(cluster_data, chr="CHR", bp="BP", snp="rsid", p="p_value", col = c("blue", "orange"), main = main_title)
}
```







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

```
top_10_merged_sig_lm <- merge(top_10_pcs_sig_snps, map_data, by = "rsid", all.x = TRUE)

top_10_merged_sig_lm <- top_10_merged_sig_lm %>%
  arrange(pc)

# # Display the first few rows of the merged data
# head(top_10_merged_sig_lm)

## ---------------- separator ----------------

top_10_merged_sig_glm <- merge(top_10_clusters_sig_snps, map_data, by = "rsid", all.x = TRUE)

top_10_merged_sig_glm <- top_10_merged_sig_glm %>%
  arrange(cluster)
#
# # Display the first few rows of the merged data
# head(top_10_merged_sig_glm)
```

PC1:

SNP	CHR	BP	Beta	P-Val	SE	Gene	Freq	Max Freq	Min Freq
rs12613530	2	31307461	0.12028175	9.005544e-19	0.011864217	CAPN14	T=0.058562/1846 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs12613530#frequency_tab) T=0.010491/47 Estonian T=0.014028/14 GoNL	0.058562	0.010491
rs213037	1	21528471	0.09544822	1.729533e-16	0.010297216	ECE1	A=0.098461/20094 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs213037#frequency_tab) A=0.037037/2 PRJEB36033 A=0.061667/37 NorthernSweden	0.098461	0.037037
rs235679	6	124304070	0.08930758	6.923165e-13	0.011766085	NKAIN2	C=0.012011/237 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs235679#frequency_tab) A=0.002396/7 KOREAN A=0.002725/46 TOMMO	0.012011	0.002396
rs3181360	9	116731379	0.09159005	2.437532e-12	0.012342818	TNFSF8	T=0.074084/24101 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs3181360#frequency_tab) T=0./0 PRJEB36033 T=0.051402/11 Vietnamese	0.074084	0

SNP	CHR	BP	Beta	P-Val	SE	Gene	Freq	Max Freq	Min Freq
rs359428	5	173190227	0.09359099	3.123404e-12	0.012008814	-	A=0.129774/3330 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs359428#frequency_tab) T=0./0 KOREAN A=0./0 Korea1K	0.129774	0
rs3787397	20	52700084	0.08918383	3.174304e-12	0.00893237	DOK5	A=0.33283/51972 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs3787397#frequency_tab) A=0.255591/160 Chileans G=0.261905/11 Siberian	0.33283	0.255591
rs4670713	2	37507459	0.08438566	6.318294e-12	0.009006862	-	C=0.244312/8354 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs4670713#frequency_tab) C=0.171667/103 NorthernSweden C=0.193825/747 ALSPAC	0.244312	0.171667
rs6850796	4	102624773	0.06652043	6.645545e-12	0.011376648	-	C=0.097767/2390 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs6850796#frequency_tab) C=0.004687/21 Estonian C=0.006012/6 GoNL	0.097767	0.004687
rs6967342	7	12318735	0.08582178	6.949173e-12	0.011536629	-	G=0.029891/569 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs6967342#frequency_tab) G=0.011607/52 Estonian G=0.011667/7 NorthernSweden	0.029891	0.011607
rs7957163	12	2675893	0.06637853	1.019880e-11	0.011317492	-	A=0.071777/10732 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs7957163#frequency_tab) A=0./0 PRJEB36033 A=0.025/1 GENOME_DK	0.071777	0

Cluster 2: [8 Significant SNPs]

SNP	CHR	BP	SE	P-Val	Beta	Gene	Freq	Max Freq	Min Freq

SNP	CHR	BP	SE	P-Val	Beta	Gene	Freq	Max Freq	Min Freq
rs11731396	4	84352339	0.66790073	6.22809E-05	2.674273254	-	C=0.144393/29546 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs11731396#frequency_tab) C=0./0 PRJEB36033 C=0.066424/1113 TOMMO	0.144393	0
rs1595361	18	4569102	0.807969804	7.40472E-05	3.201897261	-	C=0.425201/10903 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs1595361#frequency_tab) T=0.266667/104 SGDP_PRJ C=0.266667/8 PRJEB36033	0.425201	0.266667
rs3787397	20	52700084	0.645998874	4.99397E-05	2.62011272	DOK5	A=0.33283/51972 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs3787397#frequency_tab) A=0.255591/160 Chileans G=0.261905/11 Siberian	0.33283	0.255591
rs4670713	2	37507459	0.635650094	4.44686E-05	2.595324143	-	C=0.244312/8354 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs4670713#frequency_tab) C=0.171667/103 NorthernSweden C=0.193825/747 ALSPAC	0.244312	0.171667
rs5956063	23	117486408	0.509138878	6.99535E-05	2.024562099	-	T=0.042562/618 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs5956063#frequency_tab) T=0.086535/250 ALSPAC T=0.087648/325 TWINSUK	0.042562	0.086535
rs6440278	3	146124983	0.625697778	7.00049E-05	2.487942842	-	G=0.166145/35014 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs6440278#frequency_tab) G=0.110229/125 Daghestan G=0.111821/70 Chileans	0.166145	0.110229
rs7318474	13	28027431	0.632986484	4.30473E-05	2.58922144	-	G=0.358709/110575 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs7318474#frequency_tab) G=0.083905/1406 TOMMO G=0.113208/24 Vietnamese	0.358709	0.083905

SNP	CHR	BP	SE	P-Val	Beta	Gene	Freq	Max Freq	Min Freq
rs7894606	10	92956616	0.758332586	9.56854E-05	2.958467684	PCGF5	A=0.17591/5468 ALFA (https://www.ncbi.nlm.nih.gov/snp/rs7894606#frequency_tab) A=0.028037/6 Vietnamese A=0.048123/141 KOREAN A=0.061135/112 Korea1K	0.17591	0.028037

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)?

For PC1:

- The top 10 significant SNPs include rs12613530, rs213037, rs235679, rs3181360, rs359428, rs3787397, rs4670713, rs6850796, rs6967342, and rs7957163.
- Each SNP has associated information such as CHR, BP position, beta coefficient (Beta), p-value (P-Val), standard error (SE), gene, and frequency information.
- The genes associated with these SNPs include CAPN14, ECE1, NKAIN2, TNFSF8, DOK5, and others.

For Cluster 2:

- The significant SNPs include rs11731396, rs1595361, rs3787397 (appeared in PC1), rs4670713 (also in PC1), rs5956063, rs6440278, rs7318474, and rs7894606.
- Similar to the PC1 data, each SNP has associated information including CHR, BP, SE, P-Val, Beta, gene, and frequency information.
- The genes associated with there SNPs include DOK5 and PCGF5.

Comments:

- The SNPs identified for each PC and cluster seem to overlap to some extent. For example, rs3787397 and rs4670713 appear in both PC1 and Cluster 2.
- However, there are also unique SNPs for each group. For example, rs12613530, rs213037, rs235679, rs3181360, rs6850796, rs6967342, and rs7957163 are unique to PC1, while rs11731396, rs1595361, rs5956063, rs6440278, rs7318474, and rs7894606 are unique to Cluster 2.
- The analysis using PCs and clustering lead to different sets of significant SNPs due to the different statistical methods and grouping criteria used.
- The genes containing these SNPs include CAPN14, ECE1, NKAIN2, TNFSF8, DOK5, and PCGF5 among others. These genes may have potential biological relevance to the traits or phenotypes being studied.