qQTL exercise

Mahdi

September 25, 2020

Part 1

Question 1

In the `sub_geno.tab` file, 0, 1 and 2 most likely represent the two homozygous and hgeteroguzous genotypes. -1 probably means missing data.

Question 2

The `design.tab` file contains information about each column of the `sub_expr.tab` file. It says which population they belong to and other characteristics.

Question 3

Gene expression levels can be somehwat explained by the first SNP but not the second. This is because the first SNP has an R^2 value of 0.5394 and very small p value for its coefficient meaning the model explains around half the variation in the data. In contrast the second SNP has an R^2 value of 0.002019 meaning the model explains almost no variation in the data.

Question 4 Do a linear regression for snp__22__43336231 on ENSG00000100266.11

Without covariates

```
gene2 <- "ENSG00000100266.11"
snp2 <- "snp_22_43336231"

gene2_col <- t(gene_expr)[,gene2]
gene2_snp <- t(snps_filtered)[,snp2]
lm_no_cov <- lm(gene2_col ~ gene2_snp)
summary(lm_no_cov)
```

```
##
## Call:
## lm(formula = gene2_col ~ gene2_snp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.367  -5.791  -0.774   4.563  41.890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23.8641     0.5297   45.05  < 2e-16 ***
## gene2_snp     3.3238     0.6121    5.43 9.13e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.746 on 460 degrees of freedom
## Multiple R-squared:  0.06024,    Adjusted R-squared:  0.0582
## F-statistic: 29.49 on 1 and 460 DF,  p-value: 9.131e-08
```

Using the genotype PCs from pc_cvrt.tab as covariates

```
pc <- read.table("data/pc_cvrt.tab")
lm_pc <- lm(gene2_col ~ gene2_snp + pc$PC1 + pc$PC2 + pc$PC3 + pc$PC4 + pc$PC5)
summary(lm_pc)
```

```
##
## Call:
## lm(formula = gene2_col ~ gene2_snp + pc$PC1 + pc$PC2 + pc$PC3 +
##     pc$PC4 + pc$PC5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.129  -5.400  -0.454   4.568  43.137
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.521911   0.543181  43.304  < 2e-16 ***
## gene2_snp    3.941343   0.660838   5.964 4.94e-09 ***
## pc$PC1       0.012720   0.004472   2.844  0.00465 **
## pc$PC2       0.026296   0.014024   1.875  0.06142 .
## pc$PC3      -0.034836   0.014238  -2.447  0.01480 *
## pc$PC4       0.004344   0.015497   0.280  0.77934
## pc$PC5       0.007566   0.016014   0.472  0.63681
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.623 on 455 degrees of freedom
## Multiple R-squared:  0.09625,    Adjusted R-squared:  0.08433
## F-statistic: 8.076 on 6 and 455 DF,  p-value: 2.643e-08
```

Separately for african and non-africans without covariates. Hint: Use the information in the design.tab

```
get_pop_gene <- function(genes, pop, gene_mat, design_mat, inv = F){
  genes_table <- filter_snp_population(pop, gene_mat, design_mat, inv)
  genes <- t(genes_table)[,genes]
  return(genes)
}

#make african model
gene2_africa <- get_pop_gene(gene2, "YRI", gene_expr, design)
snp2_africa <- t(african_snps)[,snp2]
lm_africa <- lm(gene2_africa ~ snp2_africa)
summary(lm_africa)
```

```
##
```

```
## Call:
## lm(formula = gene2_africa ~ snp2_africa)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.0137  -4.1504  -0.3292   5.0336  19.5839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  26.3095     0.7353  35.781   <2e-16 ***
## snp2_africa  -0.7181     2.8319  -0.254      0.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.699 on 87 degrees of freedom
## Multiple R-squared:  0.0007385,  Adjusted R-squared:  -0.01075
## F-statistic: 0.0643 on 1 and 87 DF,  p-value: 0.8004
```

```
#make non african model
gene2_nonafrica <- get_pop_gene(gene2, "YRI", gene_expr, design, T)
snp2_nonafrica <- t(non_african_snps)[,snp2]
lm_nonafrica <- lm(gene2_nonafrica ~ snp2_nonafrica)
summary(lm_nonafrica)
```

```
##
## Call:
## lm(formula = gene2_nonafrica ~ snp2_nonafrica)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.922  -5.727  -0.700   4.583  42.142
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     22.8046     0.6598  34.562  < 2e-16 ***
## snp2_nonafrica   4.1310     0.6911   5.978 5.32e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.075 on 371 degrees of freedom
## Multiple R-squared:  0.08785,    Adjusted R-squared:  0.08539
## F-statistic: 35.73 on 1 and 371 DF,  p-value: 5.321e-09
```

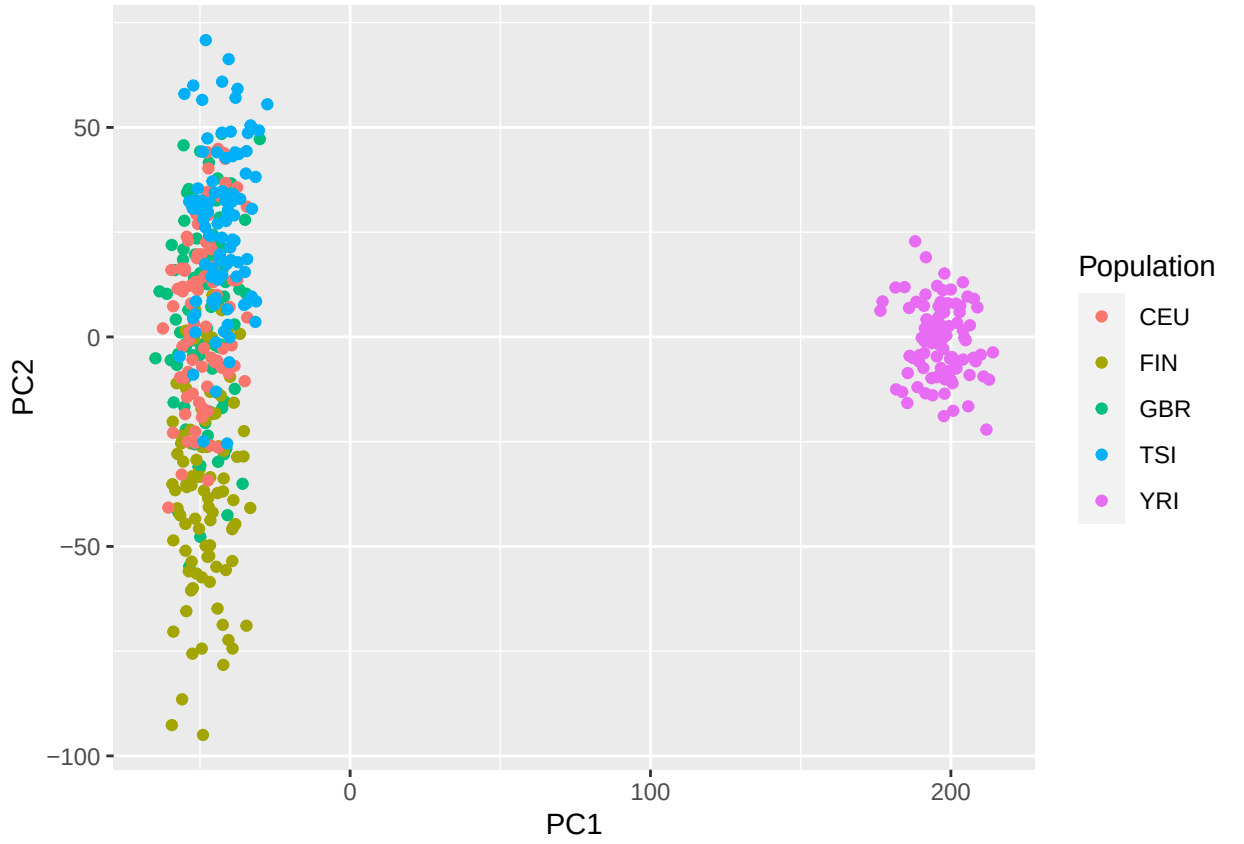
Make a dotplot of PC1 vs PC2 and color the dots by population

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## -- Conflicts ------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()    masks stats::lag()
```

```
pc_df <- data.frame(PC1=pc$PC1, PC2=pc$PC2, Population=design$Characteristics.population.)
pc_df %>%
  #gather(-Population, key="PC", value="Value") %>%
  ggplot(aes(x=PC1, y=PC2, col=Population)) +
  geom_point()
```



Question 5

1. There is no difference since both models havea very low R^2 value. (probably wrong)
2. Both models are quite poor but the african model is much worse than the nona frican model since it has a lower R^2 value and a much higher p-value. 3 We are including the principal components of the gene expression data.

Task 6 Do a linear regression on 1st snp on 1st gene, 2nd snp on 2nd gene etc.

1. Create a matrix containing the gene_id, snp_id, effect size, t.value and p.value

```
gene_t <- as.data.frame(t(gene_expr))
snps_filtered_t <- as.data.frame(t(snps_filtered))

get_lm_values <- function(expr, snps){
  model <- summary(lm(expr ~ snps))
  return(model$coefficients[2,])
}
```

```
get_lm_matrix <- function(expr, snps, fun){
  lm_matrix <- mapply(fun, expr, snps)
  lm_matrix <- as.data.frame(t(lm_matrix))
  lm_matrix$gene_id <- rownames(lm_matrix)
  lm_matrix$snp_id <- colnames(snps)
  rownames(lm_matrix) <- NULL
  lm_matrix <- lm_matrix[,c(5,6,1,3,4)]
  colnames(lm_matrix) <- c("gene_id", "snp_id", "effect_size", "t.value", "p.value")
  return(lm_matrix)
}

lm_matrix <- get_lm_matrix(gene_t, snps_filtered_t, get_lm_values)
head(lm_matrix)
```

```
##              gene_id          snp_id   effect_size     t.value   p.value
## 1 ENSG00000185386.10 snp_22_30772686  0.0151891271  0.11471394 0.9087219
## 2  ENSG00000203606.3 snp_22_34965577 -0.0026938729 -0.43502718 0.6637467
## 3  ENSG00000069998.8 snp_22_49436707  0.0303004155  0.05279978 0.9579144
## 4  ENSG00000240293.1 snp_22_34153853  0.0005987039  0.12815676 0.8980809
## 5  ENSG00000232926.1 snp_22_21970216 -0.0123130207 -1.20403852 0.2291939
## 6 ENSG00000100151.11 snp_22_48286671 -0.0689291239 -0.37137433 0.7105297
```

2. Do a multiple testing correction on the resulting p.values using fdr.

```
lm_matrix$p.adj <- p.adjust(lm_matrix$p.value, method = "fdr")
lm_matrix %>%
  filter(p.adj < 0.05)
```

```
##              gene_id          snp_id effect_size   t.value      p.value
## 1  ENSG00000205853.5 snp_22_32778467 -0.09675995 -4.362069 1.591656e-05
## 2 ENSG00000186716.14 snp_22_23454881 -0.73348131 -2.635860 8.676019e-03
## 3  ENSG00000172404.4 snp_22_41256802  0.12513490 23.209894 1.853078e-79
## 4 ENSG00000075234.12 snp_22_46686404  3.02798811 14.751187 1.335992e-40
## 5 ENSG00000100266.11 snp_22_43336231  3.32381025  5.430240 9.130826e-08
## 6  ENSG00000128408.7 snp_22_45782142 -0.27686405 -3.974314 8.193048e-05
##          p.adj
## 1 1.273325e-04
## 2 4.627210e-02
## 3 5.929849e-78
## 4 2.137587e-39
## 5 9.739548e-07
## 6 5.243551e-04
```

3. Do the same but now include the genotype PCs from pc_cvrt.tab as covariates.

```
get_lm_values_covariate <- function(expr, snps){
  model <- summary(lm(expr ~ snps + pc$PC1 + pc$PC2 + pc$PC3 + pc$PC4 + pc$PC5))
  return(model$coefficients[2,])
}

lm_matrix_cov <- get_lm_matrix(gene_t, snps_filtered_t, get_lm_values_covariate)
lm_matrix_cov$p.adj <- p.adjust(lm_matrix_cov$p.value, method = "fdr")
lm_matrix_cov %>%
  filter(p.adj < 0.05)
```

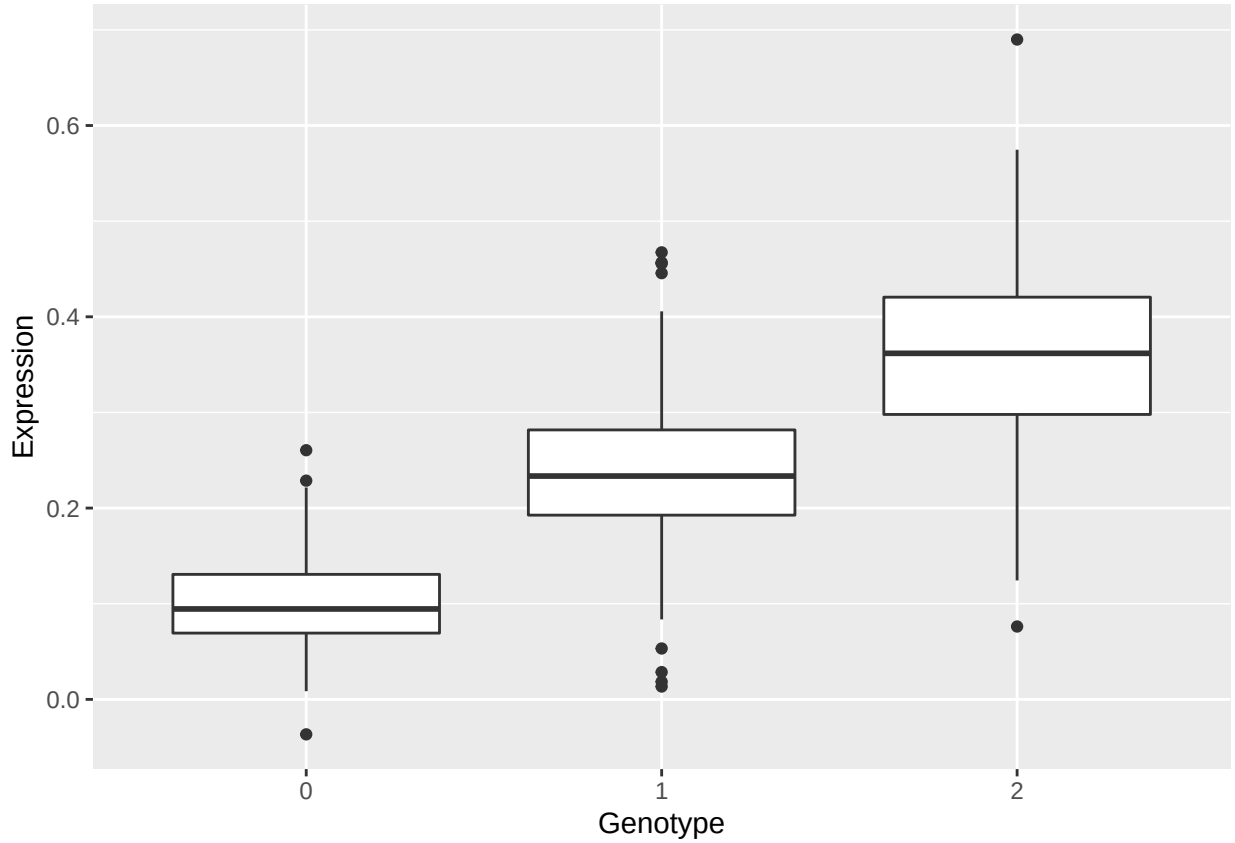
```
##              gene_id          snp_id effect_size   t.value      p.value
```

```
## 1  ENSG00000205853.5 snp_22_32778467  -0.1014790 -4.476520 9.601900e-06
## 2 ENSG00000186716.14 snp_22_23454881  -0.8623192 -2.997211 2.873675e-03
## 3  ENSG00000172404.4 snp_22_41256802   0.1382878 22.657865 1.207675e-76
## 4 ENSG00000075234.12 snp_22_46686404   3.6745686 14.935637 2.479445e-41
## 5 ENSG00000100266.11 snp_22_43336231   3.9413429  5.964158 4.942791e-09
## 6  ENSG00000128408.7 snp_22_45782142  -0.3092542 -4.407607 1.305208e-05
##          p.adj
## 1 7.681520e-05
## 2 1.532627e-02
## 3 3.864559e-75
## 4 3.967112e-40
## 5 5.272310e-08
## 6 8.353331e-05
```

4. Plot the most significant hit.

```
sig_row <- lm_matrix_cov[lm_matrix_cov$p.adj == min(lm_matrix_cov$p.adj),]
df <- data.frame(Expression = gene_t[,sig_row$gene_id],
                 Genotype = as.factor(snps_filtered_t[,sig_row$snp_id]))

df %>%
  ggplot(aes(x=Genotype, y=Expression)) + geom_boxplot()
```



Question 6

1. 32 tests were performed in a and 32 tests were performed in c.
2. Since multiple tests are being perfomred, it is possible to get a false positive by obtaining a significant

p value by chance. The false positives are corrected using a multiple test correction. In eQTL analysis, a huge number of significance tests are performed so many flase posititves can occur, so it is essential to make sure to check for false positives.

3. No both models produced the same number of significant hits.

Task 7

```
library(MatrixEQTL)
snps <- SlicedData$new()
snps$CreateFromMatrix(as.matrix(snps_filtered)) #filt_geno is your filtered genotype matrix
genes <- SlicedData$new()
genes$CreateFromMatrix(as.matrix(gene_expr)) #expr is the unchanged expression matrix
snp_pos <- read.table("data/sample_geno.pos",sep="\t",header=T)
snp_pos <- snp_pos[snp_pos$snp %in% row.names(snps_filtered),]
gene_pos <- read.table("data/sample_expr.pos",sep="\t",header=T)
all(colnames(snps) == colnames(genes))
```

```
## [1] TRUE
```

```
eQTL <- Matrix_eQTL_main(snps, genes, output_file_name=NULL,
output_file_name.cis=NULL,
pvOutputThreshold.cis=1, pvOutputThreshold=1,
snpspos=snp_pos, genepos=gene_pos,
cisDist = 0)
```

```
## Matching data files and location files
## 32 of 32 genes matched
## 32 of 32 SNPs matched
## Task finished in 0.01 seconds
## Reordering SNPs
## Task finished in 0.13 seconds
## Reordering genes
## Task finished in 0.12 seconds
## Processing covariates
## Task finished in 0 seconds
## Processing gene expression data (imputation, residualization)
## Task finished in 0 seconds
## Creating output file(s)
## Task finished in 0 seconds
## Performing eQTL analysis
## 100.00% done, 0 cis-eQTLs, 1,024 trans-eQTLs
## No significant associations were found.
## Task finished in 0.02 seconds
##
```

Question 7

1. 1024 tests were performed.
2. MatrixeQTL found no significant hits while the analysis in Task 6 yielded 6 significant hits. This may be caused by