Handin 2 part 2.1

Stefano Pellegrini

9/27/2020

```
library(MatrixEQTL)
library(tidyverse)
```

Part 1

Task 1

a. What do the -1,0,1,2 values represent in the sub_geno.tab file?
-1 rapresent a missing genotype, 0 rapresents the homozygous reference genotype (e.g. AA), 1 is the heterozygous genotype (e.g. AB), and 2 rapresents the homozygoues alternative genotype (e.g. BB).

b. What is sored in the sub_expr.tab file and what has been done with this data?
It stores the expression of 32 genes across 462 samples. The expression data has been normalized (FPKM) and it is ready to use.

c. What information is stored in the design.txt file?
It stores usuful information about the sample (samples metadata), such as organism, strain, population, phase of the project when genotype was performed, the laboratory where sequencing occurred (useful for batch effect evaluation), and other.

```
# Explore data
geno <- read.table("sub_geno.tab")
dim(geno)
```

```
## [1]  39 462
```

```
geno[c(1:5),c(1:7)]
```

```
##                HG00096 HG00097 HG00099 HG00100 HG00101 HG00102 HG00103
## snp_22_30772686       0       0       0       0       0       0       0
## snp_22_34965577       0       1       0       0       1       0       1
## snp_22_49436707       0       0       0       1       0       0       0
## snp_22_30631851       0       0       0       0       0       0       0
## snp_22_46215888       0       0       0       0       0       0       0
```

```
expr <- read.table("sub_expr.tab")
dim(expr)
```

```
## [1]  32 462
```

```
expr[c(1:5),c(1:5)]
```

```
##                        HG00096     HG00097     HG00099      HG00100     HG00101
## ENSG00000185386.10  3.39126086  3.14338583  1.75744149  4.637220667  3.31850212
## ENSG00000203606.3   0.13722244  0.06493753  0.07066341  0.135230334  0.08497032
## ENSG00000069998.8  21.78079542 27.10260305 18.10522352 29.097505289 28.91372304
## ENSG00000240293.1   0.04996629  0.24267019  0.05172027  0.007624477  0.13026634
## ENSG00000232926.1   0.05538650  0.10237401  0.09192106  0.121656570  0.13892390
```

```
design_matrix <- read.table("design.tab", sep="\t")
dim(design_matrix)
```

```
## [1] 462   7
```

```
colnames(design_matrix)[colnames(design_matrix) == "Characteristics.population."] = "Pop"
colnames(design_matrix)[colnames(design_matrix) == "Characteristics.Organism."] = "Organism"
colnames(design_matrix)[colnames(design_matrix) == "Factor.Value.laboratory."] = "Lab"
colnames(design_matrix)[colnames(design_matrix) == "Characteristics.Strain."] = "Strain"
design_matrix[c(1:5),c(1,4,5,7)]
```

```
##         Source.Name                   Strain Pop Lab
## HG00096     HG00096 lymphoblastoid cell line GBR   1
## HG00097     HG00097 lymphoblastoid cell line GBR   7
## HG00099     HG00099 lymphoblastoid cell line GBR   5
## HG00100     HG00100 lymphoblastoid cell line GBR   2
## HG00101     HG00101 lymphoblastoid cell line GBR   1
```

Task 2

a. Calculate the number of missing genotypes for each SNP across all individuals.

```
missing <- apply(geno == -1, 1, sum)
```

b. Calculate the minor allele frequency (MAF) for all SNPs across all individuals.

```
# MAF method 1
# Compute the count of each allele
pp <- apply(geno==0, 1, sum)
pq <- apply(geno==1, 1, sum)
qq <- apply(geno==2, 1, sum)
# Total number of genotypes
n <- pp + pq + qq
# Allele frequencies
p <- ((2*pp) + pq)/(2*n)
q <- 1 - p
# MAF
maf1 <- pmin(p, q)
maf1 <- data.frame(maf1)

# MAF method 2
f_alt <- apply(geno, 1, function(x) mean(x[x > -1])) /  2
f_ref <- 1 - f_alt
maf <- data.frame(MAF = pmin(f_alt, f_ref))
```

c. Filter our SNPs that have missing genotypes or a MAF<0.05 and use the filtered snps for the rest of the exercise.

```
filtered_geno <- geno[(missing == 0) & (maf >= 0.05),]
dim(filtered_geno)
```

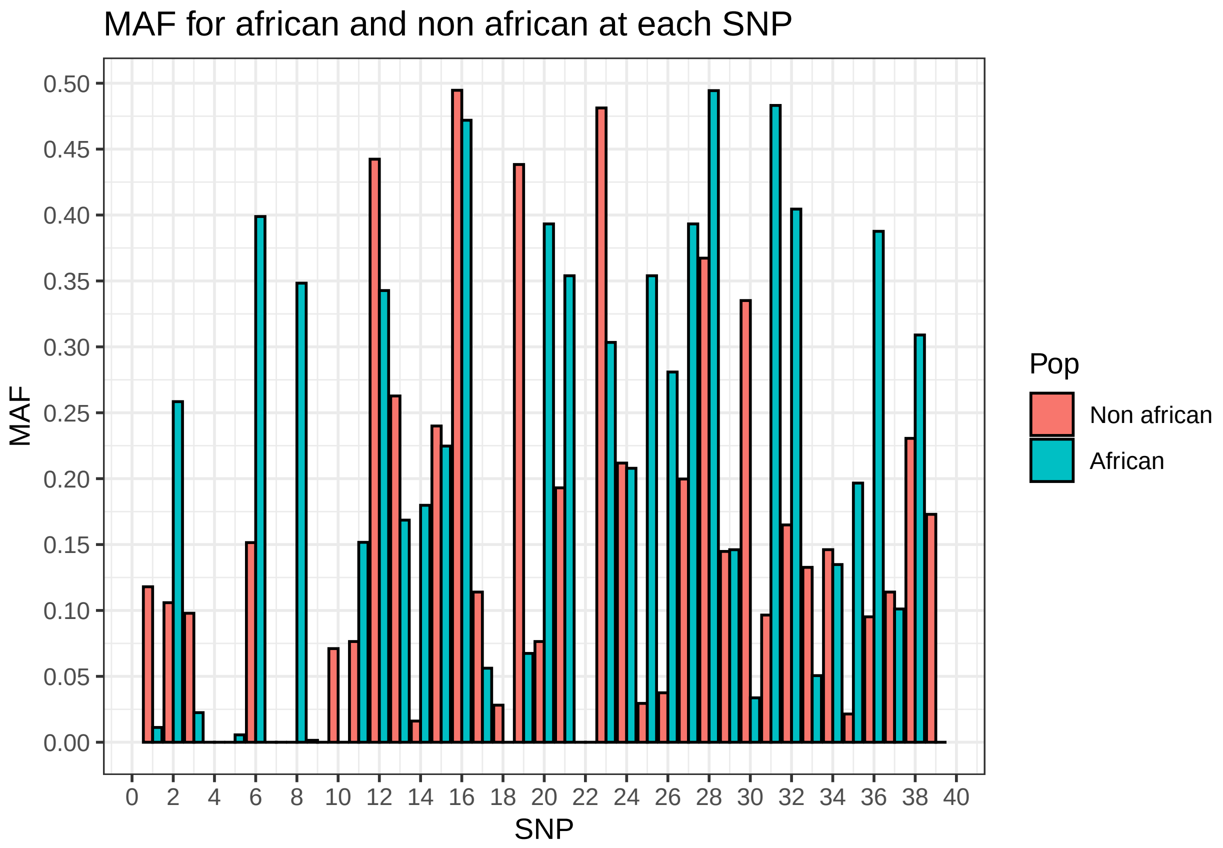
```
## [1]  32 462
```

We removed all SNPs with missing genotype and low MAF because it is very difficul to see if there is a change in these SNPs.

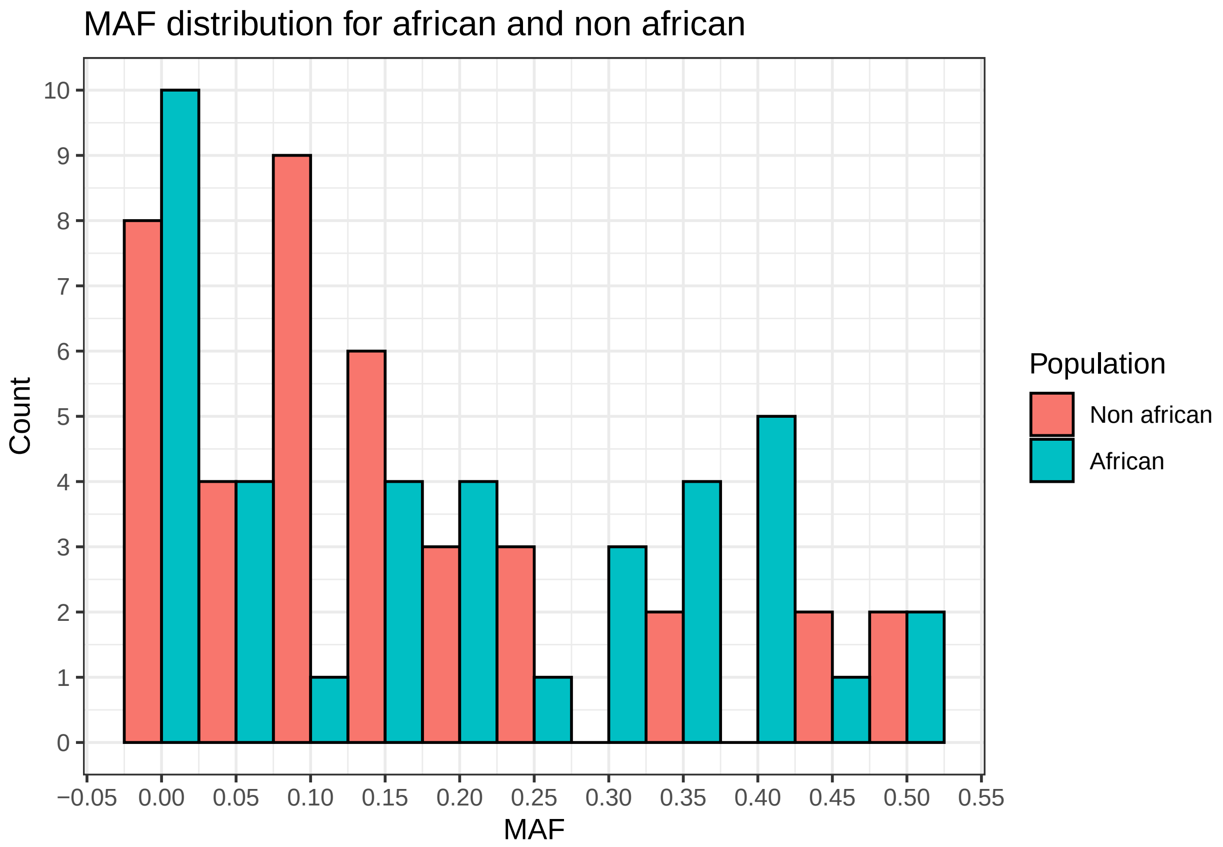
d. Calculate the MAF for africans and non-africans separately. Is there a difference?

```
# Africans
f_alt <- apply(geno[,design_matrix$Pop == "YRI"], 1, function(x) mean(x[x > -1])) /  2
f_ref <- 1 - f_alt
maf_YRI <- data.frame(MAF_YRI = pmin(f_alt, f_ref))

# Non africans
f_alt <- apply(geno[,design_matrix$Pop != "YRI"], 1, function(x) mean(x[x > -1])) /  2
f_ref <- 1 - f_alt
maf_noYRI <- data.frame(MAF_noYRI = pmin(f_alt, f_ref))
```

```
# Histogram MAF at each SNP
bind_cols(maf_noYRI, maf_YRI) %>%
  gather(key = "Pop", value = "maf", MAF_noYRI, MAF_YRI) %>%
  ggplot(aes(x=c(seq(39),seq(39)), y=maf, fill=Pop)) +
  geom_bar(stat = "identity", position = "dodge", col = "black") +
  labs(title = "MAF for african and non african at each SNP", color = "Population") +
  ylab("MAF") +
  xlab("SNP") +
  scale_fill_hue(labels = c("Non african", "African")) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
  theme_bw()
```



```
# Histogram MAF distribution
bind_cols(maf_noYRI, maf_YRI) %>%
  gather(key = "Pop", value = "maf", MAF_noYRI, MAF_YRI) %>%
  ggplot(aes(x=maf)) +
  geom_histogram(aes(fill=Pop), position = "dodge", col="black", binwidth = 0.05) +
  labs(title = "MAF distribution for african and non african", fill = "Population") +
  ylab("Count") +
  xlab("MAF") +
  scale_fill_hue(labels = c("Non african", "African")) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
  theme_bw()
```

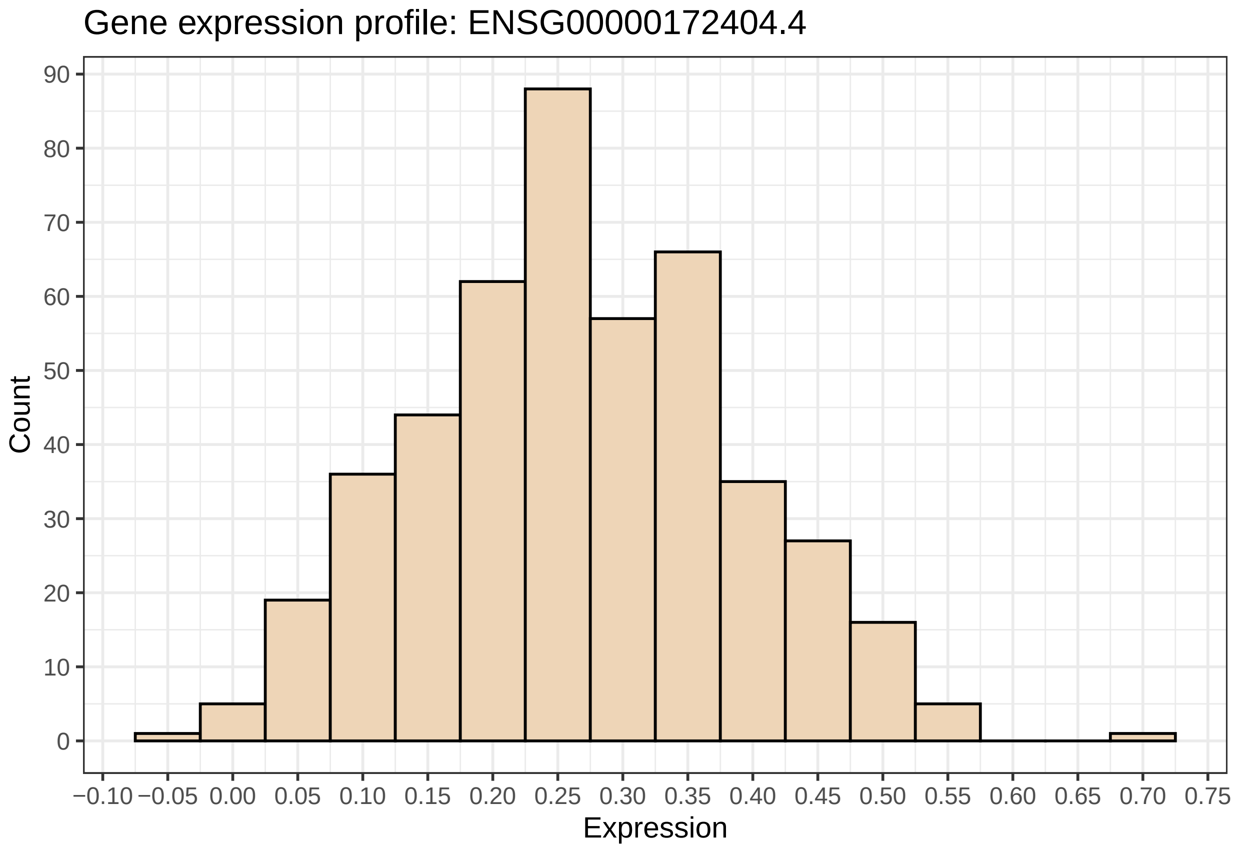


There is a difference in the MAF of africans and non africans.

Task 3: Gene expression profiles

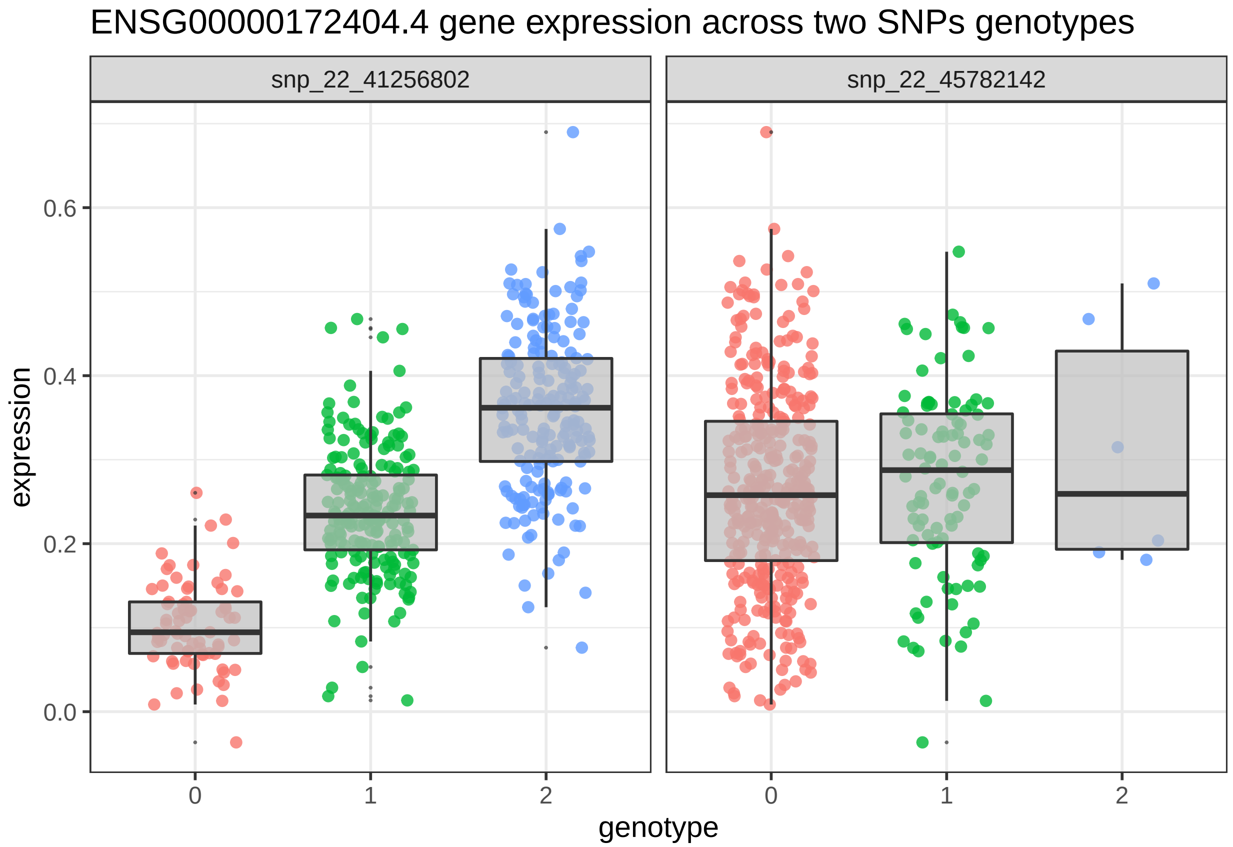
a. Plot the distribution of expression levels across all samples for the ENSG00000172404.4 gene.

```
gene = ("ENSG00000172404.4")
snps = c("snp_22_41256802", "snp_22_45782142")

# Gene expression across all samples
expr[gene,] %>% t() %>% as_tibble() %>%
  ggplot(aes(x=get(gene))) + geom_histogram(col="black", fill="bisque2", binwidth = 0.05) +
  labs(title = paste("Gene expression profile:", gene)) +
  ylab("Count") +
  xlab("Expression") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
  theme_bw()
```



b. Plot the expression levels of ENSG00000172404.4 against the genotypes of snp__22__41256802 and snp__22__45782142.

```
# Convert to long format for plotting
exprLong <- rownames_to_column(data.frame(t(expr[gene,])), var = "sample")
colnames(exprLong)[2] = "expression"
snpLong <- data.frame(t(filtered_geno[snps,])) %>%
  as_tibble() %>%
  gather(key = "snp", value = "genotype", snp_22_41256802, snp_22_45782142)
dataLong <- bind_cols(snpLong, bind_rows(exprLong, exprLong))
dataLong$genotype <- as.factor(dataLong$genotype)

# Plot
dataLong %>% ggplot(aes(x=genotype, y=expression)) +
  geom_jitter(aes(colour=genotype), alpha=0.8, position=position_jitter(width=0.25)) +
  geom_boxplot(outlier.size=0, fill="grey", alpha=0.7) +
  facet_wrap(~snp) +
  labs(title = paste(gene, "gene expression across two SNPs genotypes")) +
  theme_bw() +
  theme(legend.position = "none")
```



We can see that there is a clear variation in the expression of the gene across the first SNP genotype, but not in the second one.

Task 4: Do a linear regression of all sample genotypes on sample gene expression

a. For snp__22__41256802 on ENSG00000172404.4
b. For snp__22__45782142 on ENSG00000172404.4

```
expr_t <- t(expr)
geno_t <- t(filtered_geno)
model_a = lm(expr_t[,"ENSG00000172404.4"] ~ geno_t[,"snp_22_41256802"])
summary(model_a)
```

```
##
## Call:
## lm(formula = expr_t[, "ENSG00000172404.4"] ~ geno_t[, "snp_22_41256802"])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.28348 -0.04934 -0.00143  0.04950  0.33024
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.109393   0.007824   13.98   <2e-16 ***
## geno_t[, "snp_22_41256802"] 0.125135   0.005391   23.21   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.08216 on 460 degrees of freedom
## Multiple R-squared:  0.5394, Adjusted R-squared:  0.5384
## F-statistic: 538.7 on 1 and 460 DF,  p-value: < 2.2e-16
```

```
model_b = lm(expr_t[,"ENSG00000172404.4"] ~ geno_t[,"snp_22_45782142"])
summary(model_b)
```

```
##
## Call:
## lm(formula = expr_t[, "ENSG00000172404.4"] ~ geno_t[, "snp_22_45782142"])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.31360 -0.08515 -0.00588  0.07998  0.42486
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.265040   0.006332  41.856   <2e-16 ***
## geno_t[, "snp_22_45782142"] 0.011987   0.012425   0.965    0.335
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1209 on 460 degrees of freedom
## Multiple R-squared:  0.002019,   Adjusted R-squared:  -0.0001502
## F-statistic: 0.9308 on 1 and 460 DF,  p-value: 0.3352
```

Questions 1-4:

1. What do the -1,0,1,2 values represent in the sub__geno.tab file? (same as Task 1a)
-1 rapresent a missing genotype, 0 rapresents the homozygous reference genotype (e.g. AA), 1 is the heterozygous genotype (e.g. AB), and 2 rapresents the homozygoues alternative genotype (e.g. BB).

2. What information is stored in the design.txt file? (same as Task 1c)
It stores usuful information about the sample (samples metadata), such as organism, strain, population, phase of the project when genotype was performed, the laboratory where sequencing occurred (useful for batch effect evaluation), and other.

3. Explain the results from the linear model in Task 4. What are the important values to look at and what do they tell you?
In model a, there is a a significant positive linear relationship between the gene expression (response variable) and the SNP 22_41256802 (predictor variable). Meaning that the presence of the SNP alternative allele is associated with an increase of gene expression. While, in model b, there isn't enough evidence to state that there is a relationship between the gene expression and the SNP 22_45782142. The most important values to look at are: the p-values and the estimate of SNP coefficients. The p-value tells us if there is a significative linear relationship between the two variables. In other words, it tells us how likely is to observe such data, under the assumption that the null hypothesis is true. Where the null hypothesis states that the value of the coefficient (effect size) is 0. The estimates provide information about the size and the direction of the effect that the predictor variable has on the response variable. Also, other important values to look at are the standard error of the estimate, and the t-value, which can be thought of as a measure of the precision for the estimated coefficients values. Lastly, another important value is the R squared, which provides information about the fit of the model. Model a shows a good fit with an R squared value of 0.5, while model b has a bad fit showing a value of 0.002.

Task 5: Do a linear regression for snp_22_43336231 on ENSG00000100266.11

a. Without covariates.

```r
model = lm(expr_t[,"ENSG00000100266.11"] ~ geno_t[,"snp_22_43336231"])
summary(model)
```

```
##
## Call:
## lm(formula = expr_t[, "ENSG00000100266.11"] ~ geno_t[, "snp_22_43336231"])
##
## 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 ***
## geno_t[, "snp_22_43336231"]   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
```

b. Using the genotype PCs from pc_cvrt.tab as covariates.

```r
geno_pcs <- read.table("pc_cvrt.tab")

model = lm(expr_t[,"ENSG00000100266.11"] ~ geno_t[,"snp_22_43336231"] + ., data = geno_pcs)
summary(model)
```

```
##
## Call:
## lm(formula = expr_t[, "ENSG00000100266.11"] ~ geno_t[, "snp_22_43336231"] +
##     ., data = geno_pcs)
##
## 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 ***
## geno_t[, "snp_22_43336231"]  3.941343   0.660838   5.964 4.94e-09 ***
## PC1                          0.012720   0.004472   2.844  0.00465 **
## PC2                          0.026296   0.014024   1.875  0.06142 .
## PC3                         -0.034836   0.014238  -2.447  0.01480 *
## PC4                          0.004344   0.015497   0.280  0.77934
## 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
```

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

```r
# Africans
model = lm(expr_t[design_matrix$Pop == "YRI","ENSG00000100266.11"] ~
             geno_t[design_matrix$Pop == "YRI","snp_22_43336231"])
summary(model)
```

```
##
## Call:
## lm(formula = expr_t[design_matrix$Pop == "YRI", "ENSG00000100266.11"] ~
##     geno_t[design_matrix$Pop == "YRI", "snp_22_43336231"])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.0137  -4.1504  -0.3292   5.0336  19.5839
##
## Coefficients:
##                                                        Estimate Std. Error
## (Intercept)                                             26.3095     0.7353
## geno_t[design_matrix$Pop == "YRI", "snp_22_43336231"]  -0.7181     2.8319
##                                                        t value Pr(>|t|)
## (Intercept)                                             35.781   <2e-16 ***
## geno_t[design_matrix$Pop == "YRI", "snp_22_43336231"]  -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
```

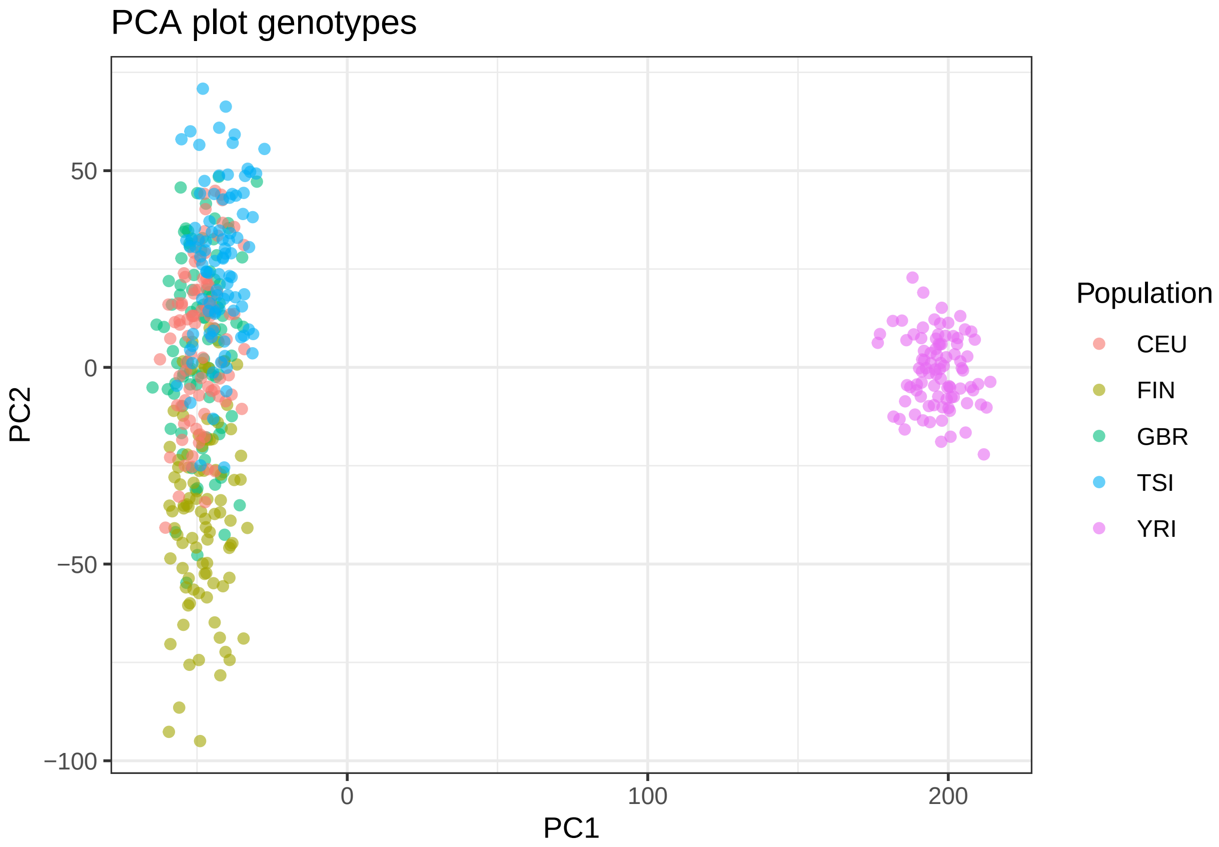
```r
# Non africans
model = lm(expr_t[design_matrix$Pop != "YRI","ENSG00000100266.11"] ~
             geno_t[design_matrix$Pop != "YRI","snp_22_43336231"])
summary(model)
```

```
##
## Call:
## lm(formula = expr_t[design_matrix$Pop != "YRI", "ENSG00000100266.11"] ~
##     geno_t[design_matrix$Pop != "YRI", "snp_22_43336231"])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.922  -5.727  -0.700   4.583  42.142
##
## Coefficients:
##                                                        Estimate Std. Error
## (Intercept)                                             22.8046     0.6598
## geno_t[design_matrix$Pop != "YRI", "snp_22_43336231"]   4.1310     0.6911
##                                                        t value Pr(>|t|)
## (Intercept)                                             34.562  < 2e-16 ***
## geno_t[design_matrix$Pop != "YRI", "snp_22_43336231"]   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
```

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

```
geno_pcs %>% ggplot(aes(x=PC1, y=PC2, col=design_matrix$Pop)) +
  geom_point(alpha=0.6) +
  labs(title = "PCA plot genotypes", color = "Population") +
  theme_bw()
```



Questions 5:

1. Is there a difference in your results in a and b? If so explain why.

It doesn't seem to be a difference between the two models. That's because the PCs added as covariates in the second model show a weak or no association with the gene expression. While in both models, there is a strong positive linear relationshiop between the response variable and the SNP. To assess if there is a significative difference between the two models, a statistical test, such as the likelihood ratio test (LRT), could be performed. Neverthless, since in the model b the PCs of the populations genotypes were included as covariates, this model explains a larger proportion of the gene expression variance with respect to one explained by model a.

2. Is there a difference between african and non-africans? If so explain why.

There is a difference between african and non-african. In fact, there is a significative positive association between the gene expression and the SNP genotypes in non africans, while there isn't a significative association in africans. Indeed, according to the "Out of Africa" model, the modern Homo sapiens originated in Africa, and then it migrated across the globe. Therefore, the non-african populations were subjected to a drastic bottleneck and to different selection pressures in respect to the african ones. Also, in the plots produced in Task 2d, it is possible to observe that the two groups shown a different minor allele frequencies.

3. What are we including in our model with the pc_cvrt.tab?

We are including the principal components (PCs) of the populations genotypes obtained by eigendecomposition (principal component analysis). The PCs are the extracted features rapresenting a linear combination of the population genotypes that captured the largest variance in the data.

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

a. Create a matrix containing the gene_id, snp_id, effect size, t.value and p.value.
b. Do a multiple testing correction on the resulting p.values using fdr.

```r
# Function to perform linear regresion of n_th snp on n_th gene expression.
pairwise_lm <- function(n){
  model <- lm(expr_t[,n] ~ geno_t[,n])
  lm_summary <- summary(model)
  row <- data.frame(gene=colnames(expr_t)[n],
                    snp=colnames(geno_t)[n],
                    t(lm_summary$coefficients[2,-2]))
  colnames(row)[5] <- "p.value"
  return(row)
}

# Create a data frame storing gene_id, snp_id, effect size, t.value and p.value
df_lm <- data.frame(gene=character(0),
                    snp=character(0),
                    estimate=numeric(0),
                    t.value=numeric(0),
                    p.value=numeric(0))
for (n in seq(dim(expr_t)[2])){
  df_lm[n,] <- pairwise_lm(n)
}

# Multiple testing correction
df_lm <- cbind(df_lm, p.adj=p.adjust(df_lm$p.value, method ="fdr"))
df_lm %>% arrange(p.adj) %>% head(10)
```

```
##                   gene             snp     estimate    t.value      p.value
## 1   ENSG00000172404.4 snp_22_41256802  0.125134902 23.2098937 1.853078e-79
## 2  ENSG00000075234.12 snp_22_46686404  3.027988106 14.7511872 1.335992e-40
## 3  ENSG00000100266.11 snp_22_43336231  3.323810251  5.4302395 9.130826e-08
## 4   ENSG00000205853.5 snp_22_32778467 -0.096759953 -4.3620695 1.591656e-05
## 5   ENSG00000128408.7 snp_22_45782142 -0.276864055 -3.9743136 8.193048e-05
## 6  ENSG00000186716.14 snp_22_23454881 -0.733481311 -2.6358605 8.676019e-03
## 7   ENSG00000213279.2 snp_22_29908154 -0.002963573 -2.2698628 2.367786e-02
## 8   ENSG00000183785.9 snp_22_44920999  0.747515263  1.3845329 1.668665e-01
## 9   ENSG00000232926.1 snp_22_21970216 -0.012313021 -1.2040385 2.291939e-01
## 10 ENSG00000100360.10 snp_22_48749371  0.133286922  0.9919806 3.217285e-01
##           p.adj
## 1  5.929849e-78
## 2  2.137587e-39
## 3  9.739548e-07
## 4  1.273325e-04
## 5  5.243551e-04
## 6  4.627210e-02
## 7  1.082417e-01
## 8  6.674662e-01
## 9  6.768613e-01
## 10 6.768613e-01
```

c. Do the same but now include the genotype PCs from pc__cvrt.tab as covariates.

```r
# Function
pairwise_lm_covariates <- function(n){
  model <- lm(expr_t[,n] ~ geno_t[,n] + ., data=geno_pcs)
  lm_summary <- summary(model)
  row <- data.frame(gene=colnames(expr_t)[n],
                    snp=colnames(geno_t)[n],
                    t(lm_summary$coefficients[2,-2]))
  colnames(row)[5] <- "p.value"
  return(row)
}

# Make df
df_lm_covariates <- data.frame(gene=character(0),
                               snp=character(0),
                               estimate=numeric(0),
                               t.value=numeric(0),
                               p.value=numeric(0))
for (n in seq(dim(expr_t)[2])){
  df_lm_covariates[n,] <- pairwise_lm_covariates(n)
}

# Multiple testing correction
df_lm_covariates <- cbind(df_lm_covariates,
                          p.adj=p.adjust(df_lm_covariates$p.value, method ="fdr"))
head(df_lm_covariates, 10)
```

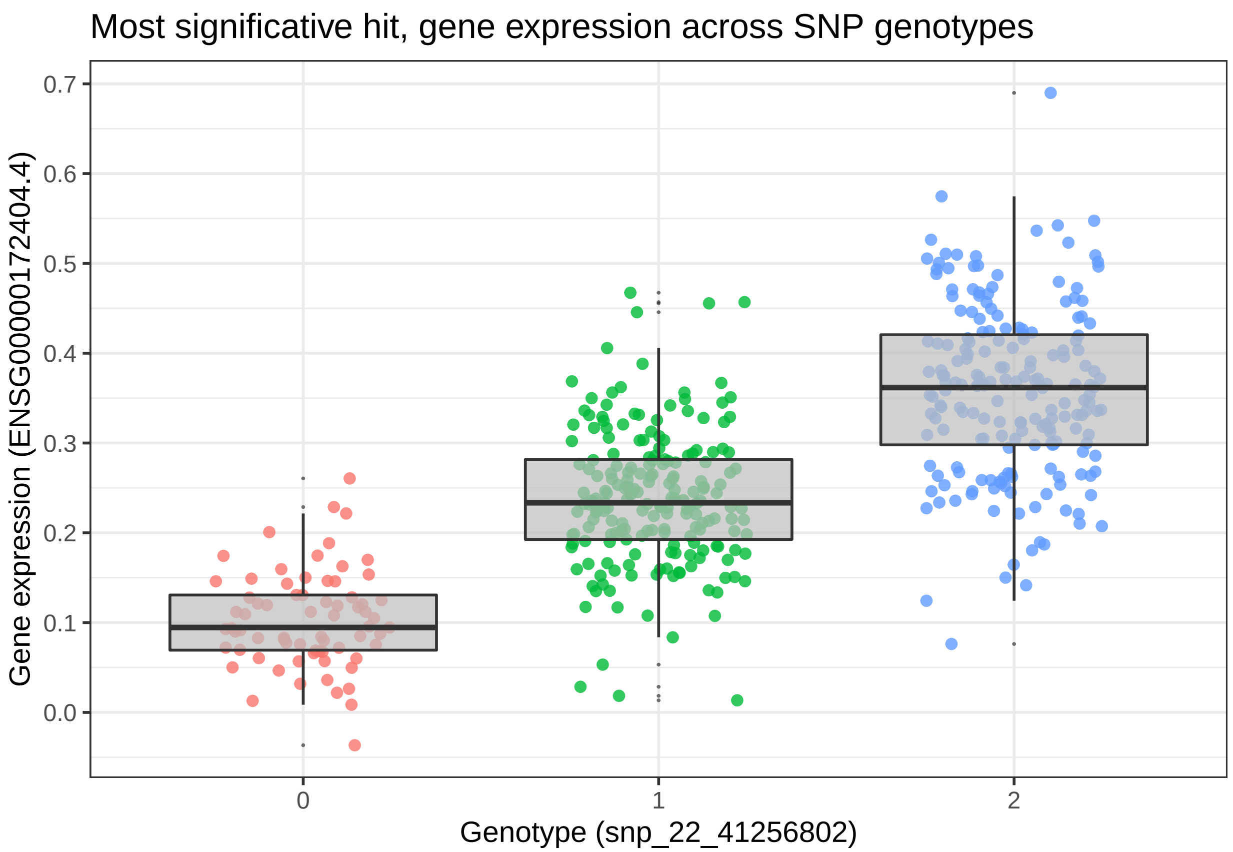
```
##                   gene             snp     estimate     t.value      p.value
## 1  ENSG00000185386.10 snp_22_30772686  0.080864278  0.59247393 0.5538275897
## 2   ENSG00000203606.3 snp_22_34965577 -0.004057665 -0.62641309 0.5313581603
## 3   ENSG00000069998.8 snp_22_49436707 -0.113814732 -0.19330769 0.8468042624
## 4   ENSG00000240293.1 snp_22_34153853  0.000575138  0.11627577 0.9074852861
## 5   ENSG00000232926.1 snp_22_21970216 -0.032702397 -2.26356595 0.0240710496
## 6  ENSG00000100151.11 snp_22_48286671 -0.016906713 -0.08843872 0.9295668990
## 7   ENSG00000205853.5 snp_22_32778467 -0.101478957 -4.47652016 0.0000096019
## 8  ENSG00000186716.14 snp_22_23454881 -0.862319228 -2.99721073 0.0028736753
## 9  ENSG00000179750.11 snp_22_45144106 -0.183938508 -0.43784252 0.6617082163
## 10 ENSG00000100360.10 snp_22_48749371  0.157422041  1.16867250 0.2431475377
##           p.adj
## 1  0.8055674032
## 2  0.8055674032
## 3  0.9675782288
## 4  0.9675782288
## 5  0.1100390839
## 6  0.9675782288
## 7  0.0000768152
## 8  0.0153262685
## 9  0.8612805784
## 10 0.5557658005
```

d. Plot the most significant hit.

```
best_hit <- df_lm_covariates %>% as_tibble() %>% arrange(p.adj) %>% head(1)
best_hit
```

```
## # A tibble: 1 x 6
##   gene              snp             estimate t.value  p.value    p.adj
##   <chr>             <chr>              <dbl>   <dbl>    <dbl>    <dbl>
## 1 ENSG00000172404.4 snp_22_41256802    0.138    22.7 1.21e-76 3.86e-75
```

```
# Convert to long format for plotting
exprLong <- rownames_to_column(data.frame(t(expr[best_hit$gene,])), var = "sample")
colnames(exprLong)[2] = "expression"
snpLong <- data.frame(t(filtered_geno[best_hit$snp,])) %>%
  as_tibble() %>%
  gather(key = "snp", value = "genotype", paste(best_hit$snp))
dataLong <- bind_cols(snpLong, exprLong)
dataLong$genotype <- as.factor(dataLong$genotype)

# Plot
dataLong %>% ggplot(aes(x=genotype, y=expression)) +
  geom_jitter(aes(colour=genotype), alpha=0.8, position=position_jitter(width=0.25)) +
  geom_boxplot(outlier.size=0, fill="grey", alpha=0.7) +
  labs(title = paste("Most significative hit, gene expression across SNP genotypes")) +
  ylab(paste("Gene expression (", best_hit$gene, ")", sep = "")) +
  xlab(paste("Genotype (", best_hit$snp, ")", sep = "")) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  theme_bw() +
  theme(legend.position = "none")
```



Questions 6:

1. How many tests did you perform in a? and c?

We performed 32 tests in both cases.

2. What are you correcting for with the fdr? Why is this important for eQTL analysis?

We are correcting for the false discovery of significant hits, or in other words, we are penalizing the p-values in order to reduce the rate of type I error (the p-values being significant by chance) when performing multiple tests. This is important when a large number of tests is performed, e.g. in eQTL analysis, where a large number of gene-SNP pairs are tested for association between gene expression and SNP allelic configuration.

3. Is there a difference in number of significant hits (FDR<0.05) in the two models?

No, the number of significant hits is 6 in both models.

```
# Significant hits first model
df_lm %>% filter(p.adj < 0.05) %>% arrange(p.adj)
```

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

```
# Significant hits second model
df_lm_covariates %>% filter(p.adj < 0.05) %>% arrange(p.adj)
```

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

Task 7: Use this Matrix_eQTL_main function to do eQTL analysis on the data.

```
snps <- SlicedData$new()
snps$CreateFromMatrix(as.matrix(filtered_geno))
genes <- SlicedData$new()
genes$CreateFromMatrix(as.matrix(expr))

snp_pos <- read.table("sample_geno.pos", sep="\t",header=T)
snp_pos <- snp_pos[snp_pos$snp %in% row.names(filtered_geno),]
gene_pos <- read.table("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 0.02 seconds

## Reordering SNPs

## Task finished in 0.29 seconds

## Reordering genes

## Task finished in 0.35 seconds
## Processing covariates

## Task finished in 0 seconds

## Processing gene expression data (imputation, residualization)

## Task finished in 0.01 seconds

## Creating output file(s)

## Task finished in 0.03 seconds

## Performing eQTL analysis

## 100.00% done, 0 cis-eQTLs, 1,024 trans-eQTLs

## No significant associations were found.

## Task finished in 0.03 seconds

##
```

Questions 7:

1. How many tests were performed in the eQTL analysis?

eQTL analysis performed 1024 tests because it tested all possible combination of gene-SNP pairs.

2. Compare the results from MatrixeQTL to your results from Task 6 a and b. Explain any similarities and/or differences that you see.

```
# Significant hits eQTL analysis
eQTL$trans$eqtls %>% as_tibble %>% filter(FDR < 0.05) %>% arrange(FDR)
```

```
## # A tibble: 10 x 6
##    snps            gene               statistic   pvalue      FDR    beta
##    <chr>           <chr>                  <dbl>    <dbl>    <dbl>   <dbl>
##  1 snp_22_41256802 ENSG00000172404.4      23.2  1.85e-79 1.90e-76  0.125
##  2 snp_22_46686404 ENSG00000075234.12     14.8  1.34e-40 6.84e-38  3.03
##  3 snp_22_43336231 ENSG00000100266.11      5.43 9.13e- 8 3.12e- 5  3.32
##  4 snp_22_32778467 ENSG00000205853.5      -4.36 1.59e- 5 4.07e- 3 -0.0968
##  5 snp_22_39406711 ENSG00000179750.11      4.02 6.86e- 5 1.40e- 2  1.43
##  6 snp_22_45782142 ENSG00000128408.7      -3.97 8.19e- 5 1.40e- 2 -0.277
##  7 snp_22_45144106 ENSG00000100299.12     -3.73 2.14e- 4 3.14e- 2 -0.311
##  8 snp_22_23699601 ENSG00000075234.12      3.64 3.06e- 4 3.91e- 2  0.736
##  9 snp_22_34896078 ENSG00000184674.7      -3.53 4.53e- 4 4.98e- 2 -2.28
## 10 snp_22_26955854 ENSG00000100299.12     -3.51 4.86e- 4 4.98e- 2 -0.447
```

The model built without using the PCs as covariates had a different p-values than the one built using the covariates, but both found 6 significative associations on the 32 gene-SNP pairs we tested. The matrixeQTL analysis found 10 significative associations, but that is because it performed 1024 tests, testing all possible combinations of gene-SNP pairs. Therefore it tested several combination that were not included in the models built in Task 6. Furthermore, the most significant hit found by eQTL analysis was found also by the models built in Task 6 a and b. Also, we can see that, except for the gene-SNP associations that were not tested in Task 6 a, and for the FDR values, the results of the eQTL analysis and the model built in Task 6 are the identical. The difference in the FDR values is due to the fact that a larger number of tests were performed in MatrixeQTL (1024 versus 32), therefore the p-values were more penalized.