Handin 2 part2.2

Stefano Pellegrini

9/29/2020

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

Part 2

Task 1 - First, we will explore the data.

```
# Expression data for chromosome 20
expr20 <- read.table("expr_ceu_chr20.tab", header = TRUE)
expr20 <- column_to_rownames(expr20, var="id")
dim(expr20)
```

```
## [1] 561  91
```

```
# Gene positions for genes on chromosome 20
expr20_pos <- read.table("expr_chr20.pos", header = TRUE)
dim(expr20_pos)
```

```
## [1] 561   4
```

```
# Genotype data for chromosome 20
geno20 <- read.table("geno_ceu_chr20_strict.tab", header = TRUE)
geno20 <- column_to_rownames(geno20, var="id")
dim(geno20)
```

```
## [1] 30000    91
```

```
# Position of genotype data for chromosome 20
geno20_pos <- read.table("geno_ceu_chr20_strict.pos", header = TRUE)
dim(geno20_pos)
```

```
## [1] 30000     3
```

```
# Genotype data for chromosome 22
geno22 <- read.table("geno_ceu_chr22_strict.tab", header = TRUE)
geno22 <- column_to_rownames(geno22, var="id")
dim(geno22)
```

```
## [1] 1001   91
```

```
# Position of genotype data for chromosome 22
geno22_pos <- read.table("geno_ceu_chr22_strict.pos", header = TRUE)
dim(geno22_pos)
```

```
## [1] 1001    3
```

1. How many samples are included in this dataset?

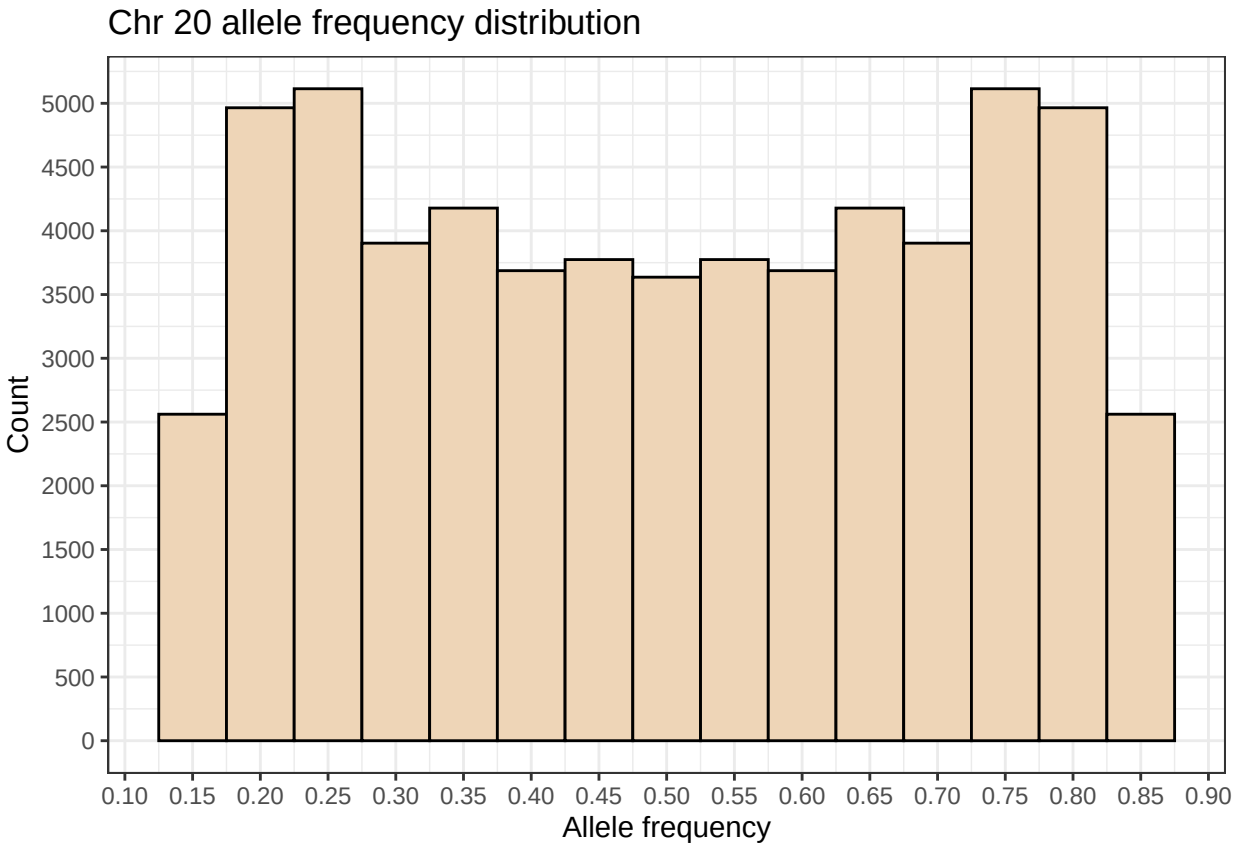
91 samples.

2. How many variants are present on chromosome 20?

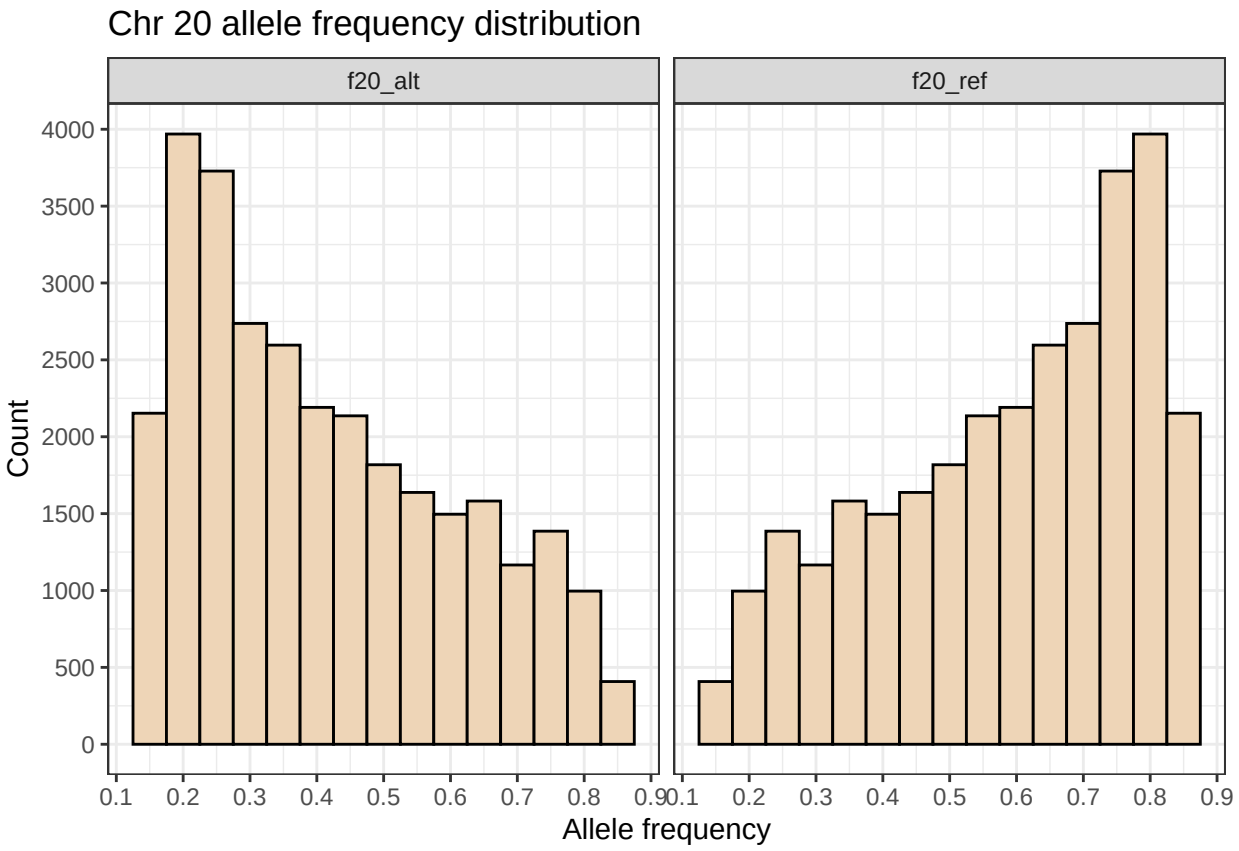
30000 variants.

3. Generate a histogram of allele frequencies for chromosome 20.

```
# Compute MAF
f20_alt <- apply(geno20, 1, function(x) mean(x[x > -1])) /  2
f20_ref <- 1 - f20_alt
maf20 <- data.frame(MAF = pmin(f20_alt, f20_ref))

# Plot histogram of allele frequencies
# Allele frequencies
cbind(data.frame(f20_alt), data.frame(f20_ref)) %>%
  as_tibble() %>%
  gather(key = "allele", value = "freq",
         f20_alt, f20_ref) %>%
  ggplot(aes(x=freq)) + geom_histogram(col="black", fill="bisque2", binwidth = 0.05) +
  labs(title = "Chr 20 allele frequency distribution") +
  ylab("Count") +
  xlab("Allele frequency") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
  theme_bw()
```



```r
# Alternative and reference allele frequencies
cbind(data.frame(f20_alt), data.frame(f20_ref)) %>%
  as_tibble() %>%
  gather(key = "allele", value = "freq",
         f20_alt, f20_ref) %>%
  ggplot(aes(x=freq)) + geom_histogram(col="black", fill="bisque2", binwidth = 0.05) +
  facet_wrap(~allele)  +
  labs(title = "Chr 20 allele frequency distribution") +
  ylab("Count") +
  xlab("Allele frequency") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 10)) +
  theme_bw()
```



4. What is the lowest allele frequency observed?

```r
maf20 %>% arrange(MAF) %>% head(1)
```

```
##                       MAF
## snp_20_1473605 0.1538462
```

The lowest allele frequency is 0.1538462.

5. How many genes are included?

561 genes are included in the data.

6. What gene shows the highest mean expression?

```
data.frame(mean_expr = apply(expr20, 1, mean)) %>%
  arrange(desc(mean_expr)) %>% head(1)
```

```
##                    mean_expr
## ENSG00000227063.4  3931.881
```

The gene with highest mean expression is ENSG00000227063.4.

Task 2 - cis-eQTL

```
# Genotype file names
SNP_file_name = "geno_ceu_chr20_strict.tab"
snps_location_file_name = "geno_ceu_chr20_strict.pos"

# Gene expression file names
expression_file_name = "expr_ceu_chr20.tab"
gene_location_file_name = "expr_chr20.pos"

# Only associations significant at this level will be saved
pvOutputThreshold_cis = 1;      # p.value threshold for cis eqtls
pvOutputThreshold_tra = 0;      # p.value threshold for trans eqtls

# Covariates file names
covariates_file_name = character(); # Set to character() for no covariates

# Distance for local gene-SNP pairs
cisDist = 1e6;                  # Define cis distance

# Load genotype data
snps = SlicedData$new();
snps$fileDelimiter = "\t";      # the TAB character
snps$fileOmitCharacters = "NA"; # Denote missing values;
snps$fileSkipRows = 1;          # One row of column labels
snps$fileSkipColumns = 1;       # One column of row labels
snps$fileSliceSize = 2000;      # Read file in slices of 2,000 rows
snps$LoadFile(SNP_file_name);
```

```
## Rows read: 2,000
## Rows read: 4,000
## Rows read: 6,000
## Rows read: 8,000
## Rows read: 10,000
## Rows read: 12,000
## Rows read: 14,000
## Rows read: 16,000
## Rows read: 18,000
## Rows read: 20,000
## Rows read: 22,000
## Rows read: 24,000
```

```
## Rows read: 26,000
## Rows read: 28,000
## Rows read: 30,000
## Rows read: 30000 done.
```

```
# Load gene expression data
gene = SlicedData$new();
gene$fileDelimiter = "\t";      # The TAB character
gene$fileOmitCharacters = "NA"; # Denote missing values;
gene$fileSkipRows = 1;          # One row of column labels
gene$fileSkipColumns = 1;       # One column of row labels
gene$fileSliceSize = 2000;      # Read file in slices of 2,000 rows
gene$LoadFile(expression_file_name);
```

```
## Rows read: 561 done.
```

```
# Load position files
snpspos = read.table(snps_location_file_name, header = TRUE, stringsAsFactors = FALSE);
genepos = read.table(gene_location_file_name, header = TRUE, stringsAsFactors = FALSE);

# Run the analysis
me = Matrix_eQTL_main(
  snps = snps,
  gene = gene,
  output_file_name=NULL,
  pvOutputThreshold = pvOutputThreshold_tra,
  useModel = modelLINEAR,
  errorCovariance =numeric(),
  verbose = TRUE,
  output_file_name.cis = NULL,    # Do not write out cis results
  pvOutputThreshold.cis = pvOutputThreshold_cis,
  snpspos = snpspos,
  genepos = genepos,
  cisDist = cisDist,
  min.pv.by.genesnp = FALSE,
  noFDRsaveMemory = FALSE,
  pvalue.hist = FALSE)
```

```
## Matching data files and location files
## 561 of 561 genes matched
## 30000 of 30000 SNPs matched
## Task finished in 0.02 seconds
## Reordering SNPs
## Task finished in 0.45 seconds
## Reordering genes
## Task finished in 0.33 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 0.01 seconds
## Performing eQTL analysis
##  6.66% done, 56,158 cis-eQTLs
## 13.33% done, 101,338 cis-eQTLs
## 20.00% done, 113,643 cis-eQTLs
## 26.66% done, 127,676 cis-eQTLs
## 33.33% done, 136,430 cis-eQTLs
## 40.00% done, 166,641 cis-eQTLs
## 46.66% done, 187,389 cis-eQTLs
## 53.33% done, 245,058 cis-eQTLs
## 60.00% done, 286,249 cis-eQTLs
## 66.66% done, 331,024 cis-eQTLs
## 73.33% done, 377,660 cis-eQTLs
## 80.00% done, 409,288 cis-eQTLs
## 86.66% done, 425,208 cis-eQTLs
## 93.33% done, 456,812 cis-eQTLs
## 100.00% done, 527,117 cis-eQTLs
## Task finished in 3.24 seconds
##
```

```
cis_eqtls = me$cis$eqtls[,-c(5)]
cis_eqtls["beta_se"] = cis_eqtls["beta"]/cis_eqtls["statistic"]
rm(me)
```

```
dim(cis_eqtls)
```

```
## [1] 527117      6
```

1. How many tests were conducted?

527117 tests were performed.

2. Using a bonferroni correction ($\alpha = 0.05$), how many genes are significant?

```
# Add column with bonferroni correction
cis_eqtls <- cbind(cis_eqtls,
                   p.adj = p.adjust(cis_eqtls$pvalue, method ="bonferroni"))

# Filter for significant hits
cis_eqtls %>% filter(p.adj < 0.05) %>% summarise(length(snps))
```

```
##   length(snps)
## 1           71
```

There are 71 gene-snp significant associations.

3. What gene-snp pair shows the lowest pvalue? What is the effect size of this snp-gene pair?

```
cis_eqtls %>% arrange(p.adj) %>% head(1)
```

```
##              snps              gene statistic       pvalue      beta   beta_se
## 1 snp_20_37055875 ENSG00000196756.5 -9.420136 5.066679e-15 -8.146604 0.8648075
##          p.adj
## 1 2.670733e-09
```

The gene-snp pair that shows the lowest p-value is SNP 20_37055875 and ENSG00000196756.5 gene. The effect size of the snp-gene pair is -8.146604.

4. What is the biotype of this gene?

It is a small nucleolar RNA non-coding host gene.

Task 3 - trans-eQTL

```
# Genotype file names
SNP_file_name = "geno_ceu_chr22_strict.tab" ;
snps_location_file_name = "geno_ceu_chr22_strict.pos" ;

# Gene expression file names
expression_file_name = "expr_ceu_chr20.tab" ;
gene_location_file_name = "expr_chr20.pos" ;

# Only associations significant at this level will be saved
pvOutputThreshold_cis = 0;                    # p.value threshold for cis eqtls
pvOutputThreshold_tra = 1;                    # p.value threshold for trans eqtls

#Covariates file names
covariates_file_name = character(); # Set to character() for no covariates

# Distance for local gene-SNP pairs
cisDist = 1e6;                        # Define cis distance

## Load genotype data
snps = SlicedData$new();
snps$fileDelimiter = "\t";            # The TAB character
snps$fileOmitCharacters = "NA";       # Denote missing values;
snps$fileSkipRows = 1;                # One row of column labels
snps$fileSkipColumns = 1;             # One column of row labels
snps$fileSliceSize = 2000;            # Read file in slices of 2,000 rows
snps$LoadFile(SNP_file_name);
```

```
## Rows read: 1001 done.
```

```
## Load gene expression data
gene = SlicedData$new();
gene$fileDelimiter = "\t";            # The TAB character
gene$fileOmitCharacters = "NA";       # Denote missing values;
gene$fileSkipRows = 1;                # One row of column labels
gene$fileSkipColumns = 1;             # One column of row labels
gene$fileSliceSize = 2000;            # Read file in slices of 2,000 rows
gene$LoadFile(expression_file_name);
```

```
## Rows read: 561 done.
```

```
#Load position files
snpspos = read.table(snps_location_file_name, header = TRUE, stringsAsFactors = FALSE);
genepos = read.table(gene_location_file_name, header = TRUE, stringsAsFactors = FALSE);

## Run the analysis
me = Matrix_eQTL_main(
  snps = snps,
  gene = gene,
  output_file_name=NULL,
  pvOutputThreshold = pvOutputThreshold_tra,
  useModel = modelLINEAR,
  errorCovariance =numeric(),
  verbose = TRUE,
  output_file_name.cis = NULL,      # Do not write out cis results
  pvOutputThreshold.cis = pvOutputThreshold_cis,
  snpspos = snpspos,
  genepos = genepos,
  cisDist = cisDist,
  min.pv.by.genesnp = FALSE,
  noFDRsaveMemory = FALSE,
  pvalue.hist = FALSE)
```

```
## 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 0.02 seconds

## Performing eQTL analysis

## 100.00% done, 561,561 eQTLs

## Task finished in 0.75 seconds

##
```

```
trans_eqtls = me$all$eqtls[,-c(5)]
trans_eqtls["beta_se"] = trans_eqtls["beta"]/trans_eqtls["statistic"]
rm(me)
```

1. How many tests were conducted?

```
dim(trans_eqtls)
```

```
## [1] 561561      6
```

561561 tests were conducted.

2. Using a bonferroni correction ($\alpha = 0.05$), how many genes are significant?

```
# Add column with bonferroni correction
trans_eqtls <- cbind(trans_eqtls,
                     p.adj = p.adjust(trans_eqtls$pvalue, method ="bonferroni"))
```

```r
# Filter for significant hits
trans_eqtls %>% filter(p.adj < 0.05) %>% summarise(length(snps))
```

```
##   length(snps)
## 1            0
```

There aren't any significant associations.

Task 4 - QQ-plot

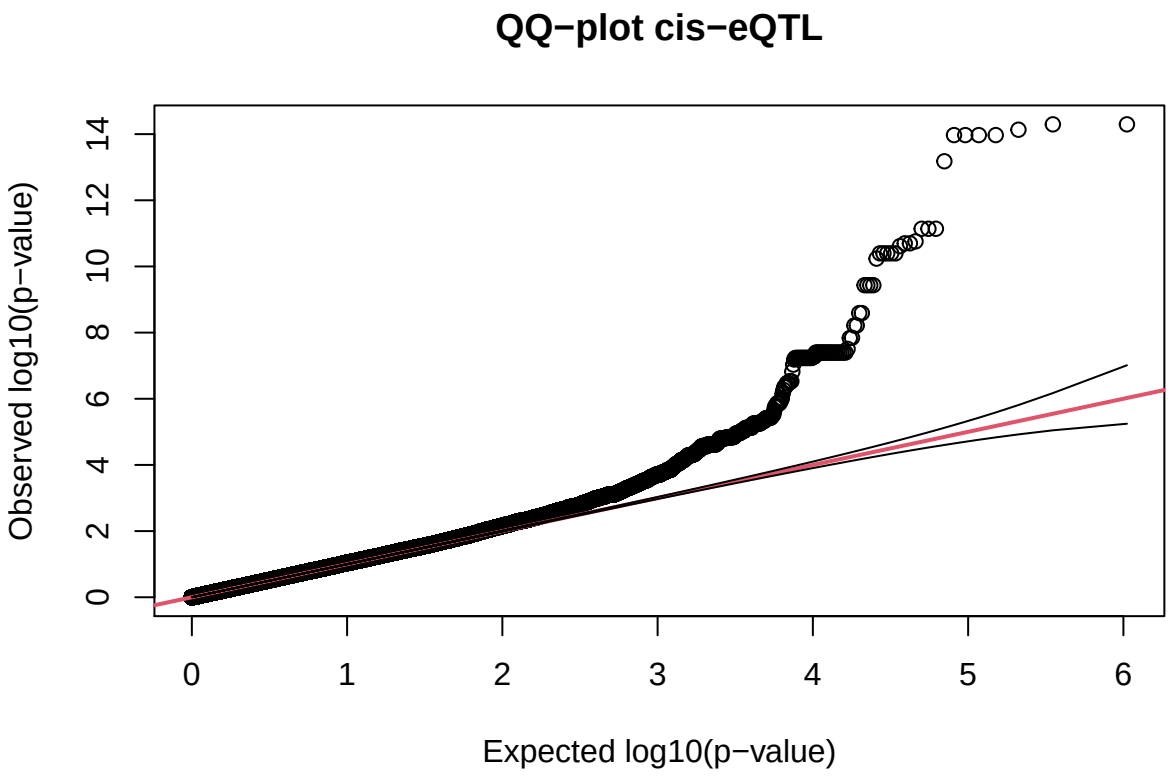
```r
qqp <- function(x, title = NaN, maxLogP = 30, ...){
  x <- x[!is.na(x)]
  if(!missing(maxLogP)){
    x[x < 10^-maxLogP] <- 10^-maxLogP
  }

  N <- length(x)
  chi1 <- qchisq(1 - x, 1)
  x <- sort(x)
  e <- -log((1:N - 0.5) / N, 10)
  plot(e, -log(x, 10),
       ylab = "Observed log10(p-value)",
       xlab = "Expected log10(p-value)",
       main = title,
       ...)
  abline(0, 1, col = 2, lwd = 2)
  c95 <- qbeta(0.95, 1:N, N - (1:N) + 1)
  c05 <- qbeta(0.05, 1:N, N - (1:N) + 1)
  lines(e, -log(c95, 10))
  lines(e, -log(c05, 10))
}
```

1. Briefly explain what a QQ-plot can be used for (2-3 sentences)

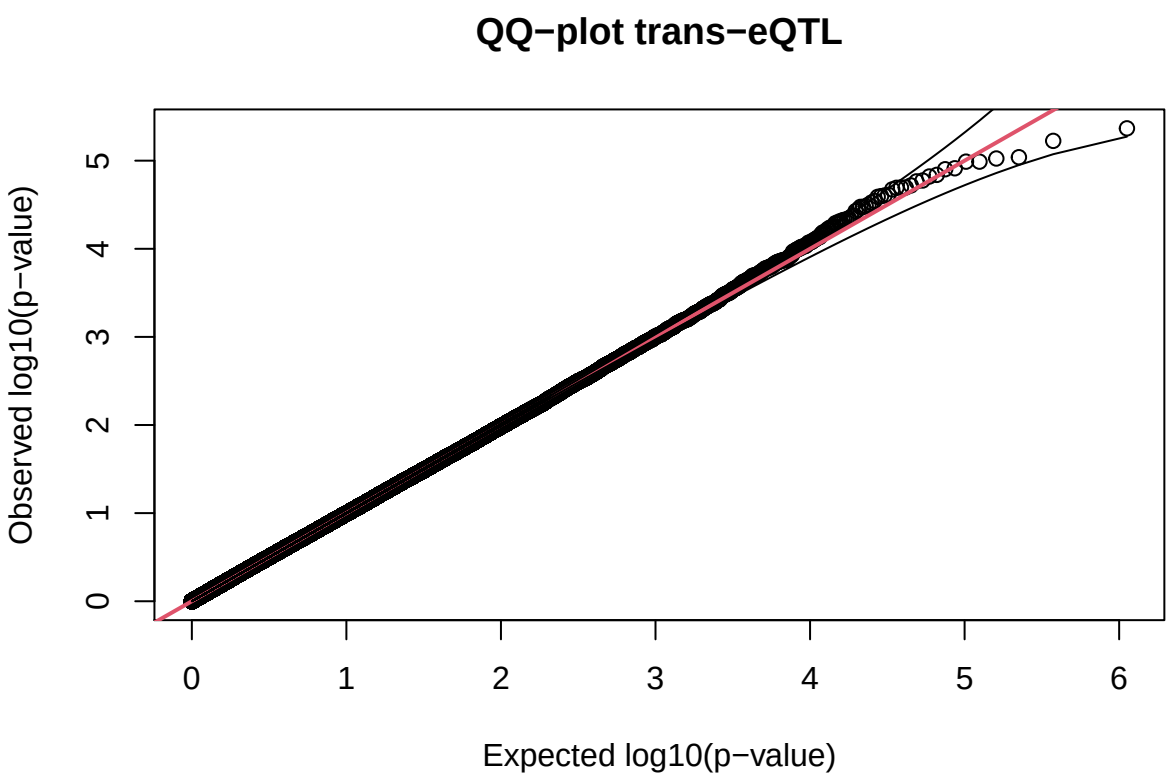
A QQ-plot is a scatter plot where the quantiles of one dataset is plotted against the quantiles of another one. The plot is used to visually compare the two distributions. If both sets of quantiles come from the same or similar distributions, the plot will approximately match the diagonal line $y = x$. Also, many distributional aspects can be visually tested, and in particular, in eQTL analysis it can be useful to check for the presence of population stratification, filtering bias, polygenicity, p-values inflation, batch effect, or other bias in the data.

2. Compute the QQ-plot for both the cis and trans eqtl separately

```r
qqp(cis_eqtls$pvalue, title= "QQ-plot cis-eQTL")
```




```
qqp(trans_eqtls$pvalue, title= "QQ-plot trans-eQTL")
```



3. Explain the plots In the case of eQTL analysis we are plotting the $\log_{10}$ of the observed p-values against the $\log_{10}$ p-values distribution that we would expect to have by chance (if there is no significant association between gene-SNP pairs), which is approximated using the beta distribution. Therefore the p-values above the diagonal line are the significant ones, while the ones that match it, or that fall within the confidence interval bounds, are not small enough to be significant.

4. What is the main difference between these two QQ-plots? We can see that the QQ plot of the cis-eQTL p-values shows a deviation from the diagonal line on the right side, the p-values above the diagonal line are the significant ones (after Bonferroni correction), and the most signifcant hit is located at the top right corner. While in the trans-eQTL QQ plot, as expected, the distribution of the observed p-values matches the expected ones, meaning that no significant gene-SNP pair association was found.

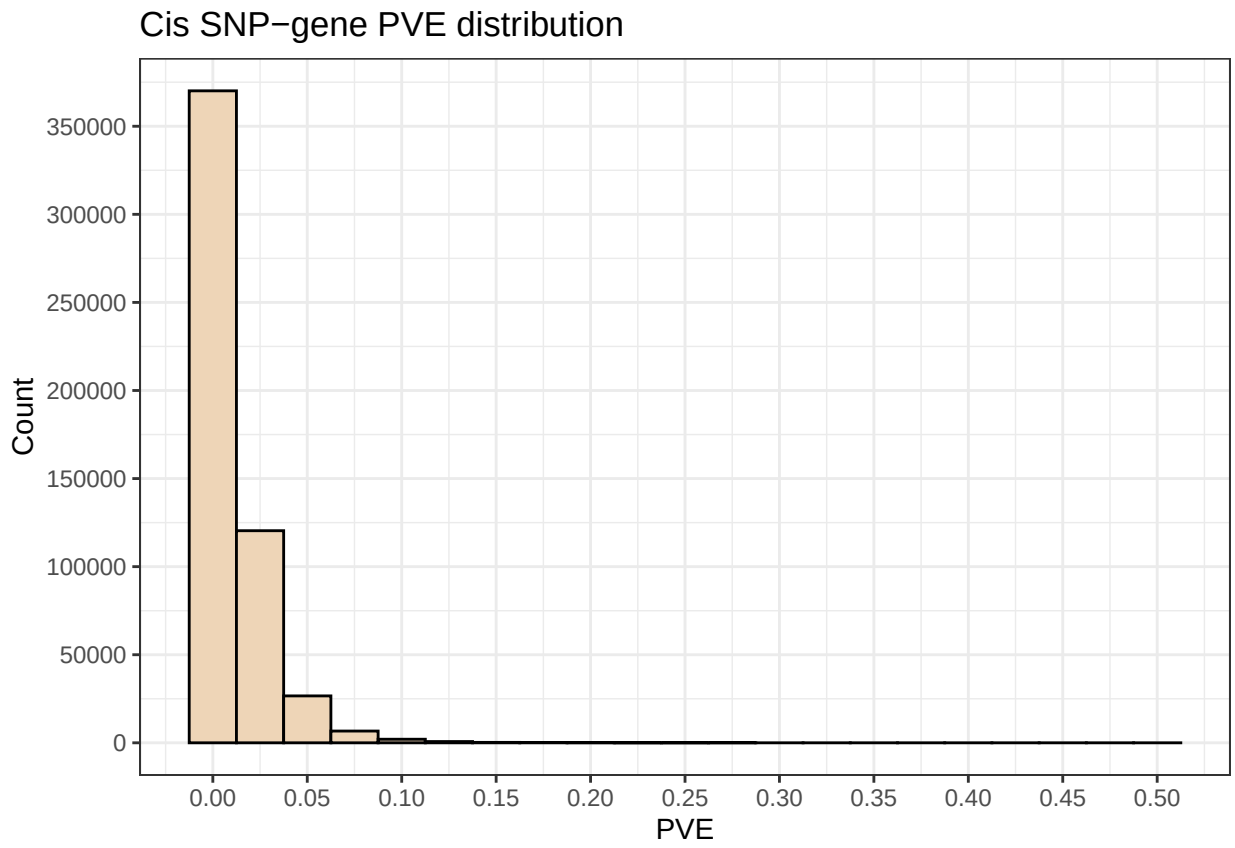
5. Explain what drives this? It is likely that there is a larger number of significant associations between genes and closely located SNPs, compared to the associations that can be found betwen genes and distant located SNPs. In fact, a mutation close to a gene is more likely to have an influence on its expression, compared to a mutation that occurred far from it. This is especially true when the gene and the SNPs are located on different chromosomes (in our case chromosome 20 and chromosome 22). Therefore, for this reason and for other technical aspects, finding distant trans eQTLs is more challenging and requires many more tests.

Task 5 - PVE

1. Calculate the PVE for all cis SNP-gene pairs and make a histogram of them

```
# Compute the PVE for all cis SNP-gene pairs
N = 91
cis_eqtls %>%
  mutate(maf = maf20[snps,1]) %>%
```

```
  mutate(pve = (2* (beta^2) * maf * (1 - maf)) /
    (2 * (beta^2) * maf * (1 - maf) + (beta_se^2) * 2 * N * maf * (1 - maf))) -> cis_eqtls_pve

# PVE histogram
cis_eqtls_pve %>% as_tibble() %>%
  ggplot(aes(x=pve)) + geom_histogram(col="black", fill="bisque2", binwidth = 0.025) +
  labs(title = "Cis SNP-gene PVE distribution") +
  ylab("Count") +
  xlab("PVE") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 10)) +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 20)) +
  theme_bw()
```



2. What gene has the highest PVE

```
cis_eqtls_pve %>% arrange(desc(pve)) %>% head(1)
```

```
##              snps              gene statistic       pvalue      beta   beta_se
## 1 snp_20_37055875 ENSG00000196756.5 -9.420136 5.066679e-15 -8.146604 0.8648075
##          p.adj       maf       pve
## 1 2.670733e-09 0.1593407 0.4937102
```

```
maf20["snp_20_37055875",]
```

```
## [1] 0.1593407
```

The gene-snp pair with the highest PVE is gene ENSG00000196756.5, associated with SNP 20_37055875.

3. what other factors can explain the remaining variance (mention 2)?

The remaining variance can be explained by the influence of several other SNPs to the gene expression, but also by environmental factors, confounders, noise and bath effect.