Project 2

Monikrishna Roy

2021-11-27

Question 1

Retrieve data matrix and patient group information of the dataset GSE19804 from NIH GEO: http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE19804 (Links to an external site.).

Hint: Use getGEO, exprs, and pData functions to retrieve the data matrix and group information.

```
dataset <- "GSE19804"
gsets <- getGEO(dataset, GSEMatrix = T, getGPL = T)
gset <- gsets[[1]]
expr <- exprs(gset)

pdata <- pData(gset)
control <- rownames(pdata[grep("Lung Normal", pdata$title), ])
cancer <- rownames(pdata[grep("Lung Cancer", pdata$title), ])
```

Question 2

Perform a t-test to compare the control and condition groups, compute the difference in mean log base 2 expression and create an output data frame that contains the following columns: gene ids (row names of expression data matrix), p-value, t-score, logFC;

```
# a function to calculate the difference in mean
cal_mean_diff <- function(x, cancer, control) {
  mean(x[cancer]) - mean(x[control])
}
# Function to calculate p-value
cal_p_value <- function(x, cancer, control) {
  t.test(x[cancer], x[control])$p.value
}
# Function to calculate t-score
cal_t_score <- function(x, cancer, control) {
  t.test(x[cancer], x[control])$statistic
}

# used apply to call the function
logFC <- apply(expr, 1, cal_mean_diff, cancer, control)
PValue <- apply(expr, MARGIN = 1, FUN = cal_p_value, cancer, control)
TScore <- apply(expr, MARGIN = 1, FUN = cal_t_score, cancer, control)

# rownames used as gene ids
geneIds <- rownames(expr)

df <- data.frame(
```

```
  row.names = NULL,
  "GeneID" = geneIds,
  "PValue" = PValue,
  "TScore" = TScore,
  "LogFC" = logFC
)
```

Question 3

Use absolute log fold change > 1 and raw p-value < 0.05 to select the differentially expressed (DE) genes. Show the Volcano plot. Color the DE genes in red.''

```
plot(
  x = df$LogFC,
  y = -log10(df$PValue),
  xlab = "logFC",
  ylab = "-log10(p-value)",
  main = "Volcano plot",
  col = ifelse(abs(df$LogFC) > 1 & df$PValue < 0.05, "red", "black")
)
abline(h = -log10(0.05), col = "red")
abline(v = -1, col = "blue")
abline(v = 1, col = "blue")
```

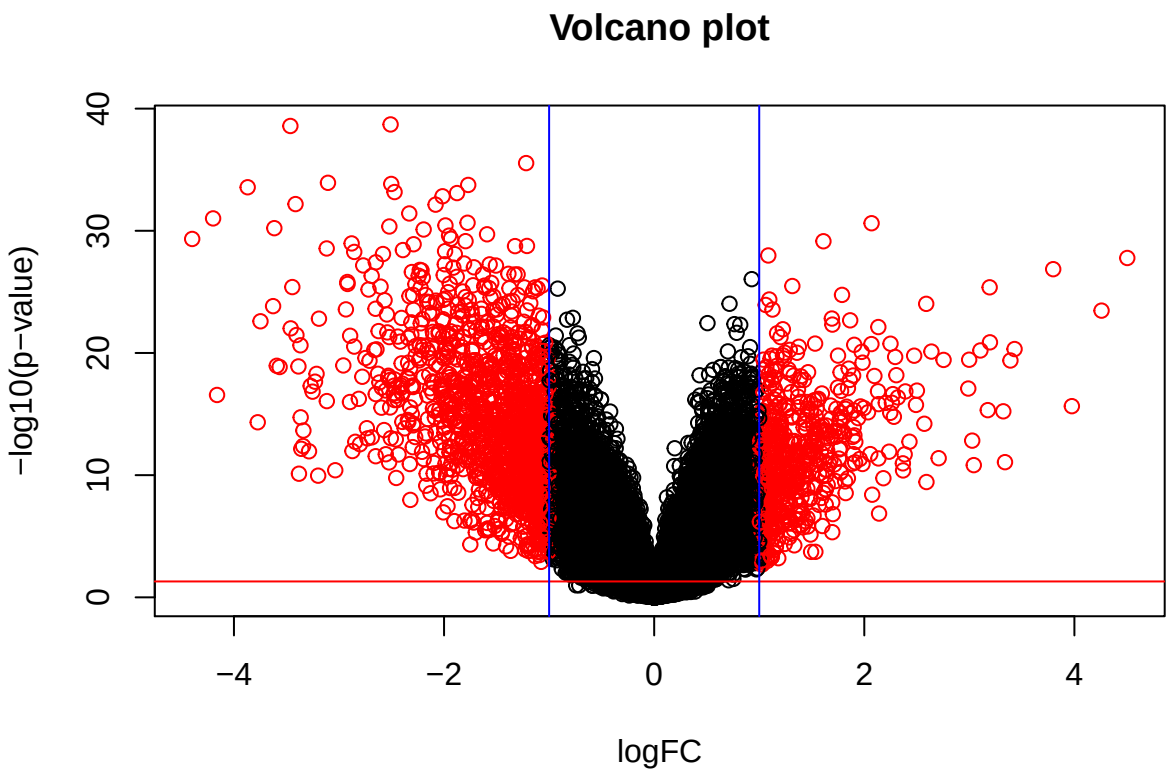

Figure 1: Volcano plot

Question 4

Calculate the t-SCORE between the two groups, for each row, and store it in a data frame called t-OBSERVED.

```r
# Function to calculate t-score
cal_t_score <- function(x, cancer, control) {
  t.test(x[cancer], x[control])$statistic
}

TScore <- apply(expr, MARGIN = 1, FUN = cal_t_score, cancer, control)
t_OBSERVED <- data.frame(TScore)
```

Question 5

Calculate the empirical p-values for the t-SCORES using a permutation analysis as follows:

- Permute the samples of the original matrix and recalculate the t-SCORE 100 times for each row.
- Store the new scores in a matrix called t_NULL_DISTRIBUTION that will have 54675 rows (total number of genes in the given matrix) and 100 columns.
- Count how many times an observed value, from vector t-OBSERVED, is more extreme than the values in the respective row in the permuted matrix
- Divide this count by the number of columns (100) and note that this value will be the empirical p-value.
- Save the p-values in the vector pT.

```r
permuteExpr <- expr
t_NULL_DISTRIBUTION <- lapply(c(1:100), function(i) {
  message(i)
  colnames(permuteExpr) <- sample(colnames(expr))
  apply(permuteExpr, MARGIN = 1, FUN = cal_t_score, cancer, control)
}) %>%
  do.call(what = cbind) %>%
  as.data.frame()

pT <- lapply(rownames(t_OBSERVED), function(r) {
  sum(abs(t_NULL_DISTRIBUTION[r, ]) > abs(t_OBSERVED[r, ])) / ncol(t_NULL_DISTRIBUTION)
}) %>% unlist()
```

Question 6

Repeat problem 4 and 5 but use Euclidean distance instead. Follow the following steps:

- Calculate the e-SCORE (Euclidean distance between the means of the two groups) for each row. Store it in a data frame called e-OBSERVED.
- Permute the samples of the original matrix and recalculate the e-SCORE 100 times for each row. Store the new scores in a matrix called e_NULL_DISTRIBUTION that will have 54675 rows (total number of genes in the given matrix) and 100 columns.
- Count how many times an observed value, from vector e-OBSERVED, is more extreme than the values in the respective row in the permuted matrix
- Divide this count by the number of columns (100) and note that this value will be the empirical p-value.
- Save the p-values in the vector pE.

```r
# Function to calculate Euclidean distance between the means of the two groups
cal_e_score <- function(x, cancer, control) {
  dist(rbind(mean(x[cancer]), mean(x[control])), method = "euclidean")
}

eSCORE <- apply(expr, MARGIN = 1, FUN = cal_e_score, cancer, control)
e_OBSERVED <- data.frame(eSCORE)

e_NULL_DISTRIBUTION <- lapply(c(1:100), function(i) {
```

```
  message(i)
  samples <- sample(ncol(expr), replace = F)
  controls <- samples[1:length(control)]
  cancers <- samples[(length(control) + 1):length(samples)]
  apply(
    expr,
    MARGIN = 1,
    FUN = cal_e_score,
    cancer = cancers,
    control = controls
  )
}) %>%
  do.call(what = cbind) %>%
  as.data.frame()

pE <- lapply(rownames(e_OBSERVED), function(r) {
  mean(abs(e_NULL_DISTRIBUTION[r, ]) > abs(e_OBSERVED[r, ]))
}) %>% unlist()
```

Question 7

Plot the histogram of the p-values (plot pT and pE separately).

```
par(mfrow = c(1, 2))
hist(pT, breaks = 100, col = "red", main = "pT")
hist(pE, breaks = 100, col = "blue", main = "pE")
```

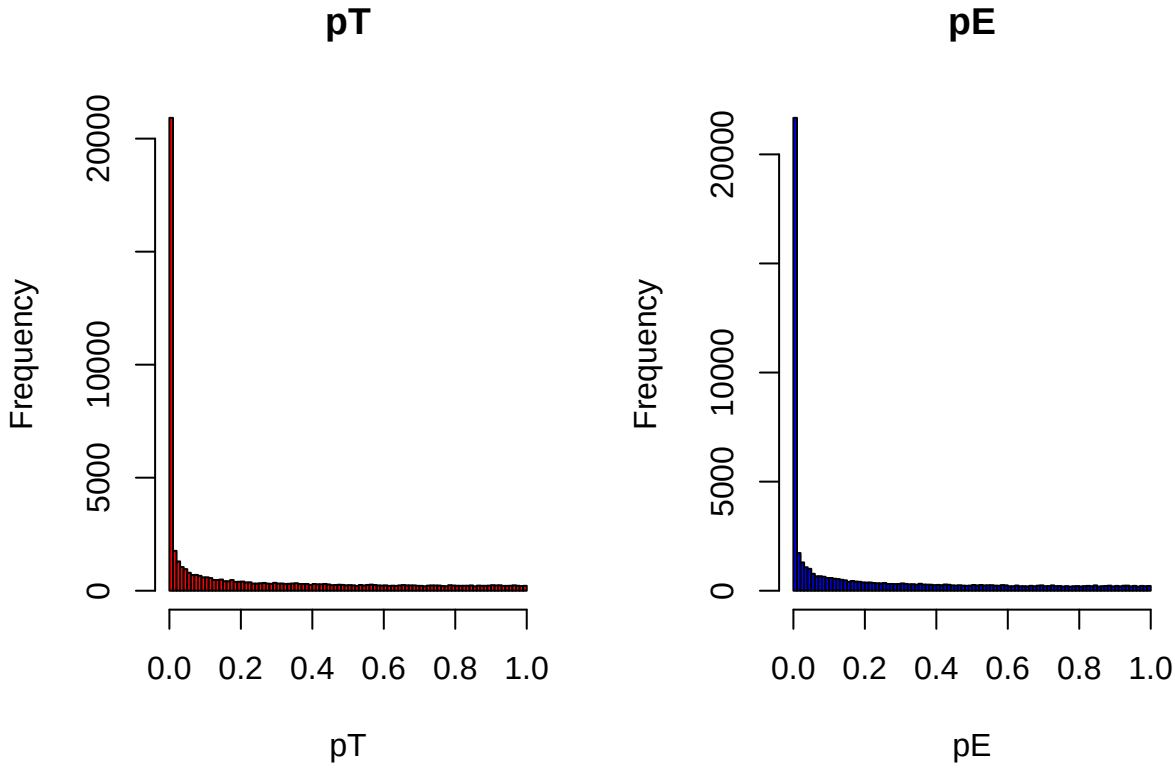

Figure 2: Plot the histogram of the p-values

Question 8

Calculate the correlation between the two p-value vectors (pT and pE).

```
cor(pT, pE)
```

```
## [1] 0.9897887
```

Question 9

Submit your code and discuss the results in the report.

- The correlation between emperical p-values is high, so they are very correlated even we used different methods to calulate the p-value.