Differential expression analysis of dermal endothelial cells of healthy and type 2 diabetic patients

Date: February 5th, 2022

Author: Daianna González Padilla

Subject: RNAseq data analysis

Content

1. Abstract

This study accomplish a differential expression analysis of RNAseq data from isolated dermal endothelial cells from diabetic and control patients.

2. Objetive

The main objective is to identify differentialy expressed genes between healthy and diabetic patients in order to gain biological insights into its possible role in processes affected by diabetes and ultimately target those genes or processes for therapeutic and pharmaceutic approaches.

3. Motivation

According to the first *Global report on diabetes* of the World Health Organization (WHO), the number of adults living with diabetes has almost quadrupled since 1980 to 422 million adults. This dramatic rise is largely due to the rise in type 2 diabetes [1].

Since diabetes has a high prevalence in the whole world and is a major cause of blindness, kidney failure, heart attacks, stroke and lower limb amputation, this disease has been one of the most studied and an important approach to understand its genetic fundaments is transcriptomics. RNAseq data analysis such as the one presented here are important to establish a complete biological scenario of diabetes causes and implicated processes.

4. Results

This analysis took the linear model *Y ~ Intercept + Disease state + Gender* to model gene expression and showed that there are 72 DE genes between healthy and diabetic patients, some implicated in insulin signaling. Most of the DE genes in diabetic patients are downregulated and KRT10 gene is highly upregulated in healthy controls.

5. Data

RNAseq data used is available on R package `recount3` as “SRP095512” project of type *data_source*, from *human*, with *gene* as data type and *gencode_v26* as selected annotation. This data are part of just one *assay* describing RNA expression data for 63,856 *genes* along 10 *samples* corresponding to isolated dermal endothelial cells from 4 diabetic patients (Pat) and 6 control individuals (Ctrl). A deeper description of this data is found in `recount3` Study Explorer and within the same data.

6. Installations

Install R 4.0.x or RStudio version 1.4 (or newer) and Bioconductor R packages.

7. Methodology

7.1 Obtain data and *RangedSummarizedExperiment* R object

Intuitively, the first step to analyze RNAseq data is to import the data through `recount3` R package. After that, a `RangedSummarizedExperiment` (RSE) object must be created with the data to analyze them in matrix-like containers called assays where rows represent ranges of interest (genes in this case) and columns represent samples (patients isolated cells in this case).

```
## Load recount3 R/Bioconductor package
library("recount3")
## Get all human projects available in recount3
human_projects <- available_projects()
## Find interest project
proj_info <- subset(
    human_projects,
    project == "SRP095512" & project_type == "data_sources"
)
## Create a RangedSummarizedExperiment (RSE) object
rse_gene_SRP095512 <- create_rse(proj_info)

## Convert raw counts to read counts
assay(rse_gene_SRP095512, "counts") <- compute_read_counts(rse_gene_SRP095512)
```

7.2 Exploratory Data Analysis and Quality Control

Once the RSE object is created, the next step is to explore the relation between the samples variables to assess overall similarity between them. This is a useful step to determine at first sight which samples are different and similar to each other, if that fits the expectation from the experiment design and to identify the major sources of variation [3]. This information helps to create a design model in future steps for DE analysis.

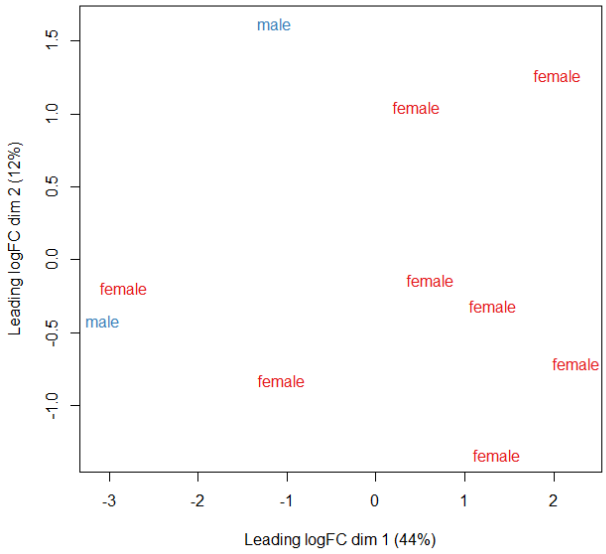
```
## Check samples attributes
rse_gene_SRP095512$sra.sample_attributes
## [1] "cell type;;endothelial cell|disease state;;Healthy control|gender;;female|source_name;;dermal blood endothelial cell"
## [2] "cell type;;endothelial cell|disease state;;Diabetic Patient|gender;;male|source_name;;dermal blood endothelial cell"
## ...
##[10] "cell type;;endothelial cell|disease state;;Healthy control|gender;;female|source_name;;dermal blood endothelial cell"

## Expand samples attributes to access them
rse_gene_SRP095512 <- expand_sra_attributes(rse_gene_SRP095512)

## Check samples information
colnames(colData(rse_gene_SRP095512))
```

As shown above, variable attributes of the samples are disease state and gender, and those are the ones to compare. Within sample information columns there are not variables describing directly the conditions of the samples but the experiment and libraries attributes.

Below samples are plotted on a two-dimensional scatterplot so that distances on the plot approximate the expression differences between the samples, so the hypothesis is that samples that belong to the same condition will be closer to each other and farther from the other conditions samples, only if these conditions determine different expression profiles . Both plots are the same, only the samples labels change in order to make same condition samples clusters more visible and obvious.

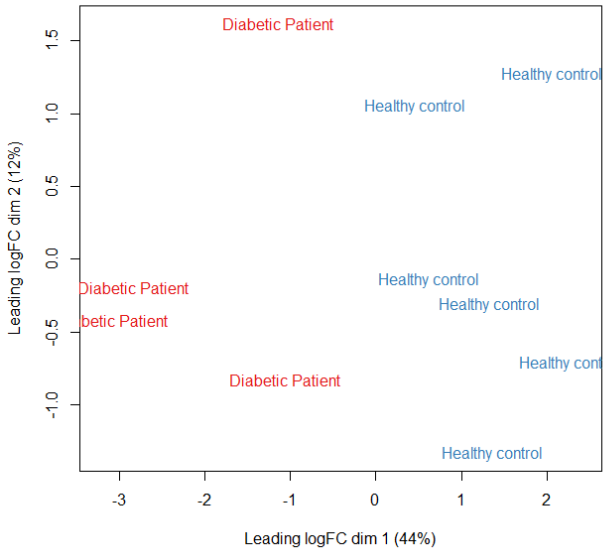
```
library("RColorBrewer")
## Each level of the condition is a different color
col.group <- factor(rse_gene_SRP095512$sra_attribute.gender)
levels(col.group) <- brewer.pal(nlevels(col.group), "Set1")
col.group
## [1] #E41A1C #377EB8 #E41A1C #E41A1C #377EB8 #E41A1C #E41A1C #E41A1C #E41A1C #E41A1C
## Levels: #E41A1C #377EB8 #4DAF4A

## Colors are chars
col.group <- as.character(col.group)
```

```
## MDS plot
plotMDS(rse_gene_SRP095512, labels = rse_gene_SRP095512$sra_attribute.gender, col=col.group)
```



By looking at the above image it is possible to conclude that gender is not an differential expression indicator since samples from the same gender do not cluster together. Even though, gender stills an interest variable since women and men could show different diabetic propensity. In fact, it has been shown that men are almost twice as likely to develop type 2 diabetes as women [2].

```
col.group <- factor(rse_gene_SRP095512$sra_attribute.disease_state)
levels(col.group) <- brewer.pal(nlevels(col.group), "Set1")
col.group
col.group <- as.character(col.group)
## Labels now given by disease sta
plotMDS(rse_gene_SRP095512, labels = rse_gene_SRP095512$sra_attribute.disease_state, col=col.group)
```

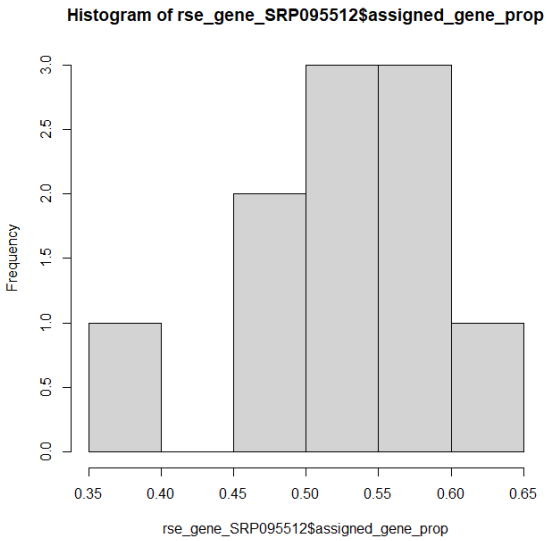


On the other hand, when samples labels are given by disease state, samples of healthy controls and diabetic patients group into two different groups at right and left, respectively. Thus, disease state works as a comparable condition that could explain differential gene expression.

Once the condition of interest is determined, the next thing to do is to get rid of samples and genes that do not overcome the minimal established values for quality describing attributes.

Initially, quality control of samples could be given by their *gene assigned reads proportion* as a measurement of RNA reads that correspond to transcripts of the samples. The closer this value is to 1 the better the sample transcription levels support given by reads.

```
## Calculate gene assigned reads proportion for each sample
rse_gene_SRP095512$assigned_gene_prop <- rse_gene_SRP095512$recount_qc.gene_fc_count_all.assigned / rse_gene_SRP095512$recount_qc.gene_fc_c
## Calculate the minimum proportion accepted: Median -3(Standard Deviation)
median(rse_gene_SRP095512$assigned_gene_prop)-3*sd(rse_gene_SRP095512$assigned_gene_prop)
## [1] 0.3239964

## General information of samples proportions
summary(rse_gene_SRP095512$assigned_gene_prop)
## Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.3665  0.4902  0.5349  0.5189  0.5608  0.6074

## Visualize graphically the frequency of samples proportions
hist(rse_gene_SRP095512$assigned_gene_prop)
```

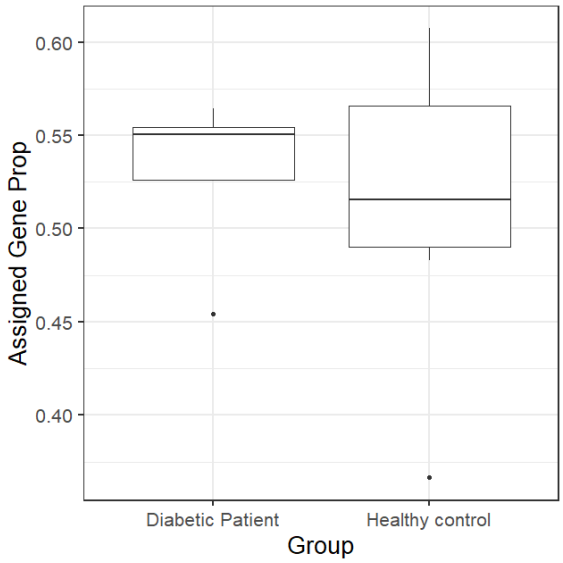


Since the minimum proportion value of the samples (0.3665) is greater than the minimum accepted value (0.32399), all samples are kept, as shown below.

```
## Samples with gene assigned proportion smaller than 0.32399
table(rse_gene_SRP095512$assigned_gene_prop < 0.32399)
## FALSE
## 10
```

Subsequently, an examination of differences between gene assigned reads proportions of disease state (cases and controls) groups is useful as a first clue to determine if there are huge differences in samples quality among these groups that could affect the differential expression results.

```
## Gene assigned reads proportion for control and cases samples
tapply(rse_gene_SRP095512$assigned_gene_prop, rse_gene_SRP095512$sra_attribute.disease_state, summary)
## $`Diabetic Patient`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4544  0.5260  0.5503  0.5298  0.5541  0.5641

## $`Healthy control`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3665  0.4902  0.5156  0.5116  0.5659  0.6074

## Boxplot of gene assigned reads proportion of controls and cases samples
library("ggplot2")
ggplot(as.data.frame(colData(rse_gene_SRP095512)), aes(y = assigned_gene_prop, x = sra_attribute.disease_state)) +
     geom_boxplot() +
     theme_bw(base_size = 20) +
     ylab("Assigned Gene Prop") +
     xlab("Group")
```



As shown above, no huge differences between the groups proportions is found and both have good quality samples. Nevertheless, a slight difference exists and that could account for those minority differentially expressed genes in the samples that lead to greater or smaller mRNA levels within samples.

Secondly, gene quality must be verified as well. To do that, it necessary to take into account the *mean expression level* of each gene.

```
## Mean of expression levels of each gene
gene_means <- rowMeans(assay(rse_gene_SRP095512, "counts"))
summary(gene_means)
##  Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.0      0.0      0.6    277.5     35.2 884351.6

## Number of genes with the first expression values
head(table(gene_means))
## gene_means
##     0   0.1   0.2   0.3   0.4   0.5
## 22606  3700  2189  1403  1141   879
```

The zero value for the first quartile of the mean expression levels of the genes lies in the fact that there are 22,606 of 63,856 total genes with mean expression zero, which means that around the 35% of genes have no expression values. Hence, those genes are discarded.

```
## Hold genes with mean expression greater than 0
rse_gene_SRP095512 <- rse_gene_SRP095512[gene_means >= 0.1, ]
## New dimensions
dim(rse_gene_SRP095512)
## [1] 41250    10
```

7.3 Data Normalization

The following step is to normalize the data for posterior differential expression analysis accounting for differences in library sizes given by sequencing depth, gene length and RNA composition of samples (*composition bias*) [3]. This is a necessary step to make accurate comparisons of gene expression between samples.

This is achieved through `edgeR` library which is able to estimate biological variation between replicate libraries and to conduct exact tests of significance.

```
## Load the necessary library
library("edgeR")
## Convert RSE object into a DGEList object to analyze it through edgeR
dge <- DGEList(
    counts = assay(rse_gene_SRP095512, "counts"),
```

```
    genes = rowData(rse_gene_SRP095512)
)
## Calculate scaling factors to convert raw library sizes into effective library sizes
dge <- calcNormFactors(dge)
```

7.4 Differential Gene Expression Analysis

The next step in the RNA-seq workflow is the differential expression analysis. The goal of differential expression testing is to determine which genes are expressed at different levels between conditions, in this case, between healthy and diabetic patients.

Fistable, a review of the samples variables integrity is necessary to discard the ones that are not informative or incomplete.

```
## Check samples variables integrity
table(rse_gene_SRP095512$sra_attribute.gender)
## female   male
##     8      2
table(rse_gene_SRP095512$sra_attribute.cell_type)
## endothelial cell
##              10
table(rse_gene_SRP095512$sra_attribute.disease_state)
## Diabetic Patient  Healthy control
##               4                6
table(rse_gene_SRP095512$sra_attribute.source_name)
## dermal blood endothelial cell
##                          10
```

As already mentioned earlier, only disease state and gender present variation between samples and are therefore informative. They are also complete since all samples fall into one of their categories.

Then, a regression model is created to model the expression of each gene as a linear combination of variables or explanatory factors from the samples: disease state and gender in this study. The outcome or response variable is the gene expression which depends on the factors parameters or coefficients estimated from the data.

Categorical data such as disease state and gender are represented as numerical variables through *dummy variables* (as 0s and 1s) with `model.matrix`, taking one as the reference and the rest as contrast.

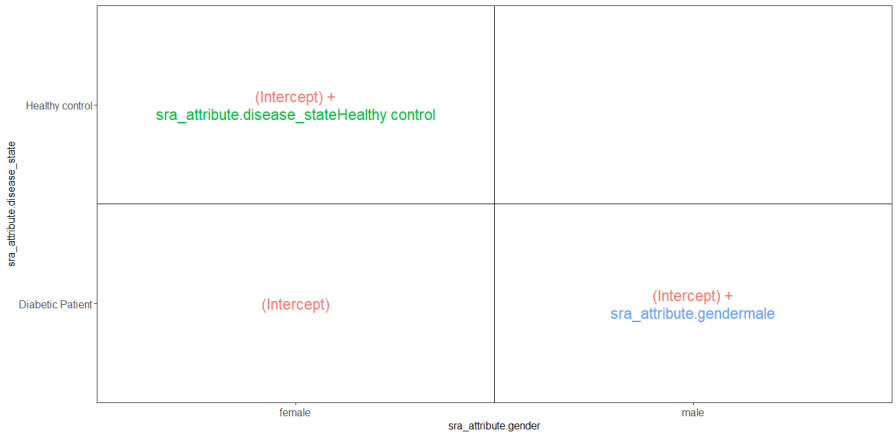
```
## Create the model Y ~ interceot + disease state + gender
mod <- model.matrix(~ sra_attribute.disease_state + sra_attribute.gender,
    data = colData(rse_gene_SRP095512)
)
```

Before using the model to get DE genes, it is important to determine if the matrix is *full rank* to avoid redundancy in the model parameters: columns must be linearly independent to estimate the coefficients values.

```
## Matrix rank must equal the number of columns
qr(mod)$rank==ncol(mod)
## [1] TRUE
```

This full rank matrix makes possible to continue with DE analysis and is also possible to visualize the matrix model design.

```
## Visualize the design of the matrix model
vd <- ExploreModelMatrix::VisualizeDesign(
    sampleData = colData(rse_gene_SRP095512),
    designFormula = ~ sra_attribute.disease_state + sra_attribute.gender,
    textSizeFitted = 4
)
cowplot::plot_grid(plotlist = vd$plotlist)
```

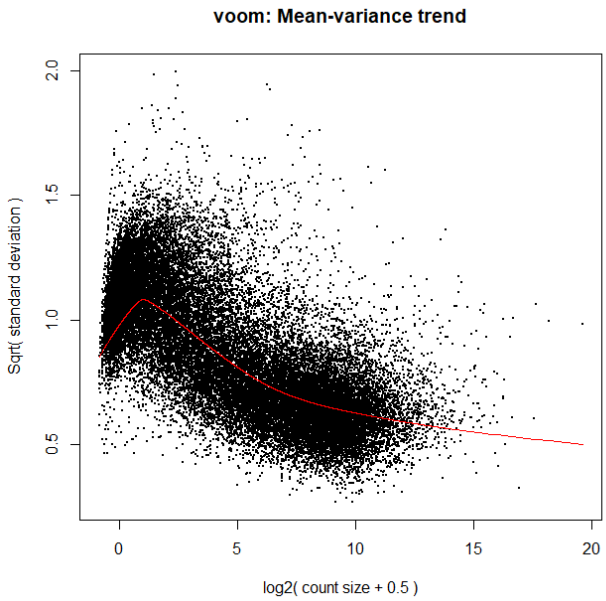


The above image makes easier to distinguish female and diabetic variables as references whose value is 0 and male and healthy variables as controls whose value is 1. In that way, just by resting the intercept value to healthy female and diabteic male coefficients, their values are obtained.

Once the linear model is created `limma` library uses it to realize the DGE analysis based on a NB distibution to model read counts and estimate gene expression variance which is essential to determine whether the changes are due to chance [4,5].

First, `voom` function estimates mean-variance relationship in the data prior to linear modelling in `limma`, then use this to compute appropriate weights for each observation [6].

```
library("limma")
## Estimate and plot mean-variance relation
vGene <- voom(dge, mod, plot = TRUE)
```



After that, the linear model fit is produced by `lmFit` and is taken by `eBayes` to compute *t-statistics* and *F-statistic* to measure the size of gene expression difference relative to the variation in the sample data. The greater these value are, the greater the evidence for differentialy expressed genes between samples of healthy and diabetic patients.

```
## Linear model fit and t values for genes
eb_results <- eBayes(lmFit(vGene))
## All DE genes with their t and p values for interest condition
de_results <- topTable(
    eb_results,
    ## Index of the interest coefficient (disease state)
    coef = 2,
    number = nrow(rse_gene_SRP095512),
    ## Conserve the original order of genes
    sort.by = "none")
dim(de_results)
## [1] 41250    16
```

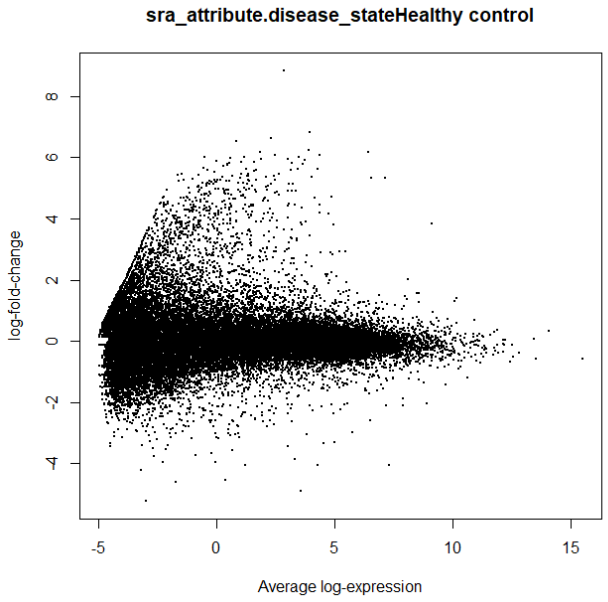
From the top ranked genes obtained above, only the ones with an adjusted p value smaller than 0.05 are considered DE genes so that the probability of observing those differences in their expression is not by chance but by disease state.

```
## DE genes between controls and cases with adjusted p value < 0.05
table(de_results$adj.P.Val < 0.05)
## FALSE  TRUE
## 41178    72
```

There are 72 DE genes that could account for the genetic differences between healthy and diabetic patients.

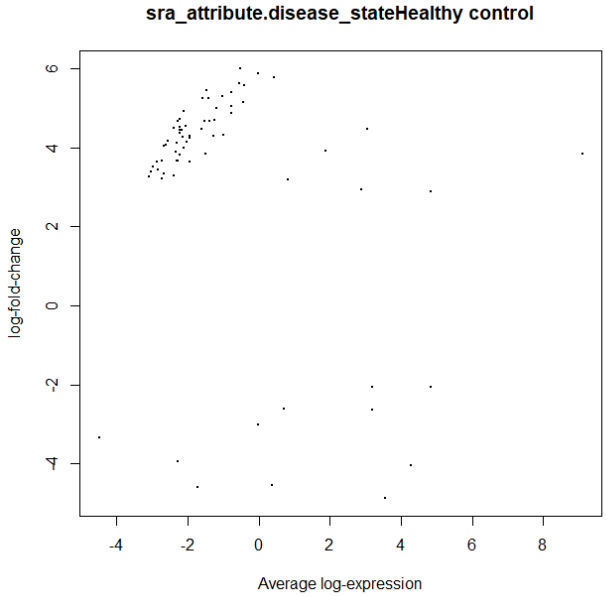
Below there is a MA plot that shows the relation between the mean expression of genes and their fold change (log ratio) regarding these two different conditions. [7]

```
## MA plot takes gene expression and disease state
limma::plotMA(eb_results, coef = 2)
```



To make clearer where DE genes are in the MA plot, they are the only ones held and plotted.

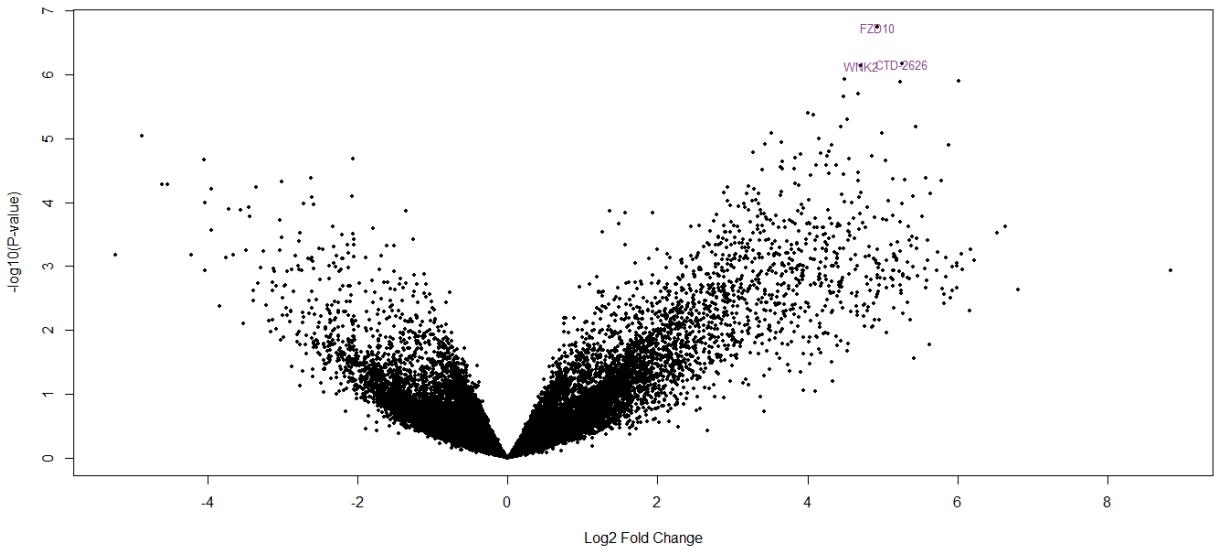
```
## Hold only DE genes
limma::plotMA(eb_results[which(de_results$adj.P.Val < 0.05),], coef=2)
```



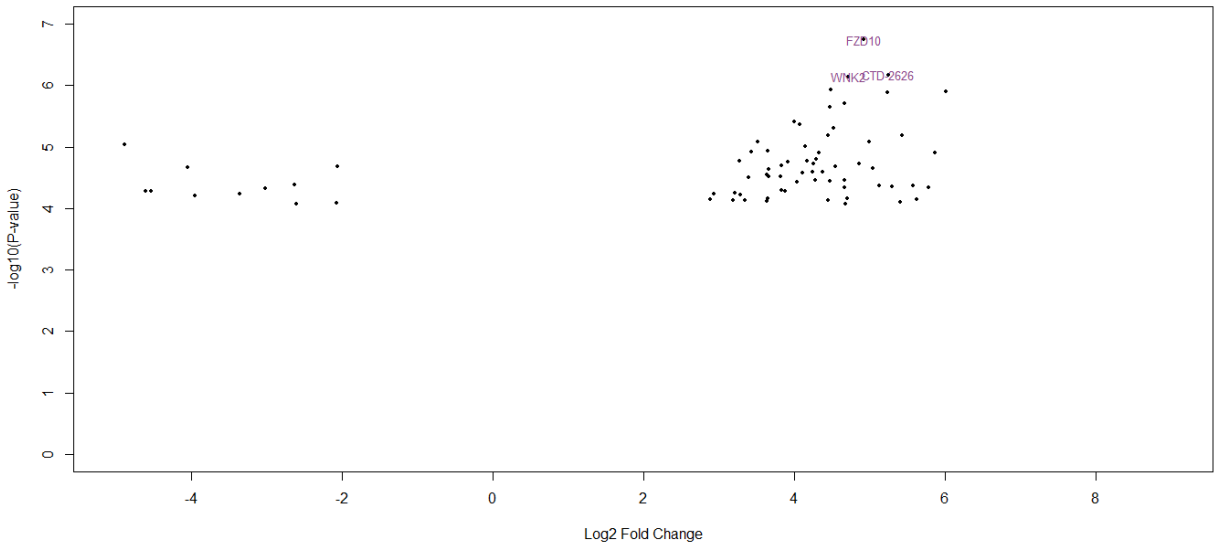
As seen in the plot, DE genes are the ones that point away from M=0 line, having positive and negative values for log fold change which is the same to have expression ratios far away from 1 so that they are upregulated and downregulated if they are above and below M=0 line, respectively.

Even if MA plots help to visualize DGE, they do not consider statistical measures (*p* values or adjusted *p* values) and therefore can not take into account statistically significant differences between cases and controls. In that way, a volcano plot results convenient because it shows the relation between the negative logarithm of p-value and log fold of each gene. In this plots, DE genes have high values in y axis since they have low p values.

```
## Volcano plot showing the top 3 genes with lowest p values
volcanoplot(eb_results, coef = 2, highlight = 3, names = de_results$gene_name, cex=0.5, hl.col = "orchid4")
```

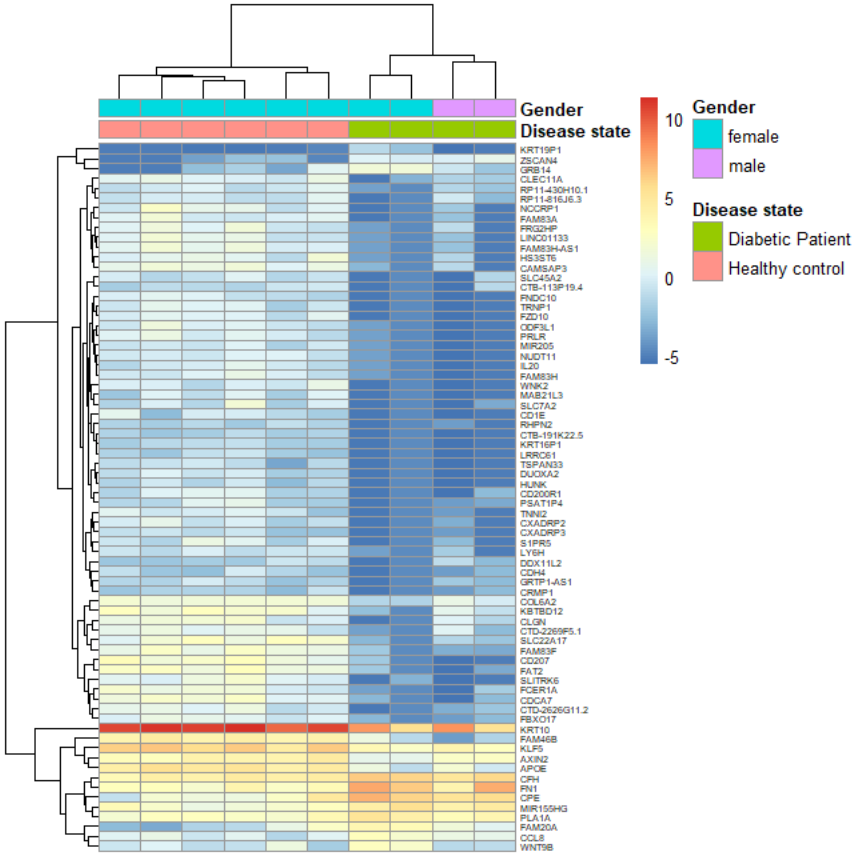


```
## Hold only DE genes
volcanoplot(eb_results[which(de_results$adj.P.Val < 0.05),], coef = 2, highlight = 3, names = de_results[which(de_results$adj.P.Val < 0.05)
```



Finally, a heatmap of DE genes expression levels across all samples is a simple way to visualize the way these genes present lower or bigger expression levels in one disease state with respect to the other.

```
## Expression of DE genes along the 10 samples
exprs_heatmap <- vGene$E[de_results$adj.P.Val <= 0.05, ]
## DE genes in de_results and vGene are in the same order
identical(rownames(exprs_heatmap), de_results[de_results$adj.P.Val <=0.05, "gene_id"])
## [1] TRUE

## DE genes names as row names of heatmap
rownames(exprs_heatmap) <- de_results[de_results$adj.P.Val <=0.05, "gene_name"]
## Data frame with interest variables of samples
col_df <- as.data.frame(colData(rse_gene_SRP095512)[,c("sra_attribute.disease_state","sra_attribute.gender")])
colnames(col_df) <- c("Disease state", "Gender")
## Heatmap with clustered rows and columns
library("pheatmap")
pheatmap(
    exprs_heatmap,
    cluster_rows = TRUE,
    cluster_cols = TRUE,
    show_rownames = TRUE,
    fontsize_row = 5,
    show_colnames = FALSE,
    annotation_col = col_df
)
```



8. Conclusion

On the one hand, this DE analysis demostrated that disease state is efectively a factor that explains differences in transcriptomic profiles of healthy and diabetic patients. On the other hand, gender did not prove to be a determinant variable in this study. From the DE genes found, 2 of the top 3 were FDZ10, and WNK2, all downregulated in diabetic patients.

WNK2 is a serine/threonine protein kinase that has been implicated in protein trafficking highly correlated with insulin resistance or diabetes. Accordingly to what this study revealed, it has been demostrated that the expression levels of WNK2 in skeletal muscle significantly decrease in *db/db* mice, a type II diabetic model, compared to that of wild type mice [8]. FDZ10 is a Frizzled family protein that serves as receptor in the canonical β-catenin and noncanonical Wnt signaling pathways whose role in diabetes type 2 has been revealed [9]. These facts support the results here obtained.

Undoubtly, DE analysis are trascendent and have a relevant impact on the identification of genes and factors implicated in diabetes.

9. Discussion and future prospects

Even if this study achieved its purpose, they are many other variables that could be integrated in the model that account for differential expression. Unfortunately, this data present limited sample information since it do not contain information about RIN and patients age, which could be important factors.

Also variables such as race, food habits, physical activity and family history are relevant factors to model gene expression considering that diabetes is a multifactorial disease and is caused by both genetic and environmental factors . Future work must be accomplished considering all this variables and taking data sets with more samples, specially, diabetic male samples since all male in this study are healthy. Deeper analysis of these DE genes must be done in order to define its role in diabetes: if their changes in expression act as causal factors or they are only correlated somehow.

10. References

1. World Health Organization (WHO). April 2019. *Global report on diabetes.* Web site: Global report on diabetes (who.int)
2. Anna Nordström*, Jenny Hadrévi, Tommy Olsson, Paul W. Franks, Peter Nordström, *Higher Prevalence of Type 2 Diabetes in Men Than in Women Is Associated With Differences in Visceral Fat Mass*, *The Journal of Clinical Endocrinology & Metabolism*, Volume 101, Issue 10, 1 October 2016, Pages 3740–3746, https://doi.org/10.1210/jc.2016-1915
3. HBC Training. (n.d.). *Differential gene expression (DGE) analysis*. Web site: Differential gene expression (DGE) analysis | Training-modules (hbctraining.github.io)
4. Tang, M. November 2019. *Negative bionomial distribution in (single-cell) RNAseq*. Web site: DNA confesses Data speak (rbind.io)
5. Bridges, D. (n.d.). *Why do we use the negative binomial distribution for analysing RNAseq data?* Web site: Why do we use the negative binomial distribution for RNAseq? (umich.edu)
6. RDocumentation. (n.d.). *voom: Transform RNA-Seq Data Ready for Linear Modelling*. Web site: voom function - RDocumentation
7. Data science blog. (n.d.). *MA plot to visualize gene expression data using Python.* Web site: MA plot to visualize gene expression data using Python (reneshbedre.com)
8. Kim, H., Kim, J. H., Hwang, K. H., Park, K. S., Cha, S. K., & Kong, I. D. (2015). *Functional Expression of WNK Kinases and Insulin Signaling Effectors in Diabetic Skeletal Muscle.* The FASEB Journal, 29,LB706.
9. Chen, J., Ning, C., Mu, J. et al. *Role of Wnt signaling pathways in type 2 diabetes mellitus*. Mol Cell Biochem 476, 2219–2232 (2021)