Bversity
1st Ever Virtual College for Biotech

Name :Rajashree.U

Contact: rajashree5820@gmail.com

Mobile: +916383331696

Title: R for Differential Gene Expression analysis

Answer the following questions below and upload them in the google form.

1. Describe the steps involved in importing the microarray data into R and Bioconductor packages.
2. Explain the code used to filter genes based on expression levels or other criteria.
3. Outline the statistical test used for the differential expression analysis and explain its purpose and limitations.
4. Visualize the differential expression results using heatmaps.
5. Analyze the results of the differential expression analysis. How many genes are significantly differentially expressed by fetal sex? Which genes have the highest fold change?
6. Based on the findings, discuss potential biological implications of the differentially expressed genes in the context of fetal sex and tobacco smoke exposure.

R for Differential Gene Expression analysis

The steps involved in importing the microarray data into R and Bioconductor packages.

Bioconductor is a free, open-source, and open-development software project for computational biology and bioinformatics. It's not just a single software, but rather a vast ecosystem of hundreds of specialised packages tailored for analysing and making sense of genomic data generated by wet lab experiments.
It provides tools for analyzing various types of data generated by modern biological experiments:

- Microarray data
- RNA-seq data
- ChIP-seq data
- methylation data.

Bioconductor operates within the R statistical programming language, offering a powerful and flexible environment for:

- data manipulation
- statistical analysis
- visualization.

With over 2000 packages at your disposal, there's likely a dedicated tool for almost any specific analysis you need. Packages handle tasks like preprocessing, normalization, differential expression analysis, functional enrichment, and much **more.**

steps :

1. To start with we have to create a folder containing differntial_gene_expression. R code file and expression_data.txt file phenotype_data(text file) and feature_data(text file)

2. Then open the Rcode file in R script which we need to run in the Console where we can execute commands and visualize the results in the environment part of R.

3. Make sure to change the directory to the folder we created first,in the terminal also change the directory to the same folder

4. Install Bioconductor first to install packages further.
   ```
   if (!requireNamespace("BiocManager", quietly = TRUE))
   install.packages("BiocManager")
   BiocManager::install()
   ```

   it asks permission to install all or some or none. we can select according to our need here we give as (all). proper installation shows no error and will be no problem in loading packages further.

5. Then install packages from Bioconductor such as Biobase, limma,geneplotter,enrich plot, EnhavedVolcano and pheatmap

   install packages from Bioconductor

   ```
   BiocManager::install(c('Biobase','limma','geneplotter','enrichplot'))
   BiocManager::install('EnhancedVolcano')
   BiocManager::install('clusterProfiler')
   ```

   install pheatmap

   ```
   install.packages('pheatmap')
   ```

   load these packages with following command

   loading CRAN and Bioconductor packages

   ```
   library(Biobase)
   library(limma)
   library(RColorBrewer)
   library(dplyr)
   library(ggplot2)
   library(geneplotter)
   library(pheatmap)
   library(enrichplot)
   library(tidyr)
   library(EnhancedVolcano)
   library(clusterProfiler)
   ```

note:If create new project or restart the console we have to load the packages again but need not install it.

Steps in loading microarray data:

6. First, we load the normalized expression assay, the phenotype data and the feature annotation data for this dataset. by passing the following commands in the console change the path to directory with the respective files. accordingly for execution.

   Input:

   ```
   exprsData <- read.delim("C:/Users/rajas/Desktop/R programming/data_R//GSE27272Norm_exprs.txt")
   phenoData <- read.delim("C:/Users/rajas/Desktop/R programming/data_R/GSE27272Norm_phenoData.txt")
   featureData <- read.delim("C:/Users/rajas/Desktop/R programming/data_R/GSE27272Norm_featureData.txt")
   ```

we can check it details in environment after proper loading

7. We can also view the (head) of the data in these files in tabular form at R script with the following commands
   Input:
   ```
   View(head(exprsData))
   View(head(phenoData))
   View(head(featureData))
   ```

The code used to filter genes based on expression levels or other criteria.

After loading all the data, we create an ExpressionSet with the expression assay, phenotype data, and the feature annotation data. An ExpressionSet is a standardized data structure in Bioconductor (from the BioBase library) that combines several different sources of information conveniently to one object.

Creating an ExpressionSet object with all attributes

```
GSE27272_Eset<-ExpressionSet(as.matrix(exprsData))
```
This line constructs a specialized data structure called an ExpressionSet, commonly used in Bioconductor for storing and managing transcriptomic data. It's specifically designed to handle gene expression values, sample information, and feature annotations.
(as.matrix(exprsData)) Converts a data object (likely a data frame or table) containing expression values into a matrix format, which is the preferred format for the ExpressionSet object.
GSE27272_Eset: Assigns the newly created ExpressionSet object to a variable named GSE27272_Eset for further use.

pData(GSE27272_Eset)<-phenoData Assigns phenotype data: This line incorporates sample-level information (phenotype data) into the ExpressionSet object.
pData(GSE27272_Eset): Accesses the slot within the ExpressionSet object designated for storing phenotype data.
phenoData: Represents a separate data object containing relevant sample information, such as experimental conditions, group labels, or patient characteristics.

```
featureData(GSE27272_Eset) <- as(featureData,"AnnotatedDataFrame")
```
Assigns feature annotations: This line adds annotations associated with the features (usually genes or probes) to the ExpressionSet object.
featureData(GSE27272_Eset): Accesses the slot within the ExpressionSet object intended for storing feature annotations**.**

as(featureData, "AnnotatedDataFrame"): Converts a data object (likely a data frame or table) containing feature annotations into an AnnotatedDataFrame object, which is a Bioconductor-specific format for storing and managing annotations.

Filtering Data

Sometimes when we are performing a differential expression analysis we have to subset the genes we are testing based on the annotation data. For example, if we are doing a differential expression analysis by sex it would make sense to filter out the genes on the Y chromosome.
Biologically, a male has an X and Y sex chromosome while a female has two X chromosomes. Features on the Y chromosome should have no expression for females because they have no Y chromosome. Therefore, we cannot compare the difference in expression between males and females for Y-linked genes.

Filters the ExpressionSet (which includes the feature data and the expression data)to the genes that are not present in the Y chromosome

GSE27272_noY <-GSE27272_Eset[GSE27272_Eset@featureData@data$CHR!="Y",]

This code filters an existing ExpressionSet object (GSE27272_Eset) to remove features located on the Y chromosome, creating a new ExpressionSet object **(GSE27272_noY)** containing only features from other chromosomes.

GSE27272_Eset@featureData@data$CHR retrieves a column named "CHR" from the feature annotation data within the ExpressionSet. This column likely contains chromosome information for each feature.

!= "Y" filters the feature data, keeping only those rows where the "CHR" value is not equal to "Y". This excludes features located on the Y chromosome.This vector acts as a filter, keeping features that meet the condition (TRUE) and excluding those that don't (FALSE).

Slicing : [,] is used to subset the ExpressionSet object based on the filtered feature data. The first part (before the comma) specifies the features to keep, while the second part (after the comma) is empty, indicating that all samples should be retained.

GSE27272_noY <- assigns the resulting subsetted ExpressionSet to a new variable named GSE27272_noY, storing the filtered data for further analysis.

Outline the statistical test used for the differential expression analysis and explain its purpose and limitations.

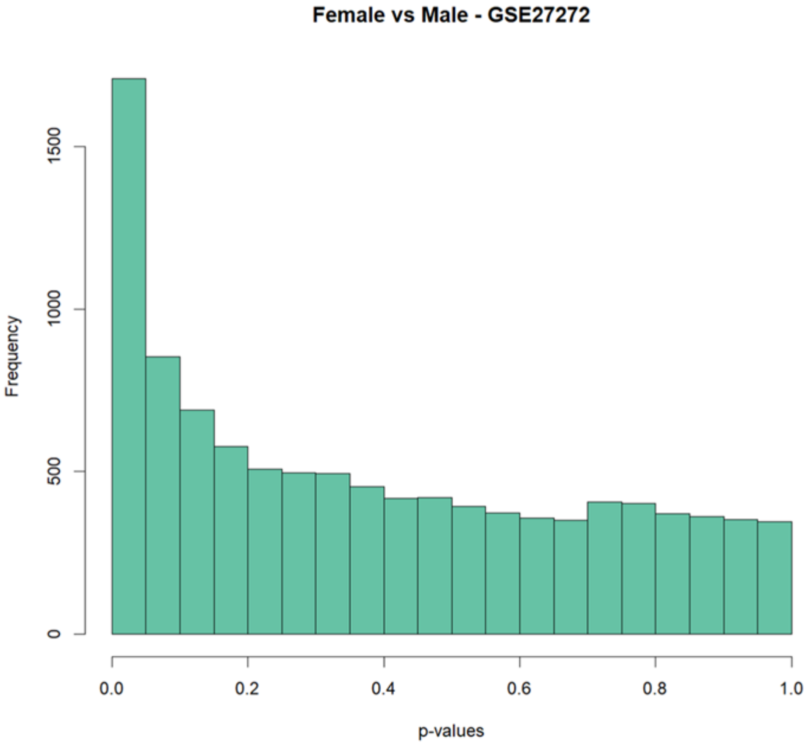
Hypothesis Testing

perform hypothesis testing on all of our genes after filtering the data. We do this by fitting a linear model for every gene and defining contrasts to test our hypotheses. In our case, our contrasts are "female" and "male" because we are interested in finding genes in the placenta are differentially expressed by the sex of the fetus.

The first step is to create a design matrix for variable of interest.

```
design <- model.matrix(~0+phenoData$sex)
colnames(design) <- c("female","male")
GSE27272_samples <-
as.character(phenoData$geo_accession)
rownames(design) <- GSE27272_samples
```

```
design <- model.matrix(~0+phenoData$sex)
colnames(design) <- c("female","male")
GSE27272_samples <-
as.character(phenoData$geo_accession)
rownames(design) <- GSE27272_samples
```



Ceate a contrast matrix that uses the function 'makeContrasts' from the 'limma' package that will take our design matrix as an input for the levels. Afterwards, we fit a linear model with function contrast. Fit() from the 'limma' package to examine the relationship between gene expression and our variable of interest. The function eBayes() on our linear model to get moderated t-test statistics. The eBayes() function performs the empirical Bayes method to squeeze the gene-wise residual variance towards a pooled estimate. Moderating the test statistics with the empirical Bayes method increases the statistical power of the differential expression analysis.

```
contrast_matrix <- makeContrasts(female-male, levels= design)
contrast_matrix <- makeContrasts(non_smoker-smoker, levels=design)

GSE27272_fit <- eBayes(contrasts.fit(lmFit(GSE27272_noY,design = design ),contrast_matrix))
```

T-TEST

A t-test (also known as Student's t-test) is a tool for evaluating the means of one or two populations using hypothesis testing. A t-test uses the sample standard deviation to estimate the standard error of the mean,A t-test is more appropriate when the sample size is small or the population standard deviation is unknown, A t-test may be used to evaluate whether a single group differs from a known value (a one-sample t-test), whether two groups differ from each other (an independent two-sample t-test), or whether there is a significant difference in paired measurements (a paired, or dependent samples t-test).

- Detecting differentially expressed genes (DEGs): The t-test identifies genes that exhibit significantly different expression levels between two experimental conditions (e.g., treated vs. control groups).
- Comparing gene expression means: It tests whether the observed difference in gene expression means between two groups is likely due to true biological variation or random chance.

Steps:

- State the null hypothesis (H0): Assumes no difference in gene expression between groups.
  Calculate the t-statistic: Measures how far apart the group means are, relative to the variability within each group.
- Determine the p-value: Probability of observing the calculated t-statistic (or a more extreme value) if H0 were true.
- Reject or fail to reject H0: If the p-value is below a pre-determined significance level (e.g., 0.05), H0 is rejected, suggesting a significant difference in expression.

Limitations of t-tests :

- T-tests have some limitations that you need to be aware of before interpreting their results. For instance, these tests can only compare the means of two groups or samples, so if you want to compare the means of more than two groups or samples, an ANOVA or Kruskal-Wallis test may be necessary.

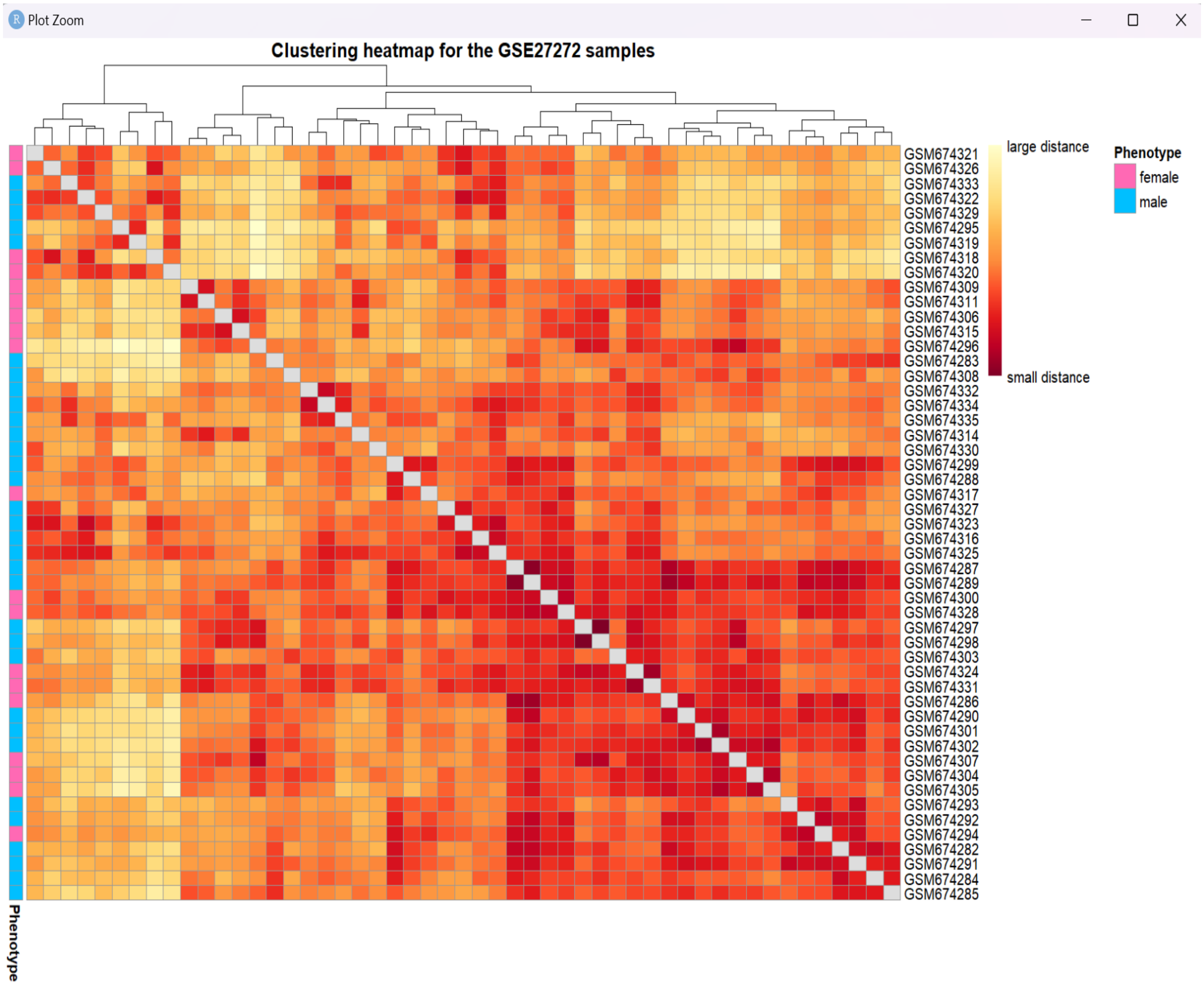
- Additionally, these tests only test the null hypothesis and not the alternative hypothesis; in this case, a confidence interval or Bayesian analysis may be used to estimate the range or probability of the true mean difference in your population.

- Sensitivity to outliers: Extreme values can disproportionately influence the t-statistic and results.Assumptions about data distribution: This relies on the assumption that data is normally distributed, which may not always hold true for gene expression data.

- Multiple testing issues: Testing thousands of genes simultaneously increases the likelihood of false positives (Type I errors). Adjustments for multiple testing (e.g., Bonferroni correction or false discovery rate control) are often necessary.

- Limited power for small sample sizes: May struggle to detect true differences in expression when sample sizes are small.

Visualize the differential expression results using heatmaps:



Observation:

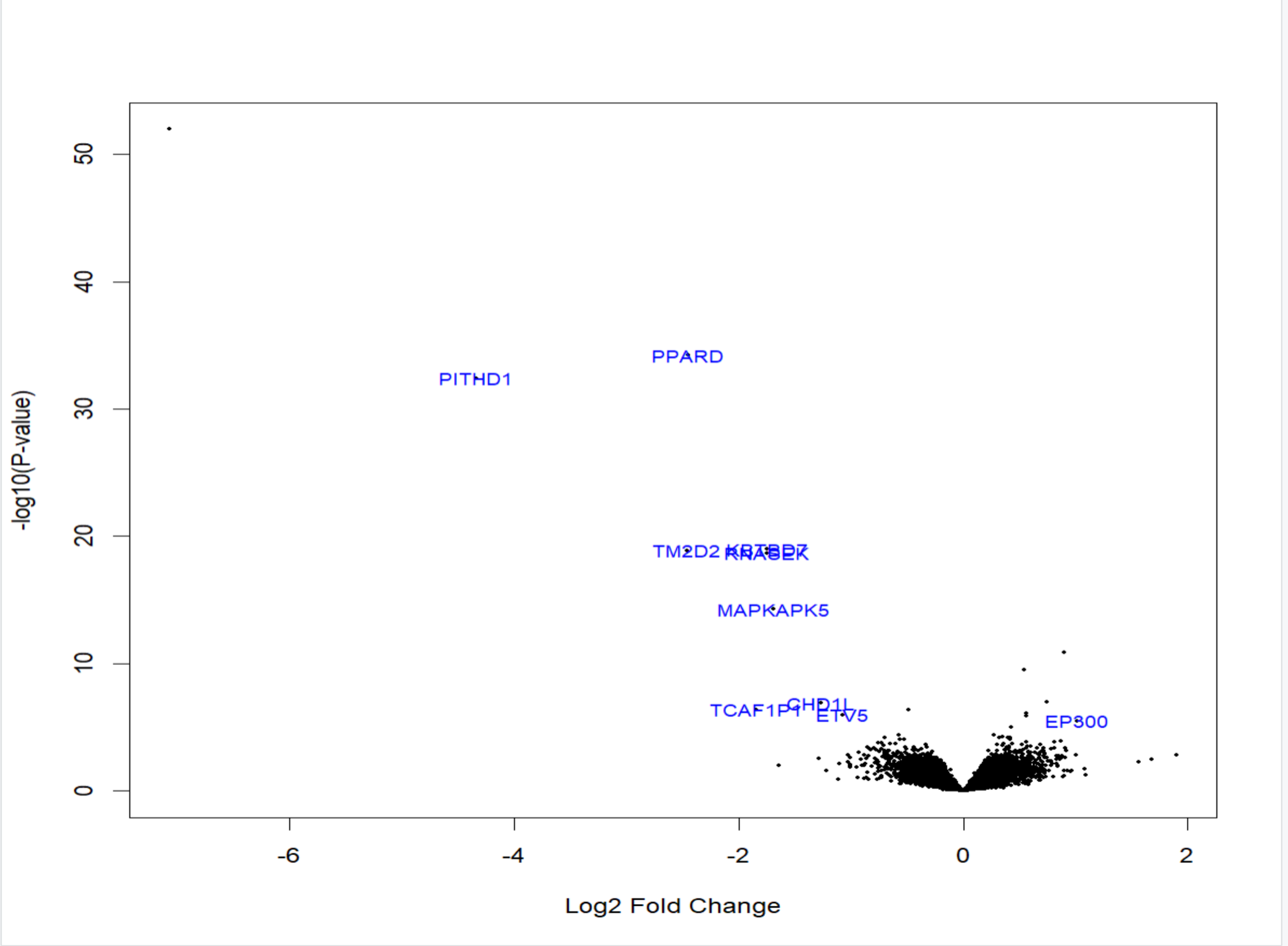
- Here we can see how samples are clustered based on the gender for the given data The blue colour denotes the “male” phenotype and pink colour denotes the “female” phenotype
- Shows the correlation between samples (mentioned on the right side of the pheatmap)the darker colour implicit stronger correlation between the samples to the end and lighter at first suggests they have lesser correlation compared to others

Genes that are significantly differentially expressed by fetal sex:

Volcano Plots:

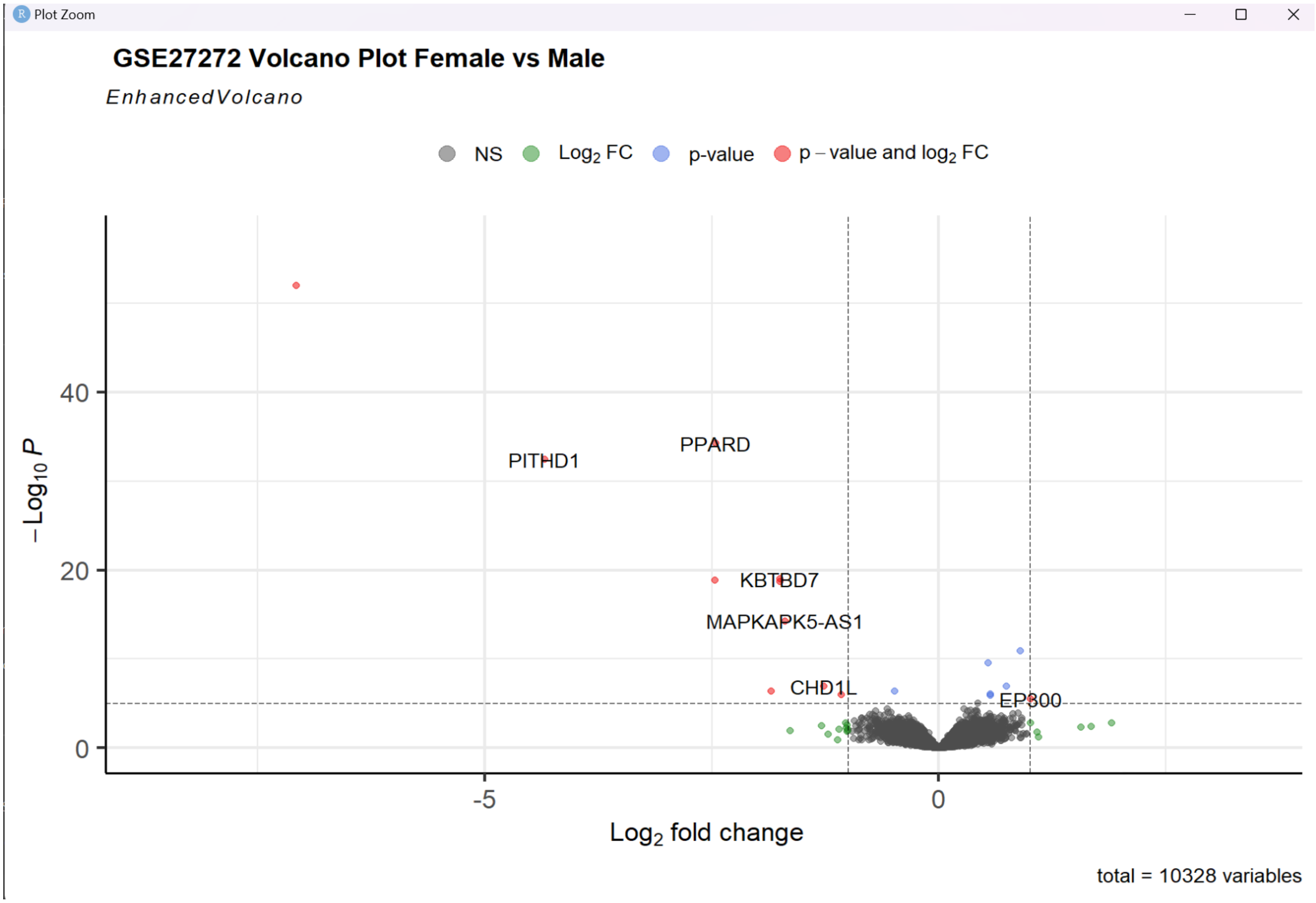
After performing hypothesis testing, it is ideal for us to try visualising our results. The most common graphs made are volcano plots, which are scatter plots of the fold change versus the p-value for each gene. P values are usually transformed on the -log10 scale. This means the more significant the p-value is (or the smaller the p-value is), the larger the value for the -log10 p value. A -log10 p-value greater than -log10(.05) is statistically significant.

```
volcano_names <- ifelse(abs(GSE27272_fit$coefficients)>=1,
as.character(GSE27272_fit$genes$Symbols), NA)
volcanoplot(GSE27272_fit, coef = 1L, style = "p-value", highlight = 100,
names = volcano_names,
xlab = "Log2 Fold Change", ylab = NULL, pch=16, cex=0.35)
```



The graph above is an output of the basic volcano plot function from the limma package. This a good plot to do if you want a quick visualization of the differentially expressed genes.

```
EnhancedVolcano(GSE27272_Results,
lab = as.character(table_GSE27272$Symbols),
x = 'Log2FC',
title=" GSE27272 Volcano Plot Female vs Male",
y = 'pvalue')
```



points that lie towards the top (highly significant p-values) and far to the left or right (large magnitude fold changes). These points represent genes with the most interesting expression patterns.

Common thresholds used are:
p-value < 0.05
|log2foldchange| > 1

A gene is statistically significant after mutliple testing correction if it has an adjusted **p value less than 0.05** .

Log2 fold change is **greater** than 0, then the gene expression is **higher** in the placenta of **female fetuses compared placenta of a male fetus.**

Log2 fold change is **less** than 0, then the gene expression is **lower** in the placentas of **female fetuses compared to males.**

If a gene's log2 fold change is too close to zero in either direction, it is hard to claim that there is a biologically meaningful effect in the gene's expression concerning fetal sex, regardless of the statistical significance of the gene. Therefore, it's important to set a threshold to determine whether a gene has a biologically significant effect size. Here **setting the criteria to be a log2 fold change that has an absolute value greater than 1.**

The **log2 fold change** determines if a gene has a **biologically significant change in expression** (or a gene having a meaningful effect size) between both groups.

For this example: Log2 fold change determines whether a gene is **upregulated or downregulated** with respect to the reference group, which is the placenta of female fetuses.

Observation:

For the volcano plot produced using EnhancedVolcano,

The genes shown in **grey** are **non-significant both statistically and in effect size**.

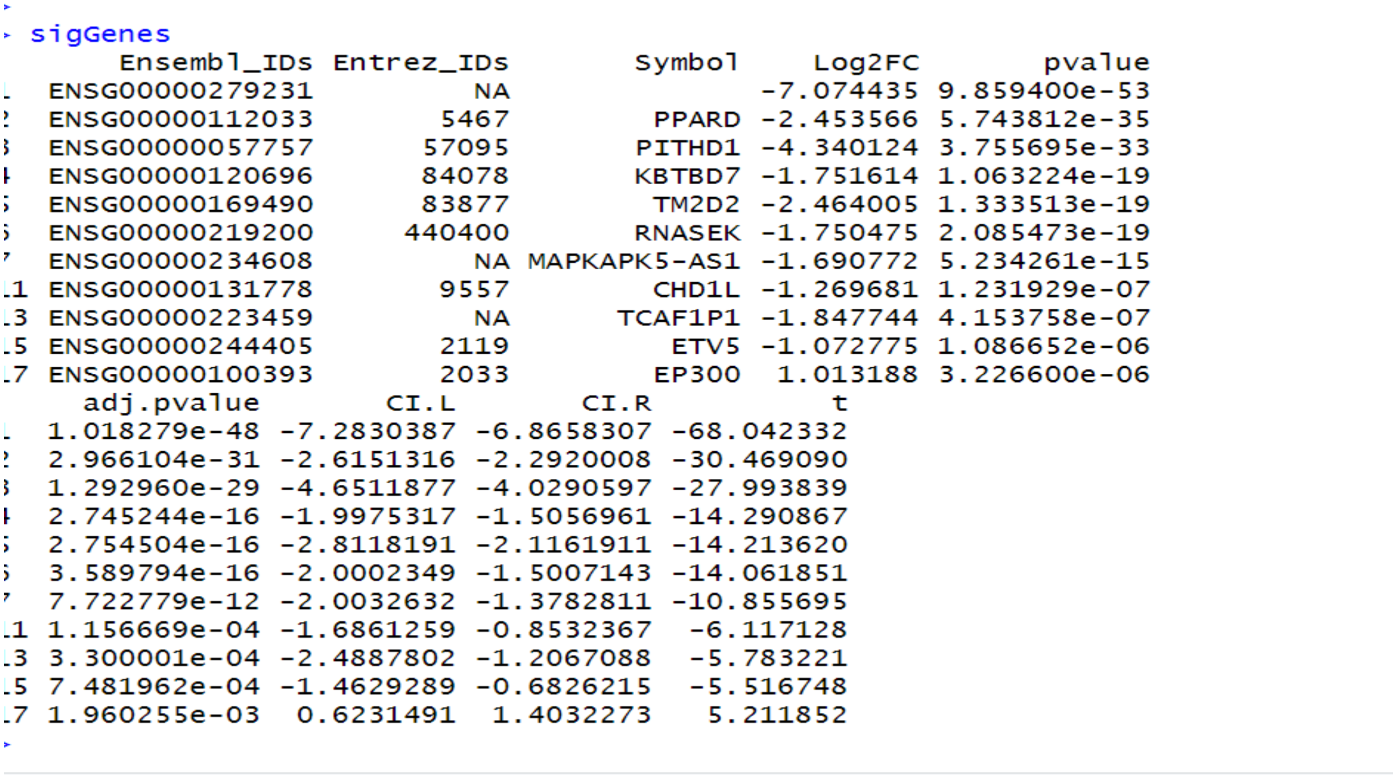
The genes in **blue** have **statistically significant p-values** but **didn't have a log2 fold change** that suggests **biological significance**.

The **green** labelled genes **aren't statistically significant** but have a **biologically meaningful log2 fold change.**

The genes coloured in **red** are both **statistically significant** and have a **biologically meaningful effect size**.

Genes that are significantly differentially expressed by fetal sex:

```
sigGenes <- GSE27272_Results[ GSE27272_Results$adj.pvalue < 0.05 &
!is.na(GSE27272_Results$adj.pvalue) &
abs(GSE27272_Results$Log2FC) > 1, ]
sigGenes
```



```
>
> sigGenes
     Ensembl_IDs Entrez_IDs         Symbol    Log2FC       pvalue
1  ENSG00000279231         NA                -7.074435 9.859400e-53
2  ENSG00000112033       5467          PPARD -2.453566 5.743812e-35
3  ENSG00000057757      57095         PITHD1 -4.340124 3.755695e-33
4  ENSG00000120696      84078         KBTBD7 -1.751614 1.063224e-19
5  ENSG00000169490      83877          TM2D2 -2.464005 1.333513e-19
6  ENSG00000219200     440400         RNASEK -1.750475 2.085473e-19
7  ENSG00000234608         NA MAPKAPK5-AS1 -1.690772 5.234261e-15
11 ENSG00000131778       9557          CHD1L -1.269681 1.231929e-07
13 ENSG00000223459         NA        TCAF1P1 -1.847744 4.153758e-07
15 ENSG00000244405       2119           ETV5 -1.072775 1.086652e-06
17 ENSG00000100393       2033          EP300  1.013188 3.226600e-06
      adj.pvalue       CI.L       CI.R          t
1  1.018279e-48 -7.2830387 -6.8658307 -68.042332
2  2.966104e-31 -2.6151316 -2.2920008 -30.469090
3  1.292960e-29 -4.6511877 -4.0290597 -27.993839
4  2.745244e-16 -1.9975317 -1.5056961 -14.290867
5  2.754504e-16 -2.8118191 -2.1161911 -14.213620
6  3.589794e-16 -2.0002349 -1.5007143 -14.061851
7  7.722779e-12 -2.0032632 -1.3782811 -10.855695
11 1.156669e-04 -1.6861259 -0.8532367  -6.117128
13 3.300001e-04 -2.4887802 -1.2067088  -5.783221
15 7.481962e-04 -1.4629289 -0.6826215  -5.516748
17 1.960255e-03  0.6231491  1.4032273   5.211852
>
```

From the enhanced volcano plot and sigGenes table we can see genes with symbol **PPARD** have the **highest fold change** with the **highest log2FC values** of **-2.453566** and **low p-values of 5.743812e-35.** The Negative sign indicates that the differential gene expression is downregulated.

Potential biological implications of the differentially expressed genes in the context of fetal sex and tobacco smoke exposure.

- The function enrichKEGG performs an enrichment analysis using the Kyoto Encyclopedia of Genes and Genomes (KEGG) database. By this, we can find which significant genes are involved in biological pathways from the data from in KEGG Database

```
sigGenes <- GSE27272_Results$Entrez_IDs[ GSE27272_Results$adj.pvalue < 0.05 &
!is.na(GSE27272_Results$adj.pvalue) &
abs(GSE27272_Results$Log2FC) > 1 ]
  sigGenes <- na.exclude(sigGenes)
  kk <- enrichKEGG(gene = sigGenes, organism = 'hsa')
  head(kk, n=10)
```

significant pathways after multiple testing corrections are:

- PPARD and EP300 genes involved in the Wnt signal pathway which is a group of signal transduction pathways with proteins that pass signals into a cell through cell receptor surfaces.
- From the significant genes table , The EP300 gene is upregulated in female fetuses compared to male fetuses. The PPARD gene is downregulated in female fetuses compared to male fetuses.

- ETV5 and PPARD genes involved in Prostate Cancer for genes.ETV5 gene is downregulated in female fetuses compared to male fetuses with log2FC value of -1.072775 and The p-value of 1.086652e-06 (indicates strong statistical significance).