PEC 1: microarray analysis

Jone Renteria

April 16, 2020

Table of Contents

In the following sections the analysis of a real study has been developed. The main objective of this study is to analyze a real problem using public data in order to learn deeper about the microarray analysis presented in the subject Omics Data Analysis of the Bioinformatics and Biostatistics Master's degree (UOC-UB).

For the reproductivity of this project, the following GitHub's repository link is given. This way, the work would be easily clonated to any computer and all the necessary files would be available at my GitHUB's page: jonerenteria_Microarray-analysis—PEC1. The clonation would be possible using GitBash, GitGUI or an R studio project, just copying the following link: https://github.com/jonerenteria/Microarray-analysis—PEC1-.git

The data has been chosen from the *Gene Expression Omnibus (GEO)* data base, and the selected study is `GSE85269`, which is public since March 20th 2018 and it was developed at the University of Oviedo (Spain). The study was published on *The Plant Cell* journal in 2013 under the title: *Preventing loss of mechanosensation by the nuclear membranes of alveolar cells reduces lung injury in mice during mechanical ventilation* by *Inés López-Alonso, Jorge Blázquez-Prieto, Laura Amado-Rodríguez, Adrián González-López, Aurora Astudillo, Manuel Sánchez, Covadonga Huidobro, Cecilia López-Martínez, Claudia C. dos Santos and Guillermo M. Albaiceta* (Lopez-Alonso et al. (2018)).

Instead, the title of `GSE85269` at the GEO data base varies from the article's one, because they have upload just the study with the microarrays used. The title is as follows: *Expression data of intact and ventilated wildtype and Zmpste24-deficient mice*.

Abstract

In order to study how changes in the nuclear mechanical properties may modify the development of ventilator-induced lung injury, in this document the analysis of Affimetrix microarray data has been carried out. For this reason, twelve mice samples with two different genotypes have been used (six of them of wild type and the rest of them mutant type). Half of the mice (three per each type) were treated as control with baseline breathing and the other six were subjected to mechanical ventilation for 2.5h with controled properties. In the document the proccesing of this data has been carried out, obtaining the most significant genes and their functions, and finally the most relevant GO terms per comparison have been obtained and displayed in different ways.

Objectives

The main objective of this study is to ilustrate the main steps of a bioinformatician to analyze Affymetrix microarrays using R and Bioconductor with a real published study. In this case, the study was comparing wild type and mutant mice to see how their lungs are affected or not after 2.5h of mechanical ventilation.

Materials and methods

Data used for the study

As mentioned before, the data selected for the current study corresponds to the *mus musculus* organism. The data used is constructed with wild type (mentioned as *WT* during the study) and mutant mice (refered as *KO* from now on). The experiment type used with the microarrays has been an *expression profiling by array*. For the experimental design, 12 samples have been analyzed: 6 from Zmpste24-deficient mice (*KO* type) where 3 have been submitted to baseline ventilation (control) and 3 to mechanical ventilation and 6 wild type mice (*WT*), with the same distribution (3 have been submitted to baseline ventilation as control and the other 3 mice to mechanical ventilation). As it could be deduced, each of the combinations has 3 repetitions.

Microarrays

The microarray used for the study has been *Affymetrix Mouse Gene 2.0 ST Array (transcript -gene- version)* and for its correct use in the R programming software, the package `mogene20sttranscriptcluster.db` has been loaded (apart from some other that would be explained later).

Methods used during the analysis

In the following subsections, each step of the process is explained, in order to understand better the workflow used for this analysis.

Environment preparation

To create the correct working environment inside our working directory, three new folders have been created. The first one is the *data* folder in which the `.CEL` and `target` files are stored. The second folder is *results*, in which as its name suggets, the results of our study will be keep inside. The last one is *figures* in which all the figures and graphs generated will be stored in order to have an easier access while writing the `Rmd` folder. The function to create all those new folders inside our working directory, is *dir.create*.

Prepare the data for the analysis

In my case, I have downloaded manually the `.CEL` archives into the *data* folder, and I have unzipped the *gz* archives to finally obtain the `.CEL` ones, which are the desired archives to work with the Affymetrix array. Nevertheless, there is an option to download directly those files with the *getGEOSuppFiles* R function indicating the GSE identification number inside brackets, which is our case is `GSE85269`.

Apart from the `.CEL` archives, the *data* folder must contain the *targets* archive in *csv* format. This archive contains three main columns: *FileName* (with the exact name of the `.CEL` files in the data folder), *Group* (which summarizes the conditions in the experiment for that sample) and *ShortName* (with a shortned version of the sample's useful name for some plots). In my particular case, I have add two more columns into the file: *Genotype*, with the shortening of the genotype used in each case (*WT* refering to the wild type and *KO* to the Zmpste24-deficient mice) and *breath*, with two levels: *BAS* (baseline) and *VEN* (mechanical ventilation).

```
> targets <- read.csv2("./data/targets.csv", header = TRUE,sep=";")
> knitr::kable(
+   targets, booktabs = TRUE,
+   caption = 'Content of the targets file used for the current
analysis')
```

Content of the targets file used for the current analysis

FileName	Group	Genotype	Breath	ShortName	date
GSM2263555_WT_baseline_1.CEL	WTBAS	WT	BAS	WTBAS.1	15/04/2015
GSM2263556_WT_baseline_2.CEL	WTBAS	WT	BAS	WTBAS.2	16/04/2015
GSM2263557_WT_baseline_3.CEL	WTBAS	WT	BAS	WTBAS.3	15/04/2015
GSM2263558_KO_baseline_1.CEL	KOBAS	KO	BAS	KOBAS.1	11/11/2014
GSM2263559_KO_baseline_2.CEL	KOBAS	KO	BAS	KOBAS.2	12/11/2014
GSM2263560_KO_baseline_3.CEL	KOBAS	KO	BAS	KOBAS.3	12/11/2014
GSM2263561_WT_ventilated_1.CEL	WTVEN	WT	VEN	WTVEN.1	15/04/2015
GSM2263562_WT_ventilated_2.CEL	WTVEN	WT	VEN	WTVEN.2	15/04/2015
GSM2263563_WT_ventilated_3.CEL	WTVEN	WT	VEN	WTVEN.3	15/04/2015
GSM2263564_KO_ventilated_1.CEL	KOVEN	KO	VEN	KOVEN.1	12/11/2014
GSM2263565_KO_ventilated_2.CEL	KOVEN	KO	VEN	KOVEN.2	12/11/2014
GSM2263566_KO_ventilated_3.CEL	KOVEN	KO	VEN	KOVEN.3	12/11/2014

```
> #str(targets)
```

Using the *Str* function into our *targets* file, we have observed that all the variables (the columns of the table above) are factors.

Packages installation

I have loaded several new packages into my R software for the development of this analysis. The most particular package have been *mogene20sttranscriptcluster.db* and *pd.mogene.2.0.st* as mentioned before for the specific array type used with *mus musculus*. Another packages used have been the following, just to mention some of them: *arrayQualityMetrics,pvca,limma,genefilter,annotate,ReactomePA,clusterProfiler.*

Read the CEL files

In order to read the raw data (`.CEL`) and store them into an objet in R (*rawData*), we have loaded the *oligo* package. The working directory for our project has already been stablished at the beginning, but for the function *list.celfiles* to work correctly, the sub-file *data* has to be specified too. With the package *Biobase* we have read again the *targets* csv file to associate the information in the `.CEL` files with the target, using the function *read.celfiles* and storing that information in an object called *rawData*.

We have obtained the `CEL` *files processing date and charged into the targets file. This has been runned just once to enable the reproductivity of the samples and not to change the targets file every time.*

In order to work with more comprenhensive names, it is possible to change the names of the files to the previously written ones in the *Targets* csv file, *ShortName*:

Data preprocessing

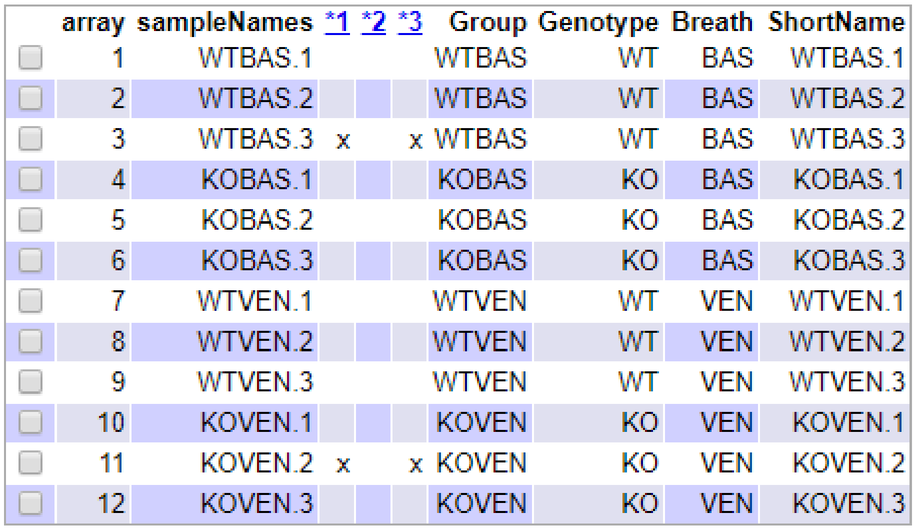
The data preprocessing step consists mainly in data visualization and quality control of raw data. After checking the quality of the data, normalization and filtration of the data may be required.

Quality control of raw data

For the quality control of raw data, the package *arrayQualityMetrics* has been used. The application of the function with the same name over the rawdata has created another folder in the working directory with all the graphics summarizing the raw data, which gives us a general overview and behaviour of the results.

The *arrayQualityMetrics* function generates a file called *index.html*, that opens the browser with access to a summarization of the analysis performed. The table shown in the first page of the webpage indicates on its three columns (with numbers 1, 2 and 3 respectively) the quality criteria used that should be verified by good quality arrays. In our example, two arrays have shown asterics in the first and third positions. As only two of the 12 arrays are showing the *problem*, at this moment all the arrays are going to keep in the analysis. This table is shown below:

```
> knitr::include_graphics("./figures/Figure1.png")
```



	array	sampleNames	*1	*2	*3	Group	Genotype	Breath	ShortName
☐	1	WTBAS.1				WTBAS	WT	BAS	WTBAS.1
☐	2	WTBAS.2				WTBAS	WT	BAS	WTBAS.2
☐	3	WTBAS.3	x		x	WTBAS	WT	BAS	WTBAS.3
☐	4	KOBAS.1				KOBAS	KO	BAS	KOBAS.1
☐	5	KOBAS.2				KOBAS	KO	BAS	KOBAS.2
☐	6	KOBAS.3				KOBAS	KO	BAS	KOBAS.3
☐	7	WTVEN.1				WTVEN	WT	VEN	WTVEN.1
☐	8	WTVEN.2				WTVEN	WT	VEN	WTVEN.2
☐	9	WTVEN.3				WTVEN	WT	VEN	WTVEN.3
☐	10	KOVEN.1				KOVEN	KO	VEN	KOVEN.1
☐	11	KOVEN.2	x		x	KOVEN	KO	VEN	KOVEN.2
☐	12	KOVEN.3				KOVEN	KO	VEN	KOVEN.3

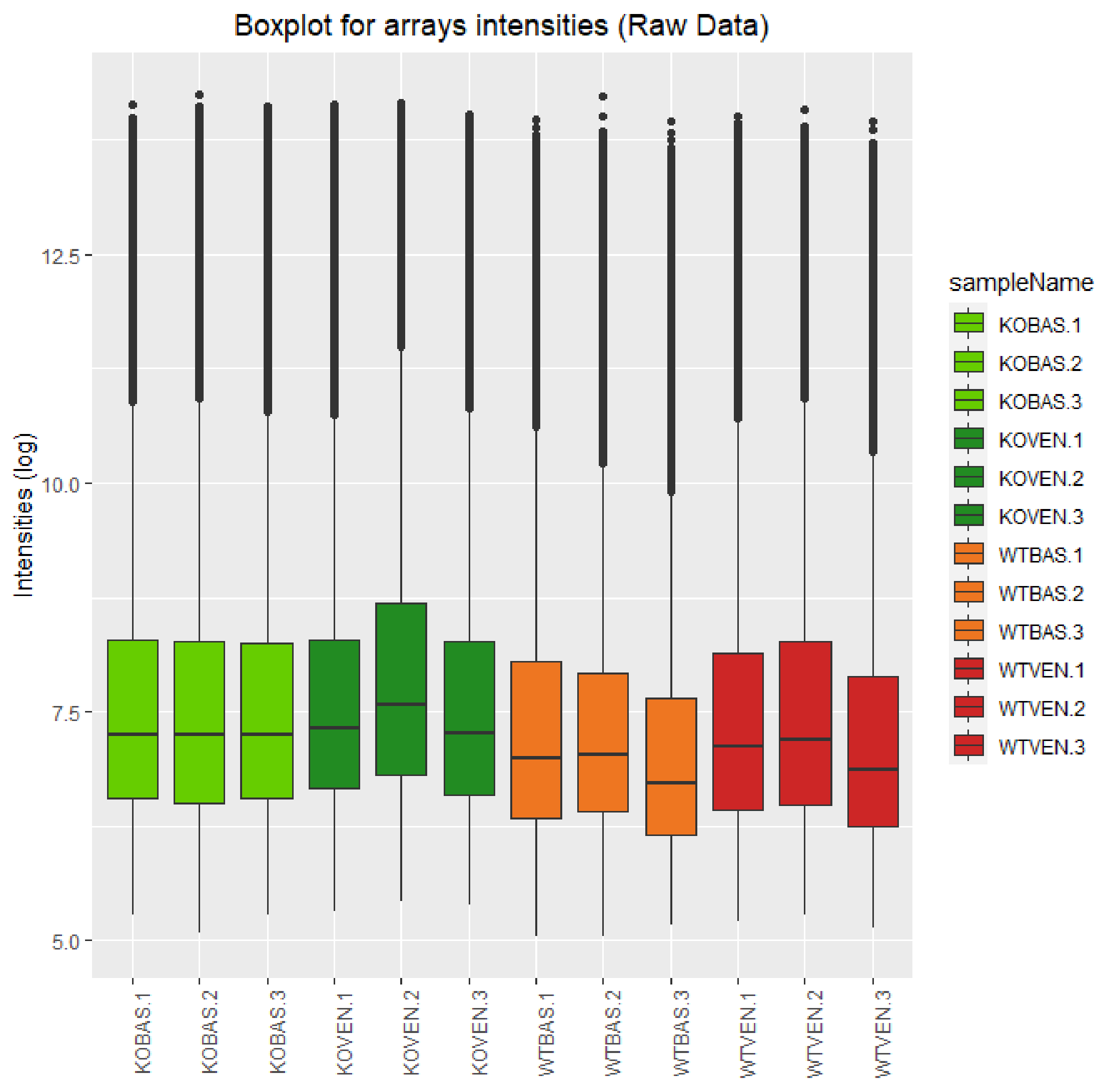
Summary table, in the index.html file, produced by the arrayQualityMetrics package on the raw data

There are different ways to visualize the raw data. As shown in the Table 2 of the Module 2 of the theory, depending on the type of microarray, there is an standarized way of visualization of the data. In our case, the array has one color and is a low level microarray, so the general visualization methods would include: boxplots, histograms and scatterplots (as the principal component analysis graph).

- **Boxplot of the raw data:**

In order to complete the boxplot with the *ggplot2* package, it is neccesary to get the data in the correct format: a data frame with log intensities in one column and sample names in the second column. As we are only going to use the PM intensities (*perfect match*), we have storaged them in an object called *pmexp* of the raw data. We have generated a *for loop* with the 12 samples, and for it's correct data storage, two empty vectors have been created (inside the loop where the information is going to be stored inside). To create the sample Names of the dataframe we have taken the information from *mytargets* object (generated earlier) and its fourth column, where the *ShortName* of the labels is stored. The *logs vector* will contain the log PM intensities of all twelve samples stacked into a single column and for the log transformation the log2() method will be used.

After storing all the data into the dataframe the boxplot is shown below.

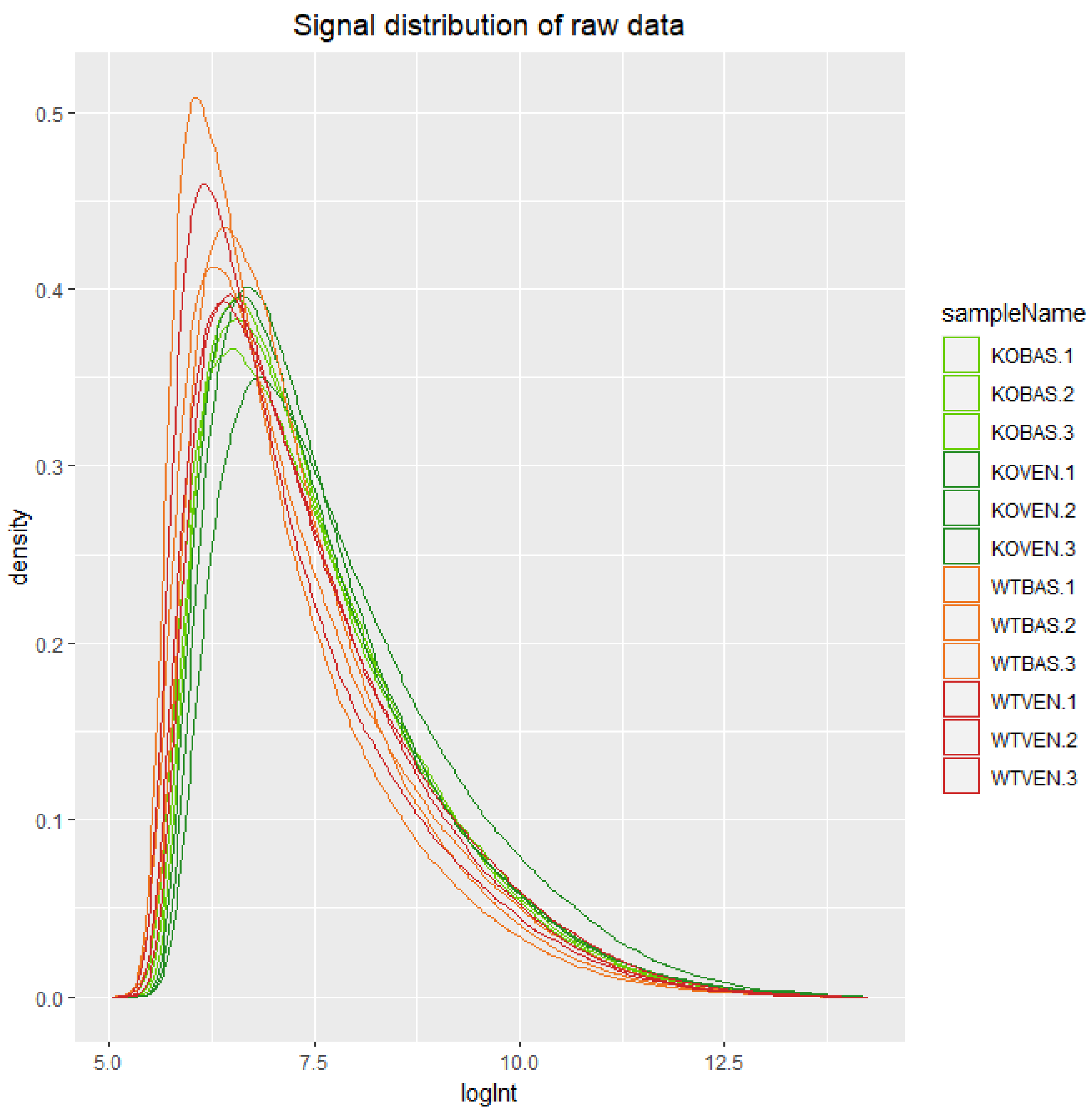


Boxplot for arrays intensities (Raw Data)

A light variation of intensity among arrays is observed, but this is the expected for raw data.

- **Histogram of the raw data:**

Another way to visualize the raw data is using histograms. Using the same dataframe as in the previos graphic, an histogram has been generated.The histogram is also called density graphic, where the original data signal distribution is shown (without normalization applied). With this histogram is also posible to see if the distribution of different arrays are similar in shape and position.

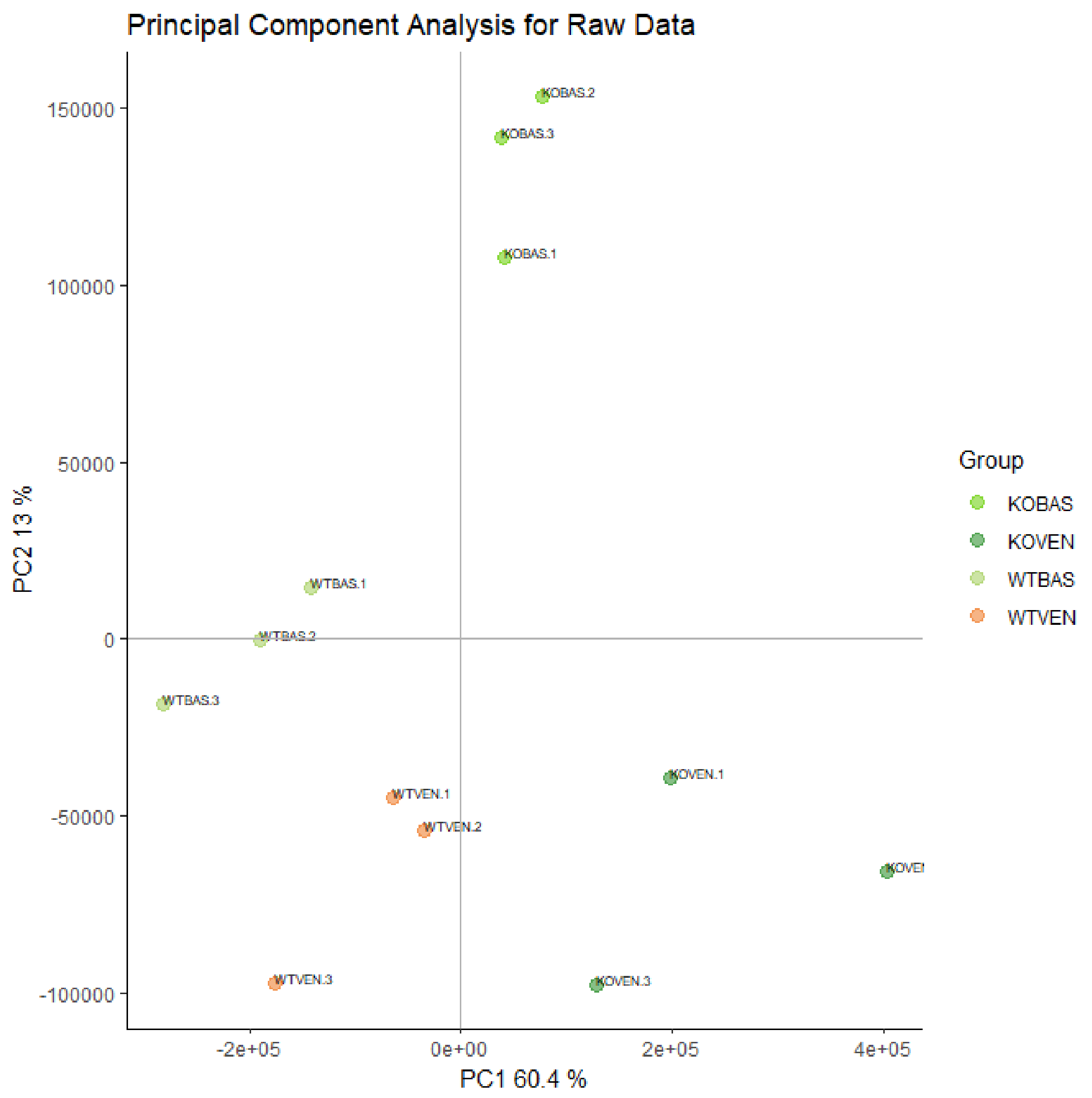


Histogram with signal intensities for raw data

This distribution shows that apparently WT types have a different distributtion than the KO mutant types. The signal distributions do not match in shape (in general terms).

- **Scatterplot of the raw data:**

In the next figure the scatterplot of each of the component is shown. To generate the figure, a function *func_scatt* has been created in order to generate the corresponding matrix.



PCA graphic for raw data

In the graph shown above it is posible to observe how data is behaved in each case. The principal component is a 60.4%, separating the KO types in the right and the WT types in the left part. The second component represents the 13%, and separates the data depending on the breathing method. The mechanical ventilation (*VEN*) samples are all of them in the lowest part, but the baseline breaths (*BAS*) are in the middle-top. The wildtype baseline is just in the middle of the line / separation, and the KO baseline mice samples are much higher than the other ones. In all of the four types of *Genotype-breath* the three biological repetitions are close to each other.

Raw data normalization

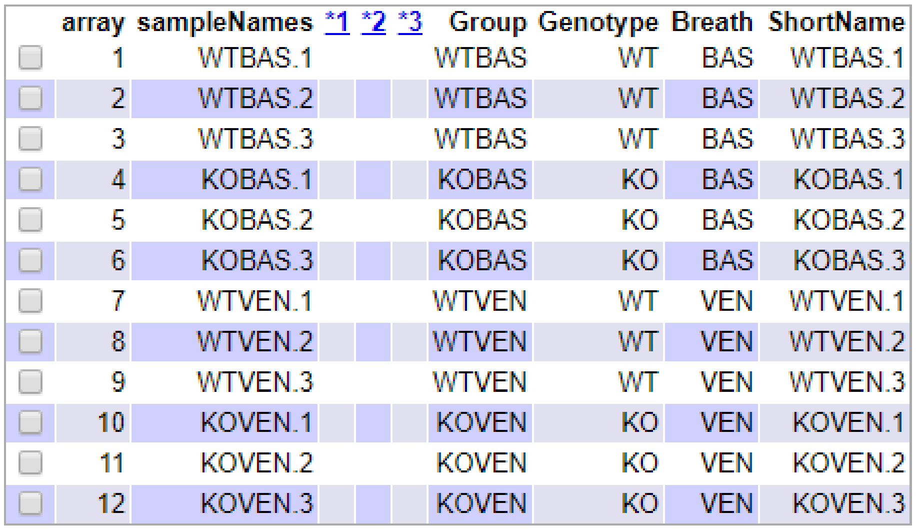
The last part of the raw data preprocessing is to make the arrays comparable among them and try to reduce, and if it is possible to eliminate, all the variability in the samples not owing to biological reasons, and that's why normalization of the raw data must be carried out. The objective of normalization is to assure that intensity differences present in the array, reflects the differential expression of the genes, rather than artificial biases due to technical issues.

In our case, RMA normalization without filtration is used. This normalization process consists of three discrete steps: background correction (with the *rma* function), normalization (to make the values in the arrays comparable), and summarization of multiple probes belonging to the same gene (or group) into one number to represent the gene expression.

Normalized data quality control

In the same way as with *rawdata*, it is interesting to perform the quality control to check how data looks after the first step of normalization. In this case, while using the *arrayQualityMetrics* function, *rawData* is substituted for *rma_data* (the name of the object that has the normalized data of the study):

In the next figure the same table as before is shown (the same as with *rawdata*). The diference with the previous one is that there is no any asteric (*) marked in this one. This gives us a general view that the normalization has been done correctly.



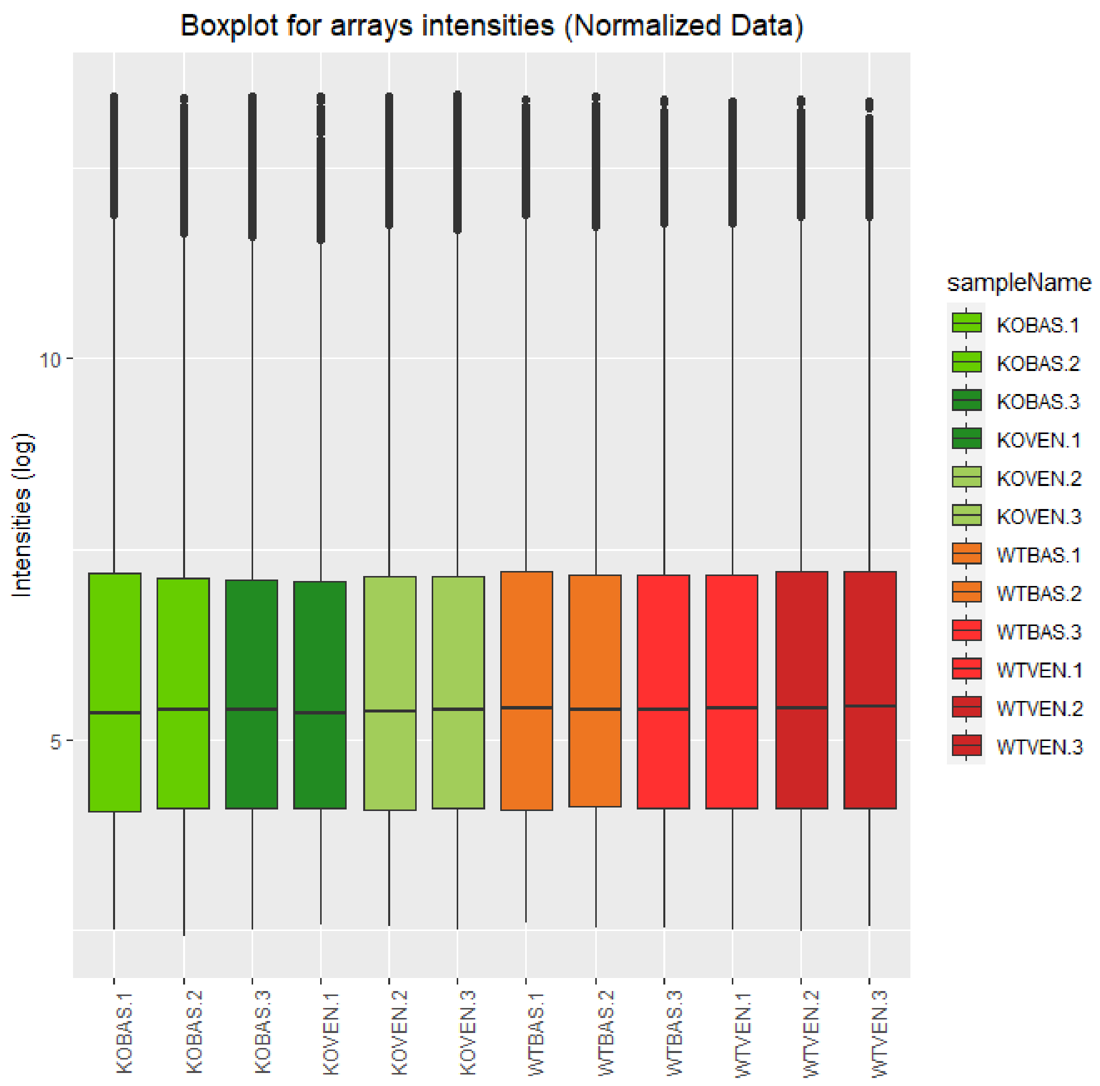
	array	sampleNames	*1	*2	*3	Group	Genotype	Breath	ShortName
☐	1	WTBAS.1				WTBAS	WT	BAS	WTBAS.1
☐	2	WTBAS.2				WTBAS	WT	BAS	WTBAS.2
☐	3	WTBAS.3				WTBAS	WT	BAS	WTBAS.3
☐	4	KOBAS.1				KOBAS	KO	BAS	KOBAS.1
☐	5	KOBAS.2				KOBAS	KO	BAS	KOBAS.2
☐	6	KOBAS.3				KOBAS	KO	BAS	KOBAS.3
☐	7	WTVEN.1				WTVEN	WT	VEN	WTVEN.1
☐	8	WTVEN.2				WTVEN	WT	VEN	WTVEN.2
☐	9	WTVEN.3				WTVEN	WT	VEN	WTVEN.3
☐	10	KOVEN.1				KOVEN	KO	VEN	KOVEN.1
☐	11	KOVEN.2				KOVEN	KO	VEN	KOVEN.2
☐	12	KOVEN.3				KOVEN	KO	VEN	KOVEN.3

Aspect of the summary table, in the index.html file, produced by the arrayQualityMetrics package on normalized data

The quality performance is analized with the same three graphics as before, as shown in the following code and figures.

- **Boxplot of the normalized data:**

The boxplot of the normalized data is shown below:

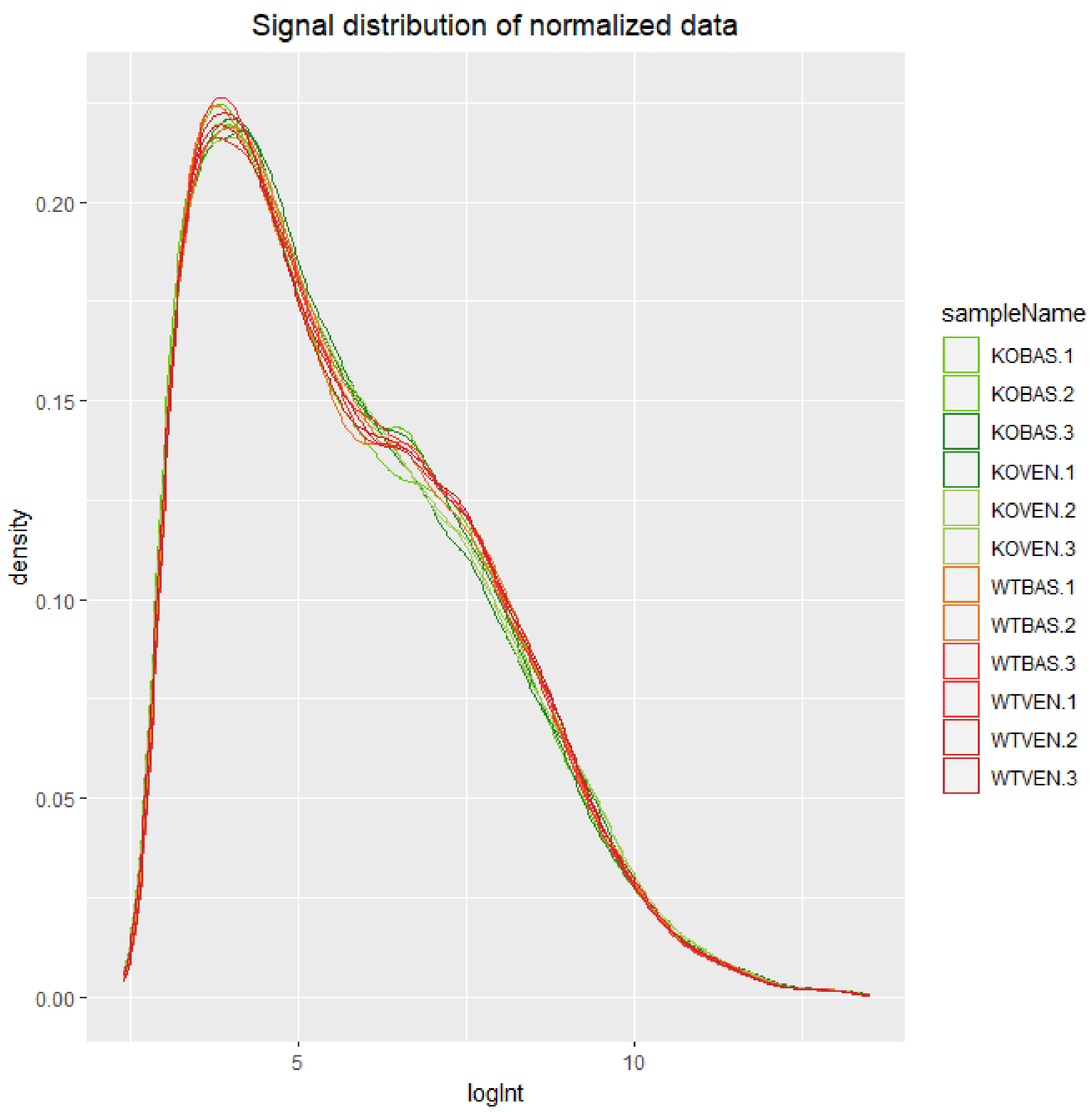


Boxplot for arrays intensities (Normalized Data)

This time, all the boxes of the 12 samples are aligned with each other. Some of the replicates are just a little bit smaller than the other ones, but it is considered normal and insignificant.

- **Histogram of the normalized data:**

The histogram of the normalized data is shown below:

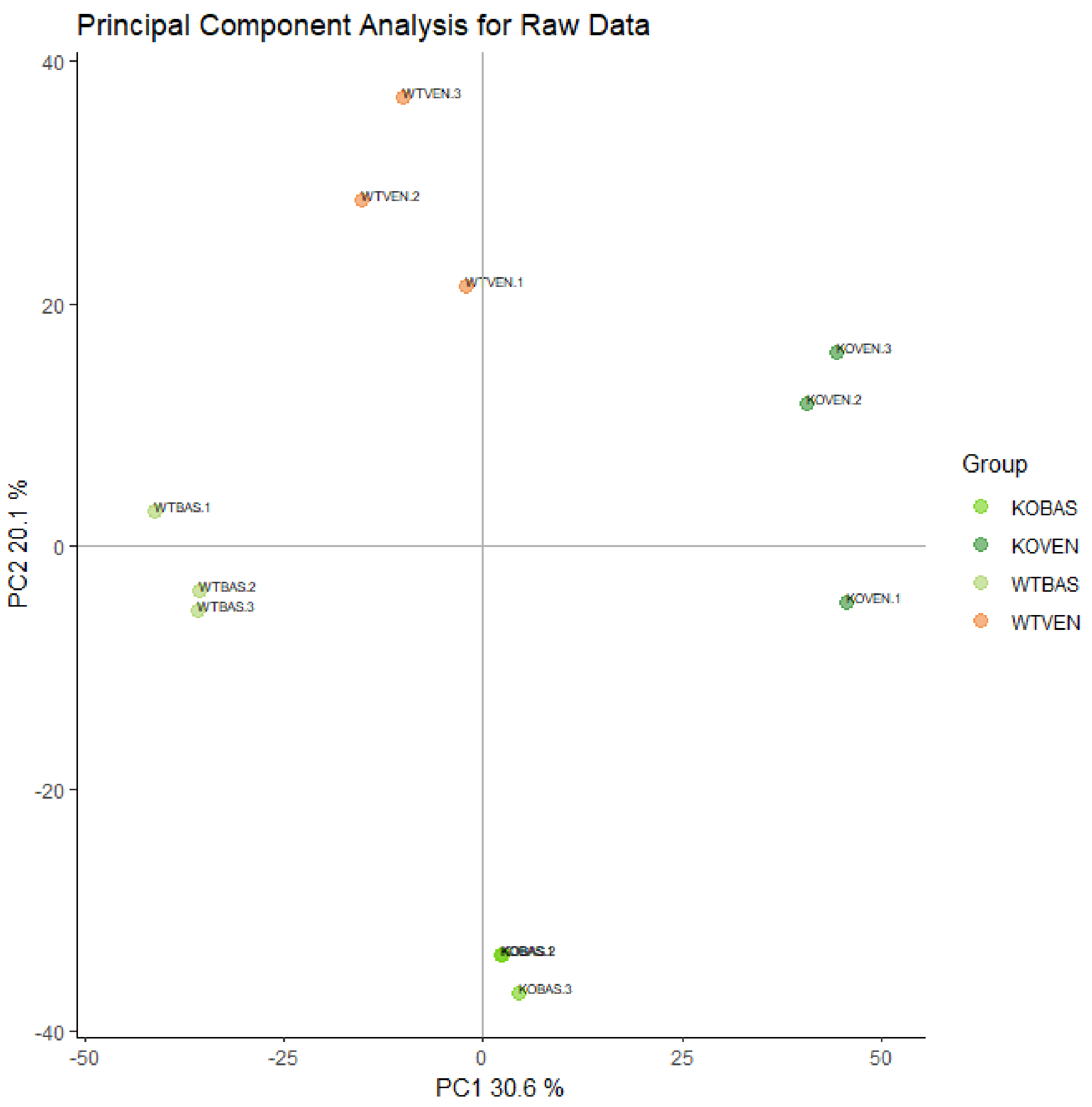


Histogram with signal intensities for normalized data

In this case, and comparing with the raw data's histogram, all the signals are aligned with each other. Nevertheless, some differences could be observed, but they are dismised, as it is considered that the data has changed from the raw one.

- **Scatterplot of the normalized data:**

Finally, the scatterplot of the normalized data is also shown:



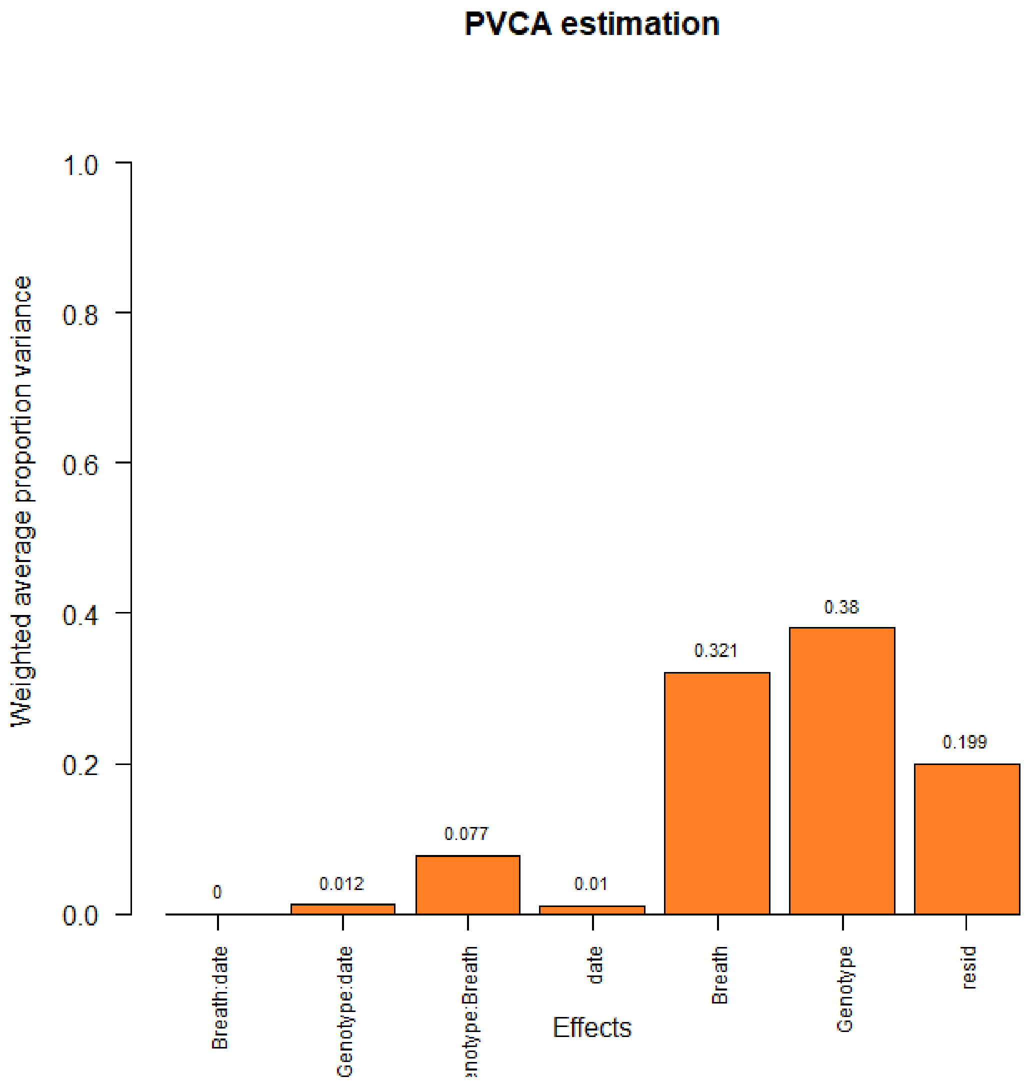
PCA graphic for normalized data

In this case, the first component of the PCA accounts for 30.6% of the total variability of the samples, and as we can observe in the plot, this variability is mainly contributed by the *genotype* condition. The *KO* samples are still in the right part of the plot and the *WT* in the left. Moreover, the *breath* factor is divided in a different way than in the case of the raw data. Each of the groups, *WTBAS*, *WTVEN*, *KOBAS* and *KOVEN* are more or less next to one of the four axis. In the previous PCA graphic, the distiction was more notorius as all the VEN types of breathing were in the lowest part and the BAS breathing were middle-top.

Batch detection

The first step for the batch detection is to defind and select the factor that would be introduced to the *pvcaBatchAssess* function of the *pvca* package. As seen with the *get.celfile.dates()* function, the processing date of the samples is not the same in all the samples, so, in order to verify if the microarray results are affected by it, the *date* is also introduced to the batch detection analysis. Apart from this variable, *genotypes* and *breath* are also introduced.

In order to visualize the results , the following graph with the *PVCA* estimation is shown:



Relative importance of the different factors affecting gene expression

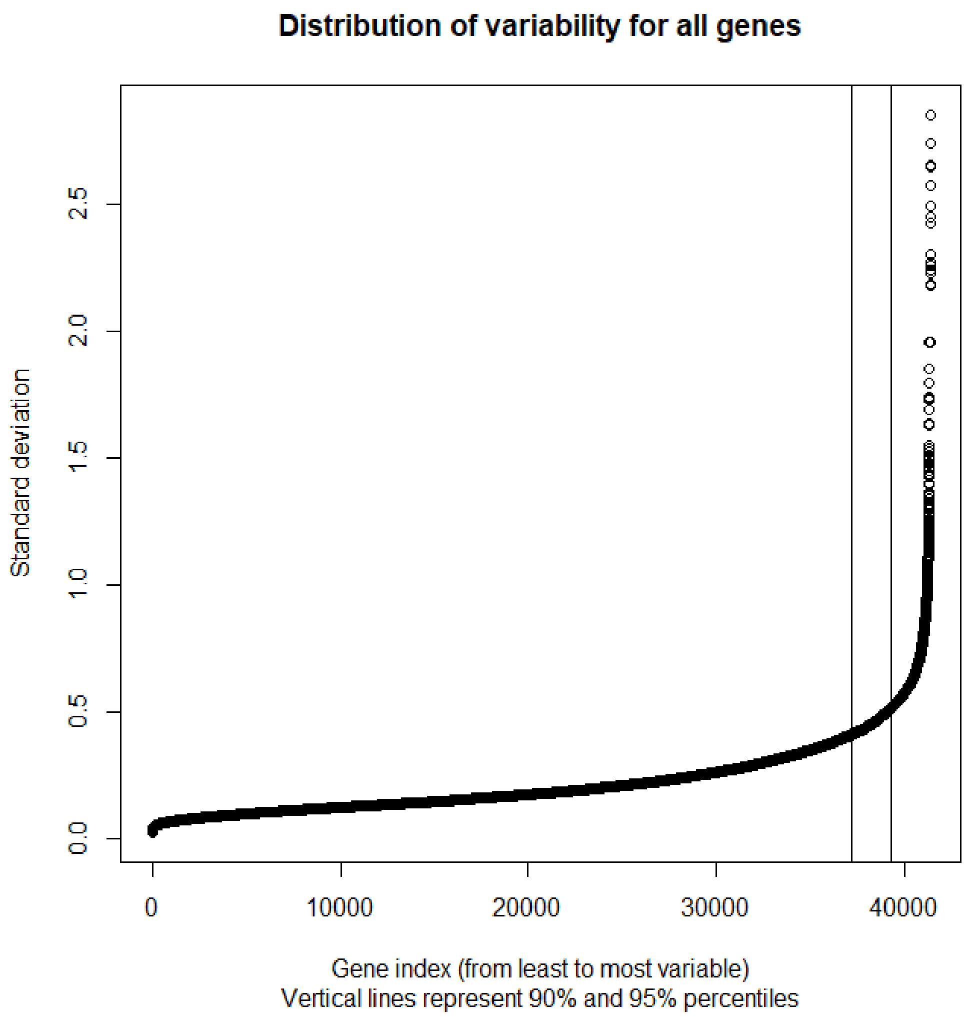
The bar diagram shows that the most relevant source of variation are *breath* and *genotype*, because of their relative size and value number are the highest. The difference between then is small. Nervertheless, instead of the high percentage of variability attributed to *breath*, it is not a batch factor, because it was an experimental factor that was included in the experimental design by the authors of the study.

We have introduced the *date* too as an experimental factor to the microarray analysis, because as the processing date was really different from one sample to another, it was expected to have a high effect in the results. Instead, it hasn't shown a high effect, so this means that it doesn't generate a high cumulative error and experimental variation. This non-biological effect then is dismiss for further analysis.

Detecting most variable genes

In the following sections the selection of the most differentially expressed genes is going to be carried out. This proccess is affected by the number of genes selected, so, the higher the number we are going to have a greater p-value adjustment we are going to obtain. The first step then is to detect the most variable genes in our data.

In the next figure, a graph of the overral variability of all genes is shown. It shows that the most variable genes are those with a standard deviation above 90-95% (taking into account all the standard deviations).



Values of standard deviations allong all samples for all genes ordered from smallest to biggest

Filtering least variable genes

Using the function *nsFilter*, the removal of those genes whose variability can be atrributed to random variation has been done. It is expected that those genes are not going to be differentially expressed, so it is better to eliminate them for further analysis.

The function *nsFilter* returns the filtered values and a report of the filtering results.

```
$numDupsRemoved
[1] 671

$numLowVar
[1] 17973

$numRemoved.ENTREZID
[1] 16710
```

After filtering there are 5991 genes left. Note that we have stored the genes left in the variable *eset_filtered*.

It is important to keep the normalized and filtered data in our *Results* folder to have a track of the most important steps carried out during the analysis. We are going to use the widely known *write.csv* function to convert the data into a *csv* format. We are saving them using *save* function.

Defining the experimental setup: The design matrix

Generating the design matrix is the first step for the analysis based on **linear models**. As we have seen in the PVCA plot, the highest effects were related to *genotype* and *breath* functions. We are going to include also the interaction between then, as seen in the Figure 2 - I of the selected article Lopez-Alonso et al. (2018), where the comparison between genotype and breath is carried out.

In this study the *Group* variable is a combination of the two types of genotypes (*WT* and *KO*). We have then 2 factors (*genotype* and *breath*). The factor *genotype* has two levels: WT (wild type) and KO (which is the mutant type). The factor *breath* has two levels too: baseline (*BAS*) and mechanical ventilation (known as *VEN*).

After loading the *limma* package, the design matrix has been created. Column names represent each of the genotype factor with the type of breathing. In the analysis, 3 repetitions per each of the treatments have been carried out.

```
        WTBAS KOBAS WTVEN KOVEN
WTBAS.1     0     0     1     0
WTBAS.2     0     0     1     0
WTBAS.3     0     0     1     0
KOBAS.1     1     0     0     0
KOBAS.2     1     0     0     0
KOBAS.3     1     0     0     0
```

```
WTVEN.1     0     0     0     1
WTVEN.2     0     0     0     1
WTVEN.3     0     0     0     1
KOVEN.1     0     1     0     0
KOVEN.2     0     1     0     0
KOVEN.3     0     1     0     0
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$Group
[1] "contr.treatment"
```

In the matrix, each row contains a one in the column of the group to which the sample belongs and a zero in the others.

Defining comparisons with the Contrasts Matrix

After designing the matrix, it is neccesary to defind the comparisons between the groups. In the Lopez-Alonso et al. (2018) article and as mentioned before, the three comparisons between the groups are the following:

- Effect of KO mice on baseline ventilation.
- Effect of KO mice on mechanical ventilation.
- Interaction between the ventilation method (breath) and KO mice.

To carry out the contrasts in the matrix, the *makeContrasts* function is used, and the results are the following:

```
       Contrasts
Levels  KOvsWT.BAS KOvsWT.VEN INT
  WTBAS         -1          0  -1
  KOBAS          1          0   1
  WTVEN          0         -1   1
  KOVEN          0          1  -1
```

As seen in the theory of the subject, a comparison between groups - called "contrast" - is represented by a "1" and a "-1" in the rows of groups to compare and zeros in the rest. If several groups intervened in the comparison would have as many coefficients as groups with the only restriction that its sum would be zero.

Model estimation and gene selection

The model estimation and the gene selection is also done with the *limma* package. The objective will be to decide if the genes could be considered differentialy expressed. First, we will have to estimate the model, then estimate the contracts and finally perform the significance tests for the final decision.

Using the function *lmFit* of *limma* package, the genes are ordered depending if they are differentialy expressed. To have control over the false positive rate (FPR), the BH procedure is used (Benjamini-Hochberg).

Obtaining lists of differentially expressed genes

Using the *topTable* function of the *limma* package, a table of each contrast is obtained. The *p-value* can be used to decide between the most to the least differentialy expressed genes. In the following tables, the order of the obtained genes is shown with this extra information apart from the *p-value*: *logFC, AveExpr, t, adj.P.Val and B*.

For comparison 1, *KOvsWT.BAS*, the first lines of *topTable* are shown below. They are genes that change their expression between KO and WT in baseline ventilation:

	logFC	AveExpr	t	P.Value	adj.P.Val	B
17481770	3.439354	7.602271	18.39173	0	5e-07	14.57473
17439029	3.067539	7.560991	18.20549	0	5e-07	14.46116
17263705	1.832571	6.954786	16.66016	0	9e-07	13.46095
17292634	2.630443	7.789606	16.40798	0	9e-07	13.28729
17428766	2.815436	7.753567	15.94746	0	9e-07	12.96200
17218060	4.428053	9.600955	15.72330	0	9e-07	12.79974

The same is done for the other two comparisons defined previously (*KOvsWT.VEN* and *INT*).

Gene Annotation

It is easier to identify the selected genes if the names are standard or the gene symbol is used. In this case, the arrays used are from the Affymetrix company. As we are working with Bioconductor, if we use an annotation package we would be able to access to the annotation packages created for the company to know and interpret easier the results obtained.

In our case, to obtain the differentialy expressed genes, the previously generated *annotatedTopTable* function is used.

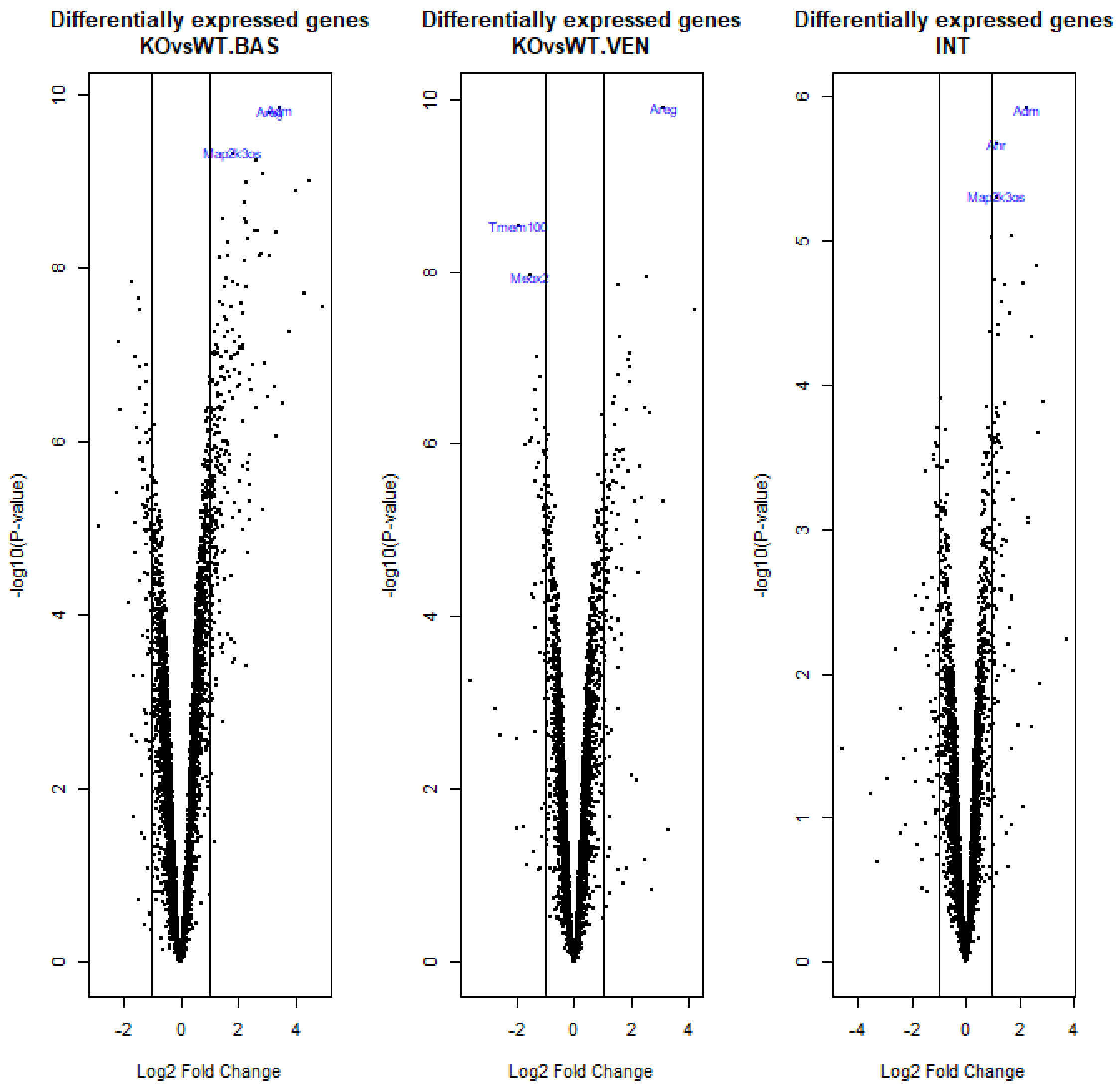
In order to know which genes are differentialy expressed in each case, the function is called for each of the comparisons described earlier. The results are once again stored in a *csv* file inside the *Results* folder.

Non of the ENTREZID columns in the tables show *NA* values so it is considered that enough genes are taken.

Visualizing differential expression

The three volcano plots (one per each of the comparison described in the contrast matrix) below show the overral differential expression. The three main genes obtained in the previous tables are shown in the graphs.

A visualization of the overall differential expression can be obtained using volcano-plots. These plots show if there are many or few genes with a large fold-change and significantly expressed or if this number is low. These graphs represent in the X-axis the changes of expression in logarithmic scale ("biological effect") and in the Y-axis the "minus logarithm" of the p-value or alternatively the `B` statistic ("Statistical effect"). The next figure shows the three volcano plots for the comparisons described. The names of the top 3 genes (i.e. the first four genes in the topTable) are shown in the plots.



Volcano plots for each of the comparisons with the names of the top 3 genes

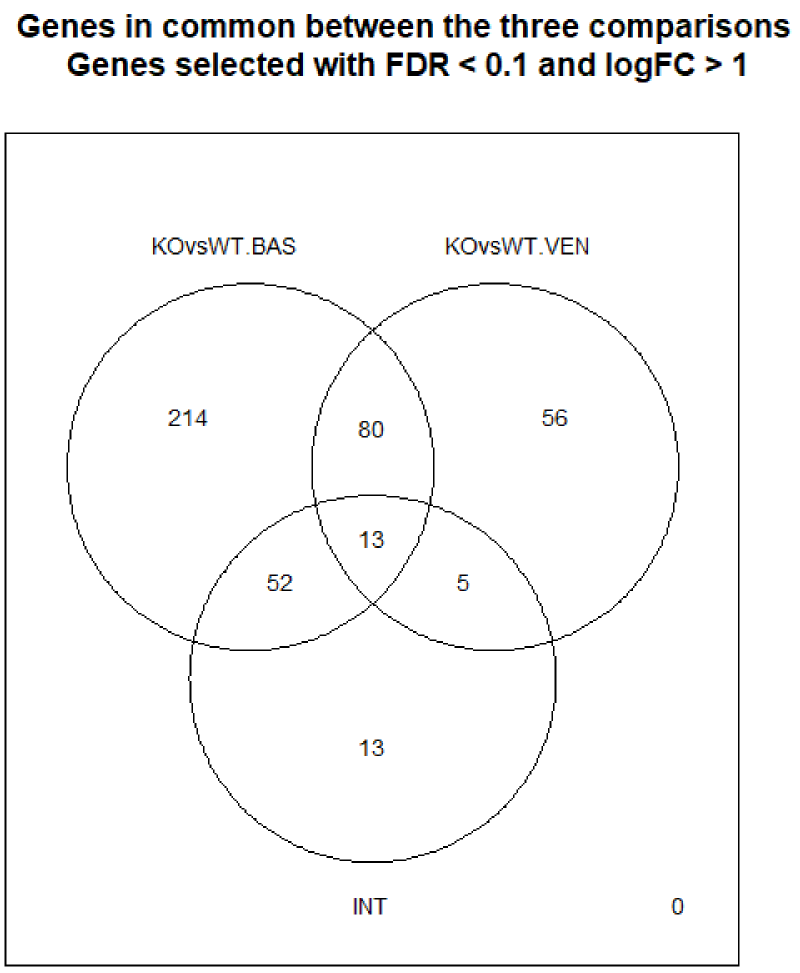
Multiple Comparisons

The function *decideTests* is used to annotate and count the genes selected in the three comparisons mentioned during this analysis.

```
       KOvsWT.BAS KOvsWT.VEN  INT
Down           82         48   19
```

```
NotSig       5632        5837 5908
Up            277         106   64
```

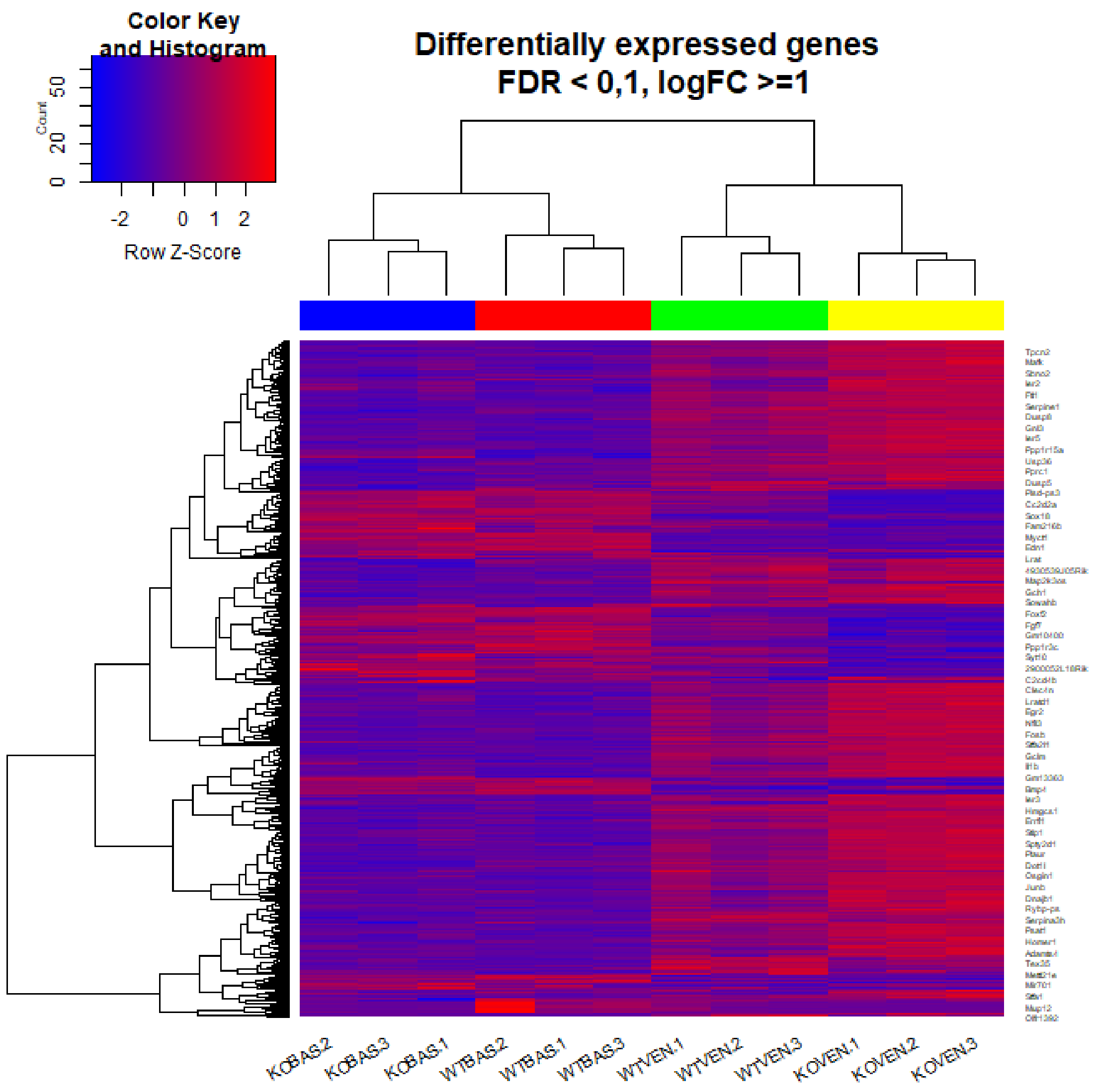
The numbers above show which genes are up and down significant in each comparison and which are not significant. To see the numbers (note that they are not the same ones) on a more ilustrative way, the function *VennDiagram* is used.



Venn diagram showing the genes in common between the three comparisons performed

Heatmaps

As a final step before the enrichment to the differentialy expressed genes, a heatmap of the genes that have been selected has been carried out. The heatmaps highlight distinct values that are significantly differential expressions (up-regulated or down-regulated). In this case, just the heatmap of the grouping has been displayed.



Heatmap for expression data grouping genes (rows) and samples (columns) by their similarity

In the *Heatmap* figure, we have seen that *WTBAS* and *KOBAS* are related with each other, as the colors between them are *similar*. Comparing with the baseline breathing, in the mechanical ventilation case (*VEN*), the wild type and KO mice are not that similar. The parts highlighted in blue and the ones highligted in red are different in the *BAS* and in the *VEN* case. One complements the other approximately: when one is red the other one is blue and viceversa.

Biological Significance of results

Even that in the subject's theory the use of *ReactomePA* and *GOstats* packages has been studied, in this case the package *clusterProfiler* is going to be used for the analysis.

List of genes that will be analyzed

The first step is to prepare a list of genes that will be analyzed. In order to introduce the correct data into de *clusterProfile* function, it is neccesay to load the `SYMBOL`, `ENTREZID` or `ENSEMBL` data. In this case *ENTREZID* is going to be introduced. For that, the three *topTab* tables generated before are selected and introduced into a *for* loop with the legnth of the whole *list*.

```
KOvsWT.BAS KOvsWT.VEN        INT
      3773       2682        549
```

We see in the numbers above how many genes are selected in each comparison. The interaction comparison has much less genes than the first two.

Application of the clusterProfiler function

Before the application of the biological significance function, note that with the *ReactomePA* function is neccesary to define the universe of all the genes that have at least one annotation in the Gene Ontology. Nevertheless, with the *clusterProfiler* function is not neccesary to do it because we will be using the *GO* keys of the function itself.

The Biological significance analysis will be applied to the three lists that we have been working with during this analysis: KOvsWT.BAS, KOvsWT.VEN and the intervention (INT). The function used for the enrichment analysis is *enrichGO*. We have performed the *enrichGO* function, and the attributes introduced have been the following: for *gene*, the list of genes created before; for *OrgDB* our organism (*Mm*), for *keyType* `ENTREZID`, in *ont*, the biological process of the genes differentially expressed, the *pAdjustMethod*, BH- Benjamini-Hochberg, and finally the p-value cut-off at 0.05.

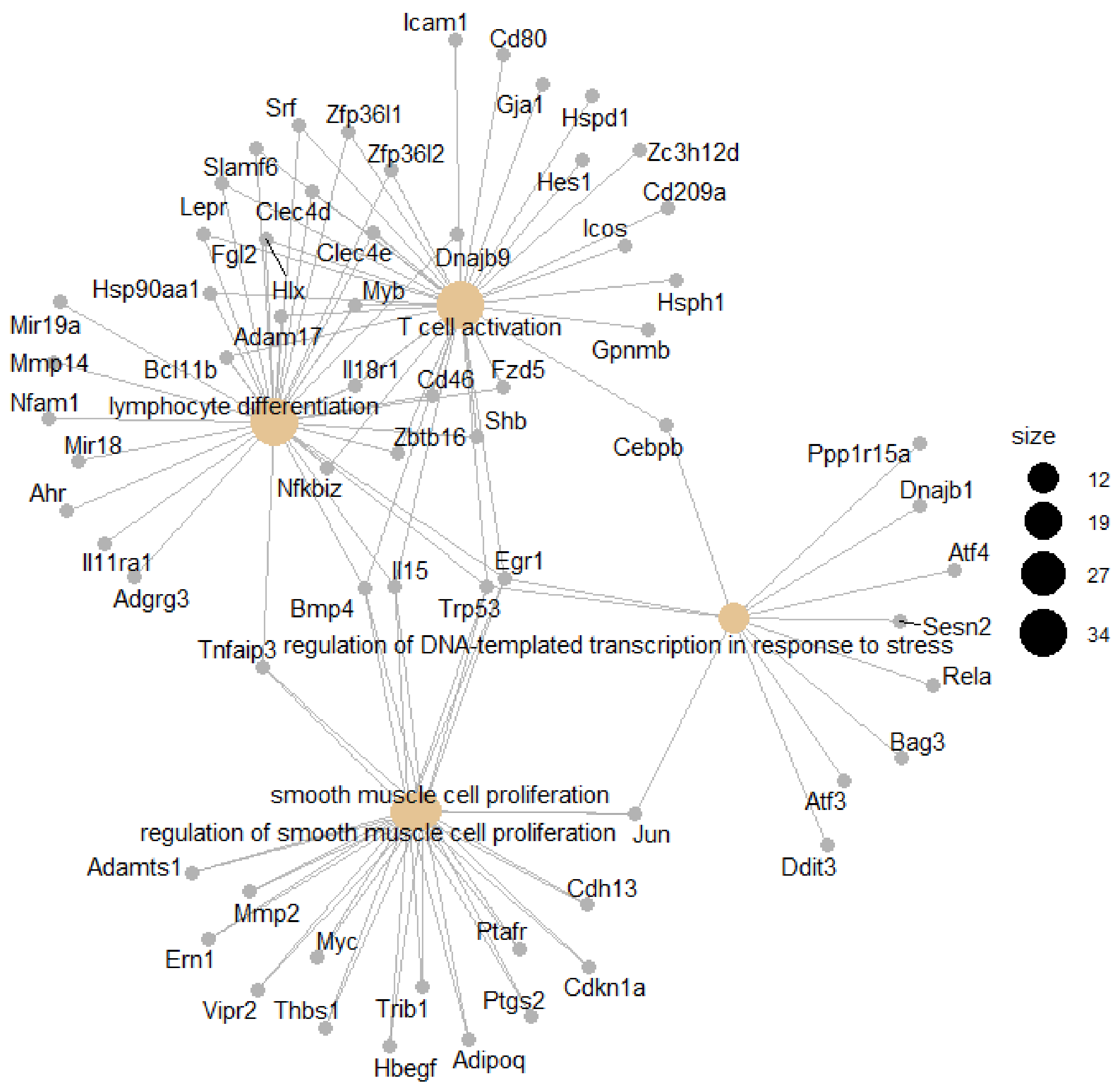
The data obtained is stored in three different *csv* files in the results folder of the working directory.

Due to the large amount of data, the output generated from the code above is not displayed in the output format of this work. This way, the document is more redable an easy to understand. Moreover, a cleaner version of this output is shown below as tables.

At the results folder, six different PDF files have been created, one per graph type (barplot - being this ones the most widely used types of graphs to visualize enriched terms- or cnetplot) and per comparison. Also, three *csv* files with the data analyzed have been created.

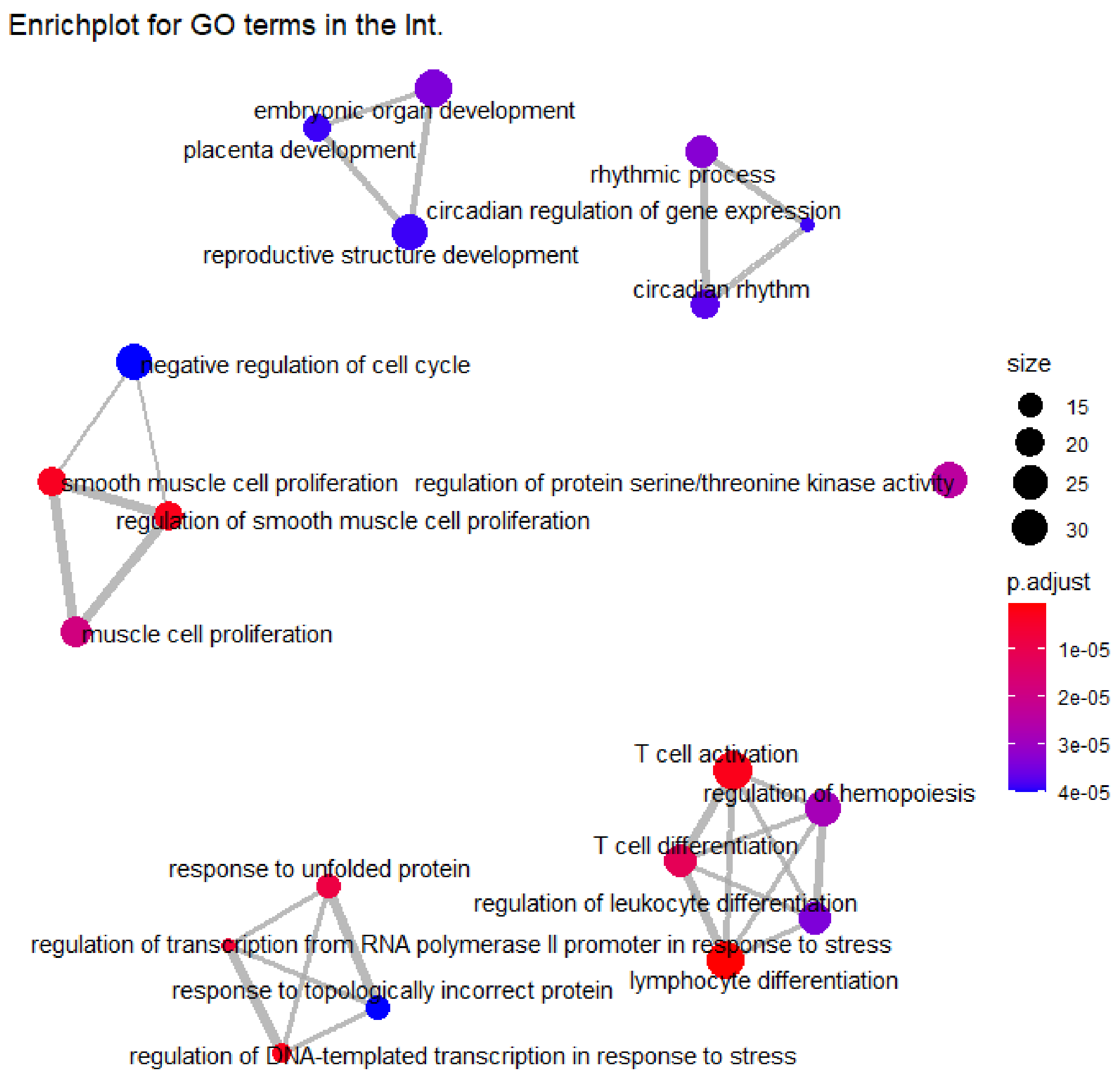
Ploting the results

As an example of the obtained results, the next figure shows the network produced from the genes selected in the comparison of the *interaction* between KO and wild type mice. The *cnetplot* function is used to extract the genes associated with the GO terms. The *cnetplot* function depicts the linkages of genes and biological concepts.



Network obtained from the enrichment analysis on the list obtained from the comparison between KO and WT in their interaction

Another way to visualize the GO Terms function is using enrichment maps and in this case, the function *emapplot* has been used to visualize the results. The enrichment map figure shows the first 20 GO terms for the interaction, colored by their p-value (depending on its value). If we visualize the PDF entiteled *ClusterProfile_Barplot.INT* from the *Results* folder, we will see that the results are the same, just representing them on a different way. The colours used are also the same as in the barplot.


Enrichplot for GO terms in the Int.
embryonic organ development
placenta development
reproductive structure development
rhythmic process
circadian regulation of gene expression
circadian rhythm
negative regulation of cell cycle
smooth muscle cell proliferation
regulation of protein serine/threonine kinase activity
regulation of smooth muscle cell proliferation
muscle cell proliferation
T cell activation
regulation of hemopoiesis
T cell differentiation
response to unfolded protein
regulation of leukocyte differentiation
regulation of transcription from RNA polymerase II promoter in response to stress
response to topologically incorrect protein
lymphocyte differentiation
regulation of DNA-templated transcription in response to stress
size
15
20
25
30
p.adjust
1e-05
2e-05
3e-05
4e-05

Enrichment analysis of the more relevant 20 GO terms in the Interaction

Another similar way of results visualitation is using the *heatplot*. This method is similar to *cnetplot*, while displaying the relationships as a heatmap. Nevertheless, it is not shown below because due to the large number of significant terms obtained, the display is not appropiate and easy to read (the data is superimposed).

Results

In the next three tables, the most relevant enriched pathways of each comparison defined in the beggining have been shown. The comparisons *KOvsWT.BAS* and *KOvsWT.VEN* have the same enriched pathways in the first, second and fifth positions (we have just plot the tables until the fifth row in each comparison). This parthways are: positive regulation of cytokine production (GO:0001819), regulation of vasculature development (GO:1901342) and regulation of angiogenesis

(GO:0045765). The interaction comparison has less counts (that`s way its cnetplot has been shown in the documents, because is has less information than the others). At least in the first five GO terms, the interaction comparison has no equal terms with *KOvsWT.BAS* and *KOvsWT.VEN*.

First rows and columns for ClusterProfiler results on KOvsWT.BAS comparison

	Description	GeneRatio	BgRatio	pvalue	p.adjust
GO:0001819	positive regulation of cytokine production	139/3451	449/23210	2.27342254111363e-18	1.40383841913767e-14
GO:1901342	regulation of vasculature development	110/3451	341/23210	3.29026841977449e-16	1.01587037460537e-12
GO:0090130	tissue migration	94/3451	284/23210	7.97036949585269e-15	1.25852348728329e-11
GO:0002683	negative regulation of immune system process	135/3451	473/23210	1.00863908681625e-14	1.25852348728329e-11
GO:0045765	regulation of angiogenesis	99/3451	307/23210	1.01904735812412e-14	1.25852348728329e-11

First rows and columns for ClusterProfiler results on KOvsWT.VEN comparison

	Description	GeneRatio	BgRatio	pvalue	p.adjust
GO:0001819	positive regulation of cytokine production	116/2445	449/23210	1.82660287931348e-20	5.73296403123163e-17
GO:1901342	regulation of vasculatur	97/2445	341/23210	1.90527219382906e-20	5.73296403123163e-17

	e development				
GO:0031349	positive regulation of defense response	99/2445	370/23210	8.73813655237896e-19	1.68404173191379e-15
GO:0001667	ameboidal-type cell migration	105/2445	406/23210	1.11933647850701e-18	1.68404173191379e-15
GO:0045765	regulation of angiogenesis	86/2445	307/23210	7.99434220616873e-18	9.62199027934468e-15

First rows and columns for ClusterProfiler results between the KO and WT interaction

	Description	GeneRatio	BgRatio	pvalue	p.adjust
GO:0030098	lymphocyte differentiation	32/498	395/23210	1.48601228701518e-10	6.34824449012884e-07
GO:0043620	regulation of DNA-templated transcription in response to stress	12/498	56/23210	1.97841111964711e-09	2.59902612790751e-06
GO:0042110	T cell activation	34/498	491/23210	2.41718552457262e-09	2.59902612790751e-06
GO:0048660	regulation of smooth muscle cell proliferation	19/498	163/23210	2.43354506358381e-09	2.59902612790751e-06
GO:0048659	smooth muscle cell proliferation	19/498	168/23210	4.04846692643276e-09	3.45901014194415e-06

Discussion

In the Lopez-Alonso et al. (2018) article, the authors have confirmed that there is evidence that the mechanical ventilation modifies the mechanical properties of the nuclear envelope, which can cause a lung injury. Even if the results haven`t been biologically analyzed between the comparisons, we have seen in general that some of the pathways obtained between both of the ventilatory strategies (baseline and mechanically ventilated) are the same. Nevertheless, the pathways when combining genotypes and ventilatory strategies (in the interaction) change completely. In our case, we have not completed the study with the neccesary biological knowledge, so we have not interpreted the results in that way. Instead, in the Lopez-Alonso et al. (2018) they have confirmed their results with confirmatory experiments in human cells, obtaining them from autopsy samples.

As a further research it would be interesting to include the data obtained from these samples to work with them in microarrays, to be able to obtain the same results as in the article previously mentioned. This way it would be easier to compare our enrichment analysis results with complete real studies.

Appendix

The code used is available at the GitHub's repository mentioned at the beggining of this document, as an R script, named *Renteria_Jone_ADO_PEC1*.

References

Lopez-Alonso, Ines, Jorge Blazquez-Prieto, Laura Amado-Rodriguez, Adrian Gonzalez-Lopez, Aurora Astudillo, Manuel Sanchez, Covadonga Huidobro, Cecilia Lopez-Martinez, Claudia C Dos Santos, and Guillermo M Albaiceta. 2018. "Preventing Loss of Mechanosensation by the Nuclear Membranes of Alveolar Cells Reduces Lung Injury in Mice During Mechanical Ventilation." *Science Translational Medicine* 10 (456): eaam7598.