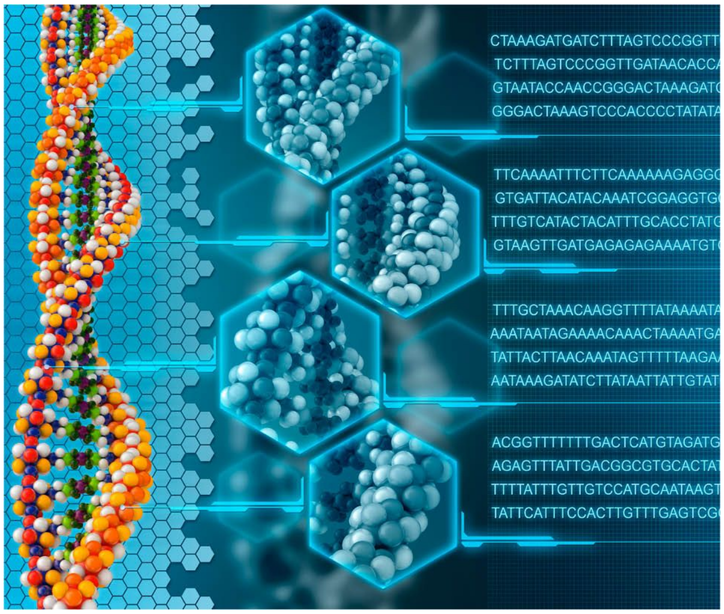
PEC 1

Omic Data Analysis

María Ponce Rodríguez

Msc in Bioinformatics and Biostatistics

24 April, 2022


CTAAAGATGATCTTTAGTCCCGGTT
TCTTTAGTCCCGGTTGATAACACCA
GTAATACCAACCGGGACTAAAGATG
GGGACTAAAGTCCCACCCCTATATA
TTCAAAATTTCTTCAAAAAAGAGGG
GTGATTACATACAAATCGGAGGTG
TTTGTCATACTACATTTGCACCTATG
GTAAGTTGATGAGAGAGAAAATGT
TTTGCTAAACAAGGTTTTATAAAATA
AAATAATAGAAAACAAACTAAAATGA
TATTACTTAACAAATAGTTTTTAAGA
AATAAAGATATCTTATAATTATTGTAT
ACGGTTTTTTTGACTCATGTAGATG
AGAGTTTATTGACGGCGTGCACTAT
TTTTATTTGTTGTCCATGCAATAAGT
TATTCATTTCCACTTGTTTGAGTCG

Contents

1 Abstract

2 Objectives

The objective of this project is to perform a microarrays analysis to determine the gene expression profiling on the different groups of the study based on the COVID-19 infectious disease and the type of symptoms.

The main objectives of this project can be defined as:

1. Differentially expressed genes between symptomatic and asymptomatic COVID-19 cases.
2. Differentially expressed genes between symptomatic COVID-19 cases and uninfected controls.
3. Differentially expressed genes between asymptomatic COVID-19 cases and uninfected controls.

3 Material and methods

3.1 Study Subjects

The study main focused in the host key mechanisms against SARS-CoV-2 infection, and to find differentially expressed genes between symptomatic, asymptomatic hosts and healthy individuals. The study found that the type I interferon pathways in in SARS-CoV-2-infected individuals (*Homo sapiens*s) from Pakistan were upregulated in asymptomatic hosts (Masood et al. 2021).

According to the information, this is an observational study where we can find different groups which are described in Table 1. This lead us to define the three groups used in the study which are control, asymptomatic and symptomatic. The control group are SARS-CoV-2 negative, or uninfected healthy individuals. The symptomatic and asymptomatic are SARS-CoV-2 positive but they can present symptoms or not present them, respectively.

Table 1: The groups of the observational study.

	SARS-CoV-2 positive	**SARS-CoV-2 negative**
With symptoms	Asymptomatic (A)	
Without symptoms	Symptomatic (S)	Control (C)

The study sample were 47 individuals, in which eleven were symptomatic COVID-19 cases, eighteen asymptomatic COVID-19 cases, and eighteen control individuals. The control group were uninfected healthy individuals. A subgroup of samples were selected randomly in this project, however, each group is going to be represented by four individual samples.

The once color microarrays where obtained using the RNA from the blood collected in Plasma/EDTA tube from the study subjects. The blood samples were taken within 24-48 hours after the confirmed COVID-19 diagnosis, and the transcriptome was determined using Clariom S RNA microarray, Affymetrix assay (Affymetrix). The information was found in GEO (*Gene Expression Omnibus*) with the code GSE177477.

3.2 Bioinformatic tools and procedures

The analysis performed in this activity, was realized using R version 4.1.2 (2021-11-01), and specific libraries developed to analyze microarray data such as Bioconductor. In order to get more fully understand the methods described in this section, you could consult (R. Gentleman 2004; Robert Gentleman et al. 2005).

The libraries used in this project can be seen in the Appendix, as well as, the whole code.

3.3 Data selection and loading data

A subgroup of samples were selected randomly in this project with the following seed `8795`, 15 out 47 RNA samples were selected, each group was fairly represented with four samples. The files selected for this project were GSM5374843_S5.CEL, GSM5374841_S3.CEL, GSM5374848_S10.CEL, GSM5374844_S6.CEL, GSM5374846_S8.CEL, GSM5374858_S20.CEL, GSM5374851_S13.CEL, GSM5374857_S19.CEL, GSM5374859_S21.CEL, GSM5374860_S22.CEL, GSM5374878_S40.CEL, GSM5374870_S32.CEL, GSM5374868_S30.CEL, GSM5374873_S35.CEL, GSM5374884_S46.CEL. An additional file called `targets.csv` contains information about the data such as the identifier of the .CEL file (row names), the group (`Groups`), and the identifier of the sample (`ShortName`).

The .CEL files with the `targets` are going to be unify in order to create an `ExpressionSet` called `rawdata` used in this project. Furthermore, the names of the .CEL files were changed to improve the interpretation of the results, the short names were used instead.

3.4 Preprocessing

The preprocessing consisted in three keys steps, (1) data exploration and quality control, (2) normalization and (3) non-specific data filtering. In addition, the data exploration and the quality control was also performed in the normalized data.

3.4.1 Data exploration and quality control

The data exploration and the quality control process were performed using `rawdata`, which contain the raw data of the microarrays. The exploration of the data was based on the visualization of the intensity of the signals using a scatterplot and boxplot. Afterwords, PCA (Principal Components Analysis), heatmap, and hierarchical cluster were performed to see how the groups were grouped.

The function `arrayQualityMetrics` creates a complex summary with more graphics, and it was used to complement the previous exploration of the data, such as the MA plots.

3.4.2 Normalization

The normalization of the data set was performed using `rawdata`. The method applied was RMA (*Robust Multiarray Average*) which perform a background correction, the normalization and a summary of the values among each group of probesets, and the result is the absolute expression value (Bolstad et al. 2003). The absolute expression matrix is essential for the analysis, and it was called `eset_rma`.

After the normalization, we replicate the exploration and quality control on the normalized data.

3.4.3 Non-specific data filtering

The non-specific data filtering is used to eliminate those genes in which the variability could be caused by randomness, and it would result in no differentially expressed genes (Hackstadt and Hess 2009). In this case, `nsFilter` function was used which eliminate the outliers using the IQR (*Interquartile range*) method, remove the most variables probesets and the one which didn't present NCBI Entrez database identifier (`ENTREZID`). The last step will ensure that all the genes are annotated in the future steps.

As a result of the process, a new `ExpressionSet` called `eset_filtered` is obtained. This `ExpressionSet` is the absolute expression matrix from the normalization but with fewer probesets.

3.5 Differentially Expressed Genes

This process is based on the (1) statistical analysis where we test the hypothesis and the possible multiple comparison tests, (2) the annotation of the data, (3) the analysis of the gene list. In the first step, we obtain the differentially expressed genes for each contrast which is the key step of this process. After this step is finish, we can perform the pathway enrichment analysis.

3.5.1 Statistical Analysis

The linear regression model of this project has a unique factor (α) with three levels (i) which are asymptomatic ("A"), healthy individuals or controls ("C") and symptomatic ("S"). In addition, the number of observations (j) goes from 1 to 12 which are the number of samples of the study.

The resulting model is:

$$Y_{ij} = \alpha_i X_{ij} + \epsilon_{ij} \qquad \text{i = 1,2,3, j = 1,..,12}$$

The reduced design matrix, obtained from the matrix contained in `eset_filtered`, of this model is:

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

The hypothesis that are going to be tested are:

- H_0: There are no significant diference in the genes expressed in the symptomatic and asymptomatic COVID19 cases.
- H_1: There are significant diference in the genes expressed in the symptomatic and asymptomatic COVID19 cases (**SvsA**).

- H_0: There are no significant difference in the genes expressed in the symptomatic COVID19 cases and healthy individuals.
- H_1: There are significant difference in the genes expressed in the symptomatic COVID19 cases and healthy individuals (**SvsC**).

- H_0: There are no significant difference in the genes expressed in the asymptomatic COVID19 cases and healthy individuals.
- H_1: There are significant difference in the genes expressed in the asymptomatic COVID19 cases and healthy individuals (**AvsC**).

As a result of the hypothesis testing, the contrast matrix is:

$$\begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

In order to test the hypothesis and create the model the `limma` package will be used (Smyth 2004). As a result, the contrasts are going to be found in `lmresult` which contain information about the log fold change (*log FC*), the average expression, t-statistic, the p-value, the adjusted p-value following Benjamini and Hochberg, and the the B-statistics for each contrast. The Benjamini and Hochberg method is used to control the false positive rate (Benjamini and Hochberg 1995).

The list of differential expressed genes are obtained using some of the parameters describe in the previous paragraph. To be more preciously, the p-value, the log FC and the adjust p-value Benjamini and Hochberg method (`adjust = "fdr"`) were used. In order to be considered a differentially expressed gene the p-value (`p.value = 0.05`) should be under the significance level ($\alpha = 0.05$), and the log FC can be less than 2 (`lfc = 2`).

The lists with the differential expressed genes, resulting of each contrast, are saved in `toptableSvsA`, `toptableSvsC`, and `toptableAvsC`.

After the hypothesis testing a **multiple comparison test** should be performed to find out which differentially expressed genes change among the different contrasts. The function `decidetests` from the `limma` package simplify this process.

In order to perform the multiple comparison test, we need to use the data from `lmresult` with the function `decidetests`. This function would tell us which genes are upregulated (+1), downregulated (-1) or are not differentially expressed (0). First, the function perform the selection of the differentially expressed data using the same parameter describe early in this section (`adjust = "fdr"`; `p.value = 0.05`; `lfc = 2`). In addition, we chose `method = "separate"` which is equivalent to use `topTable`, which is the same as treating each contrast as a different sets of data.

The result of the multiple comparison test is save as 'res.selected" which can be represented as a Venn Diagram that would represent the numbers of differentially expressed genes exclusive in each contrast and shared between the them.

3.5.2 Annotated data

The annotation package selected according to Clariom S RNA microarray, Affymetrix assay used to obtain the transciptome is `"clariomshumantranscriptcluster.db"`. The standard gene symbol ("SYMBOL"), and NCBI Entrez database ("ENTREZID") annotations were selected.

The lists with the differential expressed genes, resulting of each contrast, `toptableSvsA`, `toptableSvsC`, and `toptableAvsC`, were annotated. As a result, we obtain three different files called `SvsA_anot`, `SvsC_anot` and `AvsC_anot`. All this information was gather together in `listOfToptable`. In addition, the result of the three contrasts was obtained from `lmresult` which was annotated too, and save as `lmresult_anot`.

Finally, the gene enrichment analysis is based on the NCBI Entrez database ("ENTREZID") annotations for the universe and the gene set of the contrast. The universe is the total number of genes from the expression set of the `eset_filtered`, and save as `dataGenes`. The second list that we obtain, was selected form the annotated tables of each contrast were only the ENTREZID column was chosen. This list was called `listOfSelected`.

3.5.3 Analysis of the gene lists

According to the information in the `res` data set, we could find out which genes were upregulated (1), and downregulated (-1). We can extract which genes were upregulated and the downregulated in each contrast.

The selection of those genes could be reproduced using a loop. First of all, a list which contains all the annotated genes form the contrasts was created, and named `listOfToptables`. Secondly, the `res` data set was used to select the names of the genes that were upregulated, and downregulated. Thirdly, those genes names were selected in the differentially expressed genes from the Toptab annotated table. From the Toptab, we used the standard gene symbol (`SYMBOL`) annotation.

```r
for (i in 1:length(colnames(res))) {
    # Contrast
    comp <- colnames(res)[i]
    # Annotated Toptable
    anot_vs <- listOfToptables[[i]]
    # Upregulated genes
    up <- rownames(res[which(res[, i] == 1), i])
    genes_up <- anot_vs$SYMBOL[which(anot_vs$PROBEID %in% up)]
    # Downregulated genes
    down <- rownames(res[which(res[, i] == -1), i])
    genes_down <- anot_vs$SYMBOL[which(anot_vs$PROBEID %in% down)]

    # Saved files
    write.csv(genes_up, file = paste0("./Results/", "Upregulated_genes_", comp, ".csv"),
        row.names = FALSE)
    write.csv(genes_down, file = paste0("./Results/", "Downregulated_genes_", comp,
        ".csv"), row.names = FALSE)
}
```

The upregulated genes in the contrasts *SvsA*, *SvsC*, and *AvsC* were saved as `Upregulated_genes_SvsA.csv`, `Upregulated_genes_SvsC.csv` and `Upregulated_genes_AvsC.csv`, and the downregulated genes as `Downregulated_genes_SvsA.csv`, `Downregulated_genes_SvsC.csv`, and `Downregulated_genes_AvsC.csv`, respectively.

3.6 Pathway Enrichment Analysis

The pathway enrichment analysis performed in this project was the *Gene Set Enrichment Analysis* (GSEA), also known as *Functional Class Scoring*. This step leads the road to the biological interpretation of the results, with this analysis we could outline the biological pathways from Reactome in which the differentially expressed genes are involved. In this case, down and upregulated genes would not be separately analyzed.

The `ReactomaPA` (Yu and He 2016) package for the analysis, and the `enrichplot` package to visualize the information. In order to perform the enrichment analysis we use the list of `dataGenes` which represent the universe, and list with NCBI Entrez database ("ENTREZID") annotation of the differentially expressed genes from each contrast which was named `listOfSelected`.

After the analysis, the information of each contrast was saved in `ReactomePA.Results.SvsA.csv`, `ReactomePA.Results.SvsC.csv`, and `ReactomePA.Results.AvsC.csv`. In addition, the data is visualize with barplots (`barplot`), dotplots (`dotplot`) which can be used to see the biological pathways in which the genes are involved and see how many of them are involved, and Gene-Concept Network (`cnetplot`) which shows the genes that are involved in the biological pathways (Yu 2019).

4 Results and discussion

4.1 Preprocessing

The 15 samples selected for this study presented probeset. After the normalization applying the RMA method, the number of probeset was reduced to a 9.05% of the original ones, leaving 27189 probesets.

In this section, we are going to compare the results between the raw and normalized data simultaneously. As we can see in Figure 1 left, the samples *S1*, *S2*, *A4*, *A5*, *N3* and *N5* presented a higher standard deviation of the intensities and the median values seems to be higher than the others among the raw data. In addition, we confirmed that the normalization was effective, media and standard deviation of the intensities are homogeneous among the samples.

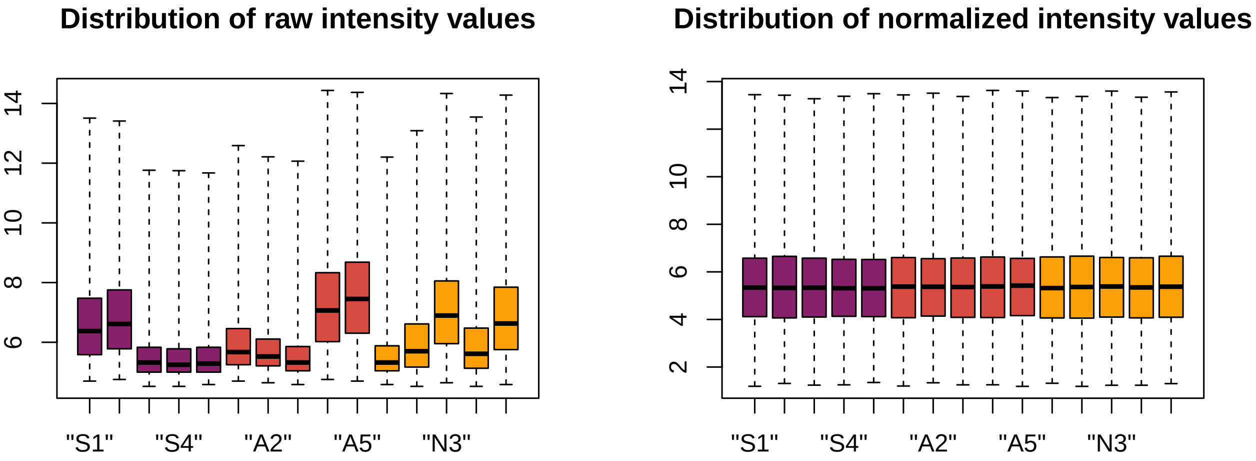


Figure 1: Boxplot of the intensisties of the microarrays of each sample. We can see the distribution of the raw data (left), and normalized data (right).

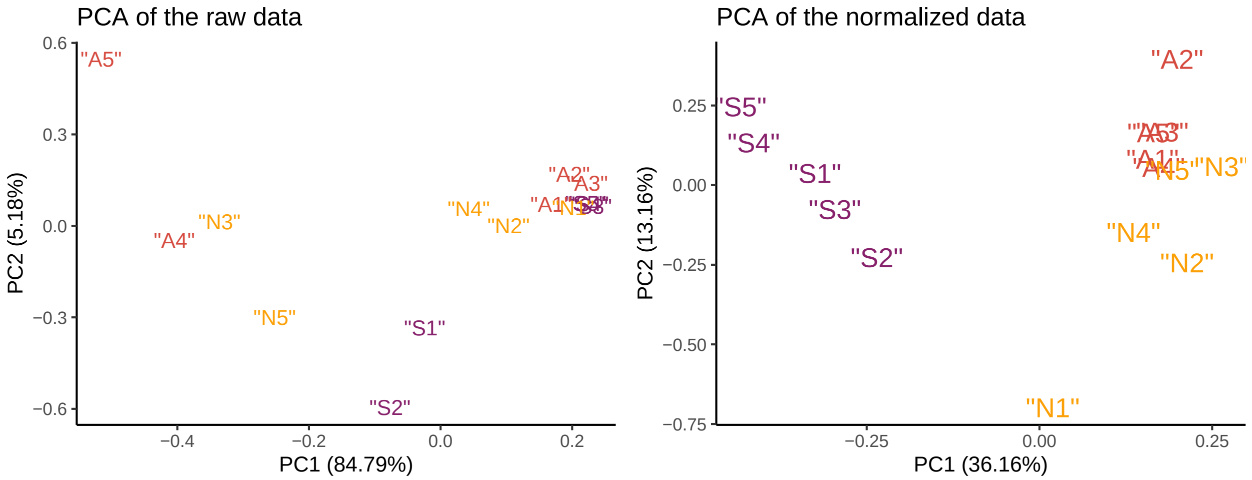


Figure 2: PCA (Principal Component Analysis) plot of the raw data (left), and normalized data (right).

Moreover, the PCA (Figure 2, left), showed that the first principal component showed that the 84.79% of the variance was caused by *A5*, *A4*, *N3* and *N5*. However, after normalizing the data the variance of the data could mainly be explained by the symptoms result of the COVID-19 infection ($PC1 = 36.16\%$). In this case, the second component ($PC2 = 13.16\%$) of the PCA could try to explain the difference between the asymptomatic COVID-19 cases, and the healthy controls but it seems to be some overlap samples, which are *A1*, *A3*, *A4*, *A5* and *N5*. This lead us to think that the asymptomatic COVID-19 cases, and the healthy controls aren't grouped according to their actual group.

Figure 3 gives us information about how the data is grouped, as well as the heatmap. As i the PCA plot of the normalized data the samples *A5*, *A4*, *N3* and *N5* are grouped together in a different group, while the rest of the samples are link together but not their true group. However, the normalized data shows us that the symptomatic COVID-19 cases can be easily separate from other groups, and what is most interesting is that the sample *N1* is isolated from the healthy control group. In addition, the asymptomatic COVID-19 cases are similar to the healthy controls but are differenciated.

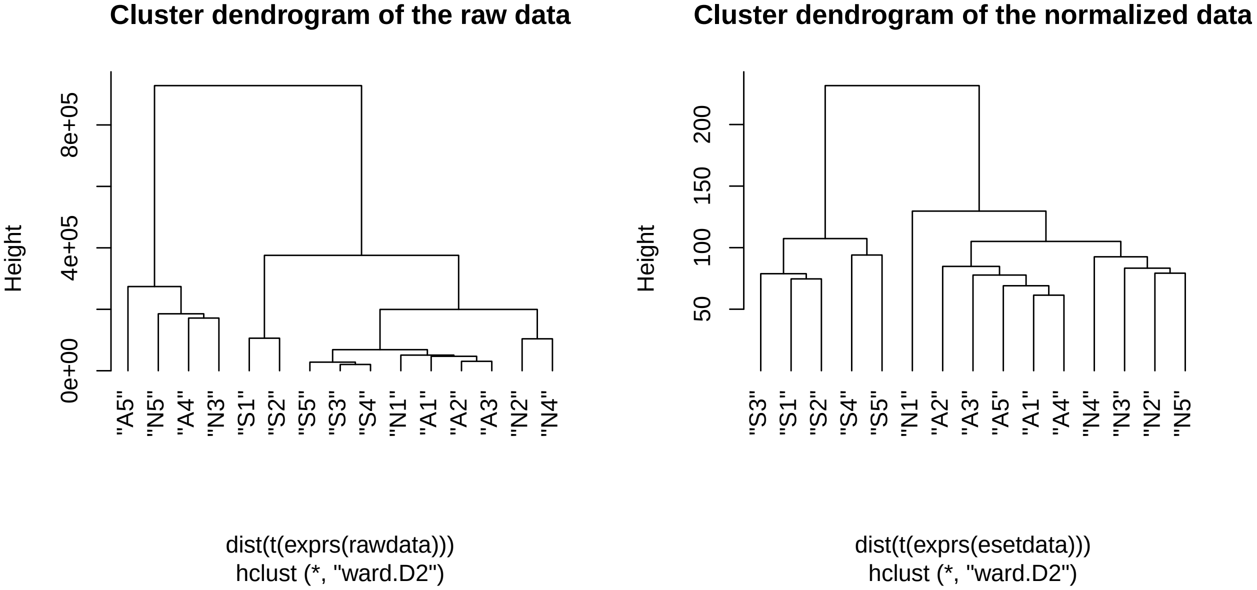


Figure 3: Cluster dendrogram of the raw data (left), and the normalized data (right).

Furthermore, the samples *A5*, *A4*, *N1*, *N3* and *N5* could be a possible problem while analyzing the data, because they don't seem to follow good standard quality metrics. The possible removal of these samples from the study should be considered when the analysis with the complete range of sample was performed. Indeed, they could difficult the descovery of the differentially expressed genes among the groups.

Afterwards the normalization, the data was filtered. The filtering process was based on the elimination of possibles outliers using the IQR method, selecting the 75% least variable genes and selecting the genes with NCBI Entrez Database identifier. According to this, the final number of genes were 4620 which means that a 16.99% of the normalized data was chosen, and a 1.54% of the raw data was selected.

The filtering process could be a reduction in the variability of the data, and could supposed the elimination of differentially expressed genes in the following contrast just be caused they don't present an Entrez identification or because the variability is over or under the third and first quartile, respectivel. This will lead as into a reduction of the genes involved in the process, and some of them could be key in some biological pathways.

4.2 Differentially expressed genes

4.2.1 Statistical Analysis

A total of 323 differentially expressed genes were identify between the three data set based on the logFC ($logFC > 2$), and the p-value ($p - value \leq 0.05$) from the 4620 prepossessed genes. The symptomatic and asymptomatic COVID-19 cases ($SvsA$) presented 179 differentially expressed genes, the symptomatic COVID-19 cases and uninfected controls ($SvsC$) a total of 140, and the asymptomatic COVID-19 cases and uninfected controls ($AvsC$) presented 4 differentially expressed genes.

Further dissection among the differentially expressed gene (Figure 4) showed that between uninfected controls and COVID-19 cases, there wasn't a singular differentially expressed genes. On the contrary, there were 77 (23.84%) differentially expressed genes between the symptomatic and asymptomatic COVID-19 cases, 36 (11.15%) between the symptomatic COVID-19 cases and uninfected controls, and 2 (0.62%) between asymptomatic COVID-19 cases and uninfected controls. In addition, 102 (31.58%) differentially expressed genes can be found among the symptomatic and asymptomatic COVID-19 cases, and symptomatic COVID-19 cases and controls.

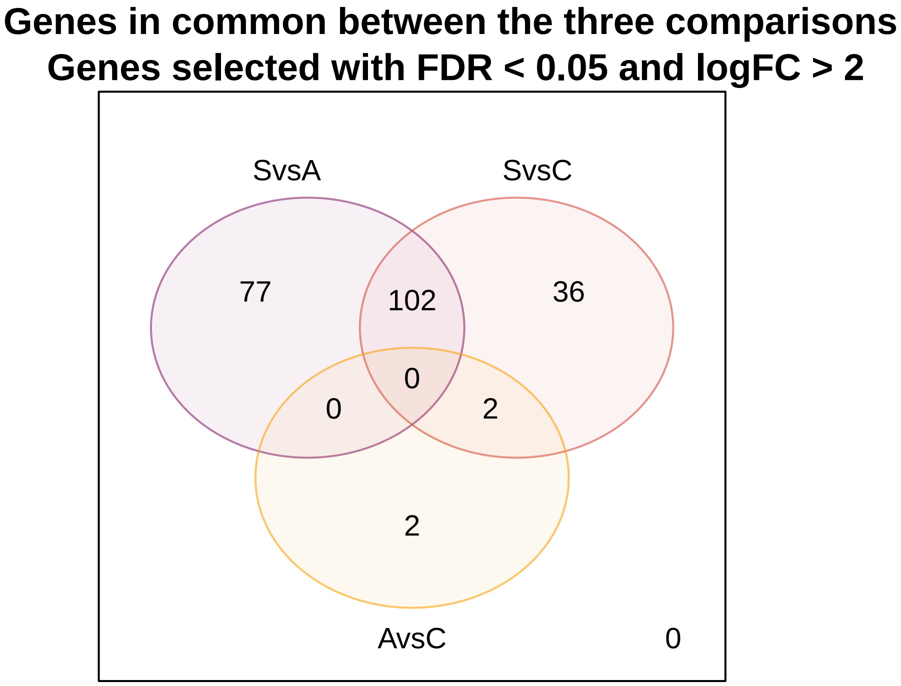


Figure 4: Venn Diagram decipts the overlapping differentially expressed genes between the symptomatic and asymptomatic COVID-19 cases (SvsA, purple), the symptomatic COVID-19 cases and uninfected controls (SvsC, orange), and the asymptomatic COVID-19 cases and uninfected controls (AvsC, yellow).

Indeed, the 102 differentially expressed genes shared between the symptomatic and asymptomatic COVID-19 cases, and symptomatic COVID-19 cases and controls, were actually genes that are exclusively expressed in the symptomatic COVID-19 cases. An the 2 differentially expressed genes shared between the asymptomatic COVID-19 cases and uninfected controls, and the symptomatic COVID-19 cases and uninfected controls were exclusively expressed in the uninfected controls. groups.

Table 2: Summary table with the differentially expressed genes as down-regulated (Down), and up-regulated (Up), and the non-differentially expressed genes (NotSig).

	SvsA	SvsC	AvsC
Down	31	54	3
NotSig	4441	4480	4616
Up	148	86	1

The vast majority of the genes weren't differentially expressed as we can see in Table 2. However, the differentially expressed genes could be upregulated or downregulated. There were a total of 148 upregulated genes in the symptomatic and asymptomatic COVID-19 cases (*SvsA*), 86 in the symptomatic COVID-19 cases and uninfected controls (*SvsC*), and 1 in the asymptomatic COVID-19 cases and uninfected controls (*AvsC*).

On the contrary, downregulated genes were 31 in the symptomatic and asymptomatic COVID-19 cases ($SvsA$), 54 in the symptomatic COVID-19 cases and uninfected controls ($SvsC$), and 3 in the asymptomatic COVID-19 cases and uninfected controls ($AvsC$).

4.2.2 Analysis of the gene lists

After the annotation process, we found out which genes were differentially expressed in each group. The Volcano plot (Figure 5) will outline some of the selected genes, but in order to really know which genes were up or downregulated will list the genes.

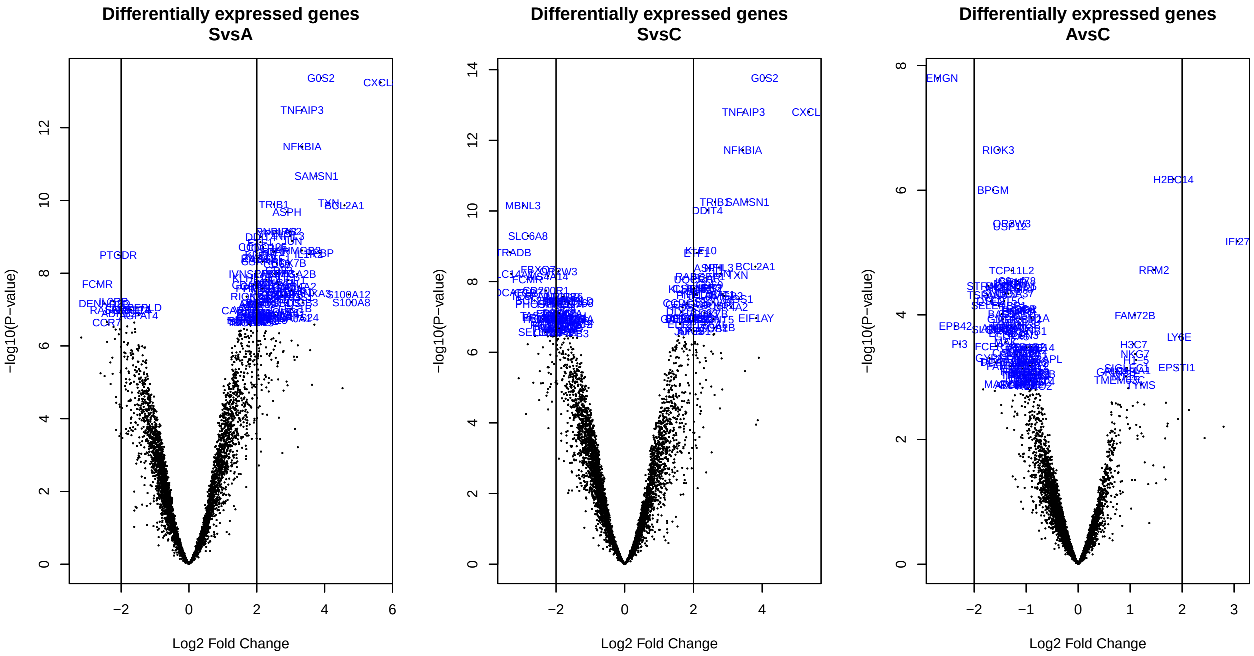


Figure 5: Volcano plot of the symptomatic and asymptomatic COVID-19 cases, symptomatic COVID-19 cases and uninfected controls and asymptomatic COVID-19 cases and uninfected controls.

First, the upregulated genes among the asymptomatic COVID-19 cases and uninfected controls ($AvsC$) are IFI27, and the downregulated, HEMGN, EPB42, PI3.

Secondly, the symptomatic and asymptomatic COVID-19 cases ($SvsA$) showed the highest number of differentially expressed genes. The up and downregulated genes were 148 and 31. On the following list we can see the upregulated downregulated genes.

- **Upregulated genes in SvsA**: RPL11, PLK3, UQCRH, S100A9, DNM3, RGS18, RGS1, RGS2, G0S2, SYF2, JUN, LAMTOR5, S100A12, S100A8, RIT1, SELP, PTGS2, RPS7, YPEL5, RPL31, IL1R2, HAT1, ZC3H15, NR4A2, TFPI, IL18R1, CALM2, LSM3, CTDSPL, CSTA, B3GNT5, CSRNP1, LTF, PROS1, CXCL8, PF4V1, ANXA3, RPL34, LARP7, GUCY1B1, PF4, PPBP, PDE5A, HMGB2, SUB1, AP3S1, RPL37, PDE4D, ELOVL7, ETF1, SPARC, DUSP1, NSA2, ZMAT2, CD83, AIF1, SH3BGRL2, ARG1, TNFAIP3, F13A1, IER3, CRISP3, VNN1, CCDC126, GNG11, PRKAR2B, SBDS, RABGEF1, PNPLA8, ENY2, TRIB1, ASPH, RPL7, UQCRB, KLF10, DNAJA1, ANXA1, HSDL2, LCN2, NFIL3, TXN, SAT1, COX7B, NDUFA1, CETN2, RPL36A, RPL36A-HNRNPH2, DDX3Y, EIF1AY, PFKFB3, DDIT4, RPS24, IDI1, ANKRD22, RGS10, RPL21, TCN1, MMP8, CLEC4D, RPL41, MRPL51, CLEC4E, CD69, CLEC2B, CLEC1B, GPR84, CD63, MYL6, POMP, OLFM4, KLF5, HMGB1, COMMD6, IRS2, NFKBIA, RPL36AL, NDUFB1, PSMA6, THBS1, PSMA4, SRP14, COPS2, RSL24D1, RPS27L, BCL2A1, RPL9, RPS17, HP, RPL27, IFT20, ITGA2B, ITGB3, RPL26, PSTPIP2, RPL17-C18orf32, JUNB, PPP1R15A, FCAR, GMFG, MCEMP1, MYL9, MMP9, RBM39, SLPI, MAP3K7CL, ETS2, SAMSN1, NDUFA6

- **Downregulated genes in SvsA**: CD2, ZNF683, DENND2D, SLAMF6, SH2D1B, FCMR, GNLY, CX3CR1, CISH, IL7R, TCF7, ADRB2, CD180, MANEA, TLR7, GPR174, CTSW, CD3E, CD3G, MPEG1, KLRG1, KLRD1, TESPA1, CD27, KLF12, PTGDR, RASGRP1, ADGRG1, CCR7, NFATC2, IL2RB

Thirdly, the symptomatic COVID-19 cases and uninfected controls (*SvsC*) differentially expressed genes, 86 were upregulated genes, and 54, downregulated.

- **Upregulated genes in SvsC**: UQCRH, S100A9, RGS1, G0S2, JUN, LAMTOR5, S100A12, S100A8, RIT1, PTGS2, YPEL5, IL1R2, NR4A2, IL18R1, CSTA, B3GNT5, CSRNP1, LTF, CXCL8, ANXA3, RPL34, PPBP, HMGB2, SUB1, AP3S1, EGR1, RPL37, ETF1, DUSP1, CD83, ARG1, TNFAIP3, IER3, VNN1, PRKAR2B, SBDS, RABGEF1, PNPLA8, TRIB1, ASPH, RPL7, UQCRB, KLF10, DNAJA1, ANXA1, LCN2, NFIL3, TXN, SAT1, COX7B, RPL36A, DDX3Y, EIF1AY, DDIT4, RPS24, IDI1, RPL21, TCN1, MMP8, CLEC4D, RPL41, CLEC4E, CD69, CLEC1B, GPR84, CD63, CKAP4, MYL6, POMP, IRS2, NFKBIA, RPL36AL, NDUFB1, THBS1, SRP14, BCL2A1, HP, ITGA2B, ITGB3, RPL26, FCAR, GMFG, MCEMP1, MMP9, SAMSN1, NDUFA6
- **Downregulated genes in SvsC**: DENND2D, SELENBP1, SLAMF6, SH2D1B, FCMR, GYPC, STRADB, FAM117B, CD28, PARP15, CX3CR1, CISH, CD200R1, SNCA, TSPAN5, LEF1, GYPB, GYPA, IL7R, ERAP2, TCF7, CD180, MANEA, DCAF12, HEMGN, TLR7, SLC6A8, SLC38A5, ALAS2, MBNL3, MARCHF8, CD3E, MPEG1, NPAT, MS4A14, CD4, KLRG1, YBX3, TESPA1, HVCN1, CD27, TAS2R14, KLF12, EPB42, PHLPP2, ERVK13-1, CCR7, SLC4A1, PHOSPHO1, SLC14A1, FECH, BCL2L1, PSMF1, FBXO7

As a consequence of the *multiple comparison test*, we observed that differentially expressed genes were shared among the different groups which actually meant that they were exclusively differentially expressed in the symptomatic COVID-19 cases and the uninfected controls. Now we can see which genes were differentially expressed exclusively in those groups.

In the case of symptomatic COVID-19 cases and uninfected controls, they shared 102 differentially expressed genes symptomatic and asymptomatic COVID-19 cases. Those genes were differentially expressed among the **symptomatic COVID-19 cases**, which were:

> UQCRH, S100A9, RGS1, G0S2, JUN, LAMTOR5, DENND2D, S100A12, S100A8, RIT1, SLAMF6, SH2D1B, PTGS2, FCMR, YPEL5, IL1R2, NR4A2, IL18R1, CSTA, B3GNT5, CSRNP1, CX3CR1, LTF, CISH, CXCL8, ANXA3, RPL34, PPBP, HMGB2, SUB1, IL7R, AP3S1, TCF7, RPL37, ETF1, DUSP1, CD180, CD83, MANEA, ARG1, TNFAIP3, IER3, VNN1, PRKAR2B, SBDS, RABGEF1, PNPLA8, TRIB1, ASPH, RPL7, UQCRB, KLF10, DNAJA1, ANXA1, LCN2, NFIL3, TXN, TLR7, SAT1, COX7B, RPL36A, DDX3Y, EIF1AY, DDIT4, RPS24, IDI1, RPL21, CD3E, MPEG1, TCN1, MMP8, CLEC4D, KLRG1, RPL41, CLEC4E, CD69, CLEC1B, GPR84, TESPA1, CD63, CD27, MYL6, POMP, KLF12, IRS2, NFKBIA, RPL36AL, NDUFB1, THBS1, SRP14, BCL2A1, HP, CCR7, ITGA2B, ITGB3, RPL26, FCAR, GMFG, MCEMP1, MMP9, SAMSN1, NDUFA6.

In the case of the asymptomatic COVID-19 cases and uninfected controls shared 2 differentially expressed genes with the symptomatic COVID-19 cases and uninfected controls. Those genes were differentially expressed among the **uninfected control cases**, which were HEMGN, EPB42.

4.3 Pathway Enrichment Analysis

Some of the pathways are commonly shared among the symptomatic COVID-19 cases and uninfected controls and the symptomatic and asymptomatic COVID-19 cases groups, which correspond to the shared genes between them. In Figure 6, the most 15 frequent pathways among the differentially expressed genes are showed for the previous contrasts.

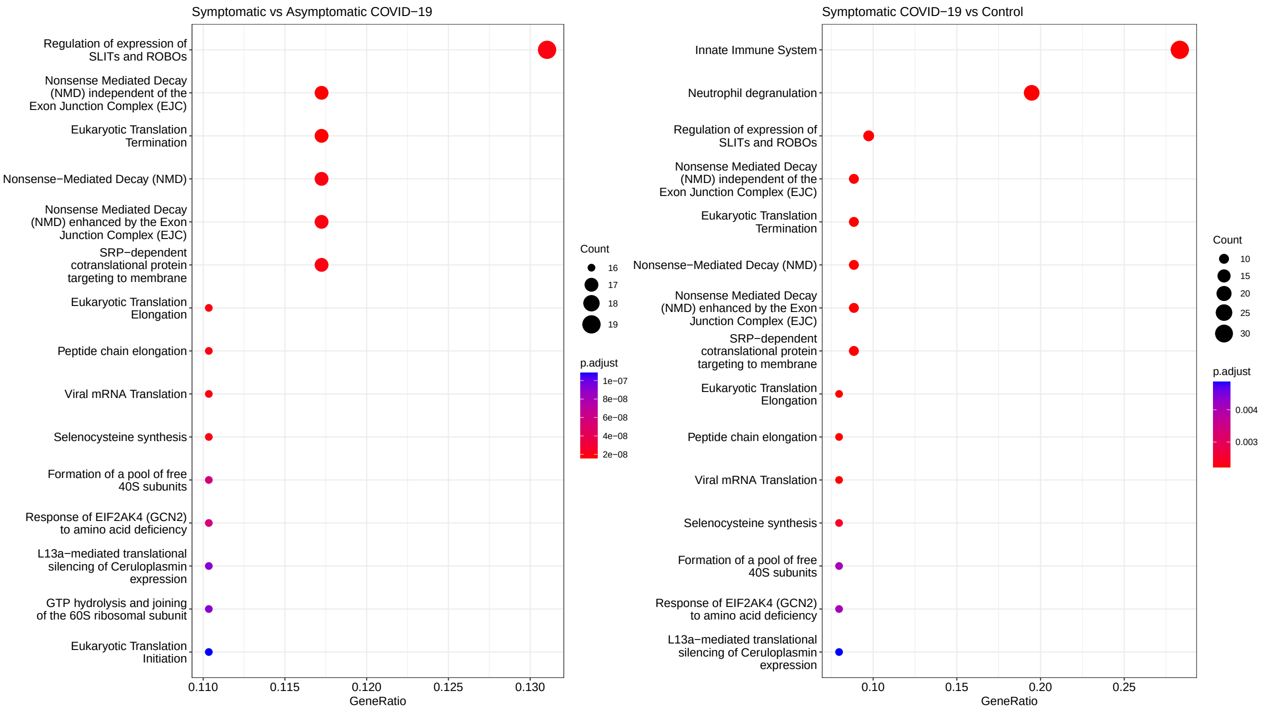


Figure 6: Dotplot for the symptomatic and the asymptomatic COVID-19 cases (Symptomatic vs Asymptomatic COVID-19), and the symptomatic COVID-19 and the uninfected control cases (Symptomatic COVID-19 vs Control)

The regulation of expression of SLITs and ROBOs was the most frequent pathway in the symptomatic and the asymptomatic COVID-19 cases, which might be related to the COVID-19. Indeed, some pathways such as the viral mRNA translation, the eukaryotic translation termination and initiation, the formation of a pool of free 40S subunits, the GTP hydrolysis and joining of the 60S ribosomal subunit, the activation of the L13a-mediated translational silencing of Ceruloplasmin expression, the response of EIF2AK4 (GCN2) to amino acid deficiency and the peptide chain elongation which could be involved with the viral cycle of the *SARS-Cov* infectious agent of the COVID-19 infection (Kash et al. 2006; Mazumder et al. 2003; Hao et al. 2005). Indeed, this interaction can be seen in the left Figure 7. In addition, the selenocysteine synthesis seems to be involved in the coagulatory complications of the symptomatic COVID-19 cases (Vavougios, Ntoskas, and Doskas 2021).

The innate immune system was the most frequent pathway in the symptomatic COVID-19 and the uninfected control cases followed by neutrophil degranulation, which makes sense because the patients with *SARS-CoV* have been infected by a pathogen, and can be seen in the right Figure 7. However, this pathway doesn't seem to appear in symptomatic COVID-19 cases which could be due to some reasons such as this patients are vaccinated, or have suffered the infectious previously, or that they don't detect the virus.

The symptomatic COVID-19 and the uninfected control cases were involved in the process of antimicrobial peptides, keratinization and, formation of the cornified envelope were downregulated, and the interferon α/β signaling is suppressed in the symptomatic COVID-19 cases, however, is the upregulated pathway in the asymptomatic cases which could be studied in the future (Bouayad 2020). This information can be seen in Figure 8.

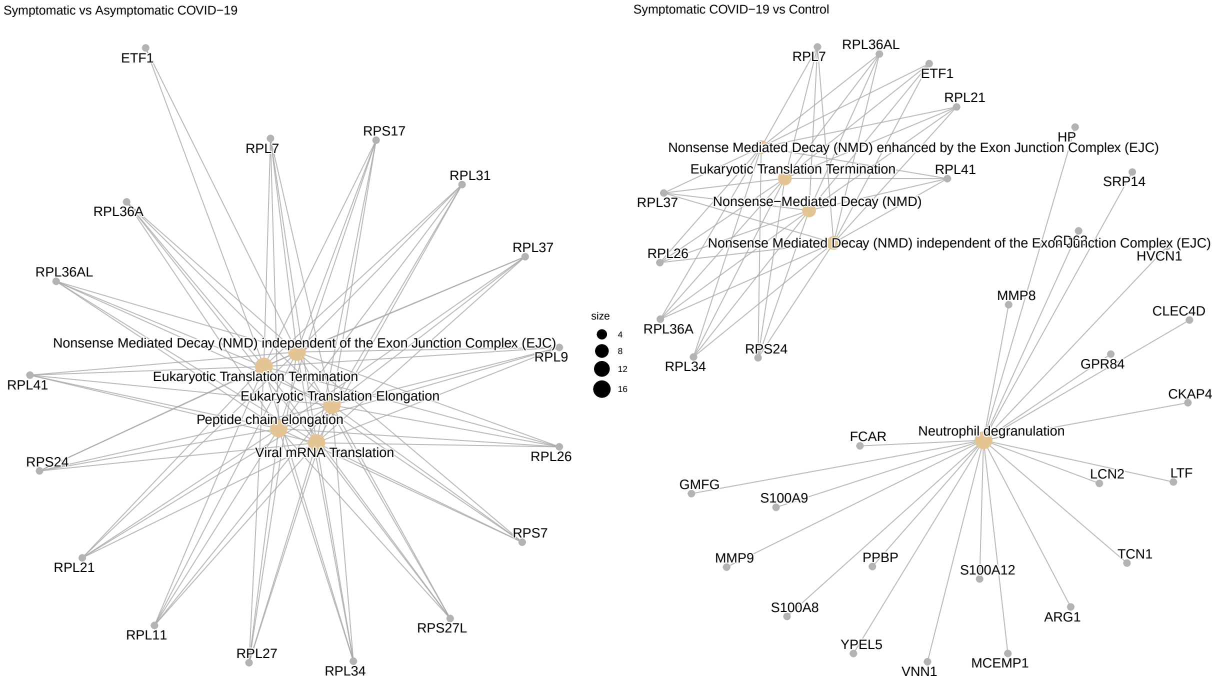

Symptomatic vs Asymptomatic COVID−19
ETF1
RPL7
RPS17
RPL31
RPL36A
RPL37
RPL36AL
Nonsense Mediated Decay (NMD) independent of the Exon Junction Complex (EJC)
RPL9
RPL41
Eukaryotic Translation Termination
Eukaryotic Translation Elongation
Peptide chain elongation
Viral mRNA Translation
RPS24
RPL26
RPS7
RPL21
RPL11
RPS27L
RPL27
RPL34
size
4
8
12
16
Symptomatic COVID−19 vs Control
RPL7
RPL36AL
ETF1
RPL21
HP
Nonsense Mediated Decay (NMD) enhanced by the Exon Junction Complex (EJC)
Eukaryotic Translation Termination
RPL41
SRP14
RPL37
Nonsense−Mediated Decay (NMD)
CD63
Nonsense Mediated Decay (NMD) independent of the Exon Junction Complex (EJC)
RPL26
HVCN1
MMP8
CLEC4D
RPL36A
RPS24
GPR84
RPL34
CKAP4
FCAR
Neutrophil degranulation
GMFG
S100A9
LCN2
LTF
MMP9
PPBP
S100A12
TCN1
S100A8
ARG1
YPEL5
VNN1
MCEMP1
size
5
10
15
20

Figure 7: Gene-Concept Network for the symptomatic and the asymptomatic COVID-19 cases (Symptomatic vs Asymptomatic COVID-19), and the symptomatic COVID-19 and the uninfected control cases (Symptomatic COVID-19 vs Control)

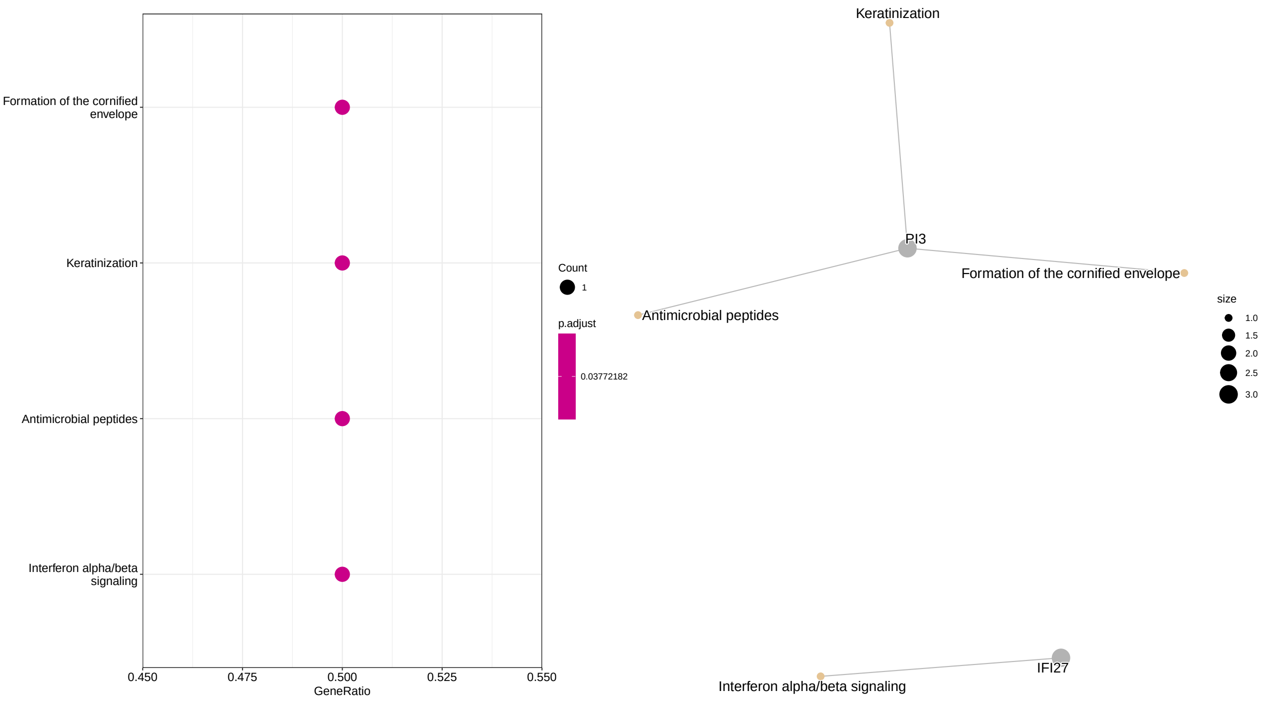


Figure 8: Dotplot and Gene-Concept Network for the asymptomatic COVID-19 and the uninfected control cases (Asymptomatic COVID-19 vs Control).

5 Conclusion

The main objective of the project was to determine the differentially expressed genes among the three groups, symptomatic and asymptomatic COVID-19 cases, and uninfected controls. According to this the conclusions extracted from the project are the following ones:

- The 1.54% of the genes from the raw data were selected to be analyzed.
-
-
-
-
-

asymptomatic COVID-19 cases and uninfected controls ($AvsC$)

symptomatic and asymptomatic COVID-19 cases ($SvsA$)

symptomatic COVID-19 cases and uninfected controls ($SvsC$)

6 References

Benjamini, Yoav, and Yosef Hochberg. 1995. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." *Journal of the Royal Statistical Society: Series B (Methodological)* 57 (1): 289–300.

Bolstad, Benjamin M, Rafael A Irizarry, Magnus Åstrand, and Terence P. Speed. 2003. "A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Variance and Bias." *Bioinformatics* 19 (2): 185–93.

Bouayad, Abdellatif. 2020. "Innate Immune Evasion by SARS-CoV-2: Comparison with SARS-CoV." *Reviews in Medical Virology* 30 (6): 1–9.

Gentleman, R. 2004. "Using GO for Statistical Analysis." *Bioconductor's Compendiums (Www.bioconductor.org)*.

Gentleman, Robert, Vince Carey, Wolfgang Huber, Rafael Irizarry, and Sandrine Dudoit. 2005. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. New York: Springer.

Hackstadt, Amber J, and Ann M Hess. 2009. "Filtering for Increased Power for Microarray Data Analysis." *BMC Bioinformatics* 10 (1): 1–12.

Hao, Shuzhen, James W Sharp, Catherine M Ross-Inta, Brent J McDaniel, Tracy G Anthony, Ronald C Wek, Douglas R Cavener, et al. 2005. "Uncharged tRNA and Sensing of Amino Acid Deficiency in Mammalian Piriform Cortex." *Science* 307 (5716): 1776–78.

Kash, John C, Alan G Goodman, Marcus J Korth, and Michael G Katze. 2006. "Hijacking of the Host-Cell Response and Translational Control During Influenza Virus Infection." *Virus Research* 119 (1): 111–20.

Masood, Kiran Iqbal, Maliha Yameen, Javeria Ashraf, Saba Shahid, Syed Faisal Mahmood, Asghar Nasir, Nosheen Nasir, et al. 2021. "Upregulated Type i Interferon Responses in Asymptomatic COVID-19 Infection Are Associated with Improved Clinical Outcome." *Scientific Reports* 11 (1): 1–14.

Mazumder, Barsanjit, Prabha Sampath, Vasudevan Seshadri, Ratan K Maitra, Paul E DiCorleto, and Paul L Fox. 2003. "Regulated Release of L13a from the 60s Ribosomal Subunit as a Mechanism of Transcript-Specific Translational Control." *Cell* 115 (2): 187–98.

Smyth, Gordon K. 2004. "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments." *Statistical Applications in Genetics and Molecular Biology* 3 (1).

Vavougios, George D, Konstantinos T Ntoskas, and Triantafyllos K Doskas. 2021. "Impairment in Selenocysteine Synthesis as a Candidate Mechanism of Inducible Coagulopathy in COVID-19 Patients." *Medical Hypotheses* 147: 110475.

Yu, Guangchuang. 2019. "Enrichplot: Visualization of Functional Enrichment Result." *R Package Version* 1 (1).

Yu, Guangchuang, and Qing-Yu He. 2016. "ReactomePA: An r/Bioconductor Package for Reactome Pathway Analysis and Visualization." *Molecular BioSystems* 12 (2): 477–79.

7 Appendix

```
##################### LOADING LIBRARIES #####################

if (!require(BiocManager)) install.packages("BiocManager")

library(Biobase)
library(oligo)
library(arrayQualityMetrics)
library(clariomshumantranscriptcluster.db)
library(genefilter)
library(limma)
library(dplyr)
library(xtable)
library(gplots)
library(ggplot2)
library(ggfortify)
library(ggpubr)
library(cowplot)
library(pvca)
library(GOstats)
library(AnnotationDbi)
library(ReactomePA)
library(enrichplot)

##################### DIRECTORIES #####################

# Work directory
dir <- getwd()

# Input directory
datadir <- file.path(dir, "Input")

# Result directory
resultsdir <- file.path(dir, "Results")

##################### SELECTING AND LOADING DATA #####################

# Subsetting data
set.seed(8795)

# List CEL selected
celFiles <- list.celfiles(datadir)[c(sample(1:11, 5), sample(12:29, 5),
                                     sample(30:47, 5))]

# Targets file
targets <- data.frame(fileName = celFiles,
                      Groups = c(rep("Symptomatic", 5), rep("Asymptomatic", 5),
                                 rep("Control", 5)),
                      ShortName = c(paste(rep("S", 5), seq(1:5), sep = ""),
                                    paste(rep("A", 5), seq(1:5), sep = ""),
                                    paste(rep("N", 5), seq(1:5), sep = "")))

# Save targets
```

```r
write.csv(targets, file.path(datadir, "targets.csv"), row.names = FALSE)

## ExpressionSet

# PhenoData
targets <- read.AnnotatedDataFrame(file.path(datadir, "targets.csv"),
                                   header = TRUE, row.names = 1, sep = ",")

# Loading CEL files
rawdata <- read.celfiles(file.path(datadir, celFiles), phenoData = targets)

# Changing row names
rownames(pData(rawdata)) <- targets@data$X.ShortName.
colnames(rawdata) <- rownames(pData(rawdata))
##################### DATA EXPLORATION #####################

# Selecting colors for each sample group
color <- c(rep("#87216BFF", 5), rep("#D64B40FF", 5), rep("#FCA007FF", 5))

# Selecting Groups
group <- pData(rawdata)$X.Groups.

# Sample number
nsamples <- nrow(pData(rawdata))

# Sample names
lsamples <- paste(rownames(pData(rawdata)))

# Density diagram for the signal distribution
pdf(file.path(resultsdir,"Figure1.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

hist(rawdata, main = "Signal distribution of the raw data",
     col = color, lty = 1:nsamples)
legend (x = "topright",
        legend =  lsamples, col = color, lty = 1:nsamples)

dev.off()

# Boxplot of raw intesity values
pdf(file.path(resultsdir,"Figure2.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

boxplot(rawdata, las = 2,  which = "all",
        col = color,
        main = "Distribution of raw intensity values")

dev.off()

# Principal Component Analysis Plot
pdf(file.path(resultsdir,"Figure3.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

autoplot(prcomp(t(exprs(rawdata))),
         colour = color,
         main = "PCA of the raw data",
         label = TRUE, shape = FALSE)+
  theme_classic()

dev.off()

# Heatmap based on euclidean distances
manDist <-  dist(t(exprs(rawdata)))

pdf(file.path(resultsdir,"Figure4.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

heatmap(as.matrix(manDist), col = heat.colors(16),
        main = "Heatmap of the raw data")

dev.off()

# Hierarchical cluster
clust.euclid.average <- hclust(dist(t(exprs(rawdata))), method = "ward.D2")

# Dendrogram of the cluster
pdf(file.path(resultsdir,"Figure5.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

plot(clust.euclid.average, hang = -1,
     main = "Cluster dendrogram of the raw data")

dev.off()
##################### QUALITY CONTROL #####################

arrayQualityMetrics(rawdata, outdir = file.path(resultsdir, "arrayQualityMetrics_rawdata"),
                    force = TRUE)

##################### NORMALIZED DATA #####################

# RMA Method
esetdata <- rma(rawdata)

# Save normalized data
write.exprs(esetdata, file.path(resultsdir, "Normdata.txt"))

# Quality Metrics
arrayQualityMetrics(esetdata, outdir = file.path(resultsdir, "arrayQualityMetrics_normdata"),
                    force = TRUE)
```

```
# Density diagram for the signal distribution
pdf(file.path(resultsdir,"FigureN.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

hist(esetdata, main = "Signal distribution of the normalized data",
     col = color, lty = 1:nsamples)
legend (x = "topright",
        legend =  lsamples, col = color, lty = 1:nsamples)

dev.off()

# Boxplot of normalized intensity values
pdf(file.path(resultsdir,"FigureN2.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

boxplot(esetdata, cex.axis = 0.5, las = 2,  which = "all",
        col = color,
        main = "Distribution of normalized intensity values")

dev.off()

# Principal Component Analysis Plot
pdf(file.path(resultsdir,"FigureN3.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

autoplot(prcomp(t(exprs(esetdata))),
         main = "PCA of the normalized data",
         colour = color,
         label = TRUE, shape = FALSE, label.size = 5)+
  theme_classic()

dev.off()

# Heatmap based on euclidean distances
manDist <-  dist(t(exprs(esetdata)))

pdf(file.path(resultsdir,"FigureN4.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

heatmap(as.matrix(manDist), col = heat.colors(16),
        main = "Heatmap of the normalized data")

dev.off()

# Hierarchical cluster
clust.euclid.average <- hclust(dist(t(exprs(esetdata))), method = "ward.D2")

# Dendrogram of the cluster
pdf(file.path(resultsdir,"FigureN5.pdf"),
    width = 8, height = 7,
    bg = "white", family = "Courier")

plot(clust.euclid.average, hang = -1,
     main = "Cluster dendrogram of the normalized data")

dev.off()
# Annotated package
annotation(esetdata) <- "clariomshumantranscriptcluster.db"

# Filtered and normalized data
filterdata <- nsFilter(esetdata,
                       var.func=IQR, # Remove outlier using Interquartile range
                       var.cutoff=0.75, # 25% least variable
                       var.filter=TRUE,
                       filterByQuantile=TRUE,
                       require.entrez = TRUE # Must have EntrezID
                       )

eset_filtered <- filterdata$eset

##################### DIFFERENTIALLY EXPRESSED GENES #####################

#### Linear model regression

# Design matrix
desingmatrix <- model.matrix(~0+X.Groups., pData(eset_filtered))
colnames(desingmatrix) <- c("A","C","S")

## Contrast matrix
# Hypothesis testing
contrastmatrix <- makeContrasts(SvsA = S - A,
                                SvsC = S - C,
                                AvsC = A - C,
                                levels = desingmatrix)

# Model
lm <- lmFit(eset_filtered, desingmatrix)

# Contrast
lmresult <- contrasts.fit(lm, contrastmatrix)

# Bayes empirical method to upgrade the T-statistics from the contrast
lmresult <- eBayes(lmresult)

# Differentially expressed genes between symptomatic and asymptomatic COVID-19 cases
# SvsA
toptable_SvsA <- topTable(lmresult,
                             number = nrow(lmresult), # Genes total number
                             coef = "SvsA",
                             p.value = 0.05, # p-value < 0.05
                             adjust = "fdr", # False Discovery Rate
                             lfc = 2) # Log fold change > 2
```

```
# Differentially expressed genes between asymptomatic COVID-19 cases and uninfected controls
# SvsC
toptable_SvsC <- topTable(lmresult,
                             number = nrow(lmresult), # Genes total number
                             coef = "SvsC",
                             p.value = 0.05, # p-value < 0.05
                             adjust = "fdr", # False Discovery Rate
                             lfc = 2) # Log fold change > 2

# Differentially expressed genes between asymptomatic COVID-19 cases and uninfected controls
# AvsC
toptable_AvsC <- topTable(lmresult,
                             number = nrow(lmresult), # Genes total number
                             coef = "AvsC",
                             p.value = 0.05, # p-value < 0.05
                             adjust = "fdr", # False Discovery Rate
                             lfc = 2) # Log fold change > 2

list_toptab <- list(SvsA = toptable_SvsA, SvsC = toptable_SvsC, AvsC = toptable_AvsC)

# Check the dimensions to see how many genes were retained for each contrast after the
# filtering
dim_toptab <- sapply(list_toptab, dim)
rownames(dim_toptab) <- c("Genes", "Parameters")

### Multiple comparison test

res <- decideTests(lmresult,
                 method = "separate",    # Equivalent topTable
                 adjust.method = "fdr",   # False Discovery Rate
                 p.value = 0.05,         # p-value < 0.05
                 lfc = 2)                # log FC > 2

sum.res.rows <- apply(abs(res),1,sum)
res.selected <- res[sum.res.rows != 0,]

# Summary of the selected genes
print(summary(res))

# Venn Diagram
pdf(file.path(resultsdir, "Venndiag.pdf"))
vennDiagram (res.selected[,1:3],
             cex = 0.9,
             circle.col = c(alpha("#87216BFF", 0.6), alpha("#D64B40FF", 0.6), alpha("#FCA007FF", 0.6)))
title("Genes in common between the three comparisons\n Genes selected with FDR < 0.05 and logFC > 2")
dev.off()

##################### GENE ANNOTATION #####################

# Annotation function
annotatedTopTable <- function(topTab, anotPackage){
  topTab <- cbind(PROBEID = rownames(topTab), topTab)
  myProbes <- rownames(topTab)
  thePackage <- eval(parse(text = anotPackage))
  geneAnots <- AnnotationDbi::select(thePackage, myProbes, c("SYMBOL", "ENTREZID"))
  annotatedTopTab <- merge(x = geneAnots, y = topTab, by.x="PROBEID", by.y="PROBEID")
  return(annotatedTopTab)
  }

### Interpretation of the list

# Annotation of SvsA
SvsA_anot <- annotatedTopTable(toptable_SvsA,
                                      "clariomshumantranscriptcluster.db")
write.csv(SvsA_anot, file = file.path(resultsdir, "toptableSvsA.csv"))

# Annotation of SvsC
SvsC_anot <- annotatedTopTable(toptable_SvsC,
                                      "clariomshumantranscriptcluster.db")
write.csv(SvsC_anot, file = file.path(resultsdir, "toptableSvsC.csv"))

# Annotation of AsvC
AvsC_anot <- annotatedTopTable(toptable_AvsC,
                                      "clariomshumantranscriptcluster.db")

write.csv(AvsC_anot, file = file.path(resultsdir, "toptableAvsC.csv"))

# List of annotated Toptables
listOfToptables <- list(SvsA_anot,SvsC_anot,AvsC_anot)

# Annotation of the model
lmresult_anot <- AnnotationDbi::select(clariomshumantranscriptcluster.db,
                        rownames(lmresult), c("SYMBOL", "ENTREZID"))

### Pathway Analysis

# Annotation of all the genes
entrezD <- AnnotationDbi::select(clariomshumantranscriptcluster.db,
                                 rownames(exprs(eset_filtered)), "ENTREZID")$ENTREZID
dataGenes <- entrezD[!duplicated(entrezD)]

# Empty selected genes
listOfSelected <- list()

# Annotation and selected gene lists
for (i in 1:length(listOfToptables)){
  # Select toptable
  topTab <- listOfToptables[[i]]
  listOfSelected[[i]] <- topTab$ENTREZID
  names(listOfSelected)[i] <- names(listOfToptables)[i]
}

##################### GENE LIST ANALYSIS #####################

# Volcano plot
```

```
for(i in 1:ncol(contrastmatrix)){
 pdf(paste0("./Results/","Volcano_",colnames(contrastmatrix)[i],".pdf"))
  volcanoplot(lmresult, coef = i, highlight = 30,
             names = lmresult_anot[,2],
             main = paste("Differentially expressed genes",
                          colnames(contrastmatrix)[i], sep = "\n"))
 abline(v = c(-2,2))
 dev.off()
}

selRSvsA <- rownames(exprs(eset_filtered)) %in% rownames(toptable_SvsA)
selDSvsA <- exprs(eset_filtered)[selRSvsA,]

selRSvsC <- rownames(exprs(eset_filtered)) %in% rownames(toptable_SvsC)
selDSvsC <- exprs(eset_filtered)[selRSvsC,]

selRAvsC <- rownames(exprs(eset_filtered)) %in% rownames(toptable_AvsC)
selDAvsC <- exprs(eset_filtered)[selRAvsC,]

# Heatmap plot
heatmap.2(selDSvsA,
          scale = "row", # Three options : row, none, column
          sepcolor = "white",
          tracecol = NULL, # Modify, I like this one more
          srtCol = 30, # Names 30º position
          main = paste("Heatmap", colnames(contrastmatrix)[1], sep = "\n"),
          cexRow = 0.5,
          cexCol = 0.9,
          Rowv = TRUE,  # Default
          Colv = TRUE,  # Default
          sepwidth = c(0.05,0.05),  # Default
          key = TRUE,   # Default
          keysize = 1.5, # Default
          density.info = "histogram"  # Default
          )
# Heatmap plot
heatmap.2(selDSvsC,
          scale = "row", # Three options : row, none, column
          sepcolor = "white",
          tracecol = NULL, # Modify, I like this one more
          srtCol = 30, # Names 30º position
          main = paste("Heatmap", colnames(contrastmatrix)[2], sep = "\n"),
          cexRow = 0.5,
          cexCol = 0.9,
          Rowv = TRUE,  # Default
          Colv = TRUE,  # Default
          sepwidth = c(0.05,0.05),  # Default
          key = TRUE,   # Default
          keysize = 1.5, # Default
          density.info = "histogram"  # Default
          )
# Heatmap plot
heatmap.2(selDAvsC,
          scale = "row", # Three options : row, none, column
          sepcolor = "white",
          tracecol = NULL, # Modify, I like this one more
          srtCol = 30, # Names 30º position
          main = paste("Heatmap", colnames(contrastmatrix)[3], sep = "\n"),
          cexRow = 0.5,
          cexCol = 0.9,
          Rowv = TRUE,  # Default
          Colv = TRUE,  # Default
          sepwidth = c(0.05,0.05),  # Default
          key = TRUE,   # Default
          keysize = 1.5, # Default
          density.info = "histogram"  # Default
          )

for (i in 1:length(colnames(res))){
  # Contrast
  comp <- colnames(res)[i]
  # Annotated Toptable
  anot_vs <- listOfToptables[[i]]
  # Upregulated genes
  up <- rownames(res[which(res[,i]==1),i])
  genes_up <- anot_vs$SYMBOL[which(anot_vs$PROBEID%in%up)]
  # Downregulated genes
  down <- rownames(res[which(res[,i]==-1),i])
  genes_down <- anot_vs$SYMBOL[which(anot_vs$PROBEID%in%down)]

  # Saved files
  write.csv(genes_up,
            file = paste0("./Results/","Upregulated_genes_",comp,".csv"),
             row.names = FALSE)
  write.csv(genes_down,
            file = paste0("./Results/","Downregulated_genes_",comp,".csv"),
             row.names = FALSE)
}

##################### GENE SET ENRICHMENT ANALYSIS OR GSEA #####################

# Rename variable
listOfData <- listOfSelected

# Contrast names
comparisonsNames <- names(listOfData)

# Universe
universe <- dataGenes
write.csv(universe, file = file.path(resultsdir, "Genes_universe.csv"))


for (i in 1:length(listOfData)){
  # Contrast
  comparison <- comparisonsNames[i]
  # Genes in the contrast
  genesIn <- listOfData[[i]]
  # Pathway Enrichment Analysis
  enrich.result <- enrichPathway(gene = genesIn,
                                 pvalueCutoff = 0.05,  # p-value < 0.05
                                 readable = T,
                                 pAdjustMethod = "BH", # False Discovery Rate
                                 organism = "human",
                                 universe = universe)  # Genes in the sample

  if (dim(enrich.result)[1] != 0) {
  write.csv(as.data.frame(enrich.result),
             file = paste0("./Results/","ReactomePA.Results.",comparison,".csv"),
             row.names = FALSE)

  pdf(file=paste0("./Results/","ReactomePABarplot.",comparison,".pdf"))
    print(barplot(enrich.result, showCategory = 15, font.size = 4,
                  title = paste0("Reactome Pathway Analysis for ", comparison,". Barplot")))
  dev.off()

  pdf(file = paste0("./Results/","ReactomePAcnetplot.",comparison,".pdf"))
    print(cnetplot(enrich.result, categorySize = "geneNum", schowCategory = 15,
                   vertex.label.cex = 0.75))
  dev.off()

  pdf(file = paste0("./Results/","ReactomePADotplot.",comparison,".pdf"))
    print(dotplot(enrich.result, showCategory = 15),
          title = paste0("Reactome Pathway Analysis for ", comparison,". Dotplot"))
  dev.off()
  }
}
```