Differential gene expression analysis from RNA seq data

Arnau Mariscal Puig

2025-09-19

Contents

1. Introduction

This project involves a differential gene expression analysis of blood samples from COVID-19 patients, bacterial pneumonia patients, and healthy individuals.

The dataset comes from McClain et al., who analyzed blood samples from patients with COVID-19 and compared them with other respiratory infections. To simplify the analysis, this project focuses only on three groups: COVID-19, bacterial pneumonia, and healthy controls (1).

1.1. Objectives

The primary objective of this project is to carry out a differential gene expression analysis using RNA-seq data. More specifically, we aim to analyze transcriptomic data from peripheral blood samples to compare gene expression between COVID-19 patients, bacterial pneumonia patients, and healthy individuals.

The secondary objectives are:

- To build a **SummarizedExperiment** object from GEO data, including count data, metadata, and gene coordinates.
- Preprocess, transform, and normalize the data to run an exploratory analysis and detect outlier samples and confounding variables.
- Identify differentially expressed genes between groups: healthy, COVID-19, and Bacterial.
- Interpret the biological significance of the results.

2. Methods

The data was downloaded from the Gene Expression Omnibus (GEO) database, under accession **GSE161731** (2). The full analysis was done using **Bioconductor**.

2.1. Data acquisition and Summarized Experiment creation

Two files are downloaded from GEO:

- GSE161731_counts.csv.gz: contains the count matrix with the gene expression data.
- GSE161731_key.csv.gz: contains the metadata of the samples.

Both files are saved into the "dades" folder and imported from there. The count matrix contained 201 samples and 60,675 genes, while the metadata included 198 samples across eight variables (Output 1): subject_id, age, gender, race, cohort, time_since_onset, hospitalized, batch.

```
## Output 1. Count matrix and metadadata: number of genes and samples
##
## • Count matrix (genes x samples):
## [1] 60675   201
##
## • Metadata (genes x samples):
## [1] 198   8
```

Since sample names had a different format in each file, they were standardized to ensure consistency. After keeping a common format, only samples present in both files were kept, resulting in 198 in total.

To facilitate data visualization in later plots, a *Shortname* is used to label each sample by combining its cohort name and a sequential number. A color is also assigned to each cohort. Shortnames are applied to both the metadata and the count matrix.

Genomic coordinates are also added using the **EnsDb.Hsapiens.v86** package. Only genes with available genomic coordinates are kept.

Finally, a SummarizedExperiment object is created, integrating gene counts, metadata and genomic coordinates (3). This object is the basis of the analysis, with 57,602 genes and 198 samples (Output 2).

```
## Output 2. SummarizedExperiment object: counts + metadata + genomic coordinates
##
## class: RangedSummarizedExperiment
## dim: 57602 198
## metadata(0):
## assays(1): counts
## rownames(57602): ENSG00000223972 ENSG00000227232 ... ENSG00000277475
##   ENSG00000268674
## rowData names(6): gene_id gene_name ... symbol entrezid
## colnames(198): Bacterial_1 Influenza_2 ... Influenza_197 Influenza_198
## colData names(10): subject_id age ... color nom_mostra
```

2.2. Metadata cleaning and cohort selection

Next, the *dplyr* package is used for metadata cleaning (5):

- Only COVID-19, Bacterial and Healthy samples are kept
- Duplicate samples are removed, keeping the first entry
- Variables are standardized: age is converted to numeric, and text variables are cleaned (spaces, dashes, and slashes replaced with underscores).
- There is a random selection of 75 samples, using a reproducible seed derived from the string "random".

Taking this into account, the SummarizedExperiment is updated.

2.3. Data preprocessing and transformation

2.3.1 Gene filtering

Before filtering low-expression genes, counts are normalized to CPM (counts per million) using **edgeR** to make samples comparable despite differences in sequencing depth (in other words, not all samples have the same number of reads, as can bee seen in Output 3).

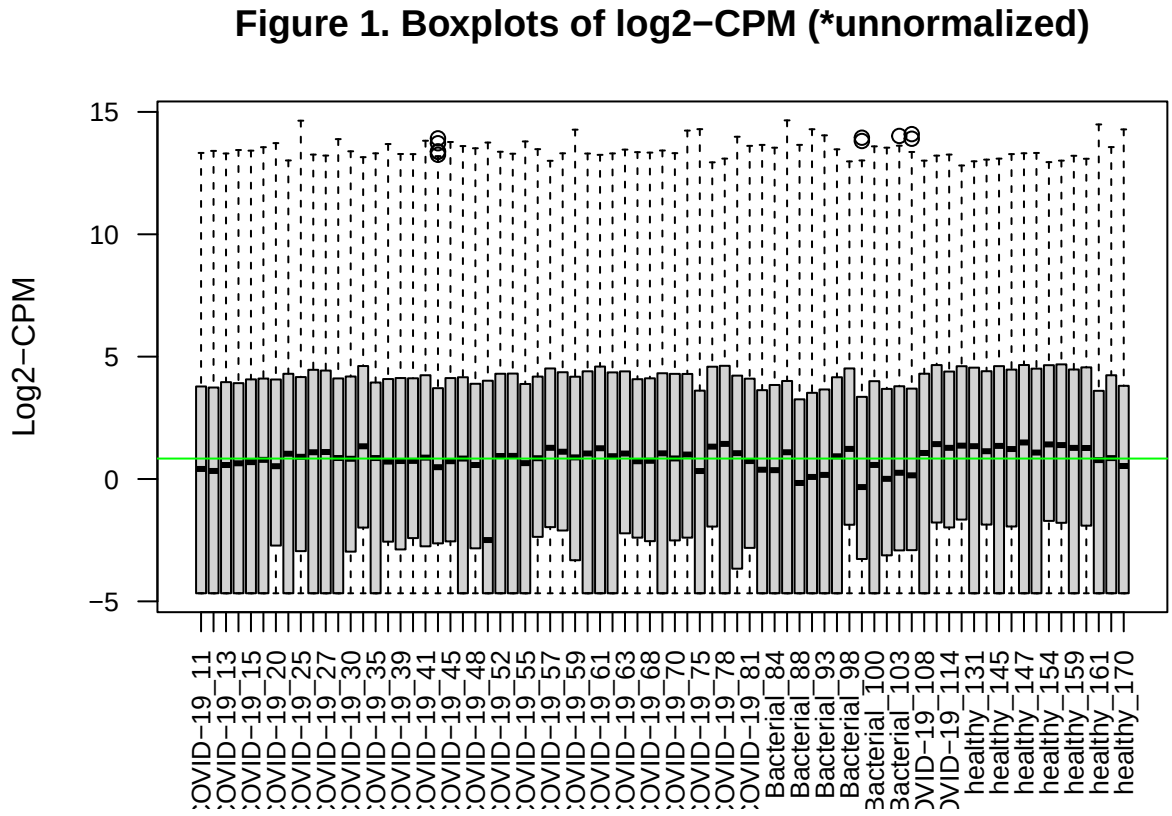
```
## Output 3. Number of counts per sample (first 5 samples)
##
## COVID-19_11 COVID-19_12 COVID-19_13 COVID-19_14 COVID-19_15
##    45734517    50138888    62895753    50638349    50862753
```

Next, genes with low expression are removed, since they provide little information to the analysis, thereby increasing the power to detect meaningful differences between other genes. Only genes with CPM > 0.5 in at least two samples are kept. After filtering, the SummarizedExperiment contains 24,934 genes.

2.3.2. Normalization

Up until now, data has been managed using the SummarizedExperiment object. However, to better handle sequencing data, the **DGElist** class from the **edgeR** package will be more useful. This structure stores count data, metadata, and gene information, also allowing to apply normalization and differential analysis functions brought by edgeR.

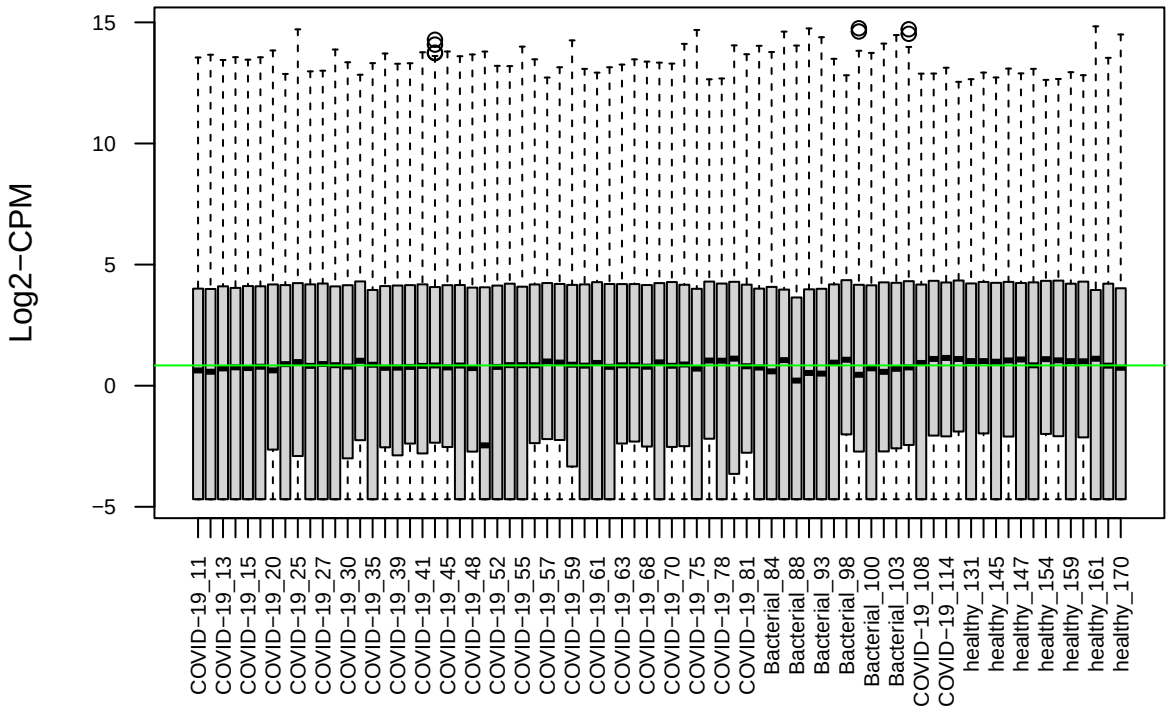
Since count data are not normally distributed, a log transformation is applied to stabilize variance and facilitate comparisons between samples. An initial exploration using boxplot indicates that most samples have similar distributions, although some outliers are observed (Figure 1).



Normalization is then performed using **TMM** (Trimmed Mean of M-values) via calcNormFactors() in edgeR. This corrects biases related to library composition, which means that some genes might be highly expressed in certain conditions, so this function ensures fair comparisons between samples.

After normalization, another boxplot shows a more similar distribution across samples (Figure 2). However, sample COVID-19_50 still seems to be an outlier, with a median noticeably lower than the general one.

Figure 2. Boxplots of log2−CPM (*normalized)



2.4. Exploratory analysis

After normalization, an exploratory analysis is performed to evaluate data quality. This step helps assess if samples from the same cohort (COVID-19, Bacterial, or Healthy) group together, or if there are any batch effects influencing the results. It also allows us to identify patterns related to confounding variables and help detect outliers.

A distance matrix is created, with pairwise comparisons between all samples using the log-transformed data. From this, the following plots are generated:

- *Heatmap of the distance matrix* (Annex, Figure A1): It is used to observe similarity patterns between samples and detect if any sample differs from the rest or fails to group with its cohort.
- *Dendrogram* (Annex, Figure A2): shows hierarchical clustering of the samples and allows visualization of groupings by cohort.
- *MDS* (Multidimensional Scaling) plot (Annex, Figure A3): represents the samples in a two-dimensional space according to their similarity.
- *PCA* plot (Annex, Figure A4): to represent variability among samples and quantify how much of it is explained by the first components (PC1 and PC2). PCA also helps explore variability related to factors such as cohort, sex, age, race, or batch.

The results of the three plots (heatmap, dendrogram, and MDS plot) show that samples from the Bacterial cohort tend to cluster, while samples from the Healthy and COVID-19 cohorts are mixed together. This pattern is most clearly visible in the MDS plot.

In addition, sample COVID-19_50 behaves atypically: in the previous boxplots it showed a median noticeably lower than the global median, while in the MDS plot it is represented far from the rest of the samples. Therefore, it is removed from the following analyses to ensure a clearer interpretation.

2.5. Identification of Confounding Variables

To identify potential confounding variables, we consider the results from PCA, contingency tables, and their associated statistical tests:

- **Race** (Annex, Table A1). The contingency table shows that most COVID-19 samples belong to the White race, while most Bacterial samples belong to the Black race. The PCA plot also reveals a separation between the two groups. Although the requirements for a chi-square test are not met, the uneven distribution suggests that **race could be considered a confounding variable**.
- **Gender** (Annex, Table A2). The chi-square test does not show significant differences between gender and cohort, and the PCA does not reveal a clear pattern either. Therefore, gender is **not considered a confounding variable**.
- **Batch** (Annex, Table A3). Most samples belong to batch 1. Even though the requirements for a chi-square test are not met, the unequal distribution suggests that batch **could be considered a confounding variable**.
- **Age** (Annex, Figure A5 and Table A4). It's the only quantitative variable. The PCA does not show a clear effect, but when age distribution is represented in a boxplot and an ANOVA test is performed, significant differences are observed. Therefore, **age is considered a confounding variable**.

2.6. Design and contrast matrices

Even though the initial plan was to adjust the model for several confounding variables, adding *batch* and *race* to the design matrix caused collinearity problems with the cohort or clinical condition (labeled as "group"). This collinearity removed the "healthy" group, making it impossible to define the contrast matrix. Therefore, the final model was adjusted only for **age** and **clinical condition (group)**, in order to identify differentially expressed genes between the healthy, COVID-19, and bacterial cohorts (Output 4).

```
## Output 4. Contrast matrix
##
##            Contrasts
## Levels      COVIDvsHealthy BacterialvsHealthy
##   age                    0                  0
##   COVID_19               1                  0
##   Bacterial              0                  1
##   healthy               -1                 -1
```

2.6.1. Voom transformation

The next step is to proceed to the differential expression analysis using **voom+limma**. The voom transformation converts count data into continuous log-CPM values with associated precision weights, which improve variance estimation and allow the use of linear models to detect differentially expressed genes.

So, after transforming the normalized DGEList object (dge_normal2) with the voom() function and the design matrix, log-CPM values with associated precision weights are obtained. The linear model in limma is then fit to the log-CPM values, with the weights applied, using the design and contrast matrices to perform group comparisons.

Finally, the eBayes() function is applied to stabilize error estimation and improve the detection of differentially expressed genes.

3. Results

3.1. Differential Gene Expression Profiles

The **topTable()** function generates a table with statistical results for each contrast. The **volcano plots** help visualize these results, while the **heatmaps** allow observation of expression profiles based on selected sets of genes.

3.1.1. COVID vs Healthy

The volcano plot appears asymmetric and concentrated, with most differentially expressed genes located on the left side (Annex, Figure A6). In the heatmap, a clear pattern is seen: genes are predominantly **overexpressed in the Healthy group**, while they tend to be underexpressed in the COVID group (Annex, Figure A7).

3.1.2. Bacterial vs Healthy

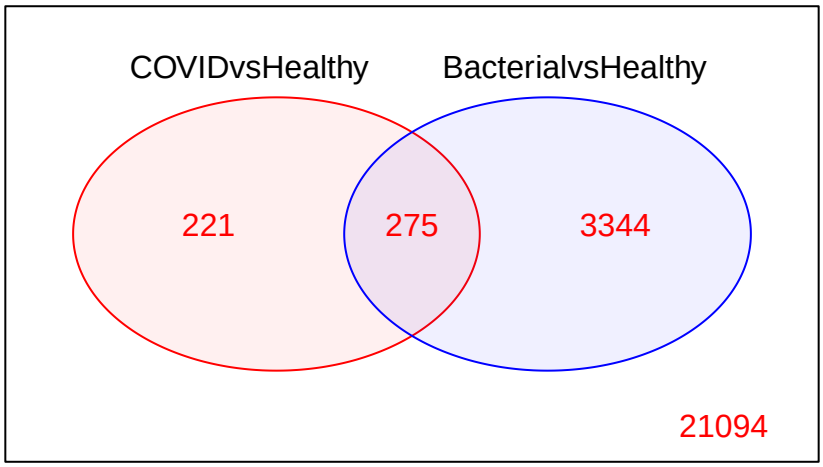
In this comparison, the volcano plot appears more symmetric and less concentrated (Annex, Figure A8). Differentially expressed genes are mainly found on the right side. In the heatmap, two distinct gene clusters can be observed for the Bacterial group (Annex, Figure A9): one with clearly underexpressed genes (upper section of the plot) and another with clearly overexpressed genes (lower section), compared to the COVID and Healthy groups.

3.1.3. Multiple Comparison

The number of differentially expressed genes in each comparison can also be assessed using the **decideTests** function and visualized with a **Venn diagram**.

A total of 496 genes were found differentially expressed in the COVID vs Healthy contrast, 3619 genes in the Bacterial vs Healthy contrast, and 275 genes were shared between the two comparisons (Figure 3).

Figure 3. Venn Diagram of differentially expressed genes



3.2. Biologic significance

Finally, an **over-representation analysis** is performed to identify biological processes (from Gene Ontology) related to overexpressed genes in COVID-19 compared to healthy samples. To do so, gene identifiers need to be in *Entrez ID* format rather than ENSEMBL IDs, so the **org.Hs.eg.db** package is used for conversion.

The analysis reveals that these genes are mainly involved in cell cycle regulation and in controlling oligodendrocyte differentiation (Annex, Figure A10). A network plot highlights key genes such as FOXM1, MELK, PKMYT1, CCNB2, and SPC24 in cell cycle processes, and HES1 and TMEM98 in oligodendrocyte differentiation (Annex, Figure A11).

4. Discussion and conclusions

In this project, a differential gene expression analysis was performed on samples from COVID-19 patients, bacterial infection patients, and healthy individuals to identify differentially expressed genes and explore their biological significance.

The analysis showed different gene activity between groups. In the COVID-19 vs healthy comparison, there was an overexpression of genes in healthy individuals, with a more limited activation in COVID-19 patients. In contrast, the Bacterial vs healthy comparison showed two clear groups of genes in bacterial samples, with some highly overexpressed and others underexpressed. Notably, the number of differentially expressed genes was seven times higher in the bacterial comparison than the covid one, indicating the immune response at a transcriptomic level is more diverse and intense. Shared genes between comparisons probably show common immune mechanisms.

Regarding the over-representation analysis related to overexpressed genes in COVID-19 vs healthy samples, it indicated these overexpressed genes in COVID-19 patients are involved in cell proliferation and oligodendrocyte differentiation, which could be related to the immune response against viral infection.

Some limitations should be considered. Confounding variables such as race and batch were not fully explored, and sample distribution across groups was uneven (e.g., most bacterial samples were from Black individuals and most COVID samples from White individuals). Therefore, future studies could incorporate these variables and ensure a more balanced sampling to improve the analysis.

5. Bibliography

1. McClain MT, Constantine FJ, Henao R, Liu Y, Tsalik EL, Burke TW, Steinbrink JM, Petzold E, Nicholson BP, Rolfe R, Kraft BD, Kelly MS, Saban DR, Yu C, Shen X, Ko EM, Sempowski GD, Denny TN, Ginsburg GS, Woods CW. Dysregulated transcriptional responses to SARS-CoV-2 in the periphery. Nat Commun. 2021 Feb 17;12(1):1079. doi: 10.1038/s41467-021-21289-y. PMID: 33597532; PMCID: PMC7889643.

2. GEO accession viewer [Internet]. GSE161731. National Center for Biotechnology Information. Available from: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE161731

3. SummarizedExperiment: Bioconductor package vignette [Internet]. Bioconductor. Available from: https://www.bioconductor.org/packages/devel/bioc/vignettes/SummarizedExperiment/inst/doc/SummarizedExperiment.html

4. EnsDb.Hsapiens.v86: Bioconductor annotation data [Internet]. Bioconductor. Available from: https://bioconductor.org/packages/release/data/annotation/html/EnsDb.Hsapiens.v86.html

5. dplyr tutorial [Internet]. RPubs. Available from: https://rpubs.com/justmarkham/dplyr-tutorial

6. AMV Casos: Ejemplo PCA 2 detección del efecto batch en datos de microarrays [Internet]. ASPteaching. Available from: https://aspteaching.github.io/AMVCasos/#ejemplo-pca-2-detecci%C3%B3n-del-efecto-batch-en-datos-de-microarrays

7. Omics Data Analysis – Case Study 1: Microarrays [Internet]. GitHub repository. ASPteaching. Available from: https://github.com/ASPteaching/Omics_Data_Analysis-Case_Study_1-Microarrays

8. Análisis de datos ómicos – Ejemplo 2 RNASeq [Internet]. GitHub repository. ASPteaching. Available from: https://github.com/ASPteaching/Analisis_de_datos_omicos-Ejemplo_2-RNASeq

9. Workflow básico de RNASeq: Preprocesado de los datos [Internet]. ASPteaching. Available from: https://aspteaching.github.io/Analisis_de_datos_omicos-Ejemplo_2-RNASeq/Workflow_basico_de_RNASeq.html#5_Preprocesado_de_los_datos

6. Annex

Figure A1. Heatmap

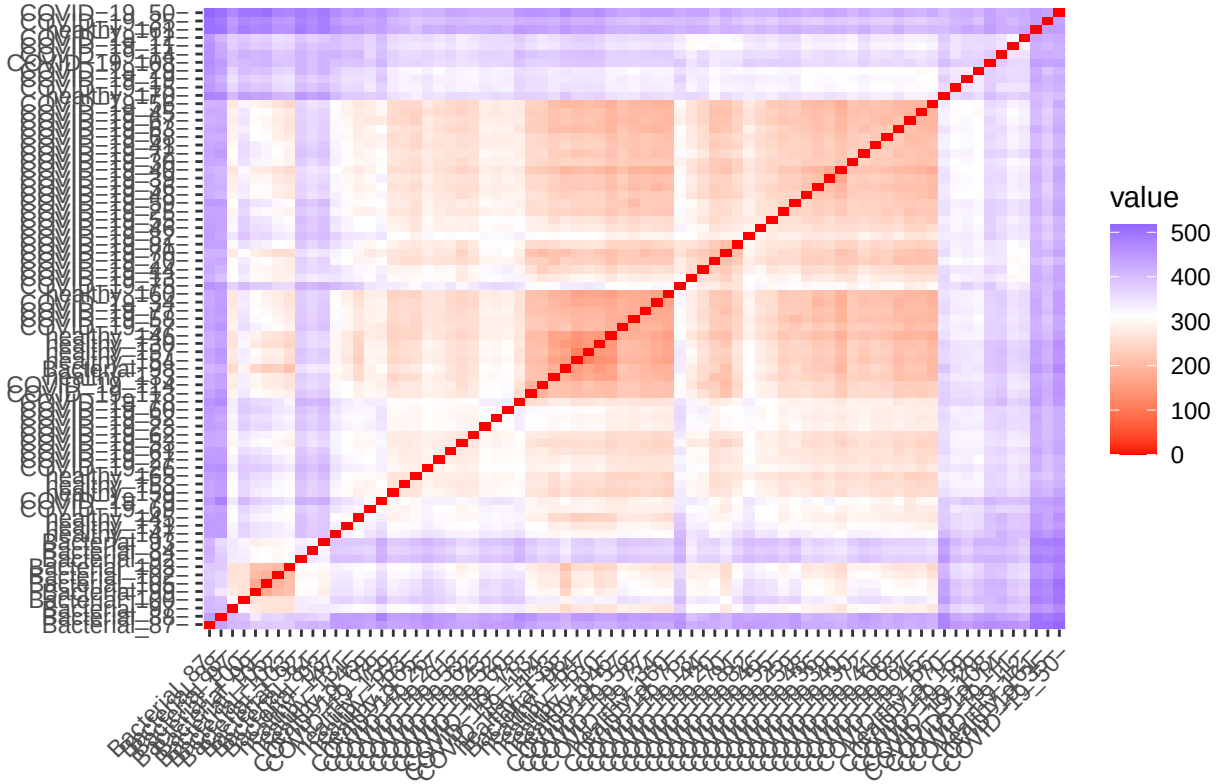


Figure A2. Hierarchical clustering of the samples

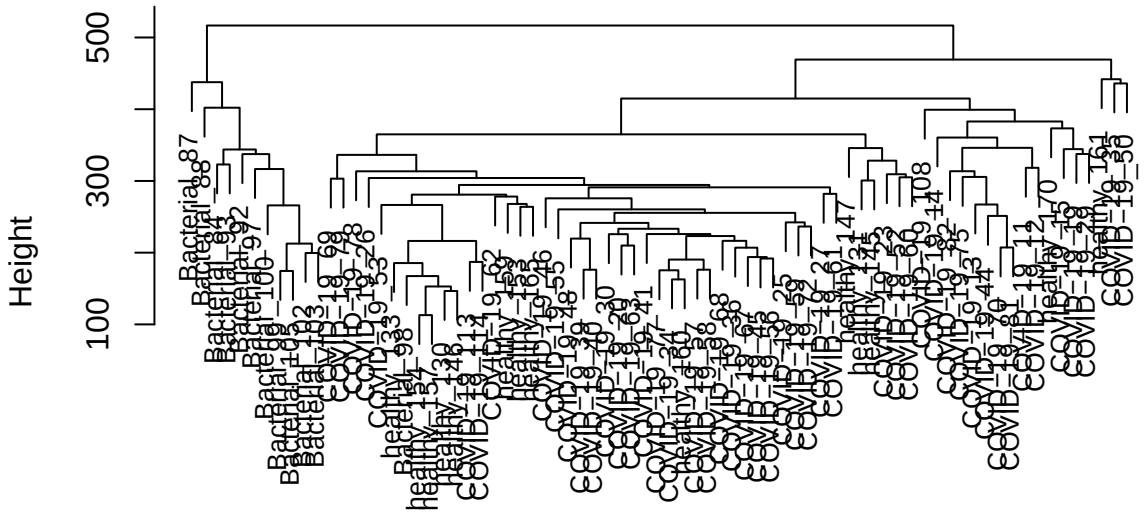


Figure A3. MDS plot

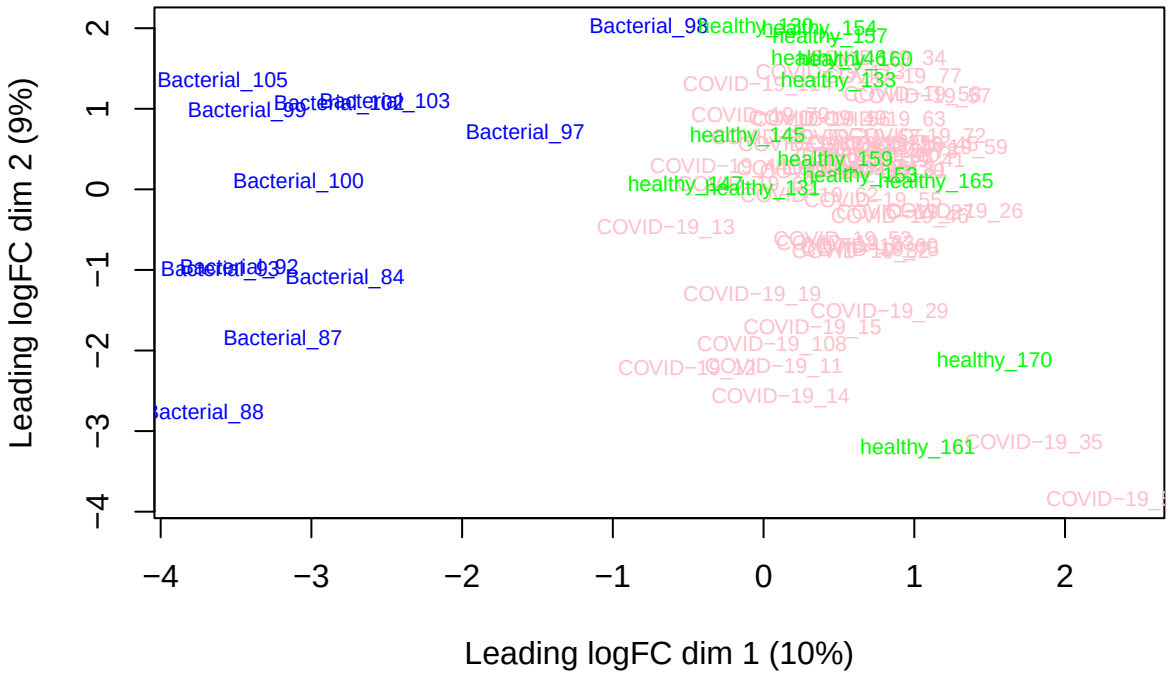


Figure A4 (A). PCA x cohort

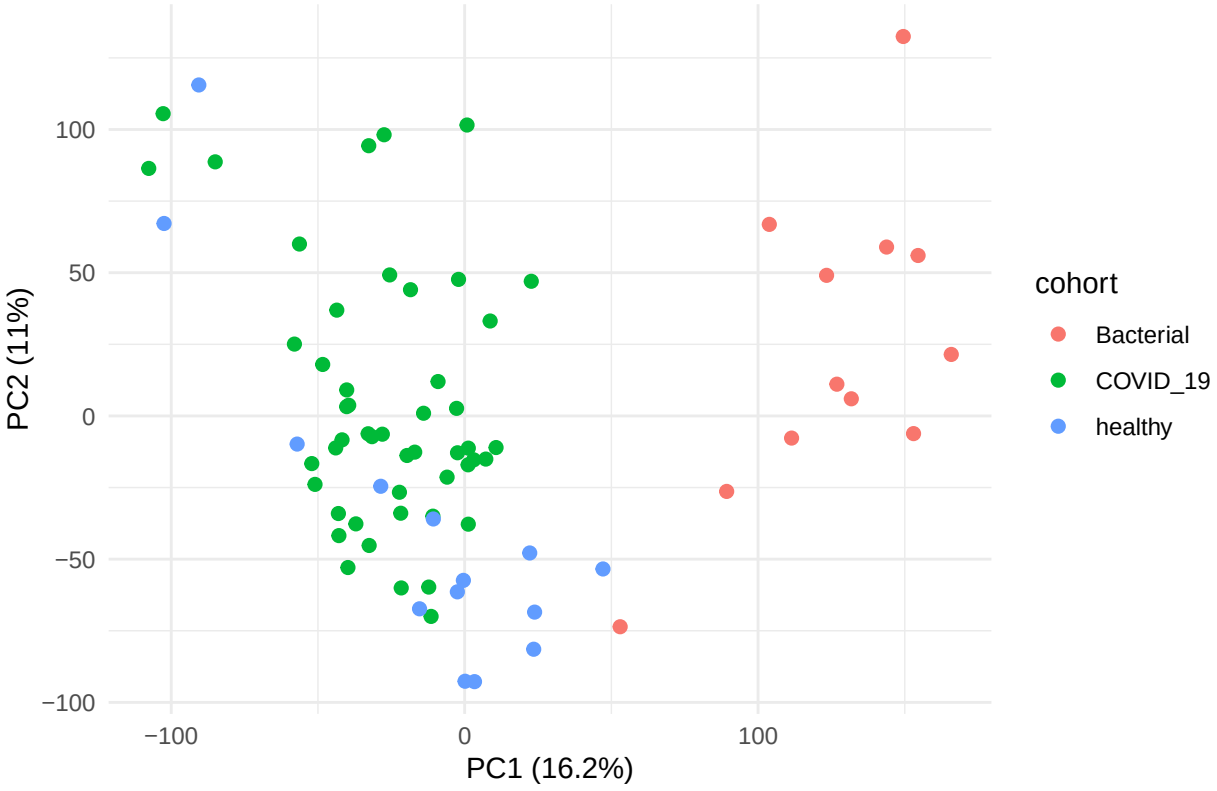


Figure A4 (B). PCA x batch

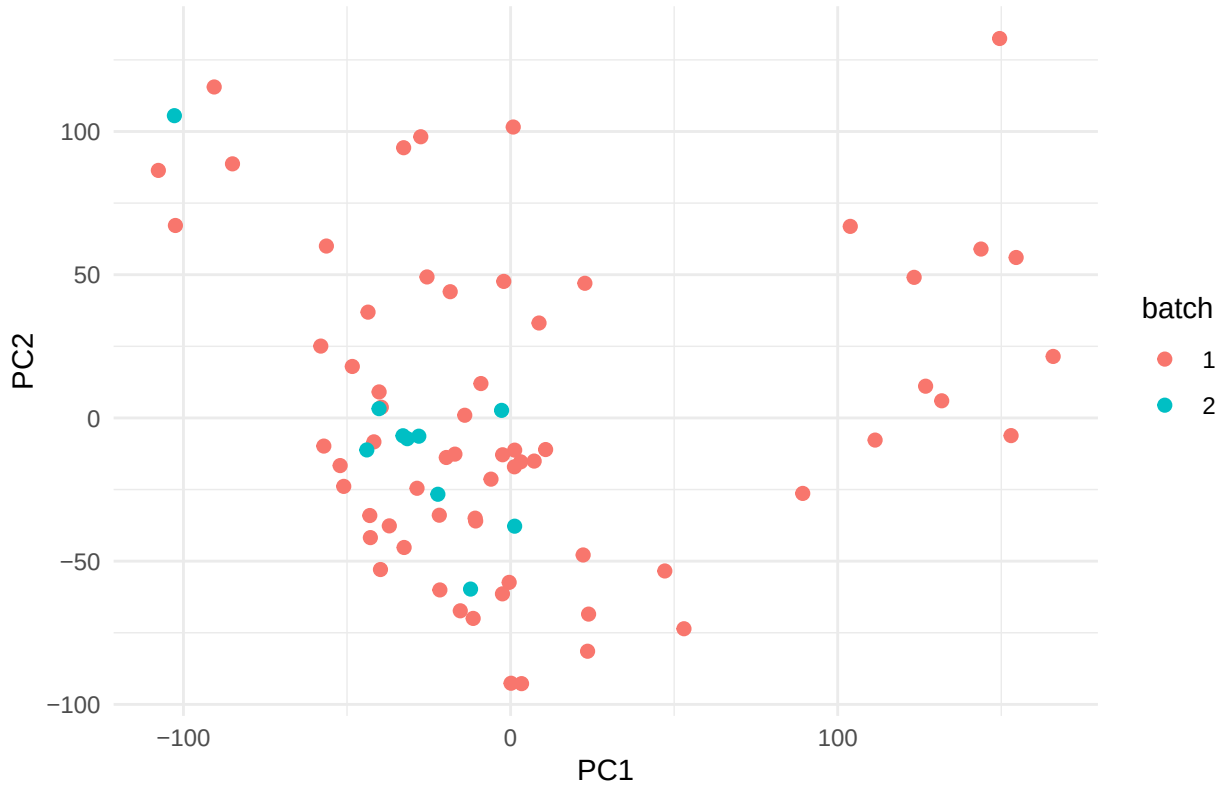


Figure A4 (C). PCA x sex

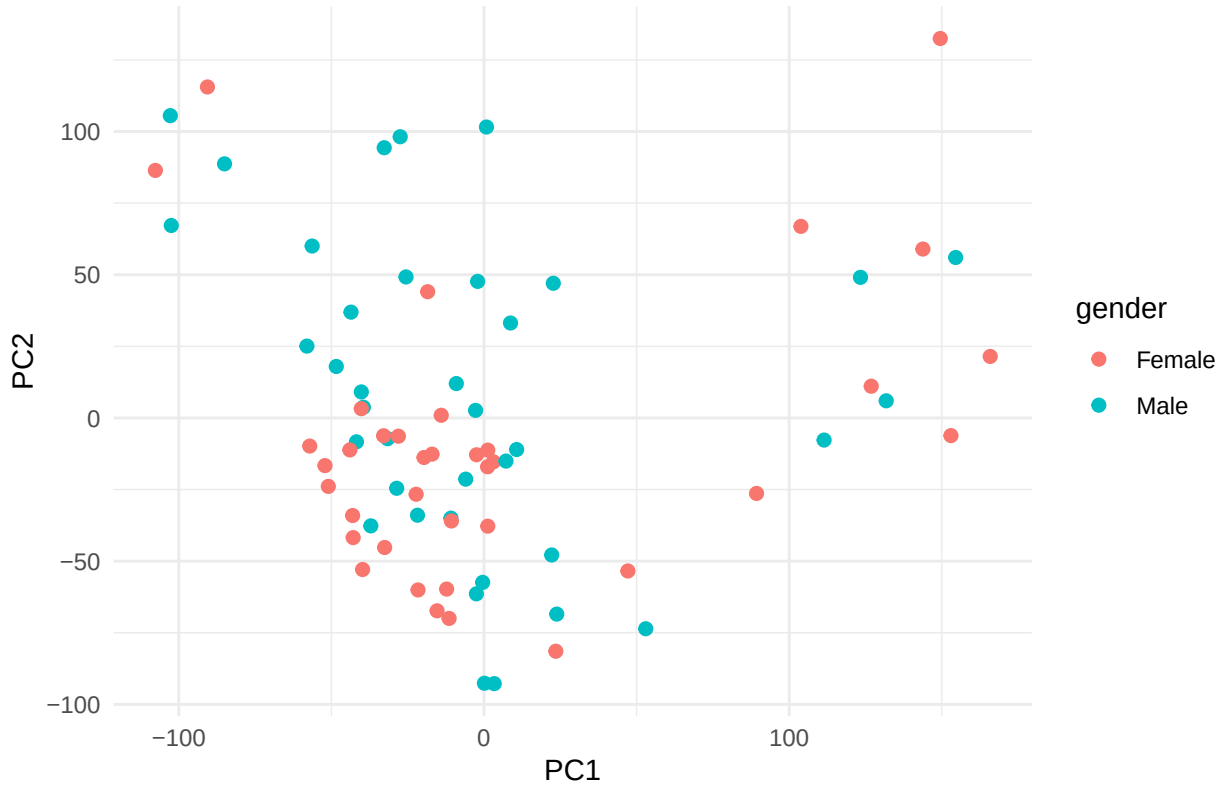


Figure A4 (D). PCA x race

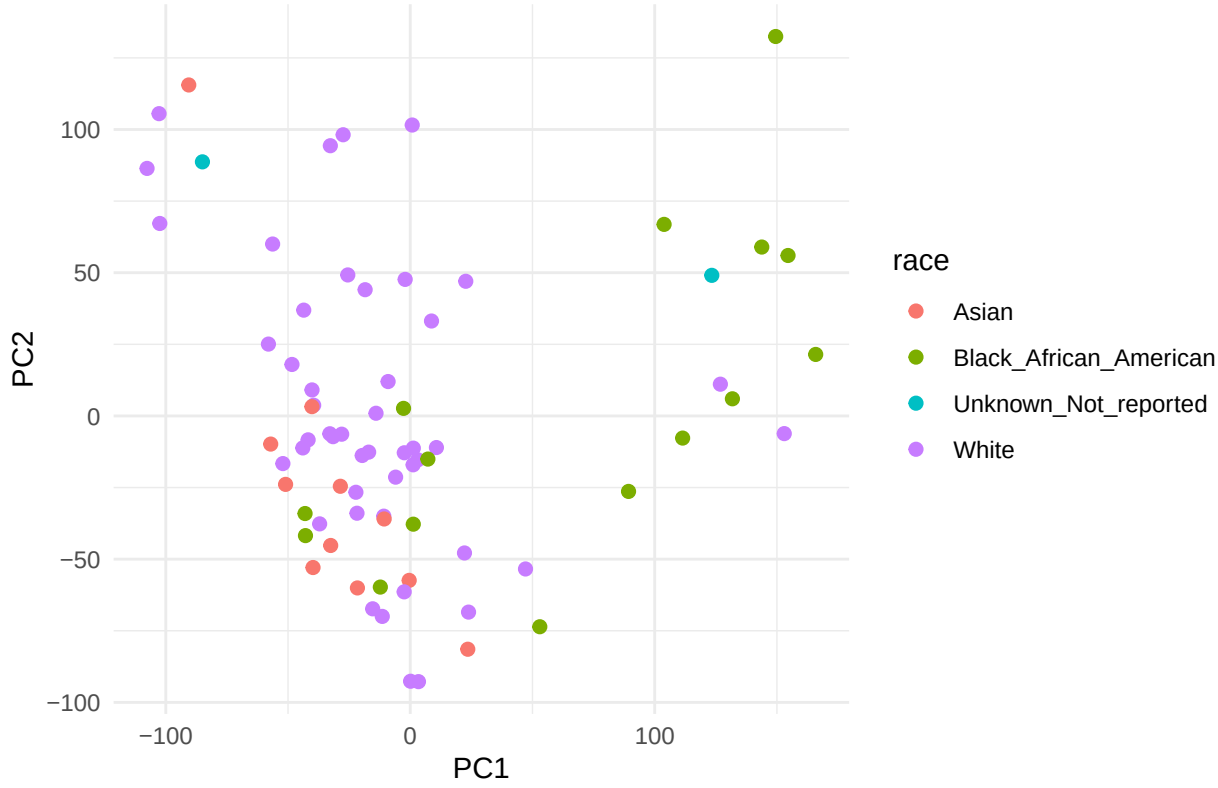


Figure A4 (E). PCA x age

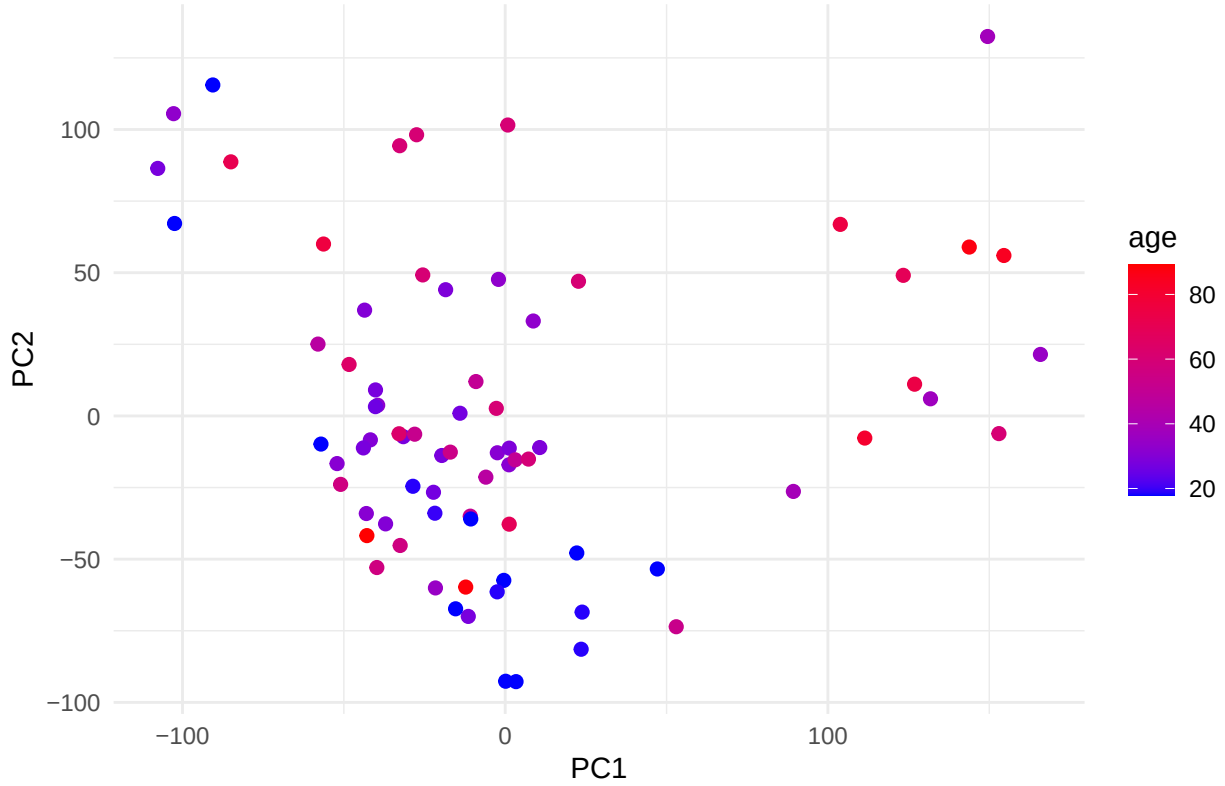


Table A1. Contingency table and chi-squared test x Race

```
##
##                                    Bacterial COVID_19 healthy
##   Asian                                    0        5       6
##   Black_African_American                   9        6       0
##   Native_Hawaiian_Pacific_Islander         0        0       0
##   Other_More_than_one_race                 0        0       0
##   Unknown_Not_reported                     1        1       0
##   White                                    2       36       8
##
##  Pearson's Chi-squared test
##
## data:  table(colData(my_se)$race, colData(my_se)$cohort)
## X-squared = NaN, df = 10, p-value = NA
```

Table A2. Contingency table and chi-squared test x Gender

```
##
##          Bacterial COVID_19 healthy
##   Female         7       24       6
##   Male           5       24       8
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(colData(my_se)$gender, colData(my_se)$cohort)
## X-squared = 0.61905, df = 2, p-value = 0.7338
```

Table A3. Contingency table and chi-squared test x Batch

```
##
##     Bacterial COVID_19 healthy
##   1        12       39      14
##   2         0        9       0
##
##  Pearson's Chi-squared test
##
## data:  table(colData(my_se)$batch, colData(my_se)$cohort)
## X-squared = 5.55, df = 2, p-value = 0.06235
```

Table A4. ANOVA test x Age

```
## Analysis of Variance Table
##
## Response: age
##           Df Sum Sq Mean Sq F value    Pr(>F)
## cohort     2  12606  6302.9  23.152 1.816e-08 ***
## Residuals 71  19329   272.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure A5. Age distribution x cohort

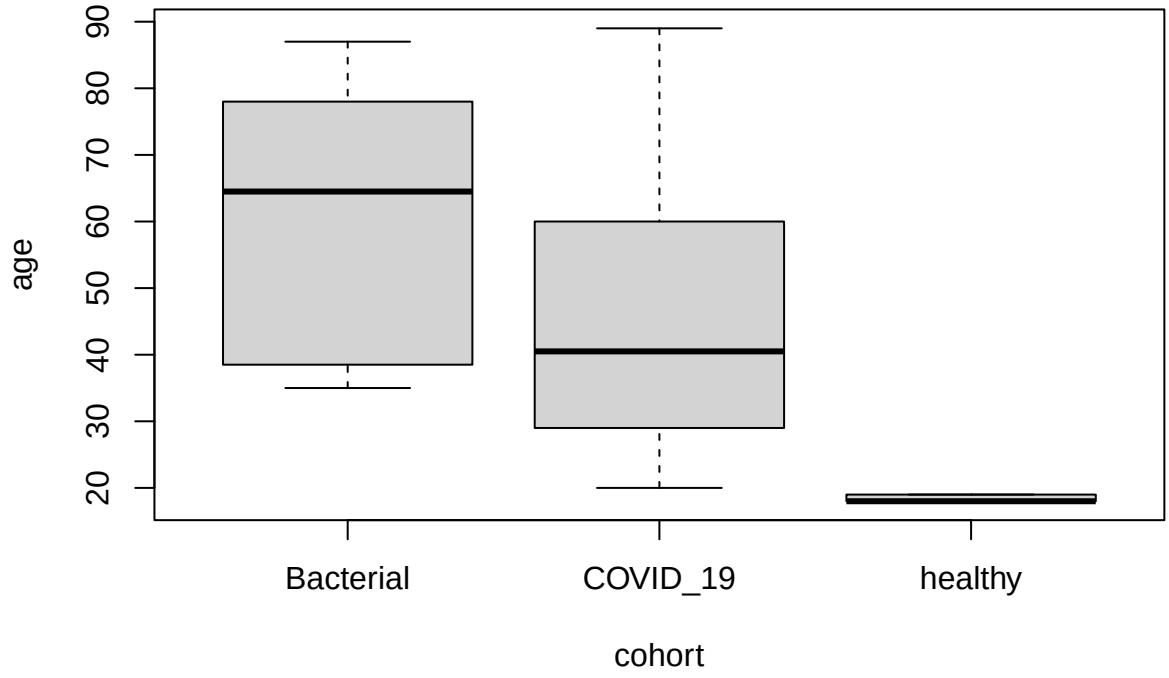


Figure A6. Volcanoplot: COVID−19 vs Healthy

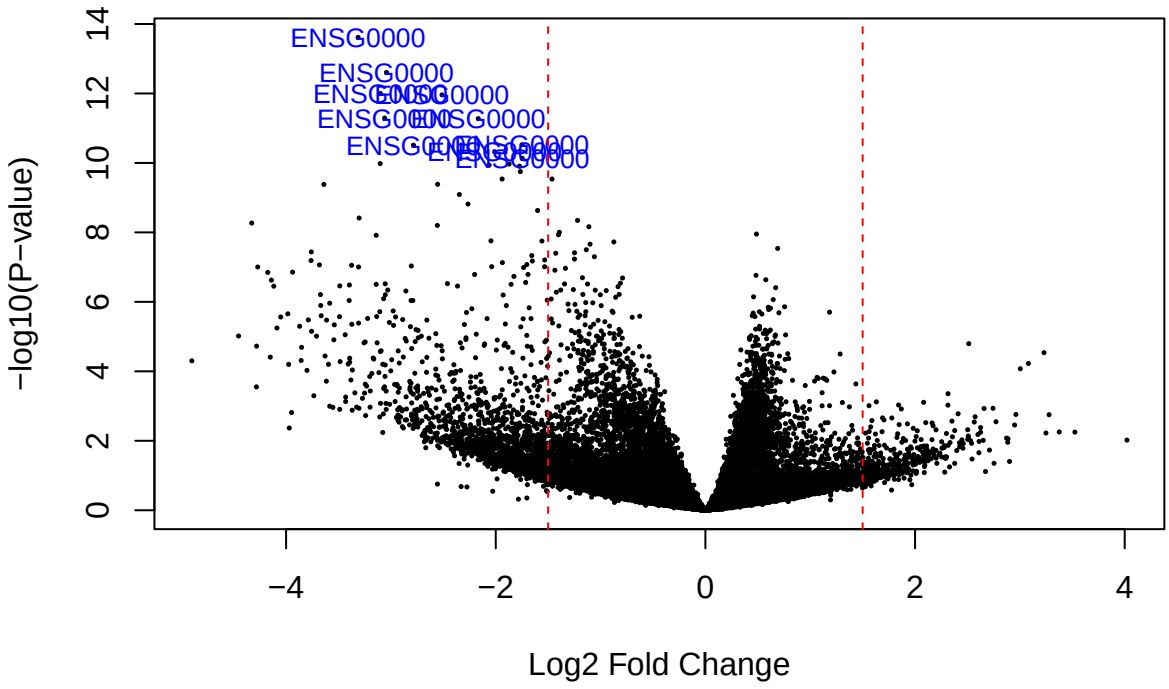


Figure A7. Heatmap: COVID−19 vs Healthy

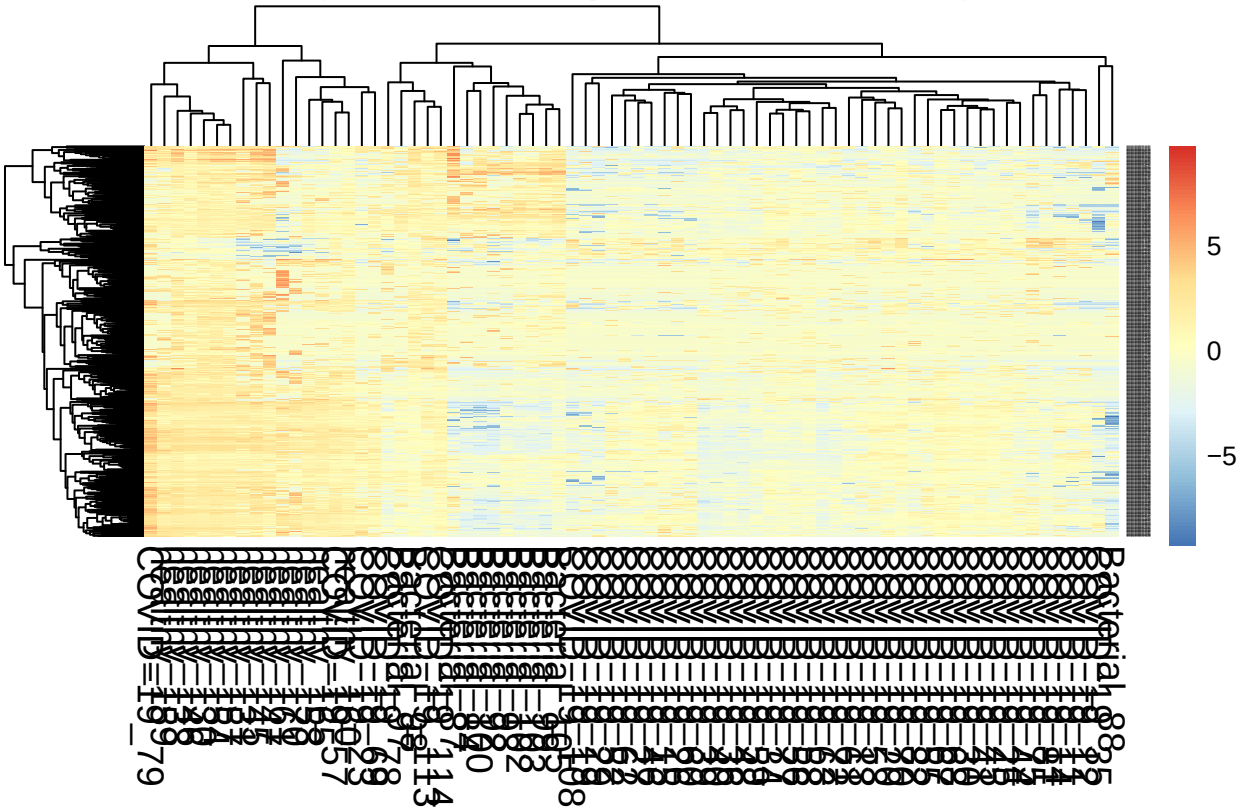


Figure A8. Volcanoplot: Bacterial vs Healthy

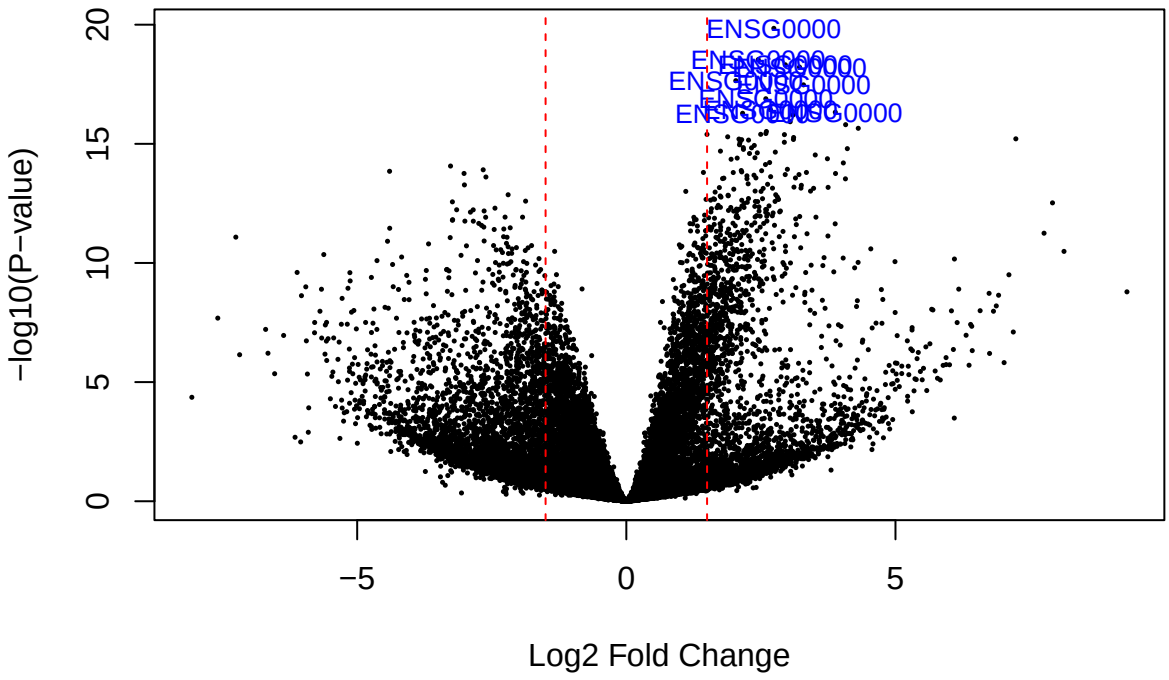


Figure A9. Heatmap: Bacterial vs Healthy

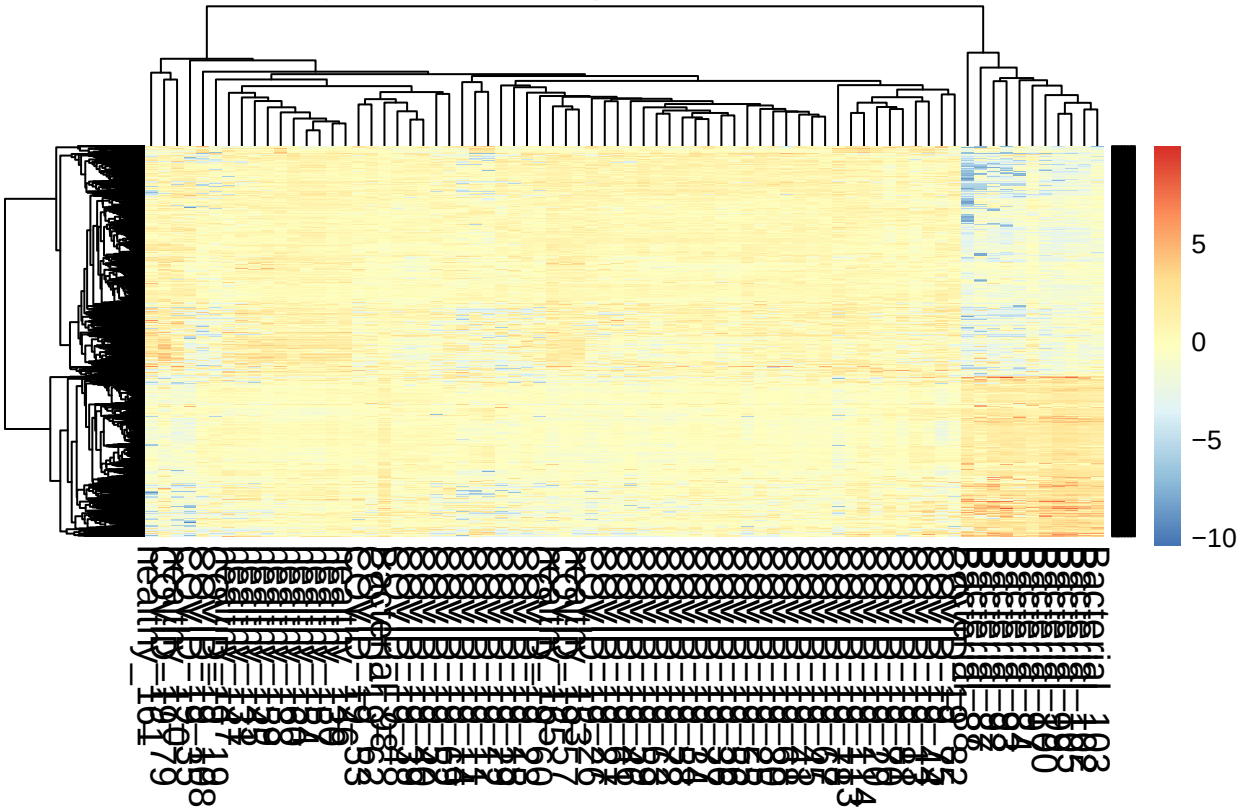


Figure A10. Enriched biological processes identified

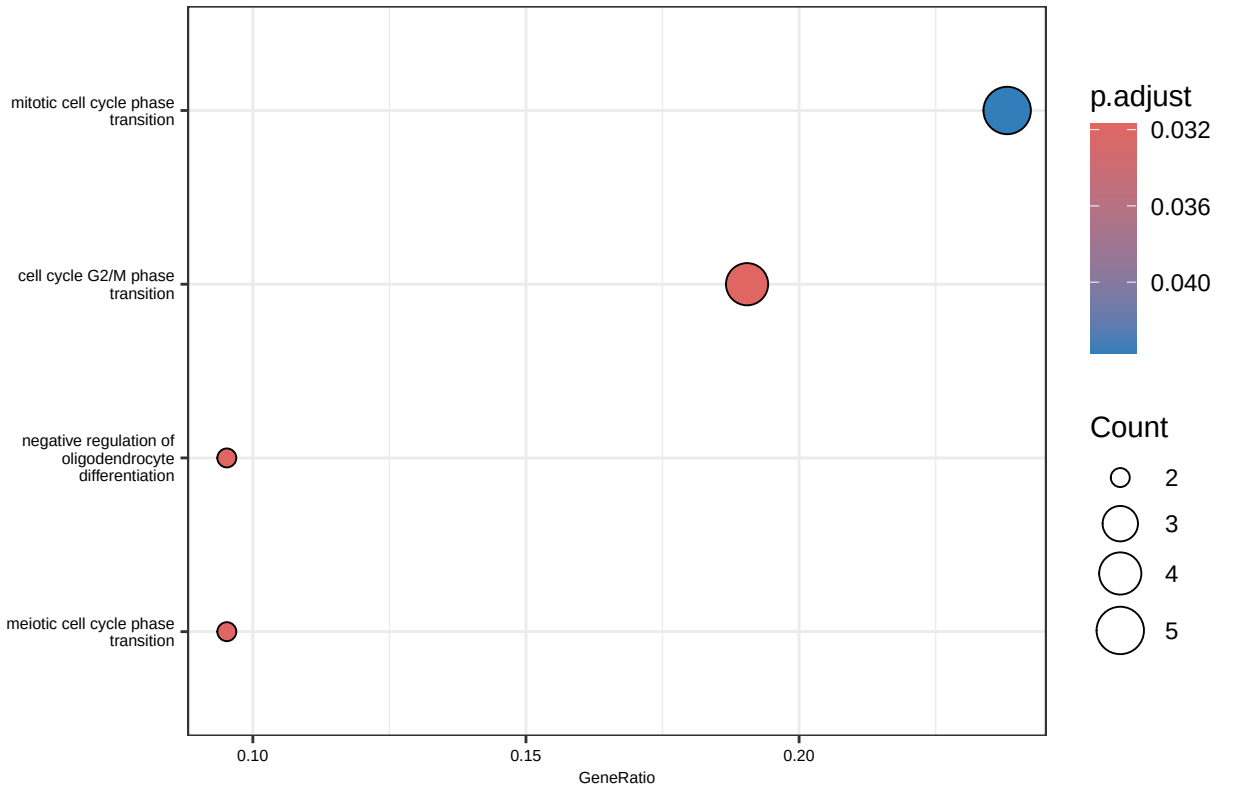


Figure A11. Network of genes associated with enriched biolog