Computational genomics project

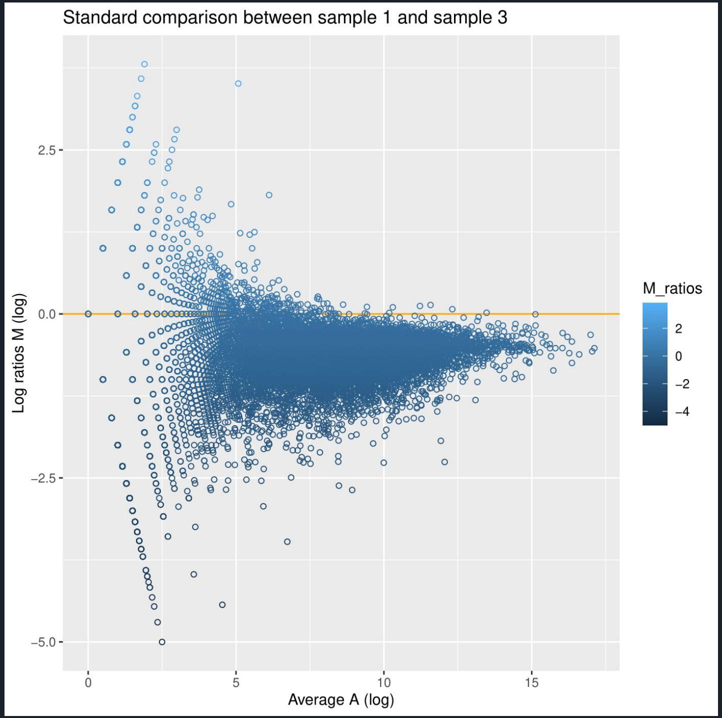
Group #7:
Matteo Moro
Elena Idi
Valentina Panizzutti
Alberto Tosadori
Vittorio Curci
Giulia Rizzoli

Topics Covered

- Data Normalization
  - Vittorio Curci
- Permutations test
  - Alberto Tosadori
  - Giulia Rizzoli
- Gene ontology and Fisher Test
  - Valentina Panizzutti
- Clustering
  - Matteo Moro
  - Elena Idi

Exercise 1

The goal of this exercise was to perform data normalization



TMM normalization

Shift every point by a scaling factor

In particular:

1. Replace all zeros with ones in the original dataframe

2. Transform the data using logarithm

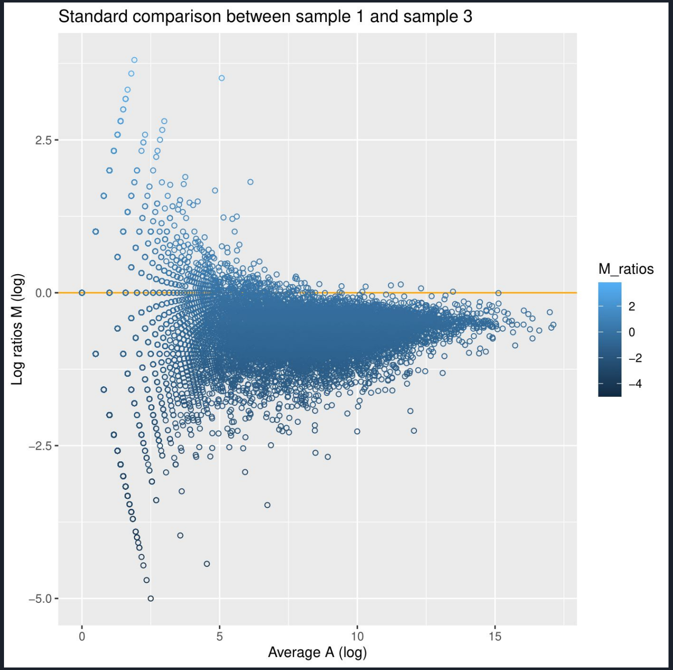
3. Compute the log-ratio M, for all the genes, between sample 1 (reference) and all the other samples

4. Estimate the scaling factor SF as the trimmed mean of M, with trim=0.05

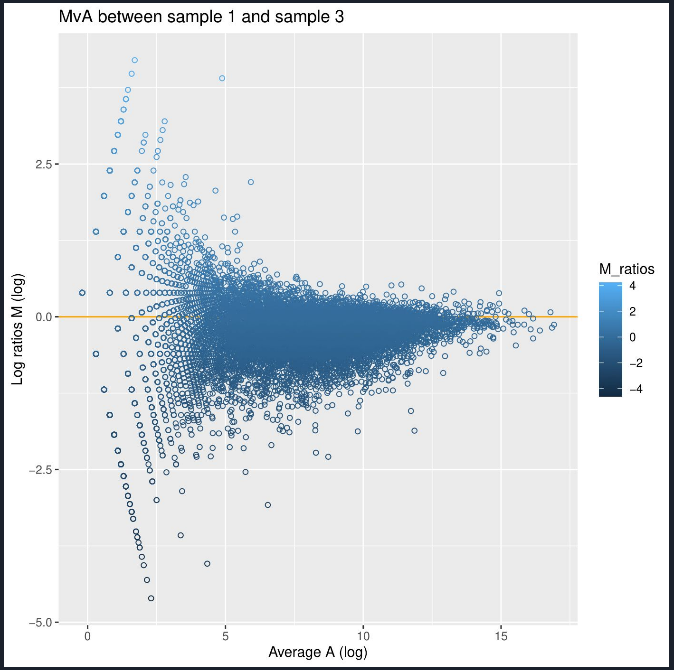
5. Add SF to all the samples except the reference

TMM normalization: results

Before normalization



After normalization



Quantile normalization

1. Create the dataframe “data_sorted”, which contains the original data with each column sorted in ascending order

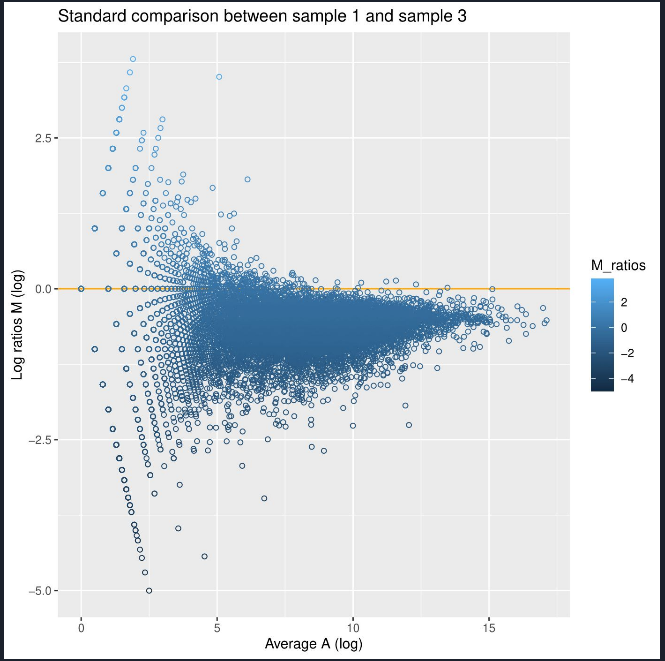
2. Create the dataframe “data_rank”, which contains the ranks of the genes positions: ties were managed using the “min” method

3. Compute the mean by rows of «data_sorted» and save the results in “data_mean”

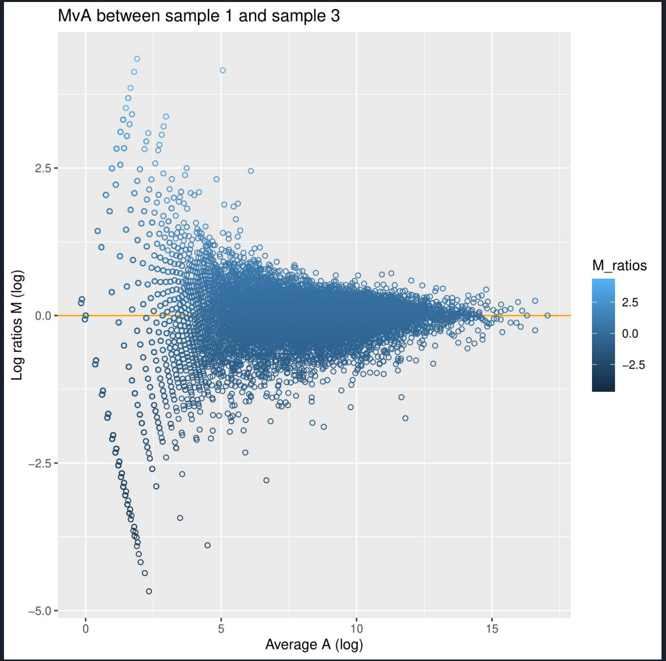
4. Create the dataframe “data_norm” with the normalized data by taking values from “data_mean”, and placing them in the new dataframe according to “data_rank”

Quantile normalization: results

Before normalization

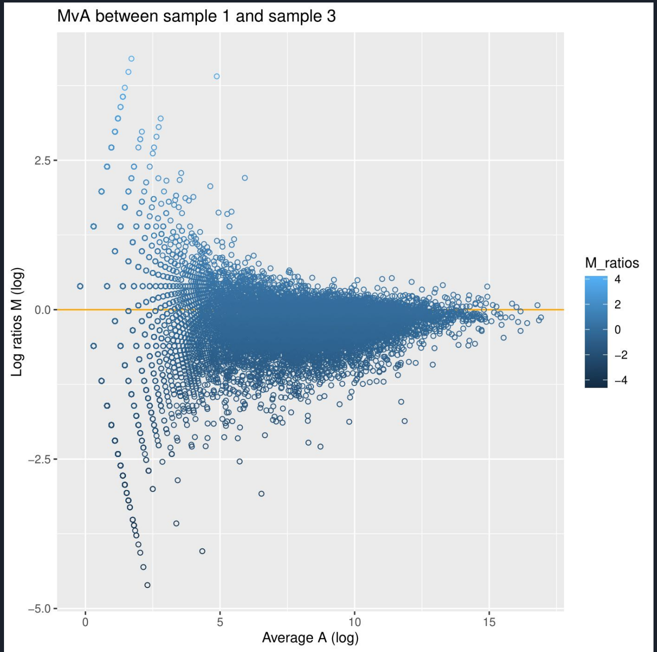


After normalization

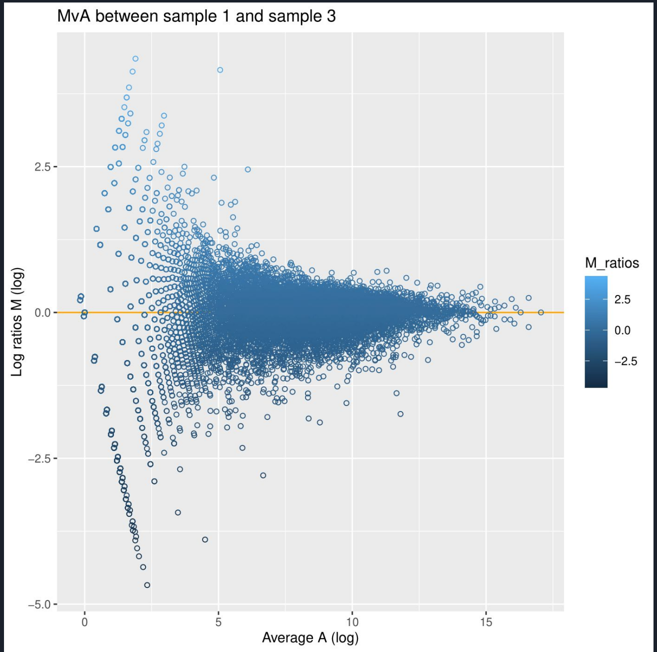


TMM vs Quantile

TMM normalization



Quantile normalization



Exercise 2: differential expression (DE) analysis

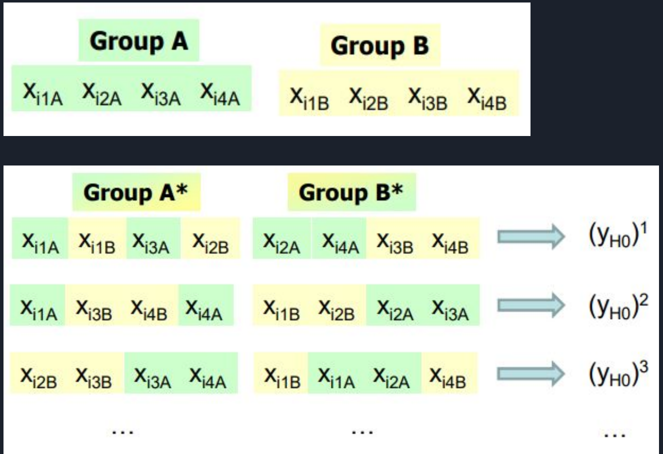
Aim: Find the set of differentially expressed genes for SARS-CoV-2

1. Calculate p-values of DE analysis between the two groups using permutation test.
2. Choose the right y statistic for the test.
3. Perform test correction:
   a. Bonferroni correction;
   b. FDR correction.

Permutation test

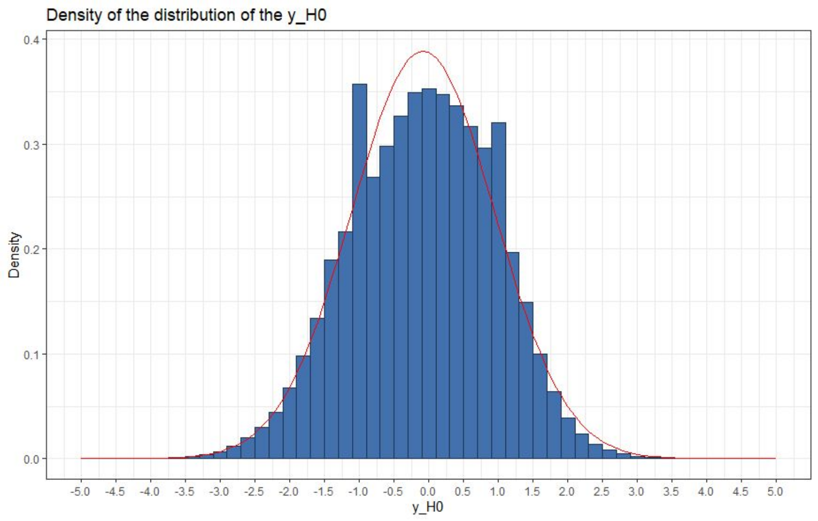
Aim: Compute y distribution under the null hypothesis using permuted data.

Healthy Infected

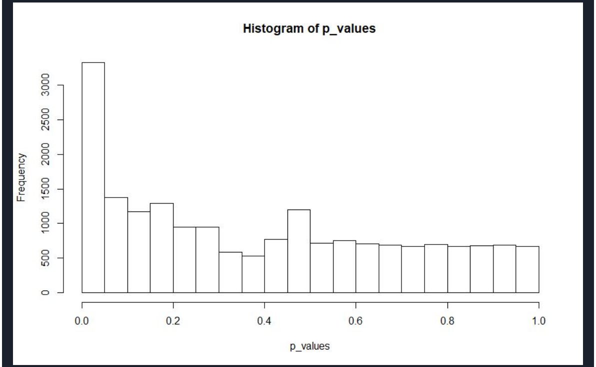


Group A				Group B			
x_{i1A}	x_{i2A}	x_{i3A}	x_{i4A}	x_{i1B}	x_{i2B}	x_{i3B}	x_{i4B}

Group A*				Group B*				
x_{i1A}	x_{i1B}	x_{i3A}	x_{i2B}	x_{i2A}	x_{i4A}	x_{i3B}	x_{i4B}	⇒ $(y_{H0})^1$
x_{i1A}	x_{i3B}	x_{i4B}	x_{i4A}	x_{i1B}	x_{i2B}	x_{i2A}	x_{i3A}	⇒ $(y_{H0})^2$
x_{i2B}	x_{i3B}	x_{i3A}	x_{i4A}	x_{i1B}	x_{i1A}	x_{i2A}	x_{i4B}	⇒ $(y_{H0})^3$
…				…				…



Test selection



Student t-test

New test
Drop the assumption:

$\sigma_A \neq \sigma_B$

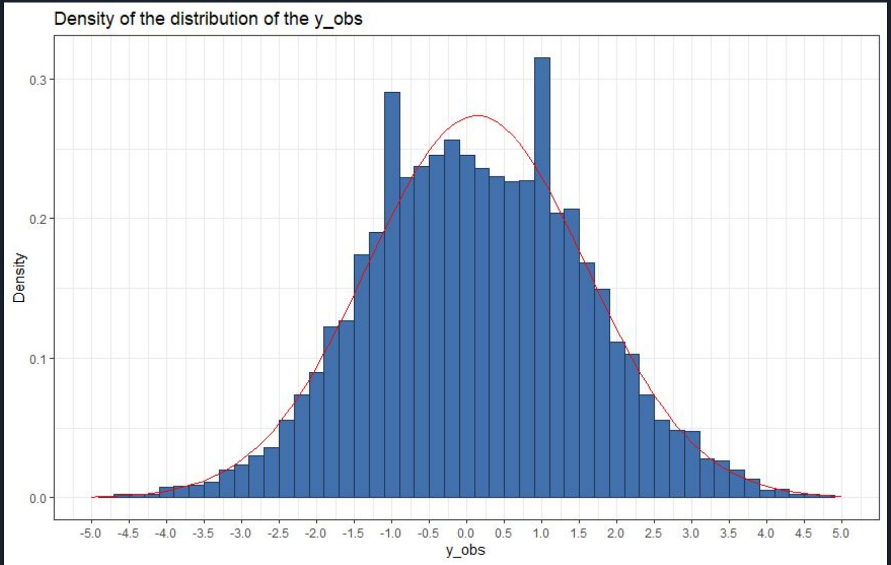
More robust statistics

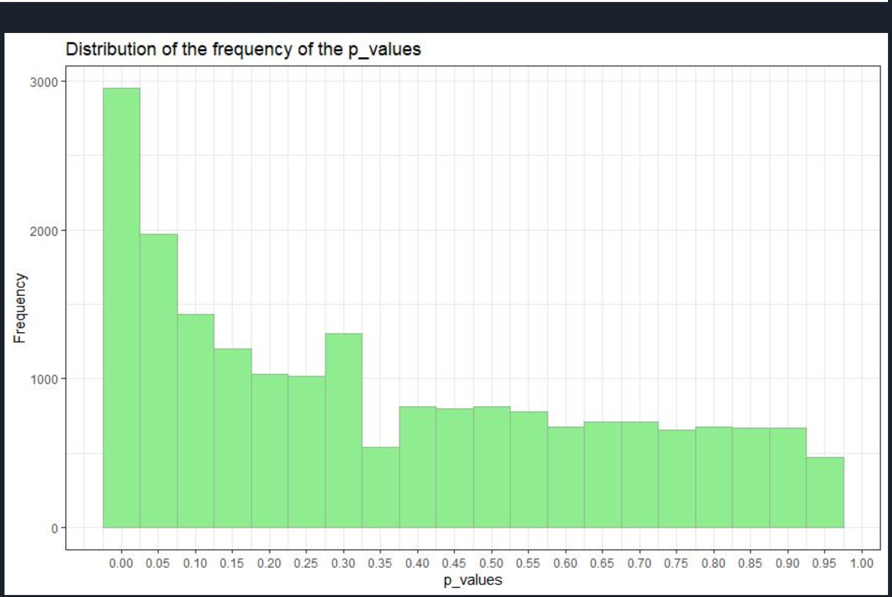
Welch-Student t-test

Test results

Using Welch-Student t-test:

$$\text{p-value} = \#|\ (y_{i,H0})^b\ | > y_{1,obs}\ /(B\ {*}N)\ \forall\ i=1,\ \ldots,N$$





Test correction (1/5)

Aim: Select a proper significance level to avoid having too many false positives (setting FP rate to 5%).

Sidak correction:

$$\alpha = 1 - (1 - desired\_glob\_FP\_rate)^{1/G}$$

Bonferroni correction:

$$\alpha = desired\_glob\_FP\_rate/G$$

Test correction (2/5) - Bonferroni vs Sidak

alpha Bonferroni = 2.51585e-06

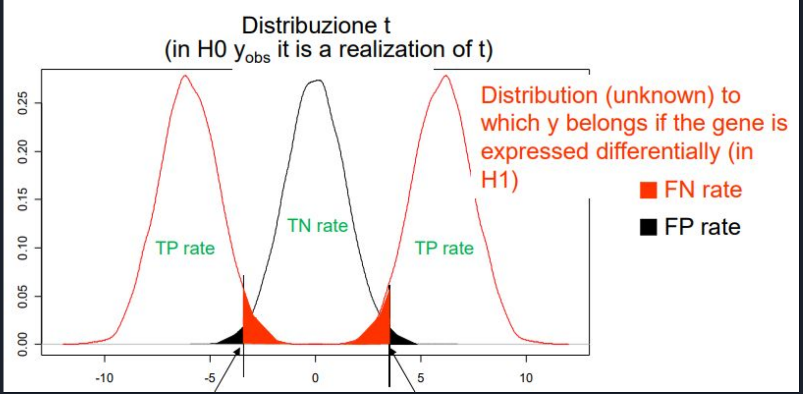
alpha Sidak = 2.580921e-06

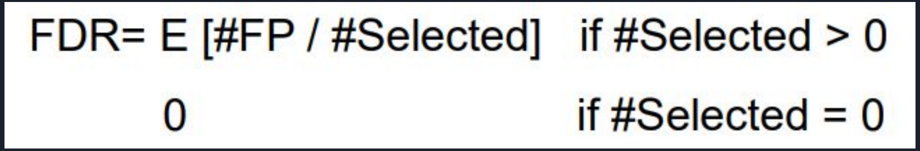
# Selected with Bonferroni = 795

# Selected with Sidak = 795

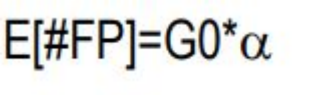
Test correction (3/5) - FDR correction

Aim: control both FP rate and FN rate by choosing alpha properly.





$$FDR = \begin{cases} E\,[\#FP \,/\, \#Selected] & \text{if } \#Selected > 0 \\ 0 & \text{if } \#Selected = 0 \end{cases}$$

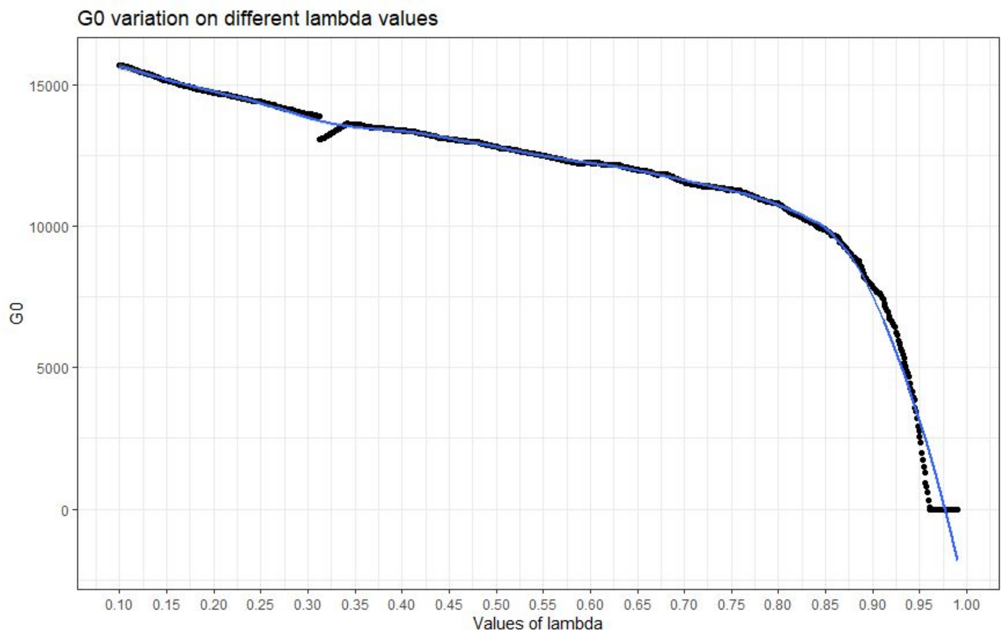


$$E[\#FP]=G0^{*}\alpha$$

Need an estimation of G0

Test correction (4/5) - Estimation of G0

Aim: find lambda for which G0 stabilizes and use that G0 for FDR correction.

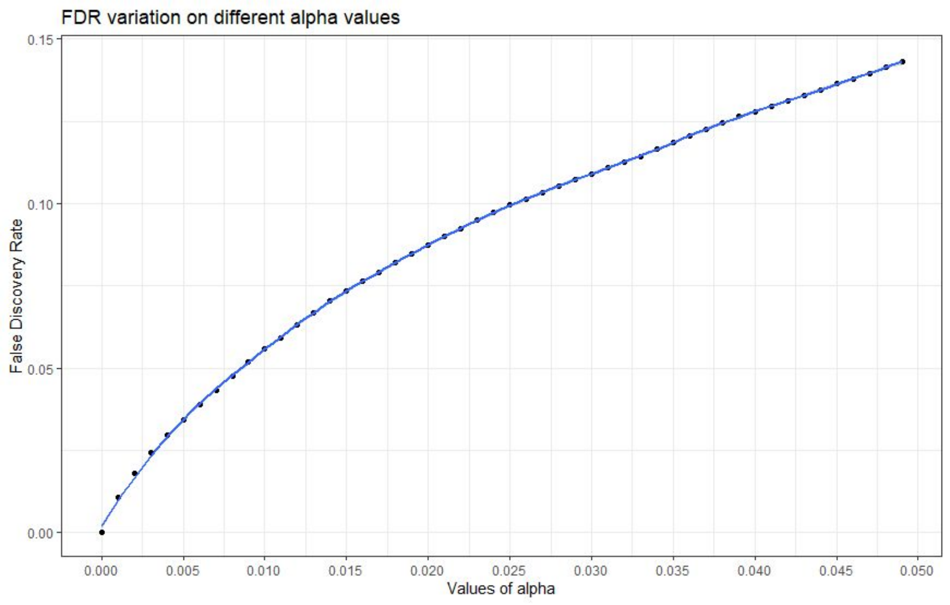


lambda = 0.687

G0 = (G-Selected_lambda)/(1-lambda)

= 11754

Test correction (5/5)

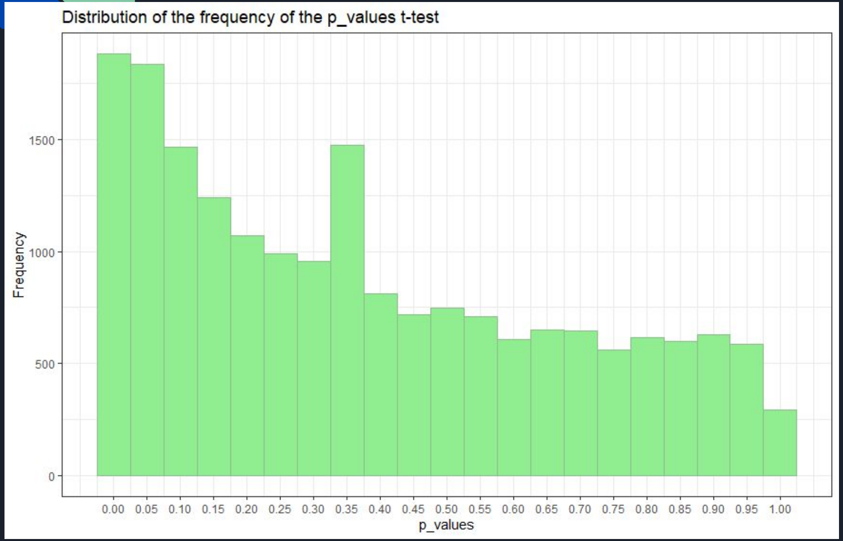


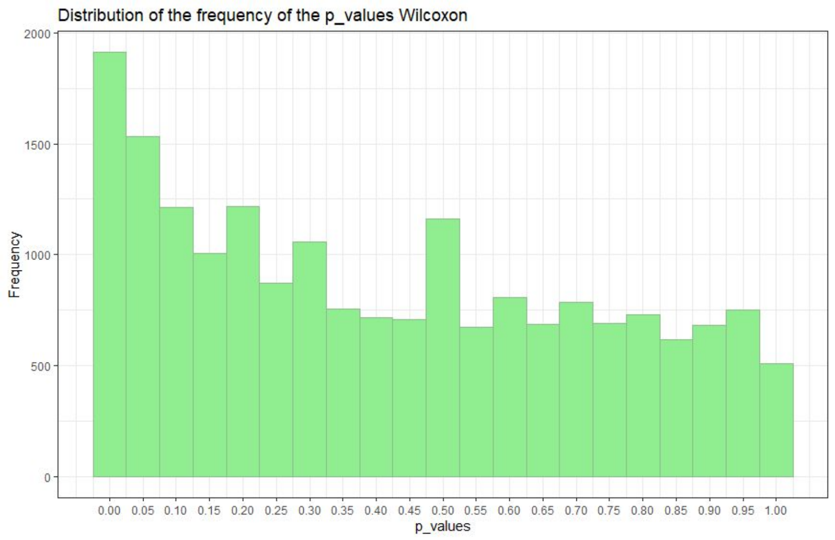
Aim: FDR = 5%

Results:
alpha = 0.0090001

# Selected = 2035

Comparison with other tests (w/o permutations)





Welch t-test: Power of test = 0.6396904

Wilcoxon: Power of test = 0.6432126

Permutations: Power of test = 0.6884979

Exercise 3

In this exercise, the goal was to implement the enrichment Analysis of GO terms starting from the list of genes selected in exercise 2.

GO Enrichment Analysis: some important concepts

It represents the best choice for the study of gene functionality by making use of a controlled vocabulary.

Gene Ontology describes the functional annotation of genes and their gene products in a stable way for each organism.

Each listed GO element contains:

- an identification code (id);

- a name associated with it (GO_id);

- the vocabulary to which it refers (3 types: Cellular Component, Biological Processes and Molecular Functions);

- a description of the concept;

- a list of possible synonyms of the concepts represented.

Having an available list of analyzed genes (in this case, a subset of differentially expressed genes), it is possible to conduct a functional enrichment analysis.

This analysis allows to verify through a statistical test (Fisher test) if the number of selected genes belonging to a certain GO term is significantly greater than which could have been obtained with a random selection of these genes.

Methods

The databases belonging to 2 specific libraries have been taken as reference: GO.db (the controlled vocabulary described above) and org.Hs.eg.db (sequenced genome of Homo Sapiens), useful, in general, for the annotation of genes.

To build the code, another library was also used, DBI, which allowed to work and write the necessary queries directly in SQL (since we work with databases).

Code organization

From org.Hs.eg.db: selection of ID, GO_id and Symbol columns (all genes) → query 1

From GO.db: selection of ID column (all genes) → query 2

Merging the query 1 and query 2 on the ID (full outer join) using the R function “merge” → merged.dbs (all genes listed and found in the 2 initial databases).

It was loaded the list of selected genes saved in exercise 2, renaming the column as ‘Symbol’ to be able to compare it with merged.dbs to obtain the list of sel_genes_GO.

Using query 4, it was assessed which of sel_genes_GO had more than 2 occurrences (it is not very effective to work with those terms whose occurrence is 0 or 1).

Fisher test

It was conducted on all GO terms with more than 2 occurrences.

A table (fisher_table) has been built to make the conduct of the test more effective.

The elements of each row are associated, in terms of meaning, to the following table in which:

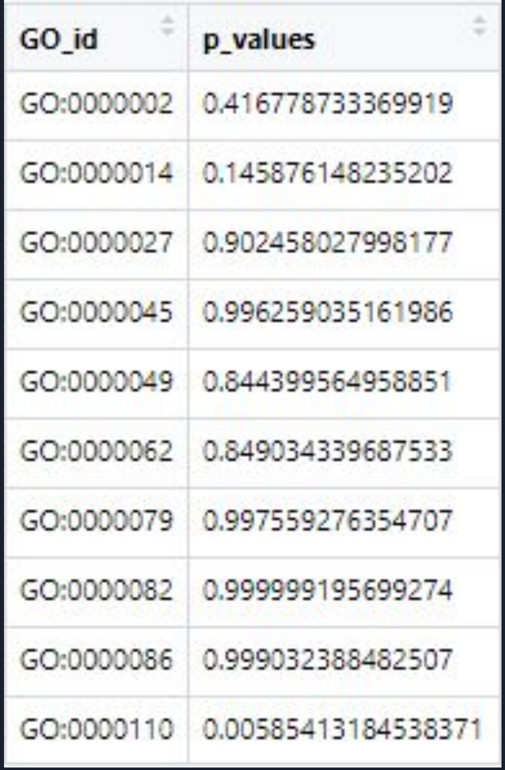
- a = total of the selected genes belonging to a specific GO_id;
- b = total of the selected genes not belonging to the specific GO_id;
- c = total of the unselected genes belonging to the specific GO_id;
- d = total of the unselected genes not belonging to the specific GO_id;

	∈GOi	!∈GOi	totRow
Selected	a	b	a+b
Non selected	c	d	c+d
TotCol	a+c	b+d	n=a+b+c+d

Through the function fisher_eval it was done the Fisher test for each GO_id term (each row of the fisher_table), returning the corresponding p_value.

Finally, a dataframe was created to contain GO_ids and their relatives p_values.

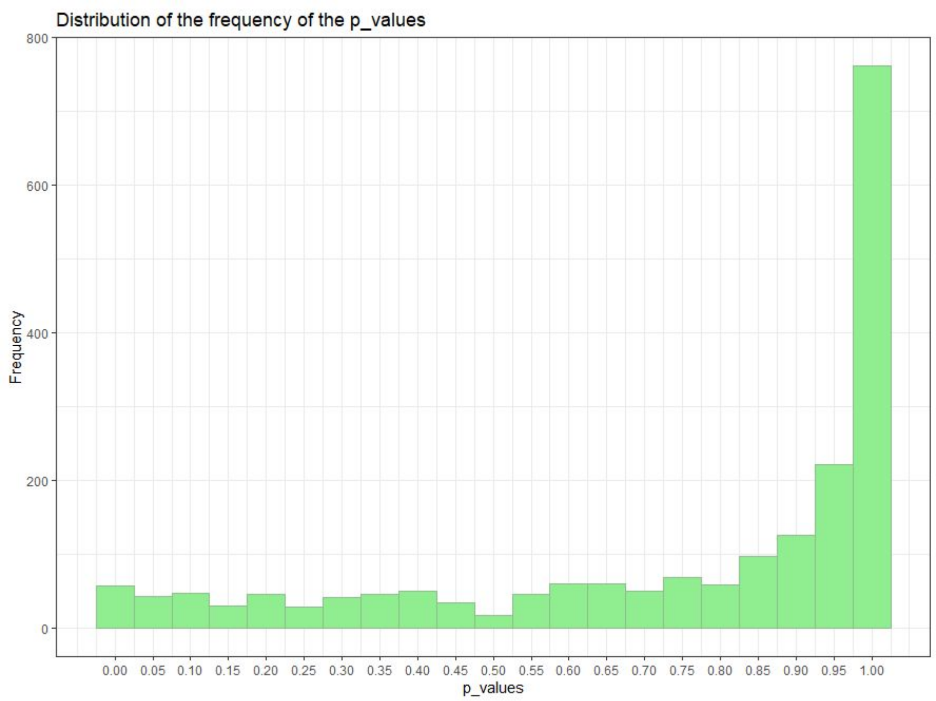
In the figure is reported part of this table.



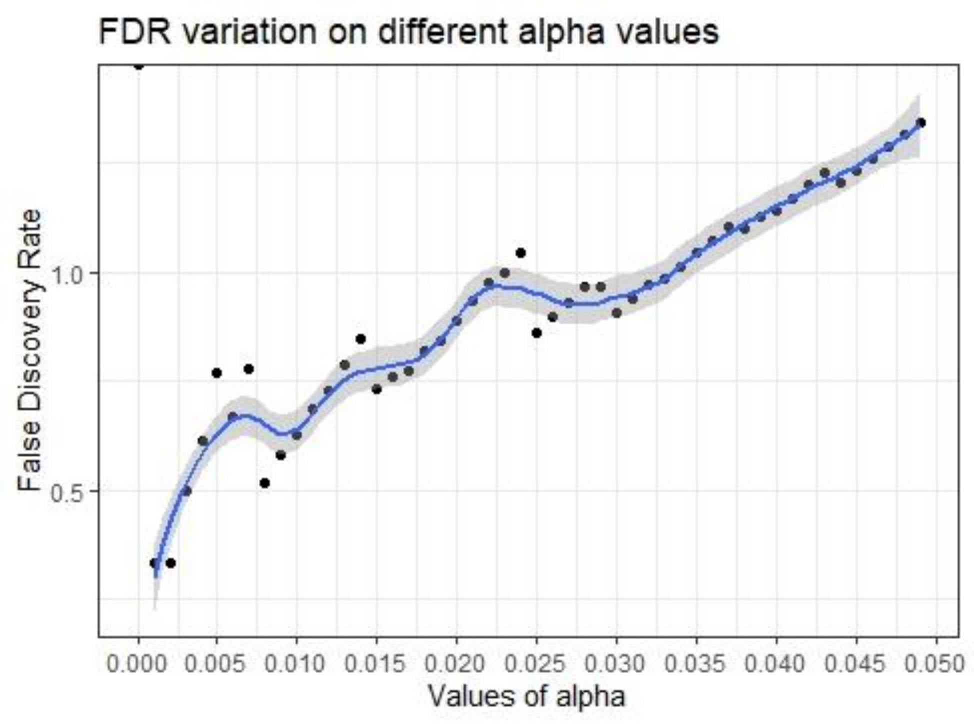
GO_id	p_values
GO:0000002	0.416778733369919
GO:0000014	0.145876148235202
GO:0000027	0.902458027998177
GO:0000045	0.996259035161986
GO:0000049	0.844399564958851
GO:0000062	0.849034339687533
GO:0000079	0.997559276354707
GO:0000082	0.999999195699274
GO:0000086	0.999032388482507
GO:0000110	0.00585413184538371

It was done a step of correction for Multiple Tests to understand which GO_ids were most characteristics and informative.

In the figure is reported the histogram of the calculated p-values.

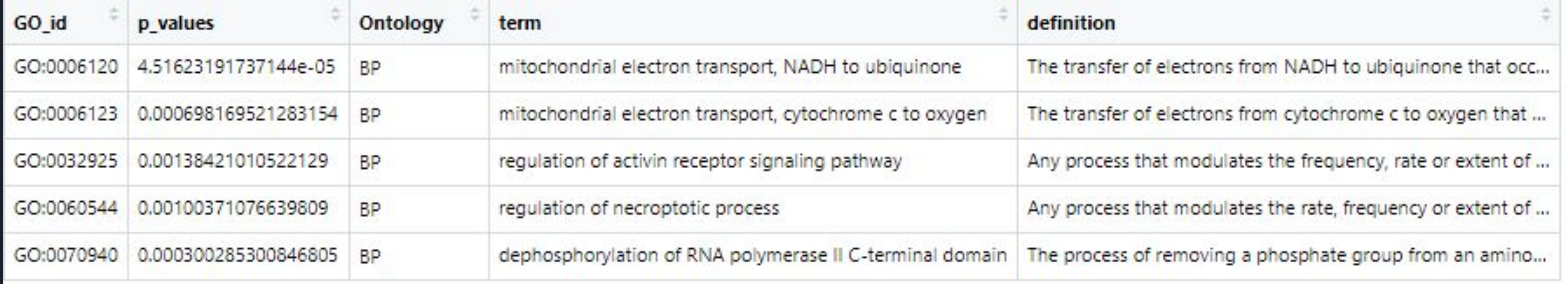


Finally, using a False Discovery Rate = 5%, we found an alpha threshold = 0.002001: this value allows to select the final list of very well represented GO_ids terms (12 with their respect p_values).

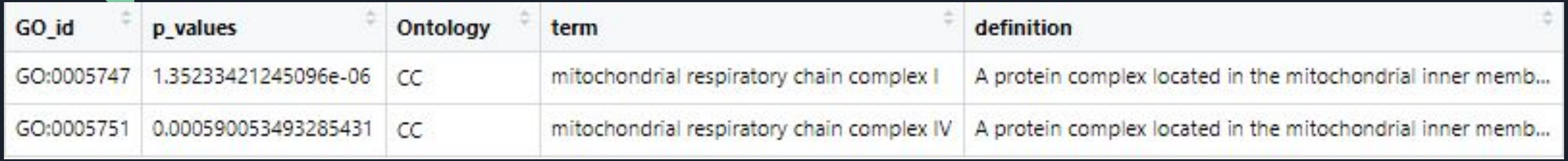


The 12 found elements were then grouped on the basis of their belonging vocabulary in tables that show the following columns: "GO_id" ,"p_values", "Ontology" ,"term" and "definition".

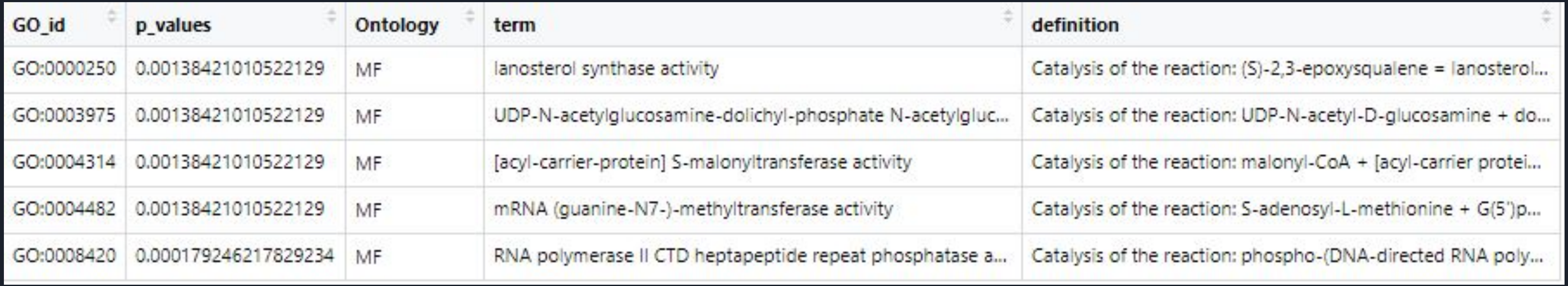
The 3 tables are shown below in the following order: BP, CC and MF.



GO_id	p_values	Ontology	term	definition
GO:0006120	4.51623191737144e-05	BP	mitochondrial electron transport, NADH to ubiquinone	The transfer of electrons from NADH to ubiquinone that occ...
GO:0006123	0.000698169521283154	BP	mitochondrial electron transport, cytochrome c to oxygen	The transfer of electrons from cytochrome c to oxygen that ...
GO:0032925	0.0013842101052129	BP	regulation of activin receptor signaling pathway	Any process that modulates the frequency, rate or extent of ...
GO:0060544	0.00100371076639809	BP	regulation of necroptotic process	Any process that modulates the rate, frequency or extent of ...
GO:0070940	0.000300285300846805	BP	dephosphorylation of RNA polymerase II C-terminal domain	The process of removing a phosphate group from an amino...



GO_id	p_values	Ontology	term	definition
GO:0005747	1.35233421245096e-06	CC	mitochondrial respiratory chain complex I	A protein complex located in the mitochondrial inner memb...
GO:0005751	0.000590053493285431	CC	mitochondrial respiratory chain complex IV	A protein complex located in the mitochondrial inner memb...



GO_id	p_values	Ontology	term	definition
GO:0000250	0.0013842101052129	MF	lanosterol synthase activity	Catalysis of the reaction: (S)-2,3-epoxysqualene = lanosterol...
GO:0003975	0.0013842101052129	MF	UDP-N-acetylglucosamine-dolichyl-phosphate N-acetylgluc...	Catalysis of the reaction: UDP-N-acetyl-D-glucosamine + do...
GO:0004314	0.0013842101052129	MF	[acyl-carrier-protein] S-malonyltransferase activity	Catalysis of the reaction: malonyl-CoA + [acyl-carrier protei...
GO:0004482	0.0013842101052129	MF	mRNA (guanine-N7-)-methyltransferase activity	Catalysis of the reaction: S-adenosyl-L-methionine + G(5')p...
GO:0008420	0.000179246217829234	MF	RNA polymerase II CTD heptapeptide repeat phosphatase a...	Catalysis of the reaction: phospho-(DNA-directed RNA poly...

Clustering

Silhouette statistics

Used to determine the optimal number of clusters

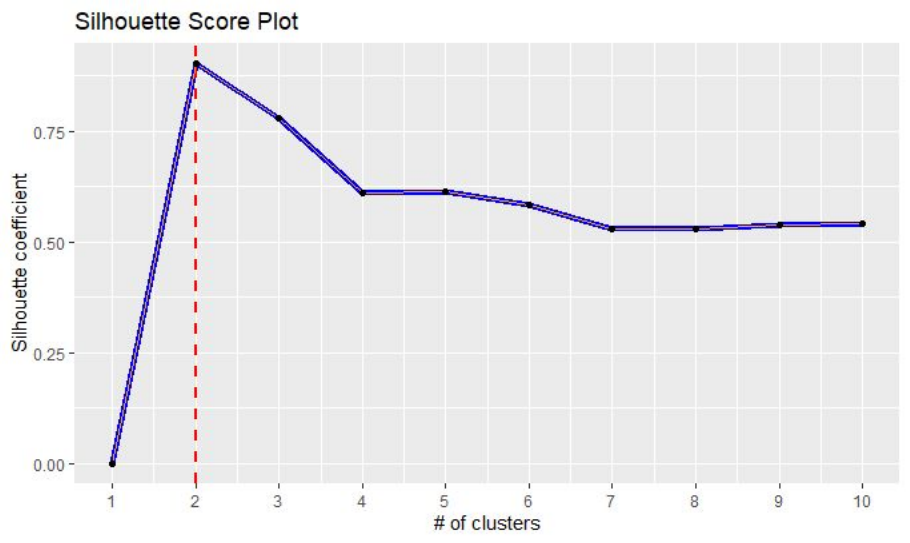
It is based on the computation of two coefficient:

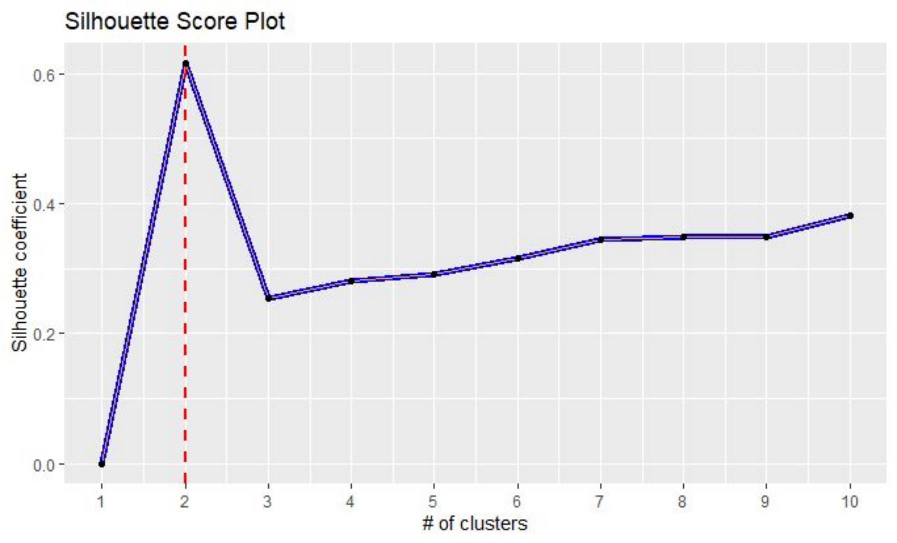
- a(i) that is the average distance of object i from other objects j in the same cluster
- b(i) that is the minimum of the average distance of a object i from the object j belonging to other clusters

Found those two coefficient we can calculate di silhouette coefficient:

$$\textit{silhouette}\;\; s(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$$

Results of the silhouette statistics





	Genes	Samples
Max Silhouette Coefficient	0.9026501	0.6152773

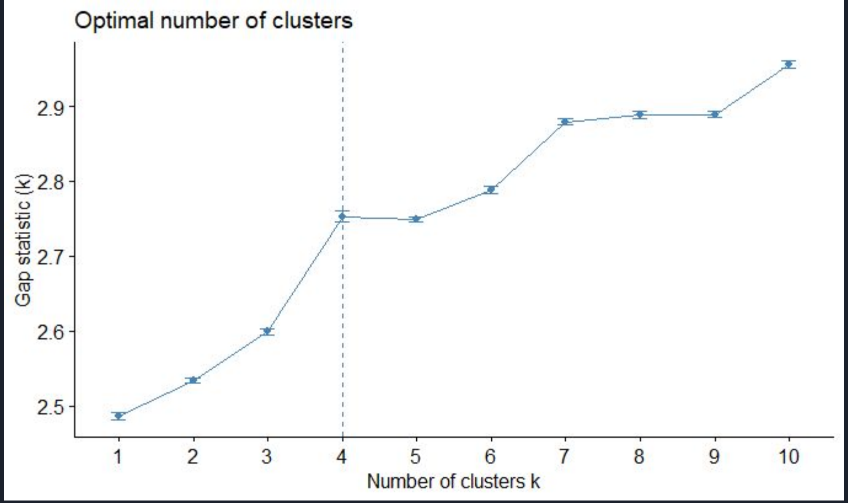
GAP statistics

- Compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data

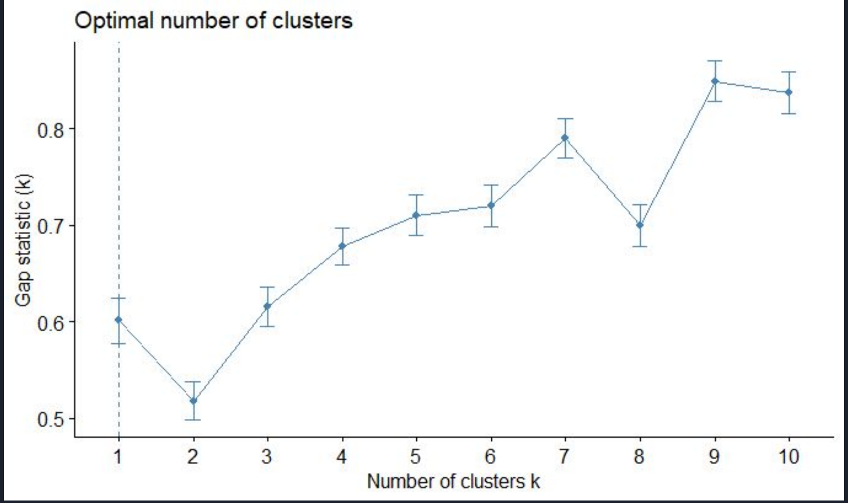
- The estimate of the optimal clusters will be the value that maximize the gap statistics
  → yields the largest gap statistics
  - This means that the clustering structure is far away from the random uniform distribution of points.

Results of the gap statistics

Optimal number of clusters :



Genes



Samples

Some consideration of the statistics

- Different optimal clusters for the considered statistics:

Optimal num. of centers	Silhouette	Gap
Genes	2	4
Samples	2	1

- We relied to the silhouette statistics to the further analysis
  - Silhouette consider both intra and extra cluster distance
  - Gap statistic K struggles with “not so well defined” clusters
    - It is harder to detect the correct point in which the rate of the Gap statistics slows down

Application of Clustering Methodologies

A brief introduction

Clustering requires techniques of unsupervised classification.

The target is to identify the best partition of objects that

- Minimizes the dissimilarity within a class
- Maximizes the dissimilarity between the classes

□ focus on distance-based clustering methodologies

Distance-based Clustering Methodologies:

Dissimilarity measure: Euclidean distance

The classification problem is solved with two methodologies:

- Combinatorial Algorithm: K-Means (Lloyd, Hartigan-Wong)
- Hierarchical Trees

K-means

- Strategy based on iterative greedy descent

  □ Optimization problem: Minimization of the sum of within cluster variances

- Euclidean distance as dissimilarity measure (although extensions exist for other distance/similarity measures)

- The number of clusters must be fixed a priori: □ Silhouette statistic

Main steps of the algorithm:

1. An initial partition is specified:
   - N objects in a M-dim space (M variables) are assigned to K clusters at random

2. At each iterative step: cluster assignment
   - Calculate the centroid of each cluster
   - FOR (i in (1:N))
     - Compute the Euclidean distance between object i and the centroid of each cluster
     - Assign i to the closest cluster
     - [H.W.] Update the centroid positions if i has changed cluster assignment

3. When none of the new partition is able to provide an improvement or the maximum number of iterations is reached, the algorithm terminates

Hierarchical trees

Two methods can be performed :

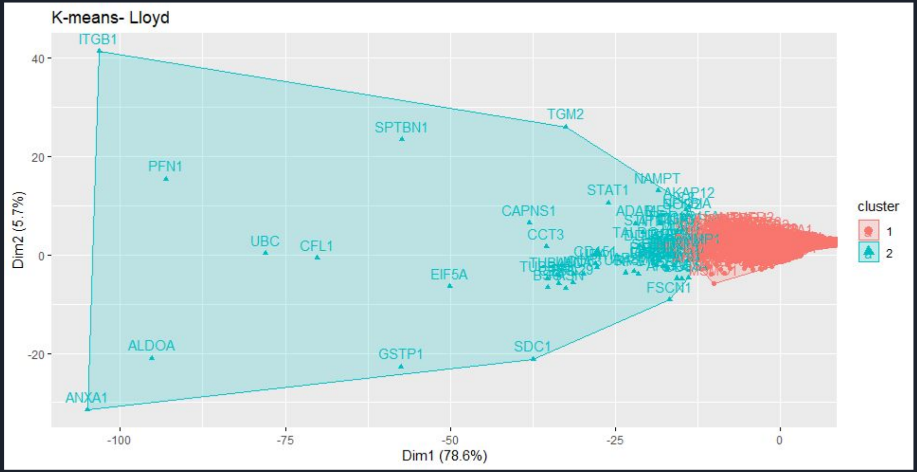
- Agglomerative Hierarchical Clustering
- Divisive Hierarchical Clustering

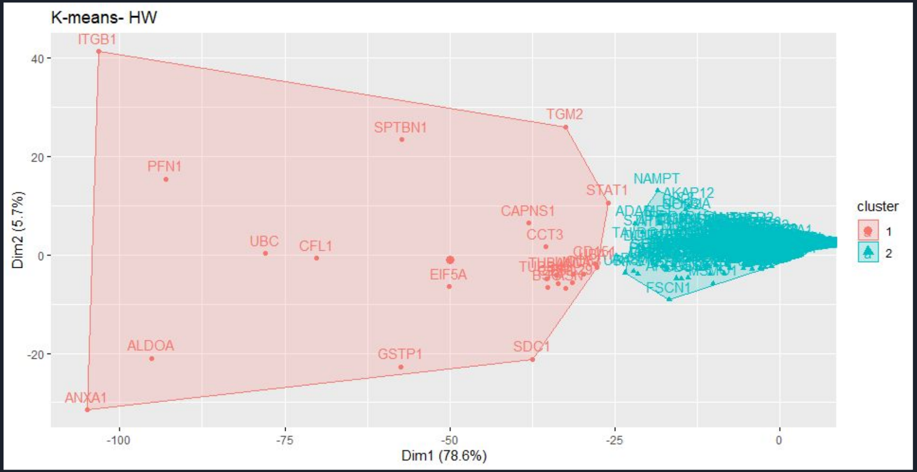
Focus on partitioning the data into a small number of clusters

□ topdown approach

Divisive clustering algorithms begin with the entire data set as a single cluster, and recursively divide one of the existing clusters into two clusters

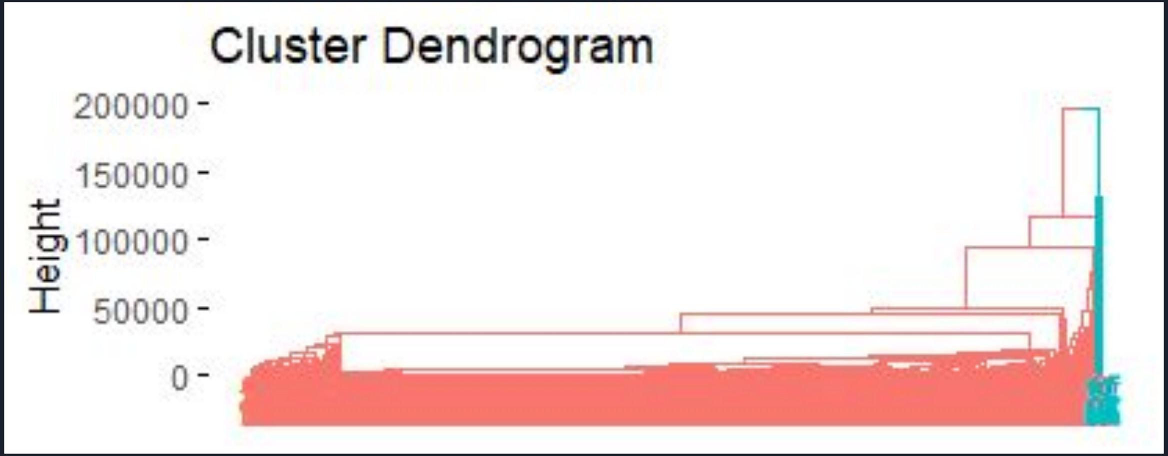
Results: Clustering of the genes





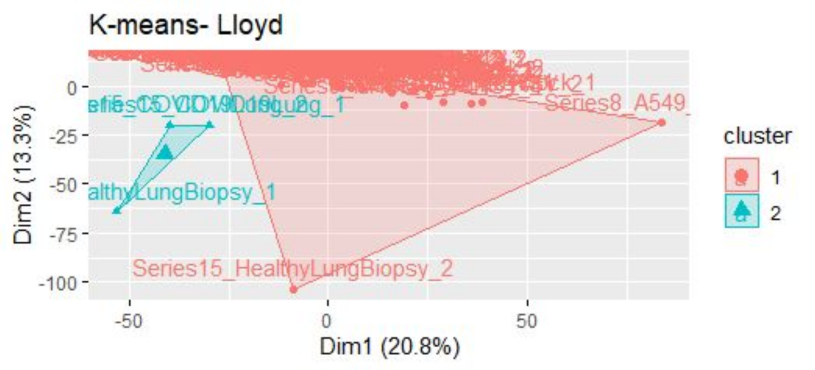
	K-means (Lloyd)	K-means (HW)
Number of genes in each cluster	73 – 1962	24 - 2011

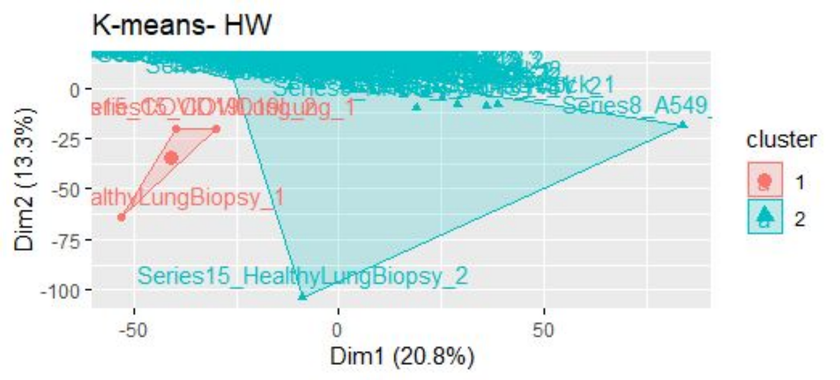
Total/within/between sum of squares	3.47e(11)
	2.11e(11)
	1.36e(11)
		3.47e(11)
		2.09e(11)
		1.39e(11)



	DIANA	K-means (HW)
Number of elements in each cluster	9 - 2026	24 - 2011
Genes in common:	Class 1: 9, Class 2: 1982	

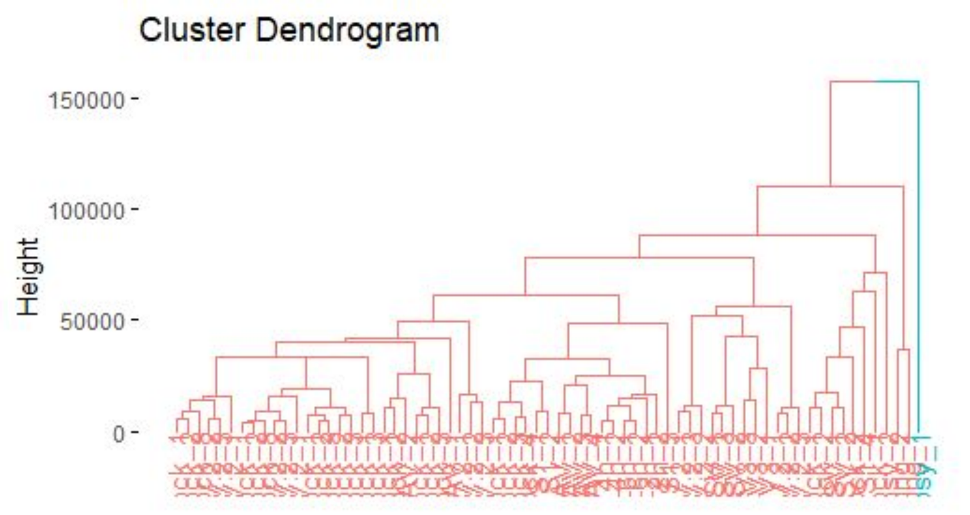
Results: Clustering of the samples





	K-means (Lloyd)	K-means (HW)
Number of samples in each cluster	3 – 66	3 – 66

Total/within/between sum of squares	9.83e(10)
	7.16e(10)
	2.67e(10)
		9.83e(10)
		7.16e(10)
		2.67e(10)



	DIANA	K-means (HW)
Number of elements in each cluster	1 - 68	3 – 66
Samples in common:	Class 1: 1, Class 2: 66	