Project_liver_cancer_cells

Alessandro Pilli, Filippo A. Mirolo, Vittoria Ossanna

2023-06-14

Settings

Task 1

Loading the required data

Task 2

The first step is to filter all the information presents in counts and annotations. To do so we extracted all the information filtering only the proteins coding (CDS). We also introduced a pseudocount of plus +1 for further analysis, to avoid problem with the logarithmization of the dataset.

```r
query <- getBM(attributes = c("ensembl_gene_id", "gene_biotype", "external_gene_name",
    "entrezgene_id"), filters = c("ensembl_gene_id"), values = list(c(r_anno_df$ensembl_gene_id)),
    mart = ensembl)

# Selecting only protein coding

query <- query[which(query$gene_biotype == "protein_coding"), ]

r_anno_df <- r_anno_df[is.element(r_anno_df$ensembl_gene_id, query$ensembl_gene_id),
    ]
raw_counts_df <- raw_counts_df[is.element(rownames(raw_counts_df), query$ensembl_gene_id),
    ]
raw_counts_df <- raw_counts_df + 1
```

Task 3

Perform differential expression analysis using edgeR package and select up- and down-regulated genes using a p-value cutoff of 0.01, a log fold change ratio >1.5 for up-regulated genes and < (-1.5) for down-regulated genes and a log CPM >1. Relax the thresholds if no or few results are available. 1. Use the workflow we developed during the course 2. Filter raw counts data retaining only genes with a raw count >20 in at least 5 Cases or 5 Control samples 3. Create a volcano plot of your results 4. Create an annotated heatmap focusing only on up- and downregulated genes

A second filtering of the raw data was performed selecting a count threshold of counts, greater then 20, and an occurrences of five samples in both groups.

```r
count_thr <- 20
repl_thr <- 5

# creating a vector of rows that we need to remove
filter_vec <- apply(raw_counts_df, 1, function(y) max(by(y, c_anno_df$condition,
    function(x) sum(x >= count_thr))))

# remove the rows from annotations and counts
filter_counts_df <- raw_counts_df[filter_vec >= repl_thr, ]
filter_anno_df <- r_anno_df[rownames(filter_counts_df), ]
```

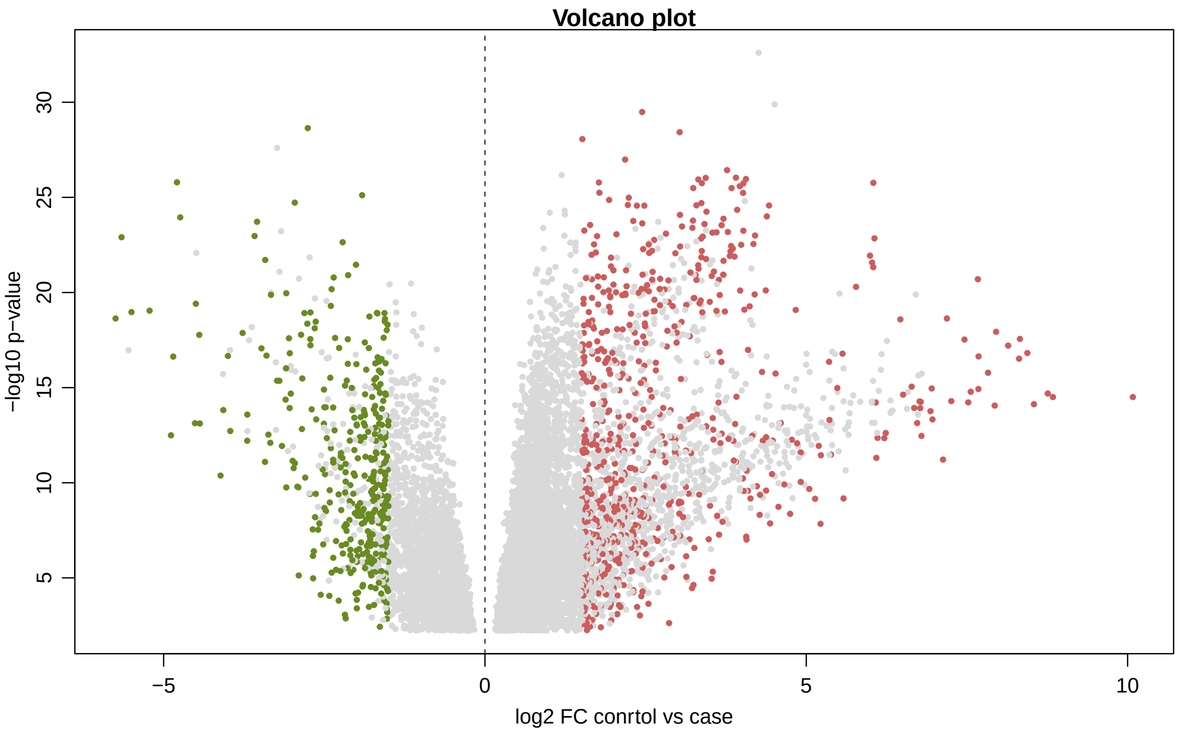
The selected data have been then normalized via TMM (trimmed mean of M values)

```r
edge_c <- DGEList(counts = filter_counts_df, group = c_anno_df$condition, samples = c_anno_df,
    genes = filter_anno_df)

# normalization with Redge [TMM method]
edge_n <- calcNormFactors(edge_c, method = "TMM")
# create normalize expression values table
CPM_table <- as.data.frame(round(cpm(edge_n), 2))
```

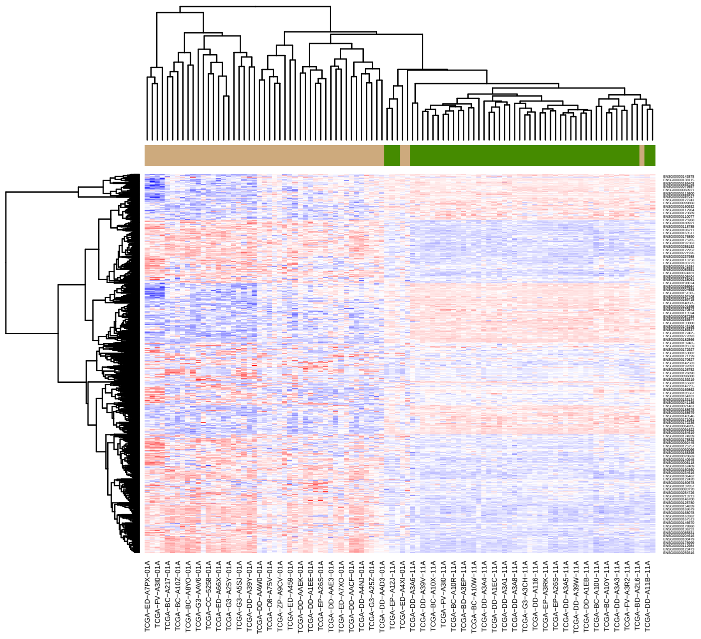
Then trough edgeR we performed a differential expression analysis, specifyng a cut-off that included a p-value of 0.01 and logFC ratio grater than 1.5 for up-regulated and lower than -1.5 for down-regulated, and a logCPM greater than 1.

```r
# design creation
design <- model.matrix(~0 + group, data = edge_n$samples)
colnames(design) <- levels(edge_n$samples$group)
rownames(design) <- edge_n$samples$sample

# calculate dispersion and fit with edgeR (necessary for differential
# expression analysis)
edge_d <- estimateDisp(edge_n, design)
edge_f <- glmQLFit(edge_d, design)

# definition of the contrast (conditions to be compared)
contrast <- makeContrasts("case-control", levels = design)

# fit the model with generalized linear models
edge <- glmQLFTest(edge_f, contrast = contrast)
DEGs <- as.data.frame(topTags(edge, n = 20000, p.value = 0.01, sort.by = "logFC"))
DEGs$class <- "="

# add labels for up (+) and down (-) regulations for certain parameters for
# -valued and FDR
DEGs$class[which(DEGs$logCPM > 1 & DEGs$logFC > 1.5)] = "+"
DEGs$class[which(DEGs$logCPM > 1 & DEGs$logFC < -1.5)] = "-"
DEGs <- DEGs[order(DEGs$logFC, decreasing = T), ]
```

Through a Vulcano plot all the up and down regulated genes can be visualized (red = up-regulated, green = down-regulated)



Thank to the pseudocount introduced earlier in the heatmap we can clearly observe a distinction between the two groups and both based on the clustering and with respect with the up and down regulated genes blocks.



Task 4

Here we obtained additional information from ensembl database and we performed cleaning and annotation on them.

```
convert <- getBM(attributes = c("ensembl_gene_id", "entrezgene_id"), filters = c("ensembl_gene_id"),
    values = DEGs$ensembl_gene_id, mart = ensembl)

DEGs <- merge(DEGs, convert, by = "ensembl_gene_id")
anyNA(DEGs)
```

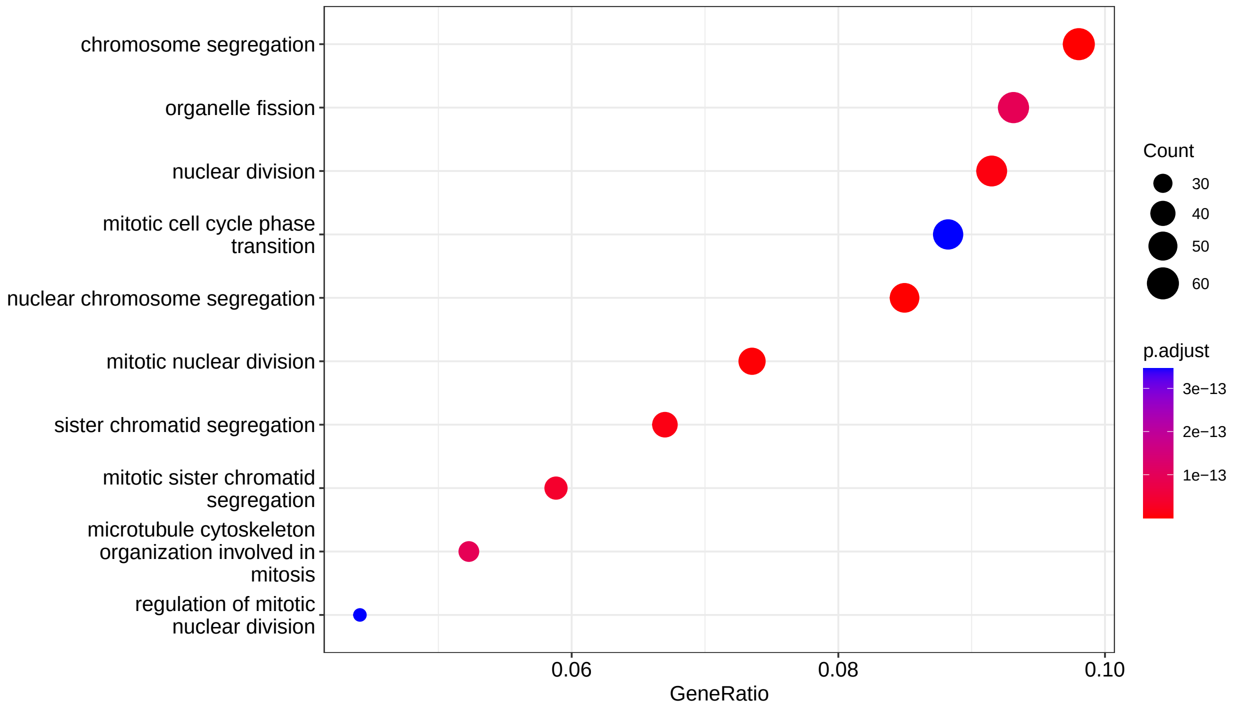
```
## [1] TRUE
```

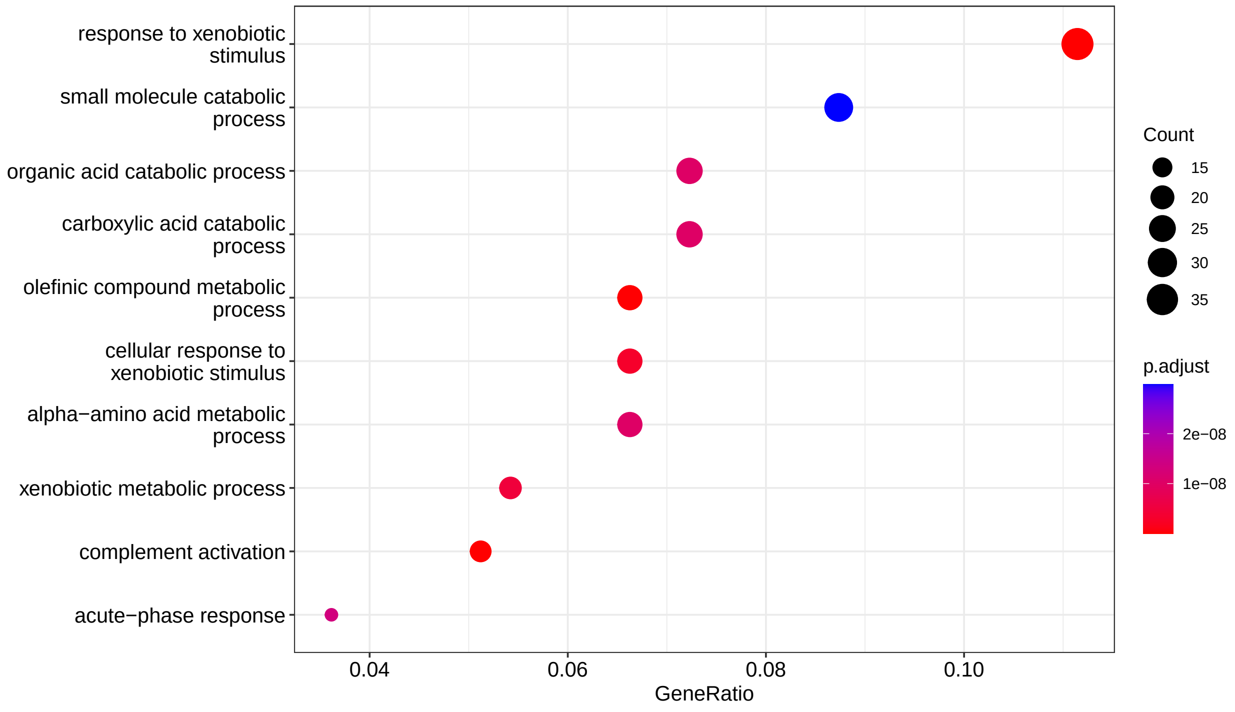
```
# removing NAs
DEGs <- DEGs[which(!is.na(DEGs$entrezgene_id)), ]
# removing duplicates
DEGs <- DEGs[-which(duplicated(DEGs$entrezgene_id)), ]

## Splitting up and down regulated ones

up_DEGs <- DEGs[DEGs$class == "+", ]
down_DEGs <- DEGs[DEGs$class == "-", ]
```

Now that the deregulated genes have been selected, enrichment analysis is performed by GSEA, obtaining additional information regarding biological processes (BP).

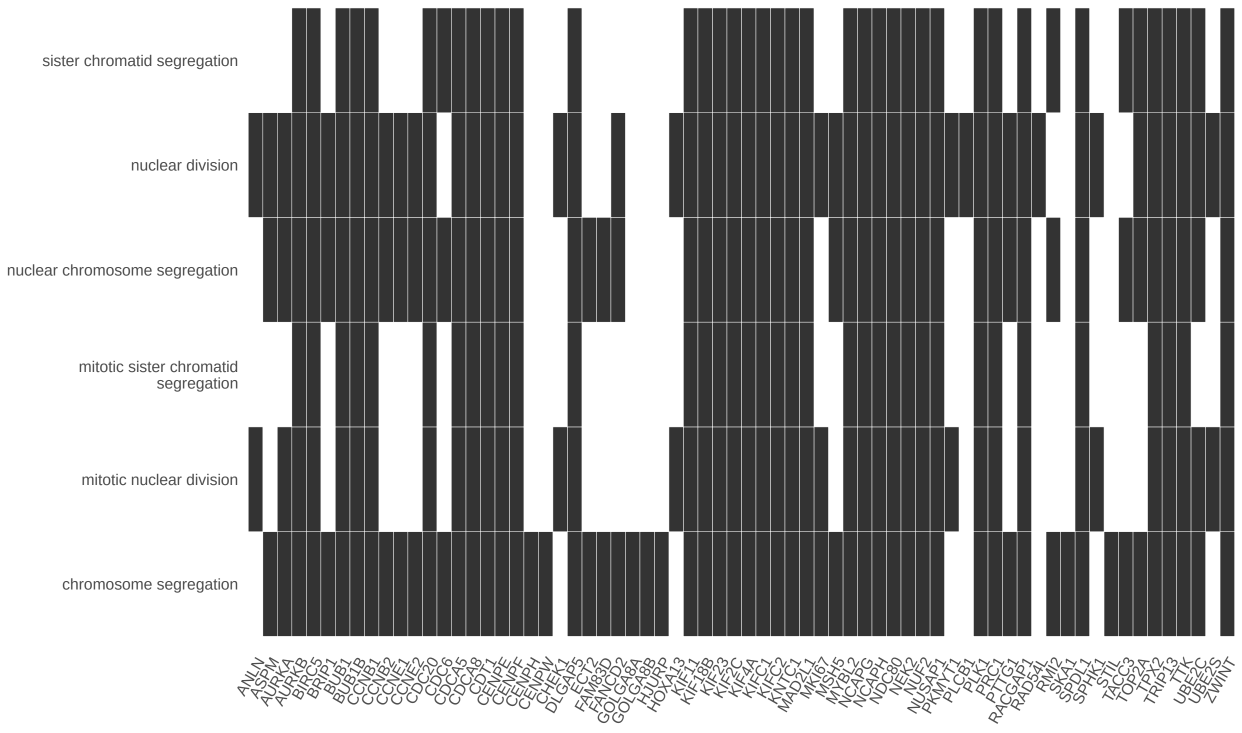
```
# Gene ontology analysis BP
up_ego_BP <- enrichGO(gene = up_DEGs$external_gene_name, OrgDb = org.Hs.eg.db, keyType = "SYMBOL",
    ont = "BP", pAdjustMethod = "BH", pvalueCutoff = 0.05, qvalueCutoff = 0.05)

down_ego_BP <- enrichGO(gene = down_DEGs$external_gene_name, OrgDb = org.Hs.eg.db,
    keyType = "SYMBOL", ont = "BP", pAdjustMethod = "BH", pvalueCutoff = 0.05, qvalueCutoff = 0.05)
```

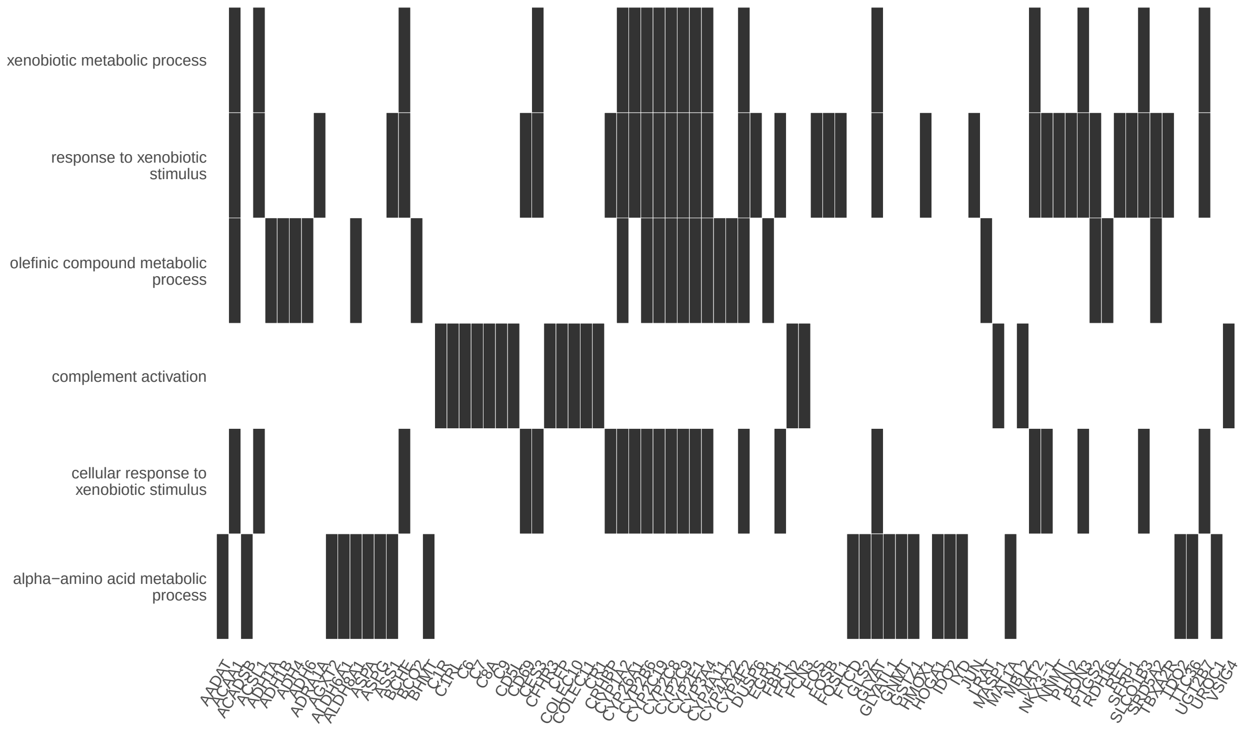
From the resulting dot plots we can observe an up regulations in those pathways that concern cell proliferation as expected in a cancerous tissue. While for the down regulated pop up mainly pathway regarding catabolic processes



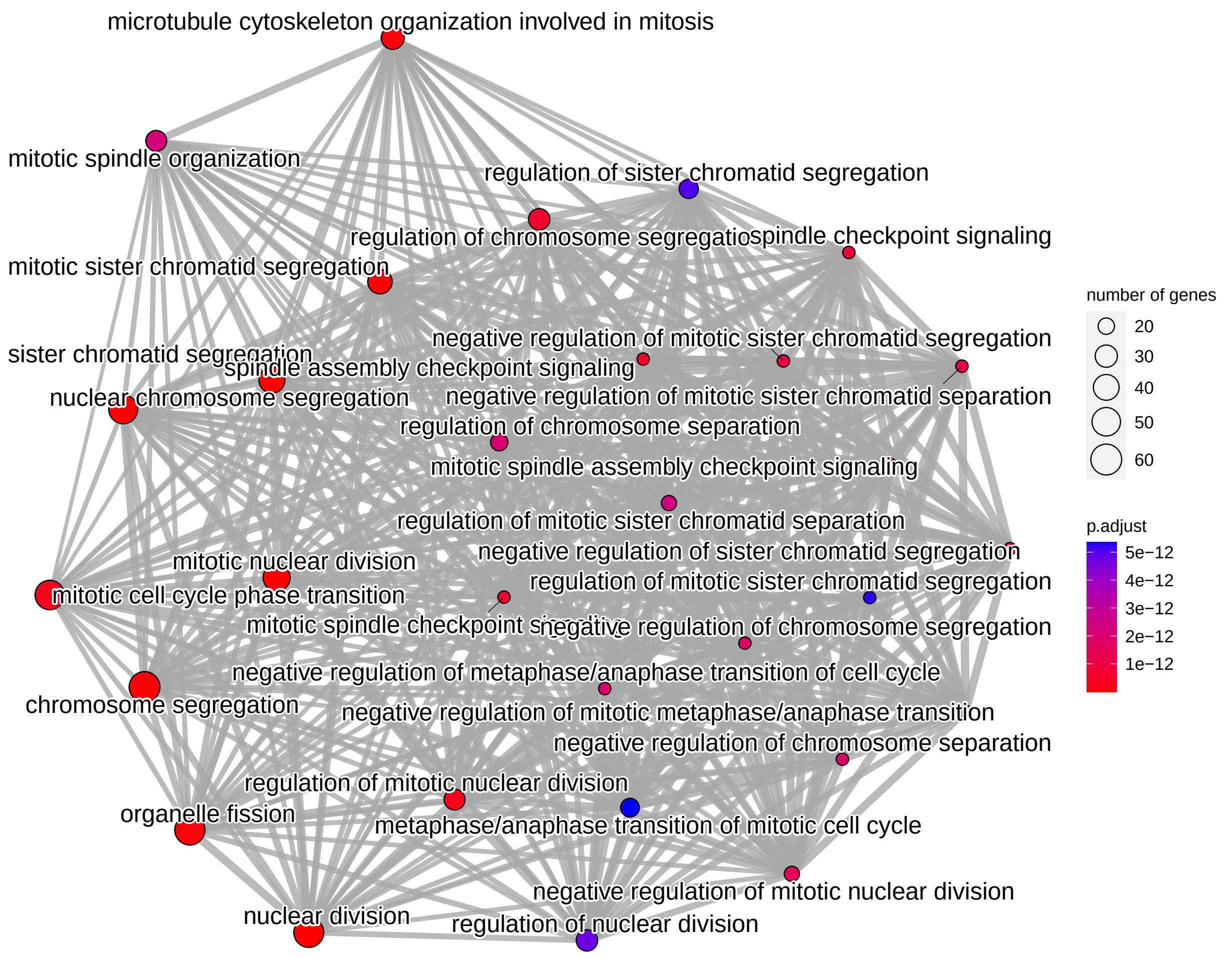


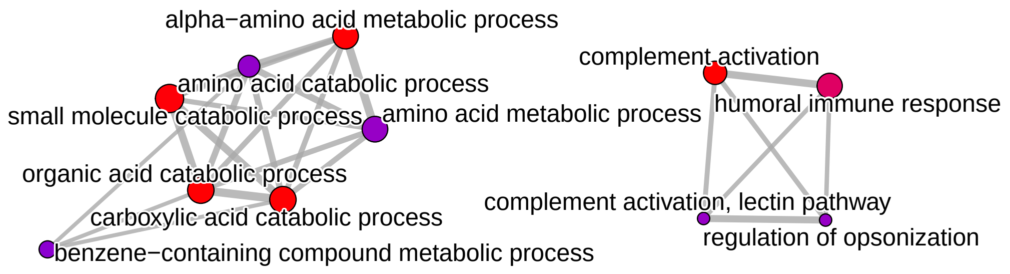
Thanks to the heat plots we can highlight those genes relatives to the pathways depicted above.

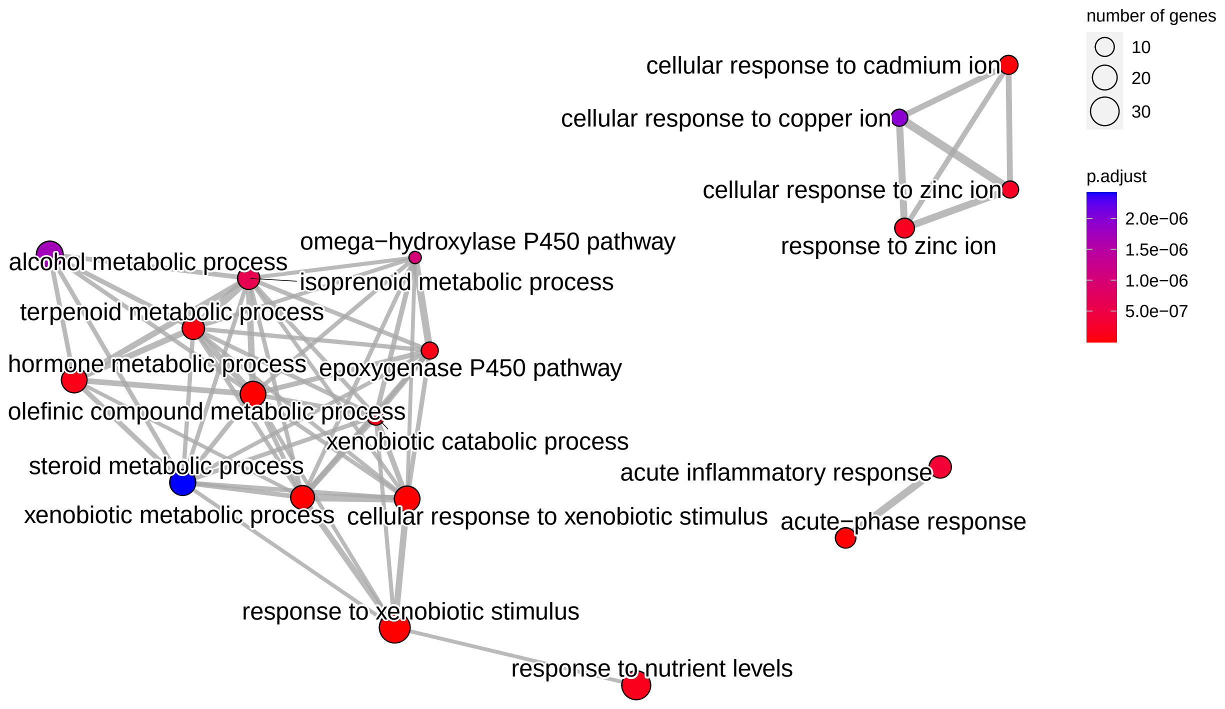




In the emaplot the pathways are connected due to the rate share of genes. Regarding the up regulated genes, we can see one big cluster due to the similar biological process. As for the down ones, we can see smaller separated clusters due to the lower connection of the metabolites produced.


microtubule cytoskeleton organization involved in mitosis
mitotic spindle organization
regulation of sister chromatid segregation
regulation of chromosome segregatio
spindle checkpoint signaling
mitotic sister chromatid segregation
negative regulation of mitotic sister chromatid segregation
sister chromatid segregation
spindle assembly checkpoint signaling
nuclear chromosome segregation
negative regulation of mitotic sister chromatid separation
regulation of chromosome separation
mitotic spindle assembly checkpoint signaling
regulation of mitotic sister chromatid separation
mitotic nuclear division
negative regulation of sister chromatid segregation
mitotic cell cycle phase transition
regulation of mitotic sister chromatid segregation
mitotic spindle checkpoint si
negative regulation of chromosome segregation
negative regulation of metaphase/anaphase transition of cell cycle
chromosome segregation
negative regulation of mitotic metaphase/anaphase transition
negative regulation of chromosome separation
regulation of mitotic nuclear division
organelle fission
metaphase/anaphase transition of mitotic cell cycle
negative regulation of mitotic nuclear division
nuclear division
regulation of nuclear division
number of genes
20
30
40
50
60
p.adjust
5e−12
4e−12
3e−12
2e−12
1e−12

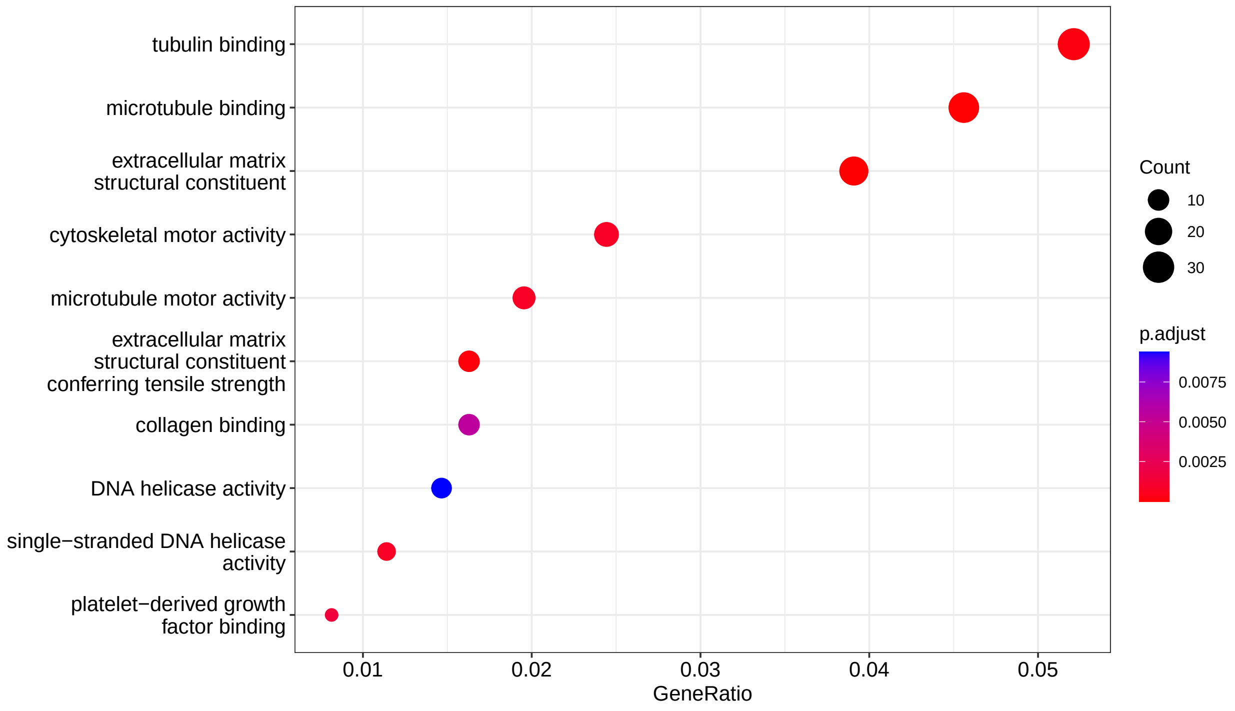




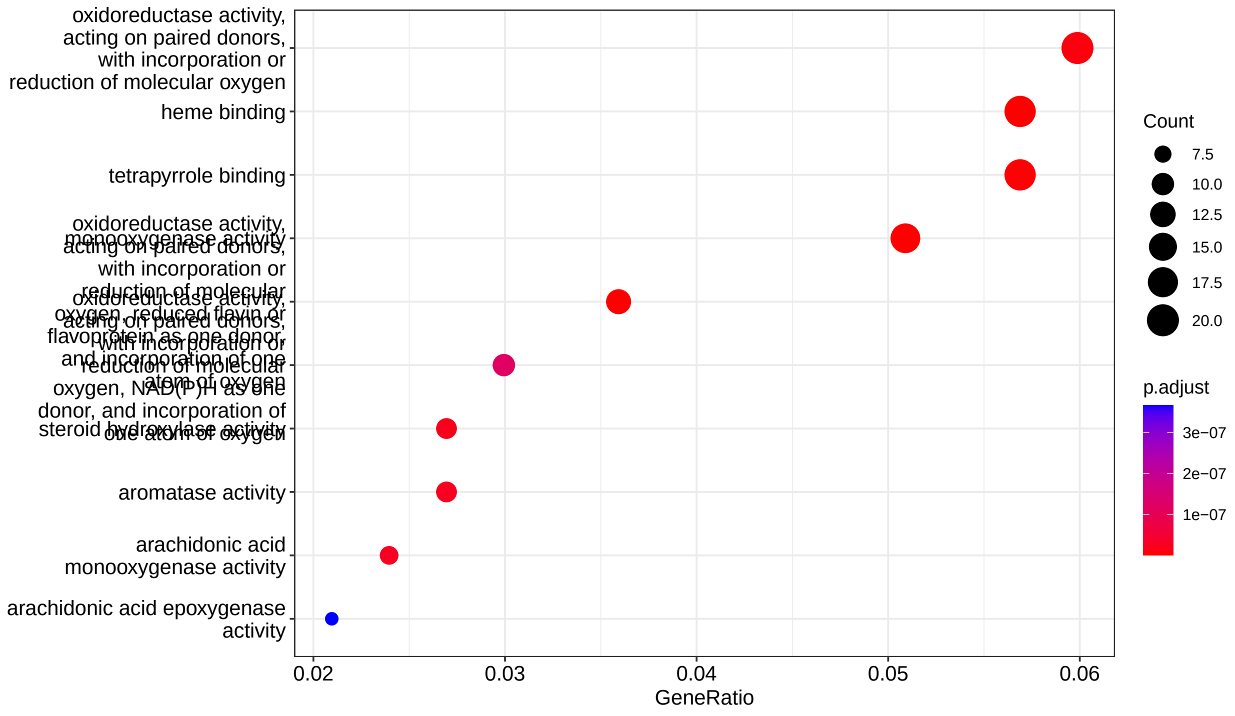
This second enrichment analysis was done take in account the molecular function (MF).

```
up_ego_MF <- enrichGO(gene = up_DEGs$external_gene_name, OrgDb = org.Hs.eg.db, keyType = "SYMBOL",
    ont = "MF", pAdjustMethod = "BH", pvalueCutoff = 0.05, qvalueCutoff = 0.05)

down_ego_MF <- enrichGO(gene = down_DEGs$external_gene_name, OrgDb = org.Hs.eg.db,
    keyType = "SYMBOL", ont = "MF", pAdjustMethod = "BH", pvalueCutoff = 0.05, qvalueCutoff = 0.05)
```

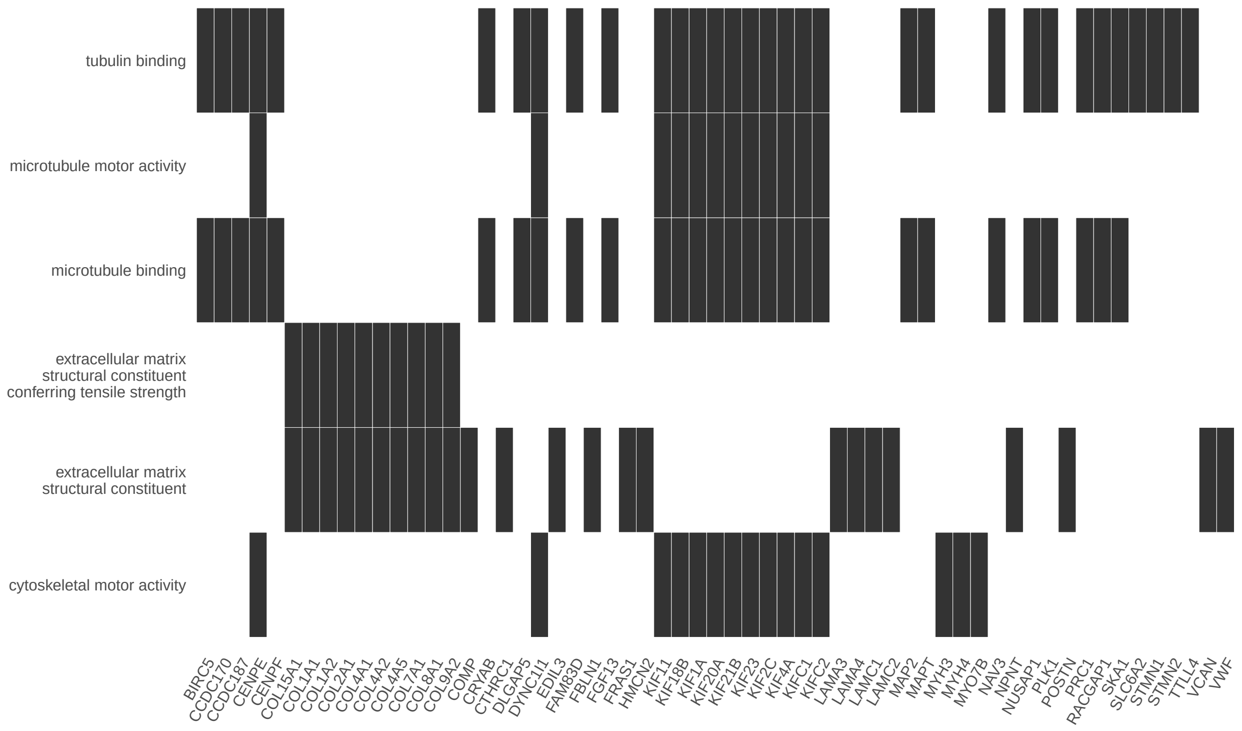
Looking at the plots regarding the molecular function there is a connection with the biological processes mentioned before.

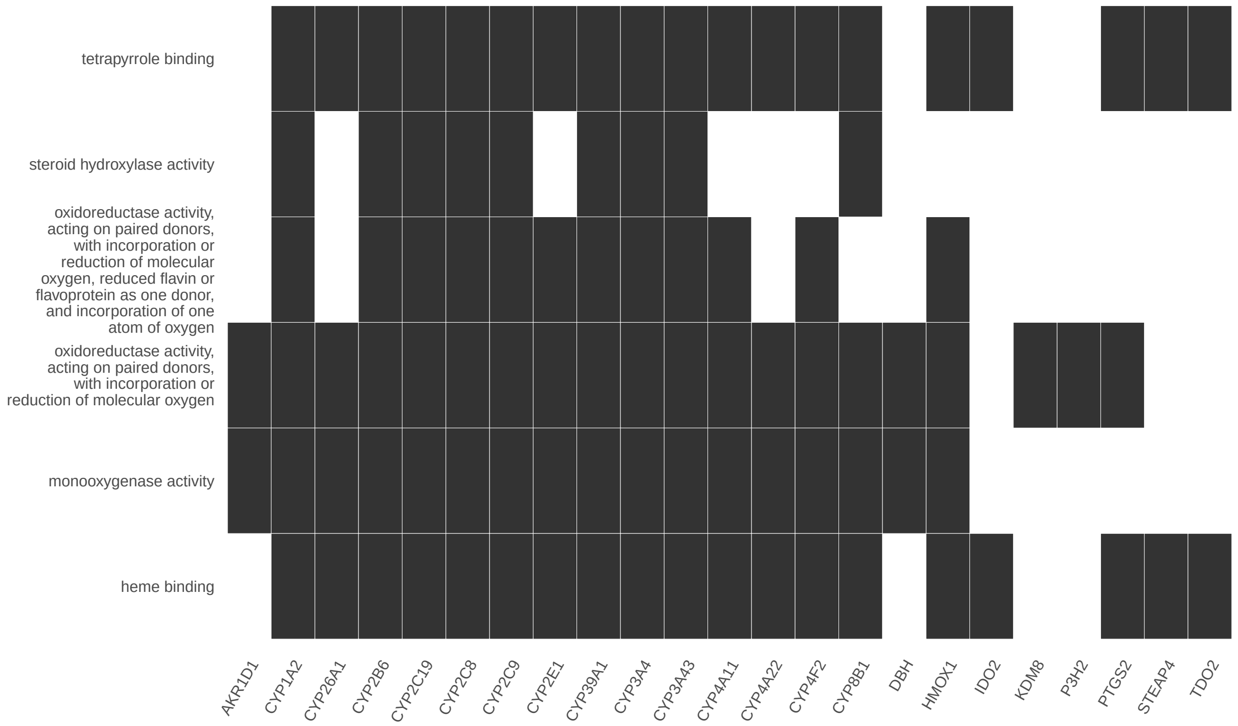


tubulin binding
microtubule binding
extracellular matrix structural constituent
cytoskeletal motor activity
microtubule motor activity
extracellular matrix structural constituent conferring tensile strength
collagen binding
DNA helicase activity
single−stranded DNA helicase activity
platelet−derived growth factor binding
0.01
0.02
0.03
0.04
0.05
GeneRatio
Count
10
20
30
p.adjust
0.0075
0.0050
0.0025

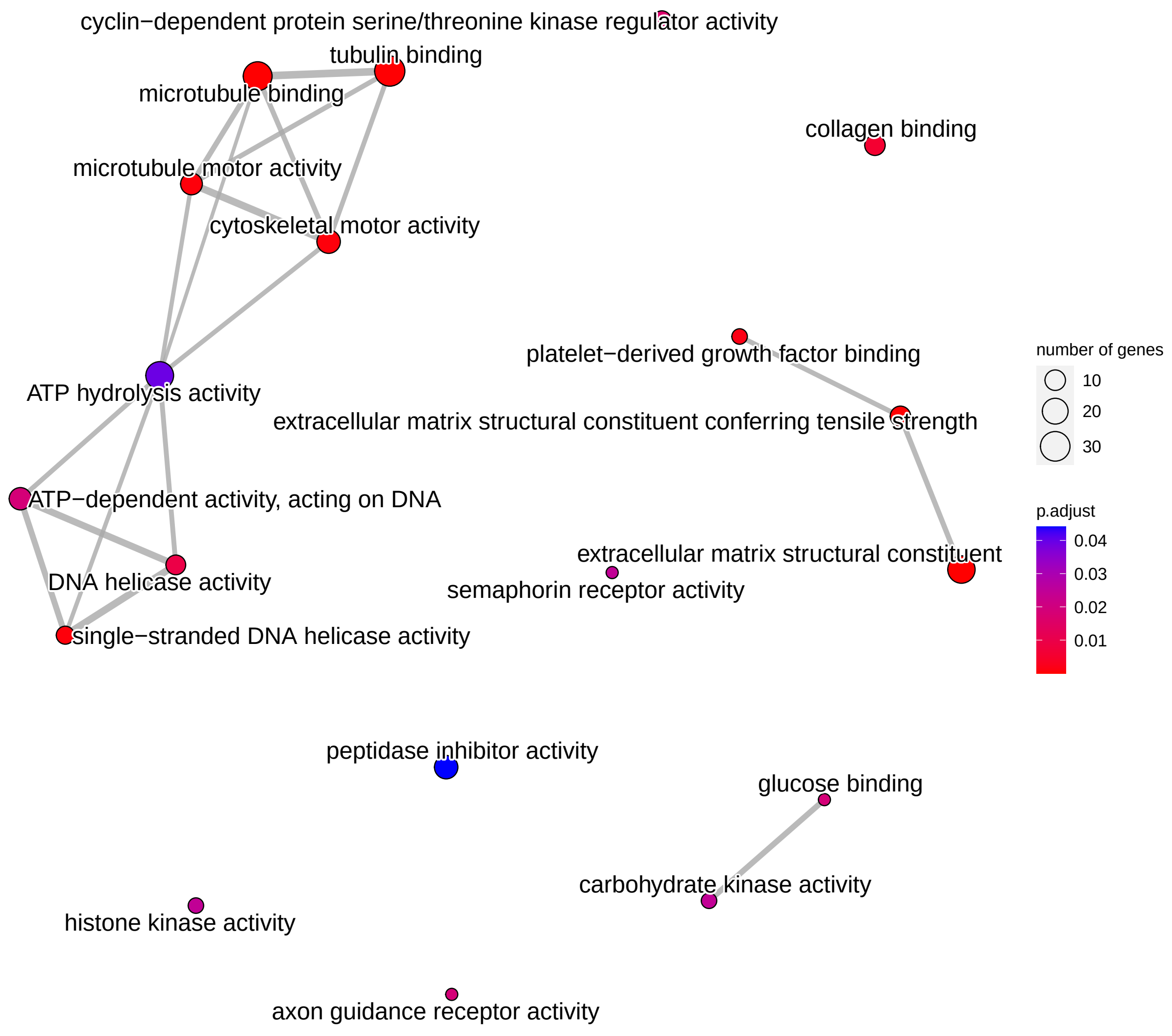


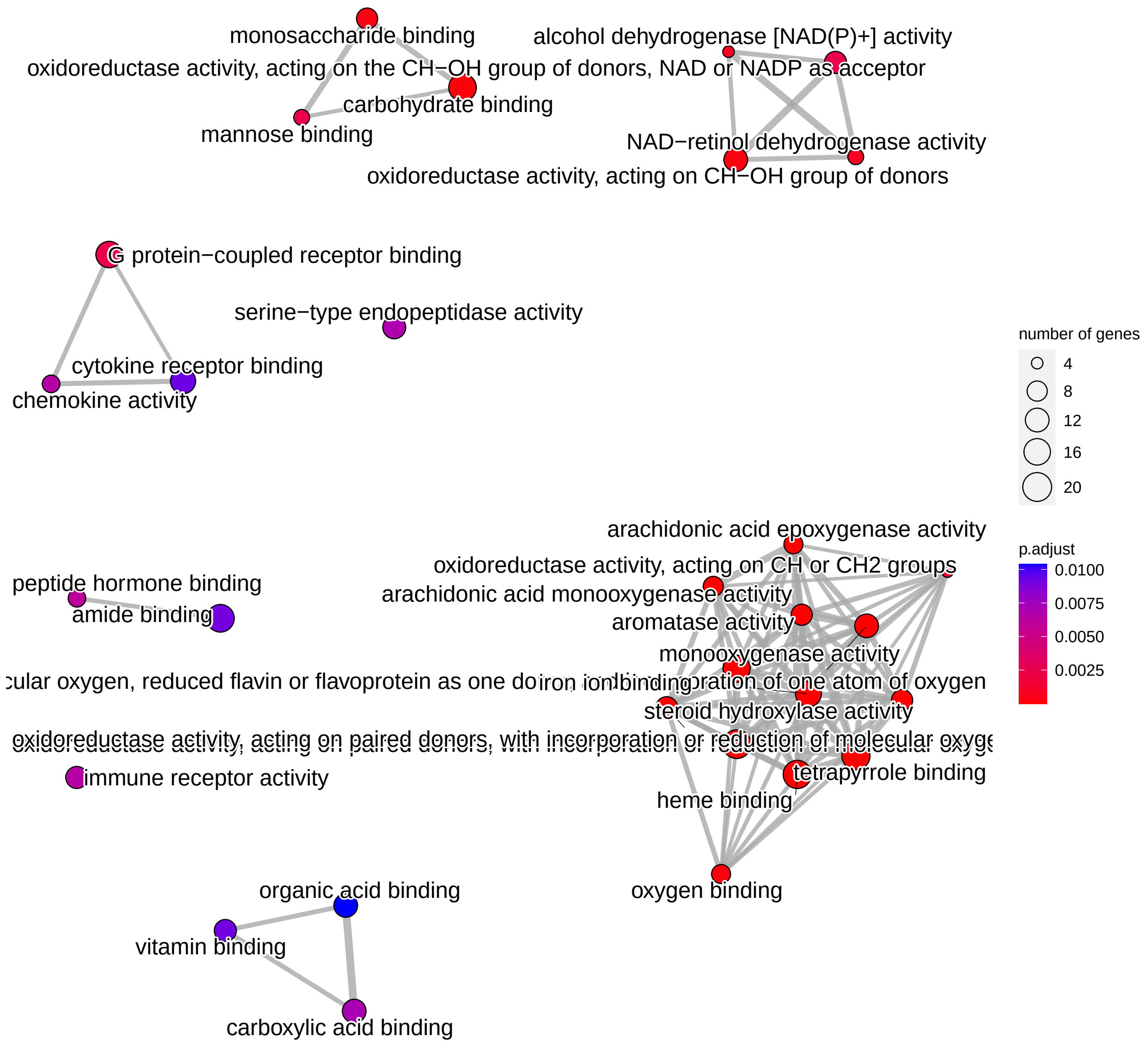
oxidoreductase activity, acting on paired donors, with incorporation or reduction of molecular oxygen
heme binding
tetrapyrrole binding
monooxygenase activity
steroid hydroxylase activity
aromatase activity
arachidonic acid monooxygenase activity
arachidonic acid epoxygenase activity
0.02
0.03
0.04
0.05
0.06
GeneRatio
Count
7.5
10.0
12.5
15.0
17.5
20.0
p.adjust
3e−07
2e−07
1e−07





From the emaplots we can see those activities which are up or down regulated of the biological processes visualized before


cyclin−dependent protein serine/threonine kinase regulator activity
tubulin binding
microtubule binding
collagen binding
microtubule motor activity
cytoskeletal motor activity
platelet−derived growth factor binding
ATP hydrolysis activity
extracellular matrix structural constituent conferring tensile strength
ATP−dependent activity, acting on DNA
extracellular matrix structural constituent
DNA helicase activity
semaphorin receptor activity
single−stranded DNA helicase activity
peptidase inhibitor activity
glucose binding
carbohydrate kinase activity
histone kinase activity
axon guidance receptor activity
number of genes
10
20
30
p.adjust
0.04
0.03
0.02
0.01



KEGG is another database which stores information about genes, pathways, enzymes and metabolites; the following commands allow to extract information give the dysregulated gene list about the specific pathways.

```
up_eKEGG <- enrichKEGG(gene = up_DEGs$entrezgene_id, organism = "human", pvalueCutoff = 0.05,
    qvalueCutoff = 0.1)

down_eKEGG <- enrichKEGG(gene = down_DEGs$entrezgene_id, organism = "human", pvalueCutoff = 0.05,
    qvalueCutoff = 0.1)
```

Task 5

From the up regulated genes we select the most significant pathway (cell cycle) that are reported in gene-pathway maps.

```
FC <- up_DEGs$logFC
names(FC) <- up_DEGs$entrezgene_id
pathview(gene.data = FC,
         pathway.id = "hsa04110",  # hsa04512 (ECM receptor interaction)
         species = "human")
```

Task 6

From now on the analysis focuses only on the up regulated genes. For each of those we evaluated the enrichment scores of the various transcription factors (TFs), selected via a window of 500 nucleotides upstream for each gene, using a position weigthed matrix (PWM). For simplicity and to save computational time, we share and load our results through a .RData file (**"enriched_transcription.RData"**).

```
sequences <- getSequence(id = up_DEGs$ensembl_gene_id, mart = ensembl, type = "ensembl_gene_id",
    upstream = 500, seqType = "gene_flank")

seq <- lapply(sequences[, 1], function(x) DNAString(x))

# data('PWMLogn.hg19.MotifDb.Hsap') enrichedTF <- motifEnrichment(seq,
# PWMLogn.hg19.MotifDb.Hsap, score = 'affinity') save(enrichedTF, file =
# 'enriched_transcription.RData')
load("enriched_transcription.RData")

report <- groupReport(enrichedTF)
# we have selected only those above the 0.975 percentage of the data
report <- report[report$p.value < 0.025, ]
report[1:5, ]
```

```
## An object of class 'MotifEnrichmentReport':
##   rank target         id raw.score      p.value top.motif.prop
## 1    1  PGAM2      PGAM2  8.020996 1.328913e-60      0.1947853
## 2    2  CEBPB M4556_1.02  2.430586 8.106795e-59      0.1917178
## 3    3  TFAP4 M2944_1.02  3.627887 9.705952e-59      0.1901840
## 4    4  PDS5A    SCC-112  2.624652 1.890000e-56      0.1871166
## 5    5   MAFK M4573_1.02  2.543173 2.060761e-56      0.1871166
```

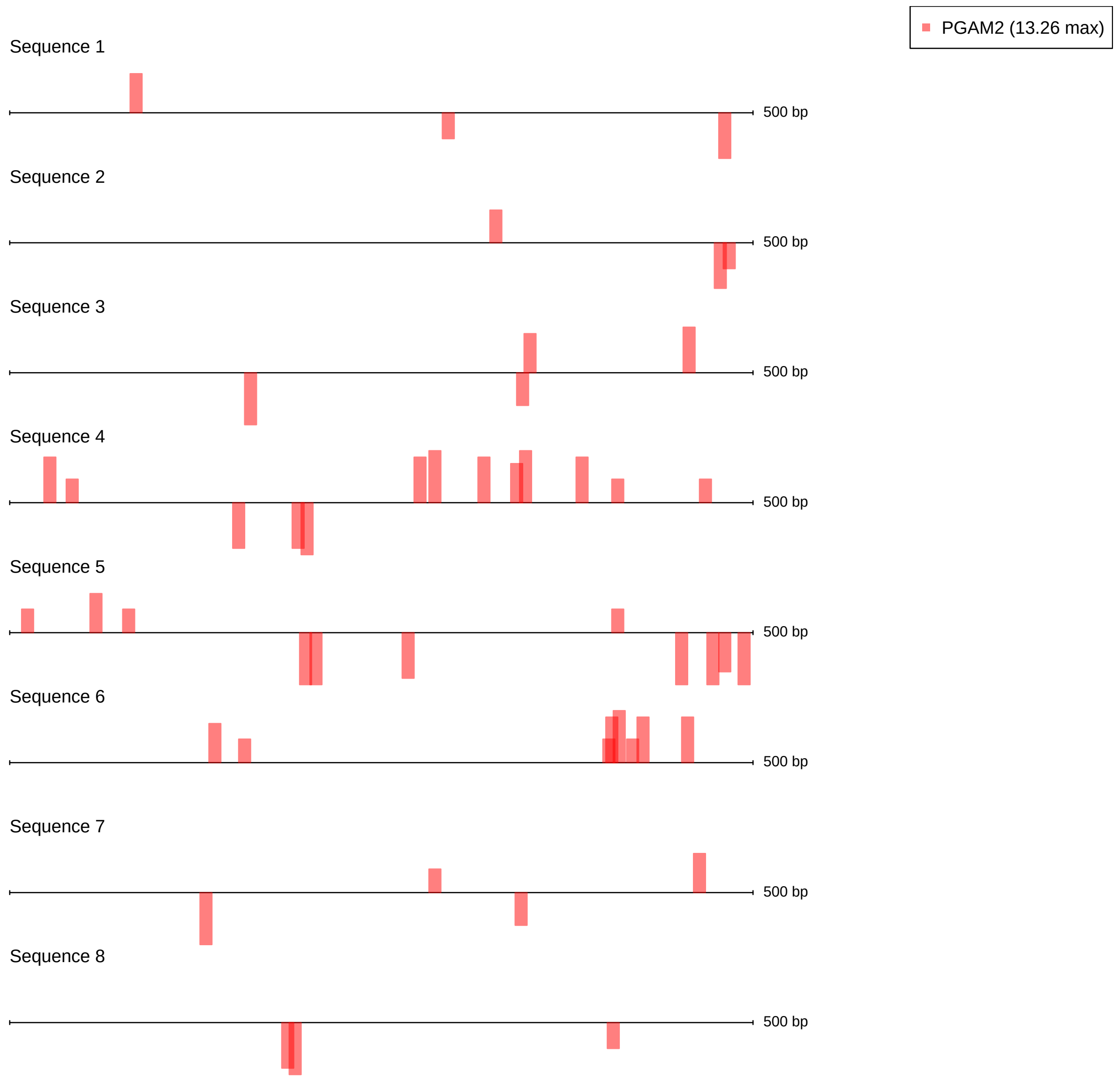
Task 7

Starting from the enriched TFs of Task 6 we selected the gene "PGAM2" as it represent the most significant result. We computed the empirical distributions of the scores for all PWMs in MotifDb against this TF and computed the distribution specific threshold with a cutoff of 99.75%.

```
mdb.human <- subset(MotifDb, organism == "Hsapiens" & geneSymbol == "PGAM2")
PWM <- toPWM(as.list(mdb.human))
```

```
names(PWM) <- sapply(names(PWM), function(x) strsplit(x, "-")[[1]][3])
scores <- motifScores(sequences = seq, PWM, raw.scores = T)

ecdf <- motifEcdf(PWM, organism = "hg19", quick = TRUE)
threshold <- log2(quantile(ecdf$PGAM2, 0.9975))
```

For graphical reasons, we are only plotting the first 10 sequences from the results.



Task 8

Starting from the list of the TFs we got before, we analyze which of the up-regulated genes have actually a binding site in their upstream sequence (via pattern matching) for the TFs. We only considered sites with a threshold above of the one calculated above. Once again for simplicity and to save computational time, we share and load our results through a .RData file (**"Scores_table.RData"**).

```
tfs <- report$target
# Some TFs gave us problem during the analysis (all the ones with the terms
# UW.M) because they cannot be found in the MotifDb, so we had to cut them out.
tfs <- tfs[!grepl("^UW.M", tfs)]
scores <- data.frame(c(1:length(seq)))


# for (i in (1:length(tfs))){ tfs_motif <- subset(MotifDb, organism ==
# 'Hsapiens' & geneSymbol == tfs[i]) PWM <- toPWM(as.list(tfs_motif))
# names(PWM) <- sapply(names(PWM),function(x) strsplit(x,'-')[[1]][3]) ecdf <-
# motifEcdf(PWM, organism = 'hg19', quick = T) threshold <-
# log2(quantile(ecdf[[tfs[i]]], 0.9975)) name <- tfs[i] sco <-
# motifScores(sequences = seq , PWM, cutoff = threshold) sco <-
# as.data.frame(apply(sco, 1, sum)) colnames(sco) <- name scores <-
# cbind(scores, sco) } save(scores, file = 'Scores_table.RData')


load("Scores_table.RData")
# The first column is the index
scores <- as.matrix(scores[, 2:length(scores)])
head(scores[, 1:10])
```

```
##      PGAM2 CEBPB TFAP4 PDS5A MAFK ZMAT2 JUN SP2 NNT CREB1
## [1,]     2    66    34     0   55     5 110  11   1    44
## [2,]     4    30    24     0   84     7  84  63   3    23
## [3,]     3    22    20     0   33     4  43   4   2    19
## [4,]    13     5    54     0   23    14  15 151  20    15
## [5,]     4    36    88     0   50    10  54  45   2    25
## [6,]     8     1    83     0   45    11  22 101  14    13
```

Given the number of regions for each upstream sequence of the genes, we perform a filtering based on the following criteria: first, if a TF has null scores for all the sequences we avoid considering it. Second, if the distribution of the TFs' values, for each sequence, has a higher median then a reference (calculated on a random sampling, dist of scores) we set a flag equal to True, False otherwise. At the end of the process we will keep only those sequences which has a considerable increase in TFs' scores and so we 'll have those up regulated genes with a greater number of TFs.

```
# filter out TFs that do not have any score greater than 0 for any of the
# sequences
columns <- c()
for (col in (1:ncol(scores))) {
    if (sum(scores[, col]) < 1) {
        columns <- append(columns, col)
    }
}
# removing the zero columns
```

```r
if (length(columns) > 0) {
    scores <- scores[, -columns]
}

# initialized a treshold for the cut, in this case the median, but can be
# increased (has been used median instead of mean because more stable).  dist
# is a sub set of all the scores to confront the line (sequence) distrbution of
# TFs
threshold <- 0.5  # 50%
dist <- c()

for (i in (1:ncol(scores))) {
    point <- sample(scores[, i], 150, replace = F)
    dist <- append(dist, point)
}

q <- quantile(dist, probs = c(threshold))
# the loop return a boolean to filter each line (gene up-strem sequence), if
# the distribution of the row has a higher value of the threshold the row is
# kept.

result <- logical(nrow(scores))
for (i in 1:nrow(scores)) {
    #
    if (quantile(scores[i, ], probs = c(threshold)) < q) {
        result[i] <- FALSE
    } else {
        result[i] <- TRUE
    }
}
summary(result)
```

```
##    Mode   FALSE    TRUE
## logical     279     373
```

Task 9

The online database **STRING** was used to find protein protein interactions (PPI) among the differentially expressed genes and exported the network in TSV format for downstream analysis.

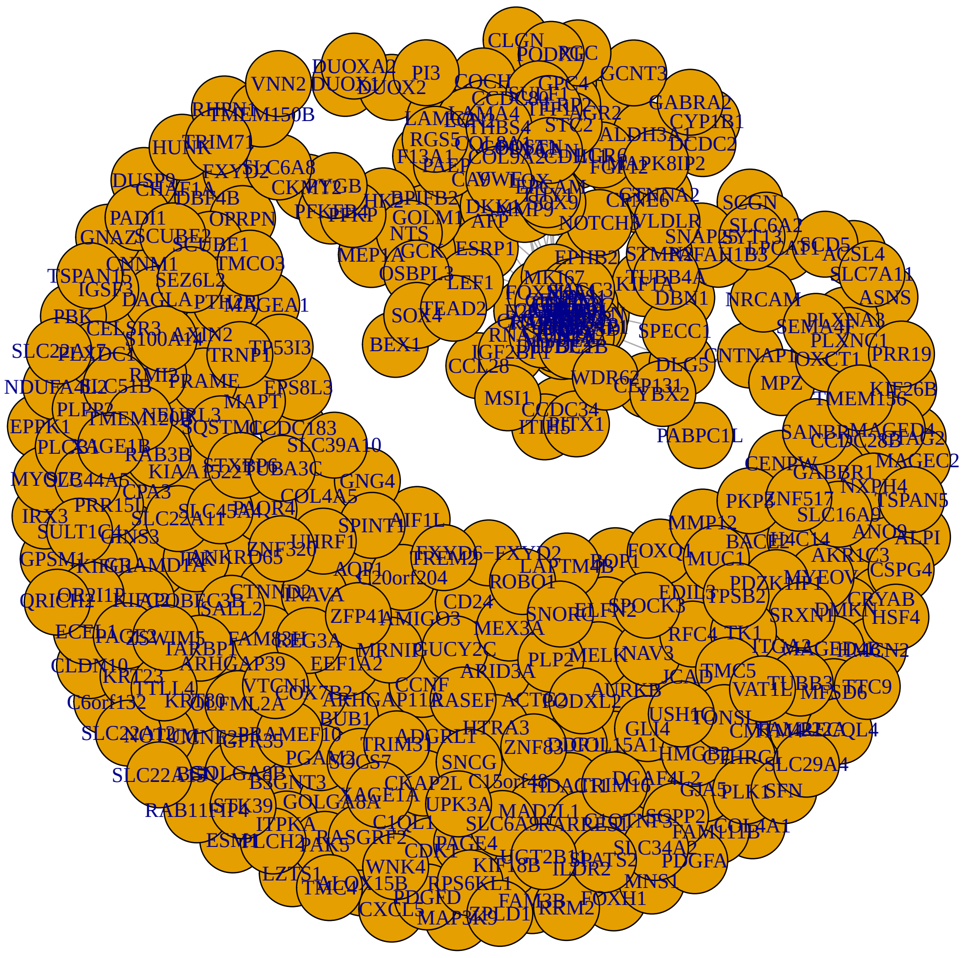
```r
top_enrich <- sequences$ensembl_gene_id[result]

up_names <- up_DEGs$external_gene_name[which(top_enrich %in% up_DEGs$ensembl_gene_id)]

## the command below produce a txt file with the gene of interest that have
## been used on STRING to retrive the TSV file.

# lapply(up_names, write, my_dir, append=TRUE, ncolumns=1000)

links <- read.delim("string_interactions_short.tsv")
```

Task 10

Via BioMart and the function getBM we extracted information on the specified attributes to then build a network. An important first step was the evaluation of the presence or absence of specific genes both in the query and nodes to keep only those in common.

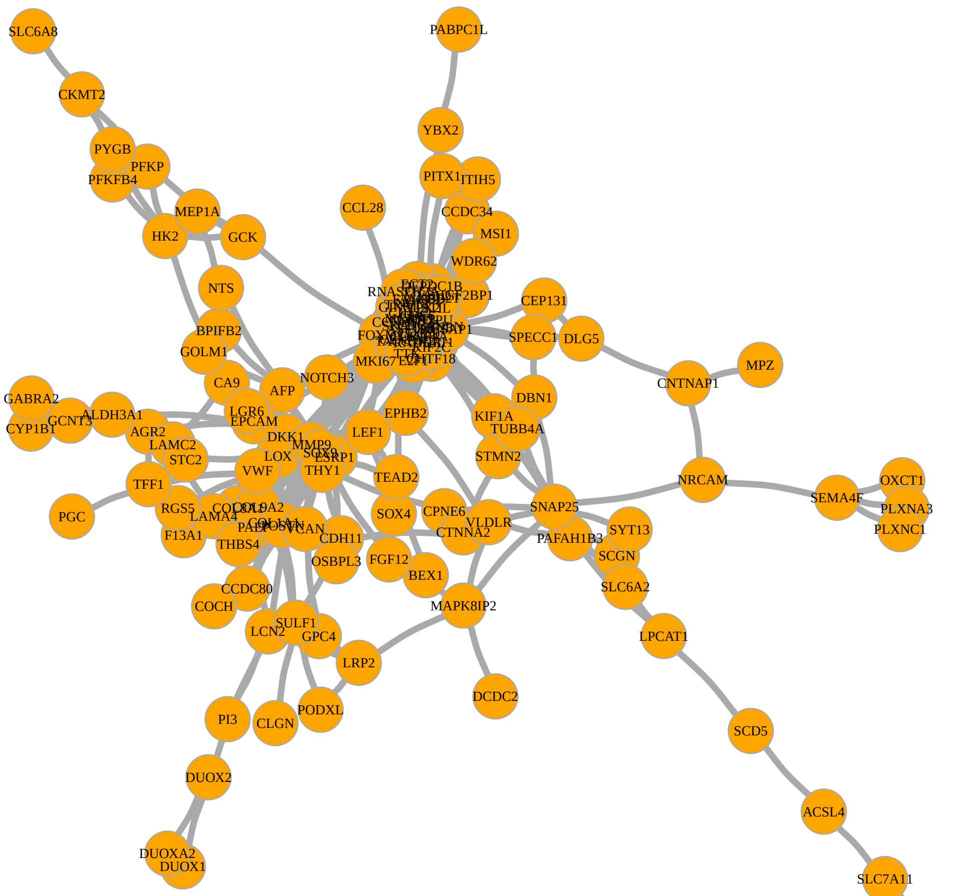
```r
query <- getBM(attributes = c("external_gene_name", "ensembl_gene_id", "description",
    "gene_biotype", "start_position", "end_position", "chromosome_name", "strand"),
    filters = c("ensembl_gene_id"), values = top_enrich, mart = ensembl)

query <- unique(query[, c(1, 3:6)]) %>%
    arrange(external_gene_name)
index <- links$node2 %in% query$external_gene_name
links <- links[index, ]
index <- links$X.node1 %in% query$external_gene_name
links <- links[index, ]
```

Here we can see a first and crowded representation of the full network.

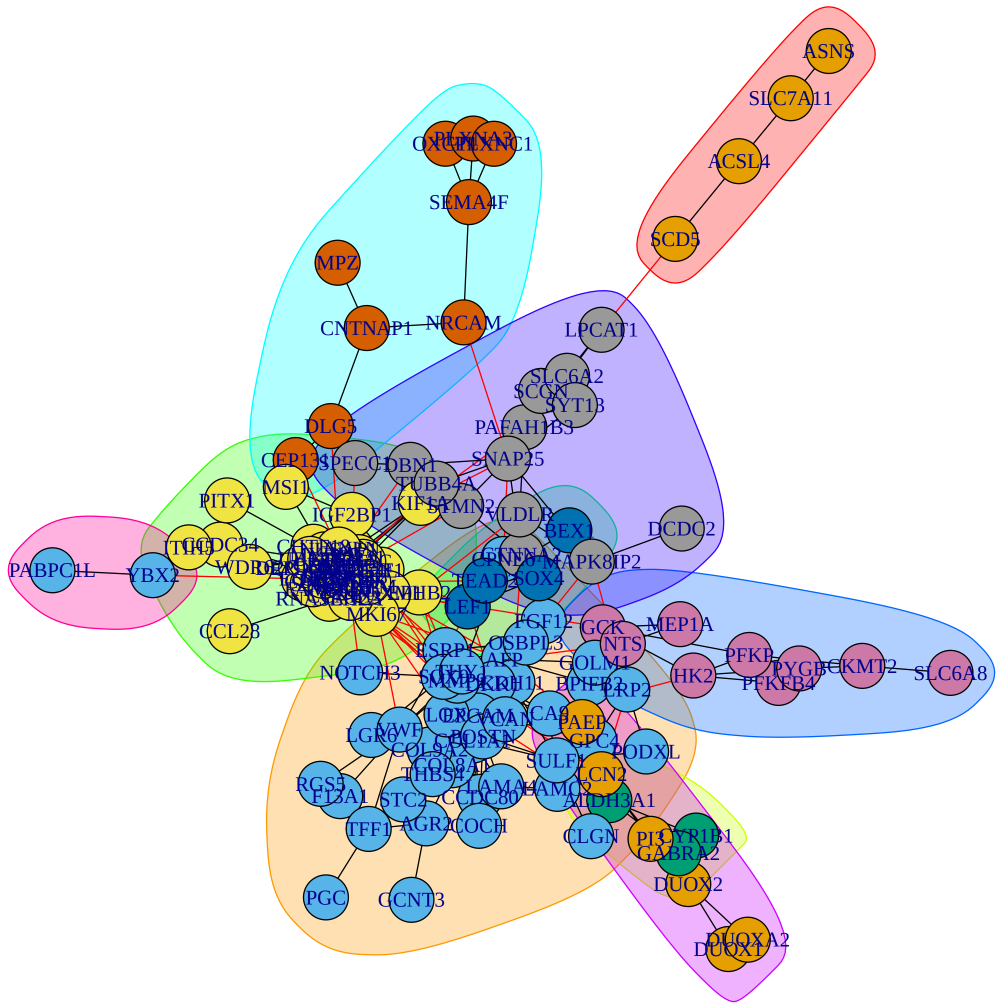


From the previous network the most connected nodes have been extracted and visualized.

```
## [1] 137
```



Taking in consideration only this extracted nodes a clustering have been performed using the edges as a reference to build the clusters



```
## Cluster 1 :  ACSL4, ASNS, SCD5, SLC7A11
## Cluster 2 :  AFP, BPIFB2, CA9, CCDC80, CDH11, CLGN, COCH, COL1A1, COL8A1, COL9A2, CPNE6, DKK1, EPCAM
## Cluster 3 :  AGR2, GCNT3, PGC, STC2, TFF1
## Cluster 4 :  ALDH3A1, CYP1B1, GABRA2
## Cluster 5 :  ANLN, ASF1B, CCDC34, CCL28, CCNB1, CDC20, CDC45, CDC6, CDCA7, CDCA8, CENPM, CENPU, CHEK
## Cluster 6 :  BEX1, LEF1, SOX4, TEAD2
## Cluster 7 :  CEP131, CNTNAP1, DLG5, MPZ, NRCAM, OXCT1, PLXNA3, PLXNC1, SEMA4F
## Cluster 8 :  CKMT2, GCK, HK2, MEP1A, NTS, PFKFB4, PFKP, PYGB, SLC6A8
## Cluster 9 :  CTNNA2, DBN1, DCDC2, LPCAT1, MAPK8IP2, PAFAH1B3, SCGN, SLC6A2, SNAP25, SPECC1, STMN2, S
## Cluster 10 :  DUOX1, DUOX2, DUOXA2, LCN2, PAEP, PI3
## Cluster 11 :  PABPC1L, YBX2
```

From the graph we analyzed the two most dense clusters (clusters 2 and 5): 1. cluster 5 present proteins that are highly related to the cell cycle confirming the pathway resulting from KEGG analysis (cell cycle), coded by the id *hsa04110* 2. cluster 2 shows proteins related to the cell shape and motility, once again verified by the second most representative KEGG pathway (ECM receptor interaction) identified by the code *hsa04512*

As expect the conclusions lead to the identification of up-regulated genes and related pathway that are easily connected to a tissue in a cancerous state, both cell cycle and deregulated of the ECM play a key role in the advance of a tumor.