# Introduction to DGE

View on GitHub

Approximate time: 120 minutes

## Learning Objectives:

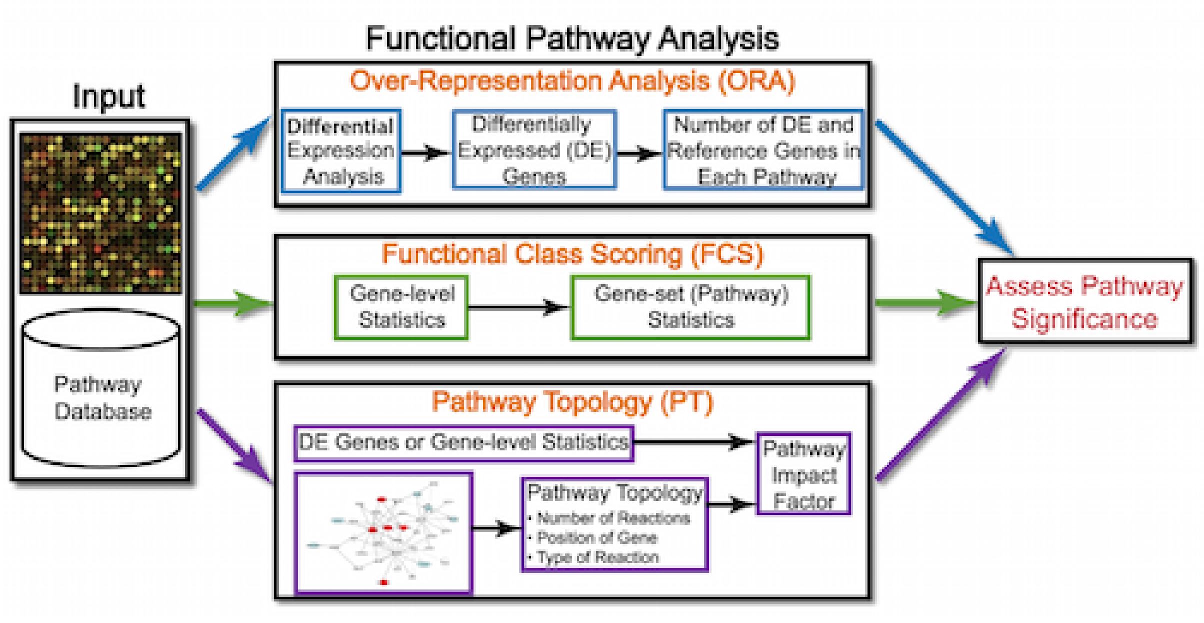
- Determine how functions are attributed to genes using Gene Ontology terms
- Understand the theory of how functional enrichment tools yield statistically enriched functions or interactions
- Discuss functional analysis using over-representation analysis, functional class scoring, and pathway topology methods
- Explore functional analysis tools for over-representation analysis

# Functional analysis

The output of RNA-seq differential expression analysis is a list of significant differentially expressed genes (DEGs). To gain greater biological insight on the differentially expressed genes there are various analyses that can be done:

- determine whether there is enrichment of known biological functions, interactions, or pathways
- identify genes' involvement in novel pathways or networks by grouping genes together based on similar trends
- use global changes in gene expression by visualizing all genes being significantly up- or down-regulated in the context of external interaction data

Generally for any differential expression analysis, it is useful to interpret the resulting gene lists using freely available web- and R-based tools. While tools for functional analysis span a wide variety of techniques, they can loosely be categorized into three main types: over-representation analysis, functional class scoring, and pathway topology [1].


Functional Pathway Analysis
Input
Pathway Database
Over-Representation Analysis (ORA)
Differential Expression Analysis
Differentially Expressed (DE) Genes
Number of DE and Reference Genes in Each Pathway
Functional Class Scoring (FCS)
Gene-level Statistics
Gene-set (Pathway) Statistics
Pathway Topology (PT)
DE Genes or Gene-level Statistics
Pathway Topology
· Number of Reactions
· Position of Gene
· Type of Reaction
Pathway Impact Factor
Assess Pathway Significance

The goal of functional analysis is provide biological insight, so it's necessary to analyze our results in the context of our experimental hypothesis: **FMRP and MOV10 associate and regulate the translation of a subset of RNAs**. Therefore, based on the authors' hypothesis, we may expect the enrichment of processes/pathways related to **translation, splicing, and the regulation of mRNAs**, which we would need to validate experimentally.

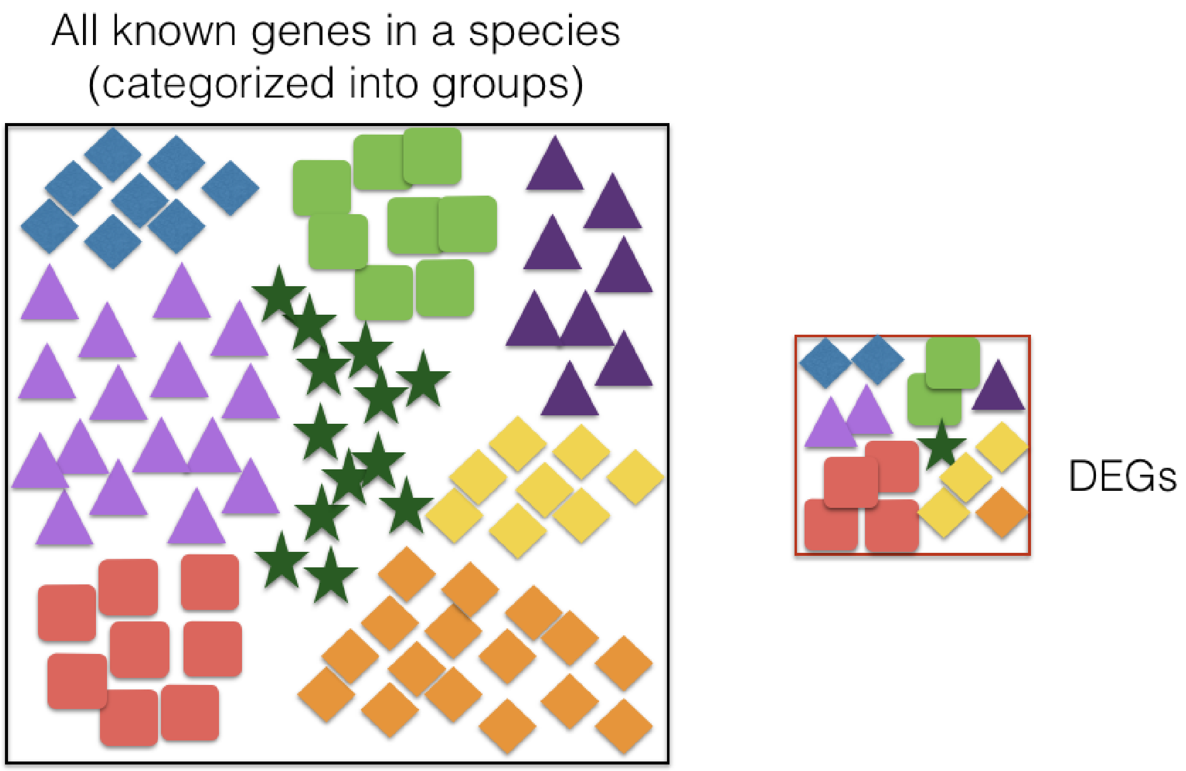
***Note that all tools described below are great tools to validate experimental results and to make hypotheses. These tools suggest genes/pathways that may be involved with your condition of interest; however, you should NOT use these tools to make conclusions about the pathways involved in your experimental process. You will need to perform experimental validation of any suggested pathways.***

## Over-representation analysis

There are a plethora of functional enrichment tools that perform some type of "over-representation" analysis by querying databases containing information about gene function and interactions.

These databases typically **categorize genes into groups (gene sets)** based on shared function, or involvement in a pathway, or presence in a specific cellular location, or other categorizations, e.g. functional pathways, etc. Essentially, known genes are binned into categories that have been consistently named (controlled vocabulary) based on how the gene has been annotated functionally. These categories are independent of any organism, however each organism has distinct categorizations available.

To determine whether any categories are over-represented, you can determine the probability of having the observed proportion of genes associated with a specific category in your gene list based on the proportion of genes associated with the same category in the background set (gene categorizations for the appropriate organism).



| Genes categories | Organism-specific Background | DE results | Over-represented? |
|---|---|---|---|
| Functional category 1 | 35/13000 | 25/1000 | Likely |
| Functional category 2 | 56/13000 | 4/1000 | Unlikely |
| Functional category 3 | 90/13000 | 8/1000 | Unlikely |
| Functional category 4 | 15/13000 | 10/1000 | Likely |
| … | | | |
| … | | | |

The statistical test that will determine whether something is actually over-represented is the *Hypergeometric test*.

## Hypergeometric testing

Using the example of the first functional category above, hypergeometric distribution is a probability distribution that describes the probability of 25 genes (k) being associated with "Functional category 1", for all genes in our gene list (n=1000), from a population of all of the genes in entire genome (N=13,000) which contains 35 genes (K) associated with "Functional category 1" [4].

The calculation of probability of k successes follows the formula:

$$P(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

This test will result in an adjusted p-value (after multiple test correction) for each category tested.

## Gene Ontology project

One of the most widely-used categorizations is the **Gene Ontology (GO)** established by the Gene Ontology project.

"The Gene Ontology project is a collaborative effort to address the need for consistent descriptions of gene products across databases" [2]. The Gene Ontology Consortium maintains the GO terms, and these GO terms are incorporated into gene annotations in many of the popular repositories for animal, plant, and microbial genomes.

Tools that investigate enrichment of biological functions or interactions often use the Gene Ontology (GO) categorizations, i.e. the GO terms to determine whether any have significantly modified representation in a given list of genes. Therefore, to best use and interpret the results from these functional analysis tools, it is helpful to have a good understanding of the GO terms themselves and their organization.

### GO Ontologies

To describe the roles of genes and gene products, GO terms are organized into three independent controlled vocabularies (ontologies) in a species-independent manner:

- **Biological process:** refers to the biological role involving the gene or gene product, and could include "transcription", "signal transduction", and "apoptosis". A biological process generally involves a chemical or physical change of the starting material or input.
- **Molecular function:** represents the biochemical activity of the gene product, such activities could include "ligand", "GTPase", and "transporter".
- **Cellular component:** refers to the location in the cell of the gene product. Cellular components could include "nucleus", "lysosome", and "plasma membrane".

Each GO term has a term name (e.g. **DNA repair**) and a unique term accession number (**GO:0005125**), and a single gene product can be associated with many GO terms, since a single gene product "may function in several processes, contain domains that carry out diverse molecular functions, and participate in multiple alternative interactions with other proteins, organelles or locations in the cell" [3].

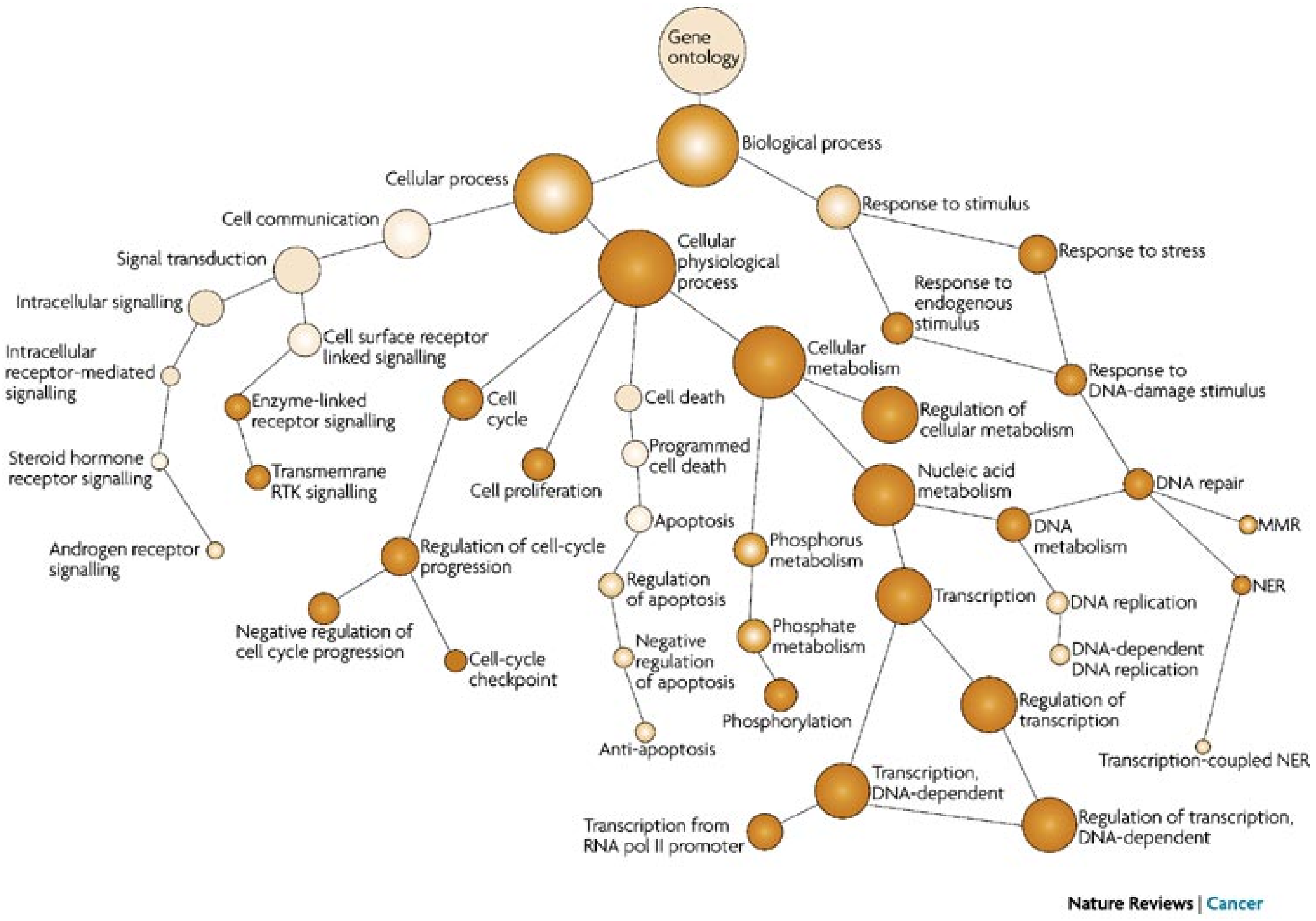
### GO term hierarchy

Some gene products are well-researched, with vast quantities of data available regarding their biological processes and functions. However, other gene products have very little data available about their roles in the cell.

For example, the protein, “p53”, would contain a wealth of information on it’s roles in the cell, whereas another protein might only be known as a “membrane-bound protein” with no other information available.

The GO ontologies were developed to describe and query biological knowledge with differing levels of information available. To do this, GO ontologies are loosely hierarchical, ranging from general, ‘parent’, terms to more specific, ‘child’ terms. The GO ontologies are “loosely” hierarchical since ‘child’ terms can have multiple ‘parent’ terms.

Some genes with less information may only be associated with general ‘parent’ terms or no terms at all, while other genes with a lot of information be associated with many terms.


Gene ontology
Biological process
Cellular process
Response to stimulus
Cell communication
Response to stress
Signal transduction
Cellular physiological process
Response to endogenous stimulus
Intracellular signalling
Cell surface receptor linked signalling
Cellular metabolism
Response to DNA-damage stimulus
Intracellular receptor-mediated signalling
Enzyme-linked receptor signalling
Cell cycle
Cell death
Regulation of cellular metabolism
Steroid hormone receptor signalling
Transmemrane RTK signalling
Cell proliferation
Programmed cell death
Nucleic acid metabolism
DNA repair
Apoptosis
DNA metabolism
MMR
Androgen receptor signalling
Regulation of cell-cycle progression
Phosphorus metabolism
Regulation of apoptosis
Transcription
DNA replication
NER
Negative regulation of cell cycle progression
Cell-cycle checkpoint
Negative regulation of apoptosis
Phosphate metabolism
DNA-dependent DNA replication
Phosphorylation
Regulation of transcription
Anti-apoptosis
Transcription-coupled NER
Transcription, DNA-dependent
Transcription from RNA pol II promoter
Regulation of transcription, DNA-dependent
Nature Reviews | Cancer

Tips for working with GO terms

## clusterProfiler

We will be using clusterProfiler to perform over-representation analysis on GO terms associated with our list of significant genes. The tool takes as input a significant gene list and a background gene list and performs statistical enrichment analysis using hypergeometric testing. The basic arguments allow the user to select the appropriate organism and GO ontology (BP, CC, MF) to test.

## Running clusterProfiler

To run clusterProfiler GO over-representation analysis, we will change our gene names into Ensembl IDs, since the tool works a bit easier with the Ensembl IDs.

Then load the following libraries:

```
# Load libraries
library(DOSE)
library(pathview)
library(clusterProfiler)
library(org.Hs.eg.db)
```

For the different steps in the functional analysis, we require Ensembl and Entrez IDs. We will use the gene annotations that we generated previously to merge with our differential expression results.

```
## Merge the AnnotationHub dataframe with the results
res_ids <- left_join(res_tableOE_tb, annotations_ahb, by=c("gene"="gene_id"))
```

> ***NOTE:*** *If you were unable to generate the* `annotations_ahb` *object, you can download the annotations to your* `data` *folder by right-clicking here and selecting "Save link as…"*
>
> *To read in the object, you can run the following code:* `annotations_ahb <- read.csv("annotations_ahb.csv")`

To perform the over-representation analysis, we need a list of background genes and a list of significant genes. For our background dataset we will use all genes tested for differential expression (all genes in our results table). For our significant gene list we will use genes with p-adjusted values less than 0.05 (we could include a fold change threshold too if we have many DE genes).

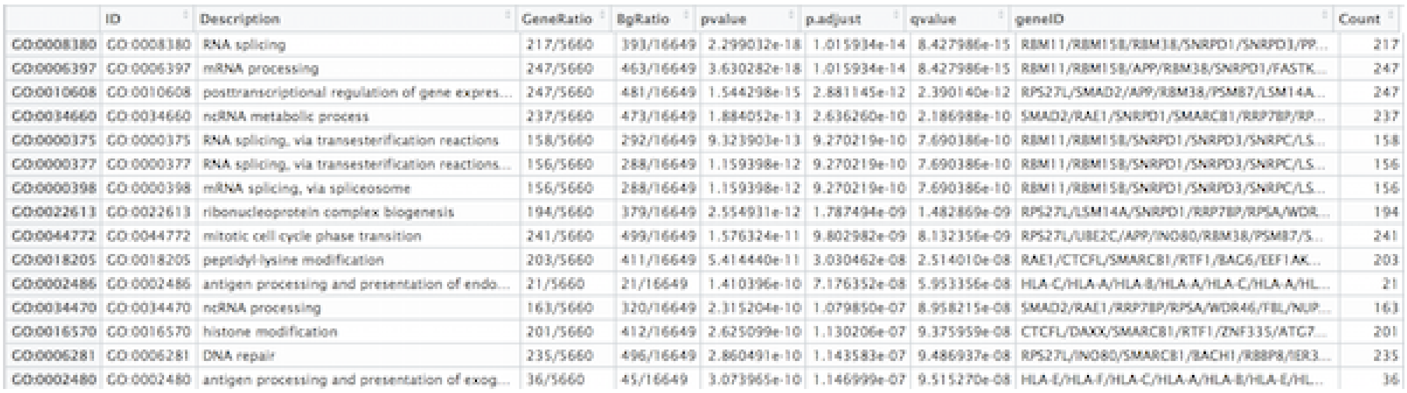
```
## Create background dataset for hypergeometric testing using all genes tested fo
allOE_genes <- as.character(res_ids$gene)

## Extract significant results
sigOE <- dplyr::filter(res_ids, padj < 0.05)

sigOE_genes <- as.character(sigOE$gene)
```

Now we can perform the GO enrichment analysis and save the results:

```
## Run GO enrichment analysis
ego <- enrichGO(gene = sigOE_genes,
                universe = allOE_genes,
                keyType = "ENSEMBL",
                OrgDb = org.Hs.eg.db,
                ont = "BP",
                pAdjustMethod = "BH",
                qvalueCutoff = 0.05,
                readable = TRUE)
```

> **NOTE:** The different organisms with annotation databases available to use with for the `OrgDb` argument can be found here.
>
> Also, the `keyType` argument may be coded as `keytype` in different versions of clusterProfiler.
>
> Finally, the `ont` argument can accept either “BP” (Biological Process), “MF” (Molecular Function), and “CC” (Cellular Component) subontologies, or “ALL” for all three.

```
## Output results from GO analysis to a table
cluster_summary <- data.frame(ego)

write.csv(cluster_summary, "results/clusterProfiler_Mov10oe.csv")
```



| | ID | Description | GeneRatio | BgRatio | pvalue | p.adjust | qvalue | geneID | Count |
|---|---|---|---|---|---|---|---|---|---|
| GO:0008380 | GO:0008380 | RNA splicing | 217/5660 | 393/16649 | 2.299012e-18 | 1.015934e-14 | 8.427986e-15 | RBM11/RBM15B/RBM38/SNRPD1/SNRPD3/PP... | 217 |
| GO:0006397 | GO:0006397 | mRNA processing | 247/5660 | 463/16649 | 3.630282e-18 | 1.015934e-14 | 8.427986e-15 | RBM11/RBM15B/APP/RBM38/SNRPD1/FASTK... | 247 |
| GO:0010608 | GO:0010608 | posttranscriptional regulation of gene expres... | 247/5660 | 481/16649 | 1.544298e-15 | 2.881145e-12 | 2.390140e-12 | RPS27L/SMAD2/APP/RBM38/PSMB7/LSM14A... | 247 |
| GO:0034660 | GO:0034660 | ncRNA metabolic process | 237/5660 | 473/16649 | 1.884052e-13 | 2.636260e-10 | 2.186988e-10 | SMAD2/RAE1/SNRPD1/SMARCB1/RRP7BP/RP... | 237 |
| GO:0000375 | GO:0000375 | RNA splicing, via transesterification reactions | 158/5660 | 292/16649 | 9.323903e-13 | 9.270219e-10 | 7.690386e-10 | RBM11/RBM15B/SNRPD1/SNRPD3/SNRPC/LS... | 158 |
| GO:0000377 | GO:0000377 | RNA splicing, via transesterification reactions... | 156/5660 | 288/16649 | 1.159398e-12 | 9.270219e-10 | 7.690386e-10 | RBM11/RBM15B/SNRPD1/SNRPD3/SNRPC/LS... | 156 |
| GO:0000398 | GO:0000398 | mRNA splicing, via spliceosome | 156/5660 | 288/16649 | 1.159398e-12 | 9.270219e-10 | 7.690386e-10 | RBM11/RBM15B/SNRPD1/SNRPD3/SNRPC/LS... | 156 |
| GO:0022613 | GO:0022613 | ribonucleoprotein complex biogenesis | 194/5660 | 379/16649 | 2.554931e-12 | 1.787494e-09 | 1.482869e-09 | RPS27L/LSM14A/SNRPD1/RRP7BP/RPSA/WDR... | 194 |
| GO:0044772 | GO:0044772 | mitotic cell cycle phase transition | 241/5660 | 499/16649 | 1.576324e-11 | 9.802982e-09 | 8.132356e-09 | RPS27L/UBE2C/APP/INO80/RBM38/PSMB7/S... | 241 |
| GO:0018205 | GO:0018205 | peptidyl-lysine modification | 203/5660 | 411/16649 | 5.414440e-11 | 3.030462e-08 | 2.514010e-08 | RAE1/CTCFL/SMARCB1/RTF1/BAG6/EEF1AK... | 203 |
| GO:0002486 | GO:0002486 | antigen processing and presentation of endo... | 21/5660 | 21/16649 | 1.410396e-10 | 7.176352e-08 | 5.953356e-08 | HLA-C/HLA-A/HLA-B/HLA-A/HLA-C/HLA-A/HL... | 21 |
| GO:0034470 | GO:0034470 | ncRNA processing | 163/5660 | 320/16649 | 2.315204e-10 | 1.079850e-07 | 8.958215e-08 | SMAD2/RAE1/RRP7BP/RPSA/WDR46/FBL/NUP... | 163 |
| GO:0016570 | GO:0016570 | histone modification | 201/5660 | 412/16649 | 2.625099e-10 | 1.130206e-07 | 9.375959e-08 | CTCFL/DAXX/SMARCB1/RTF1/ZNF335/ATG7... | 201 |
| GO:0006281 | GO:0006281 | DNA repair | 235/5660 | 496/16649 | 2.860491e-10 | 1.143583e-07 | 9.486937e-08 | RPS27L/INO80/SMARCB1/BACH1/RBBP8/ERR3... | 235 |
| GO:0002480 | GO:0002480 | antigen processing and presentation of exog... | 36/5660 | 45/16649 | 3.073965e-10 | 1.146999e-07 | 9.515270e-08 | HLA-E/HLA-F/HLA-C/HLA-A/HLA-B/HLA-E/HL... | 36 |

> **NOTE:** Instead of saving just the results summary from the `ego` object, it might also be beneficial to save the object itself. The `save()` function enables you to save it as a `.rda` file, e.g. `save(ego, file="results/ego.rda")`. The complementary function to `save()` is the function `load()`, e.g. `ego <- load(file="results/ego.rda")`.
>
> *This is a useful set of functions to know, since it enables one to preserve analyses at specific stages and reload them when needed.* More information about these functions can be found here & here.

## Visualizing clusterProfiler results

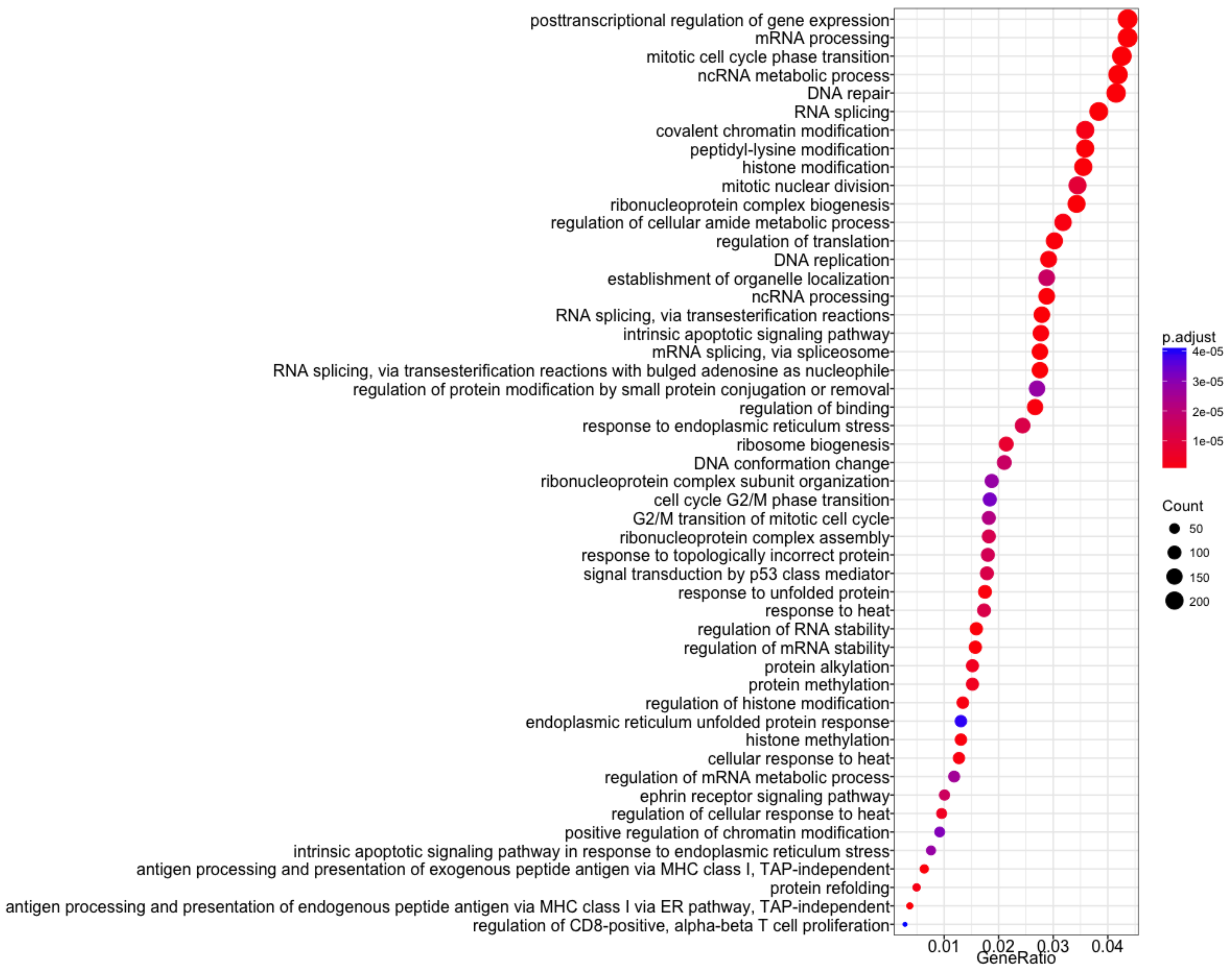
clusterProfiler has a variety of options for viewing the over-represented GO terms. We will explore the dotplot, enrichment plot, and the category netplot.

The **dotplot** shows the number of genes associated with the first 50 terms (size) and the p-adjusted values for these terms (color). This plot displays the top 50 GO terms by gene ratio (# genes related to GO term / total number of sig genes), not p-adjusted value.

```
## Dotplot
dotplot(ego, showCategory=50)
```

**To save the figure,** click on the `Export` button in the RStudio `Plots` tab and `Save as PDF...`. In the pop-up window, change:

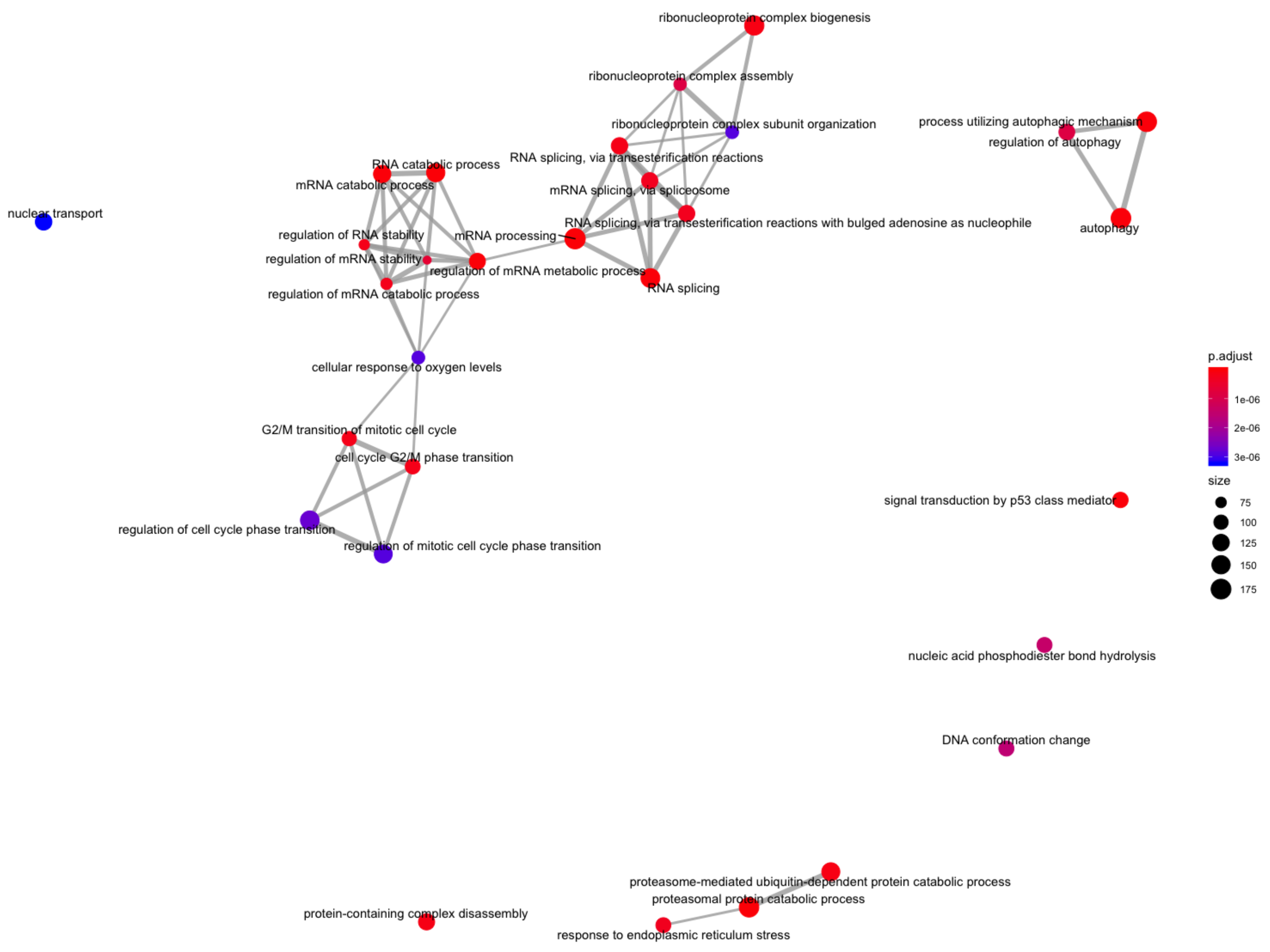
- `Orientation:` to `Landscape`
- `PDF size` to `8 x 14` to give a figure of appropriate size for the text labels



The next plot is the **enrichment GO plot**, which shows the relationship between the top 50 most significantly enriched GO terms (padj.), by grouping similar terms together. Before creating the plot, we will need to obtain the similarity between terms using the `pairwise_termsim()` function (instructions for emapplot). In the enrichment plot, the color represents the p-values relative to the other displayed terms (brighter red is more significant), and the size of the terms represents the number of genes that are significant from our list.

```
## Add similarity matrix to the termsim slot of enrichment result
ego <- enrichplot::pairwise_termsim(ego)

## Enrichmap clusters the 50 most significant (by padj) GO terms to visualize rela
emapplot(ego, showCategory = 50)
```

**To save the figure,** click on the `Export` button in the RStudio `Plots` tab and `Save as PDF...`. In the pop-up window, change the `PDF size` to `12 x 14` to give a figure of appropriate size for the text labels.

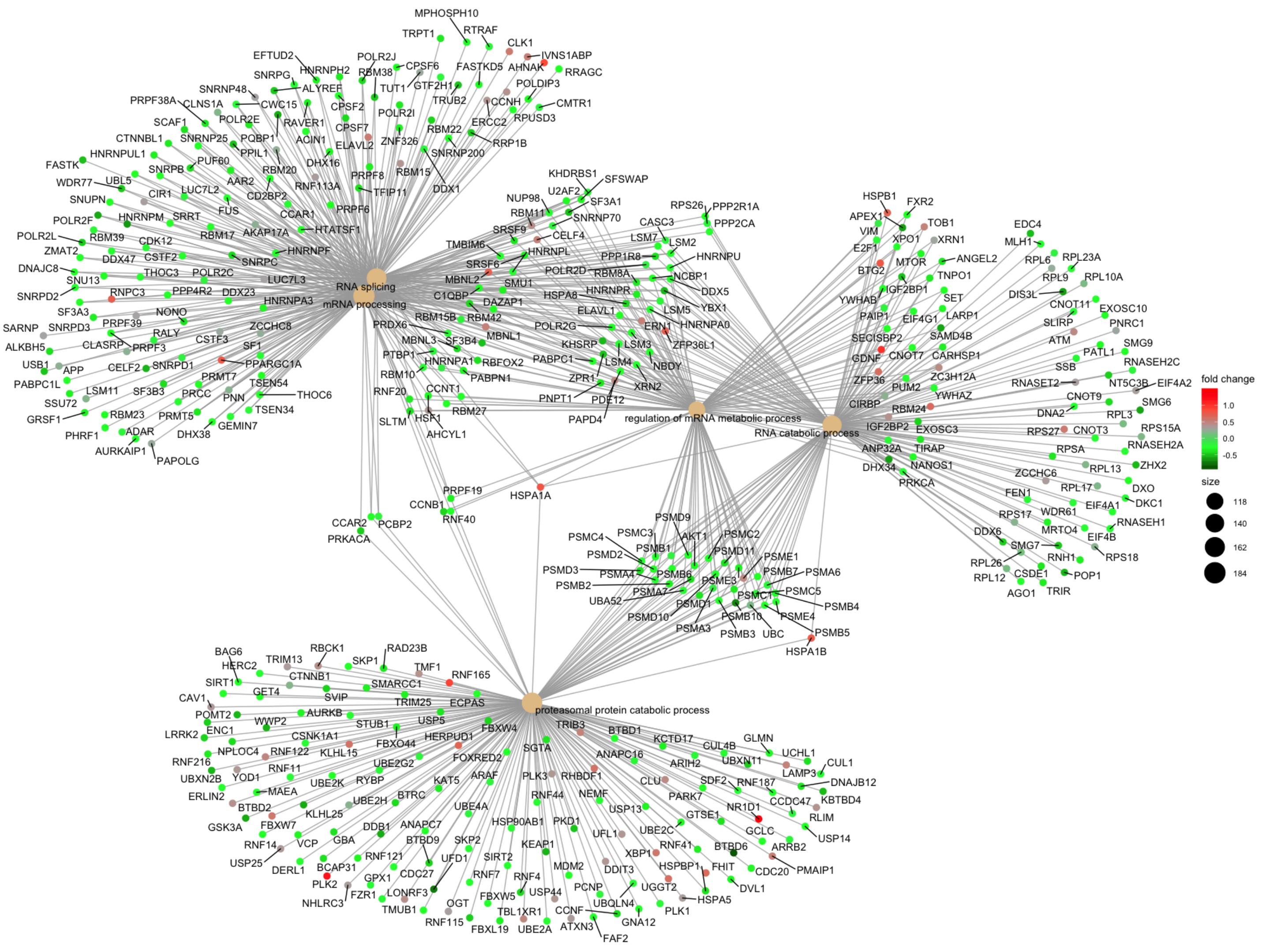


Finally, the **category netplot** shows the relationships between the genes associated with the top five most significant GO terms and the fold changes of the significant genes associated with these terms (color). The size of the GO terms reflects the pvalues of the terms, with the more significant terms being larger. This plot is particularly useful for hypothesis generation in identifying genes that may be important to several of the most affected processes.

> **Note** - You may need to install the `ggnewscale` package using `install.packages("ggnewscale")` for the `cnetplot()` function to work.

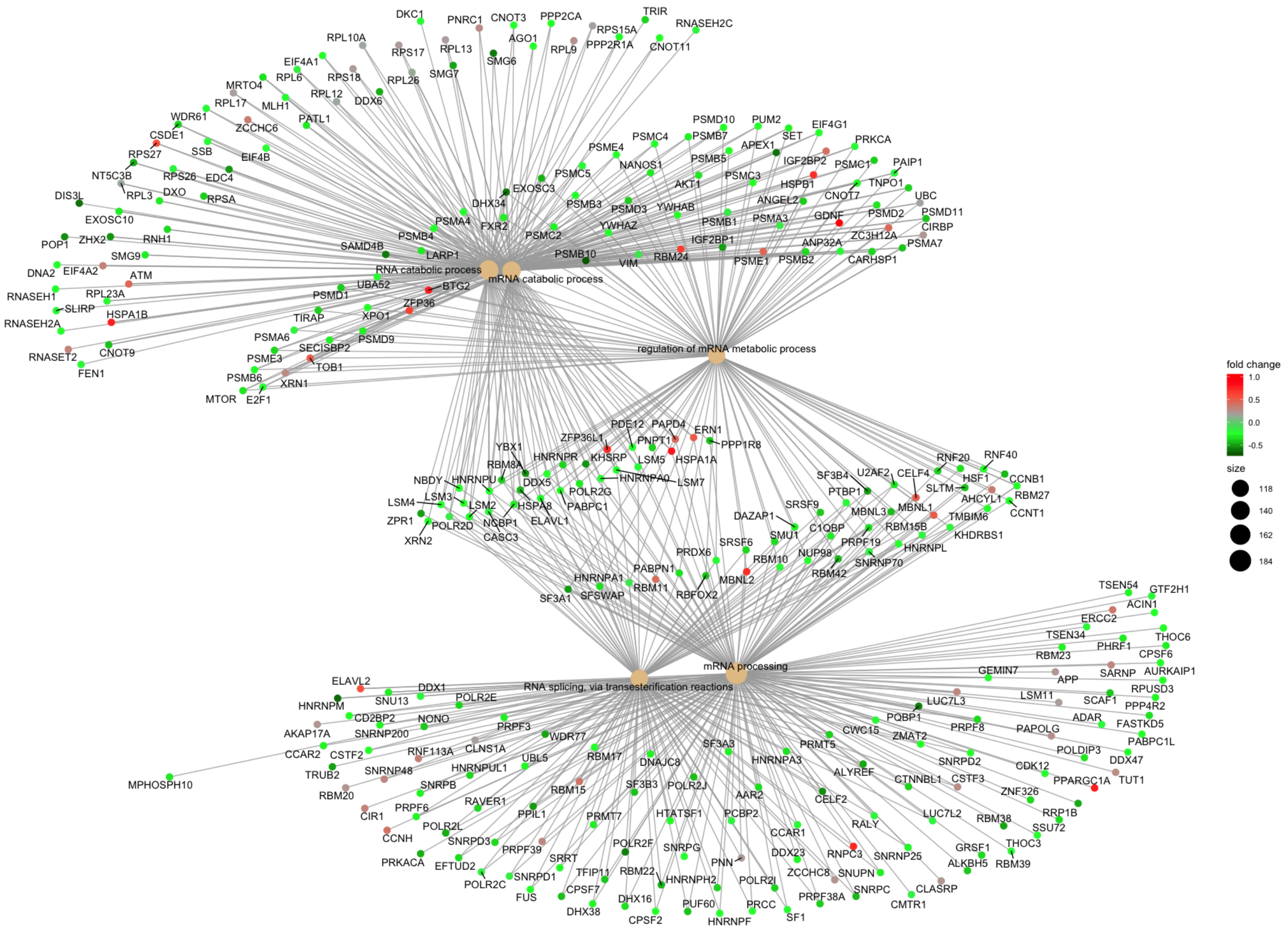
```
## To color genes by log2 fold changes, we need to extract the log2 fold changes 
OE_foldchanges <- sigOE$log2FoldChange

names(OE_foldchanges) <- sigOE$gene

## Cnetplot details the genes associated with one or more terms - by default give
cnetplot(ego,
         categorySize="pvalue",
         showCategory = 5,
         foldChange=OE_foldchanges,
         vertex.label.font=6)

## If some of the high fold changes are getting drowned out due to a large range,
OE_foldchanges <- ifelse(OE_foldchanges > 2, 2, OE_foldchanges)
OE_foldchanges <- ifelse(OE_foldchanges < -2, -2, OE_foldchanges)

cnetplot(ego,
         categorySize="pvalue",
         showCategory = 5,
         foldChange=OE_foldchanges,
         vertex.label.font=6)
```

**Again, to save the figure,** click on the `Export` button in the RStudio `Plots` tab and `Save as PDF...`. Change the `PDF size` to `12 x 14` to give a figure of appropriate size for the text labels.



If you are interested in significant processes that are **not** among the top five, you can subset your `ego` dataset to only display these processes:

```
## Subsetting the ego results without overwriting original `ego` variable
ego2 <- ego

ego2@result <- ego@result[c(1,3,4,8,9),]

## Plotting terms of interest
cnetplot(ego2,
         categorySize="pvalue",
         foldChange=OE_foldchanges,
         showCategory = 5,
         vertex.label.font=6)
```


RNA catabolic process
mRNA catabolic process
regulation of mRNA metabolic process
mRNA processing
RNA splicing, via transesterification reactions
fold change
1.0
0.5
0.0
-0.5
size
118
140
162
184
DKC1
PNRC1
CNOT3
PPP2CA
TRIR
RNASEH2C
RPS15A
PPP2R1A
CNOT11
RPL10A
RPS17
RPL13
AGO1
RPL9
SMG6
SMG7
RPL26
EIF4A1
RPL6
RPS18
RPL12
DDX6
MRTO4
RPL17
MLH1
PATL1
WDR61
ZCCHC6
CSDE1
SSB
EIF4B
RPS27
NT5C3B
RPS26
EDC4
RPL3
DXO
RPSA
DIS3L
EXOSC10
POP1
ZHX2
RNH1
SMG9
DNA2
EIF4A2
ATM
RNASEH1
RPL23A
SLIRP
HSPA1B
RNASEH2A
RNASET2
CNOT9
FEN1
MTOR
E2F1
PSMB6
XRN1
PSME3
TOB1
SECISBP2
PSMA6
PSMD9
TIRAP
XPO1
ZFP36
PSMD1
UBA52
BTG2
LARP1
SAMD4B
PSMB4
PSMA4
FXR2
DHX34
EXOSC3
PSMC5
PSMB3
PSMD3
YWHAB
PSMC2
YWHAZ
PSMB10
VIM
RBM24
PSMC4
PSMD10
PUM2
EIF4G1
PSMB7
SET
PRKCA
PSME4
APEX1
NANOS1
PSMB5
IGF2BP2
PSMC1
PAIP1
AKT1
PSMC3
HSPB1
TNPO1
UBC
ANGEL2
CNOT7
GDNF
PSMD2
PSMD11
PSMB1
PSMA3
CIRBP
IGF2BP1
ANP32A
ZC3H12A
PSMA7
PSME1
PSMB2
CARHSP1
PDE12
PAPD4
ERN1
ZFP36L1
PNPT1
PPP1R8
YBX1
HNRNPR
KHSRP
LSM5
HSPA1A
RBM8A
HNRNPU
DDX5
HNRNPA0
LSM7
NBDY
POLR2G
LSM3
LSM4
LSM2
HSPA8
PABPC1
ZPR1
POLR2D
NCBP1
ELAVL1
XRN2
CASC3
RNF20
RNF40
SF3B4
U2AF2
CELF4
HSF1
CCNB1
PTBP1
SLTM
AHCYL1
RBM27
SRSF9
MBNL3
MBNL1
CCNT1
DAZAP1
TMBIM6
SMU1
C1QBP
RBM15B
KHDRBS1
SRSF6
PRPF19
HNRNPL
PRDX6
RBM10
NUP98
SNRNP70
PABPN1
MBNL2
RBM42
HNRNPA1
RBM11
SF3A1
SFSWAP
RBFOX2
TSEN54
GTF2H1
ACIN1
ERCC2
TSEN34
THOC6
PHRF1
RBM23
CPSF6
GEMIN7
SARNP
AURKAIP1
APP
RPUSD3
LSM11
SCAF1
PPP4R2
LUC7L3
ADAR
FASTKD5
PQBP1
PRPF8
PAPOLG
CWC15
ZMAT2
PABPC1L
POLDIP3
DDX47
SNRPD2
CDK12
CSTF3
PPARGC1A
TUT1
CTNNBL1
ZNF326
RRP1B
LUC7L2
RBM38
SSU72
THOC3
GRSF1
ALKBH5
RBM39
ELAVL2
DDX1
HNRNPM
SNU13
POLR2E
CD2BP2
NONO
AKAP17A
SNRNP200
PRPF3
WDR77
CCAR2
CSTF2
RNF113A
CLNS1A
TRUB2
SNRNP48
HNRNPUL1
UBL5
RBM17
SF3A3
PRMT5
DNAJC8
HNRNPA3
ALYREF
MPHOSPH10
SNRPB
RBM15
SF3B3
POLR2J
RBM20
RAVER1
AAR2
CELF2
PRPF6
PPIL1
CIR1
POLR2L
PRMT7
HTATSF1
PCBP2
RALY
CCAR1
CCNH
SNRPD3
PRPF39
POLR2F
SNRPG
DDX23
RNPC3
SNRNP25
PRKACA
EFTUD2
SRRT
PNN
ZCCHC8
SNUPN
SNRPD1
TFIP11
RBM22
HNRNPH2
POLR2I
CLASRP
POLR2C
CPSF7
SNRPC
FUS
DHX16
PRPF38A
CMTR1
DHX38
PUF60
PRCC
CPSF2
HNRNPF
SF1