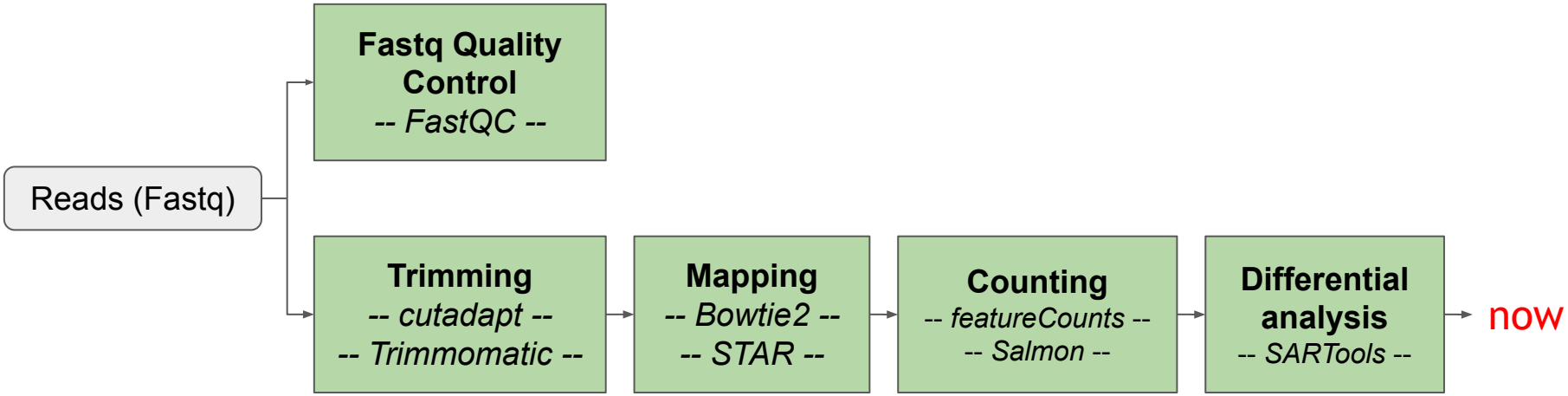
Gene Set Analysis

Thibault Dayris, Jean-Pascal Meneboo, Audrey Onfroy

École de Bioinformatique IFB-INSERM 2024

Introduction

So far…


Reads (Fastq)
Fastq Quality Control
-- FastQC --
Trimming
-- cutadapt --
-- Trimmomatic --
Mapping
-- Bowtie2 --
-- STAR --
Counting
-- featureCounts --
-- Salmon --
Differential analysis
-- SARTools --
now

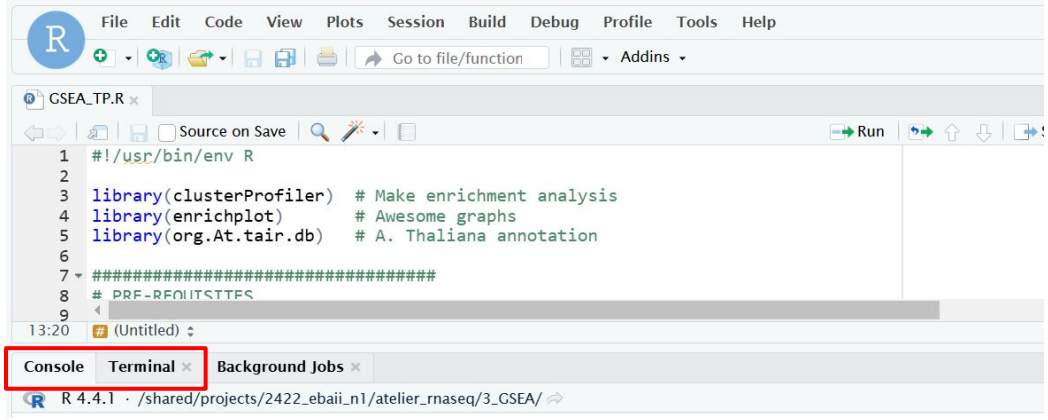
Copy the support to your folder

Execute the commands below in the **terminal**, to (1) create a directory in your $HOME and (2) copy-paste the data there.

```
mkdir -p /shared/projects/<YOUR_PROJECT>/TP_GSEA

cp -r /shared/projects/2422_ebaii_n1/atelier_rnaseq/3_GSEA/*
      /shared/projects/<YOUR_PROJECT>/TP_GSEA
```

Now we use only RStudio and the **console**.

In the **~/TP_GSEA** folder, open the **GSEA_TP.R** file (or **GSEA_TP.Rmd**).



Packages of interest

For this session, we use the following packages:

```
library(clusterProfiler)          # Make enrichment analysis
library(enrichplot)               # Awesome graphs
library(org.At.tair.db)           # A. Thaliana annotation
```

Input data

We have a large table with many **columns**:

```
deseq_genes = read.table(
     file = "tables/KOvsWT.complete.txt",
     sep = "\t",
     header = TRUE
)

colnames(deseq_genes)
```

```
 [1] "Id"            "WT1"        "WT2"        "WT3"        "KO1"
 [6] "KO2"           "KO3"        "norm.WT1"   "norm.WT2"   "norm.WT3"
[11] "norm.KO1"      "norm.KO2"   "norm.KO3"   "baseMean"   "WT"
[16] "KO"            "FoldChange" "log2FoldChange" "stat"       "pvalue"
[21] "padj"          "dispGeneEst""dispFit"    "dispMAP"    "dispersion"
[26] "betaConv"      "maxCooks"
```

Input data

We have a large table with many **rows**:

```
nrow(deseq_genes)
```

```
[1] 27655
```

```
head(deseq_genes$Id)
```

```
[1] "gene:AT1G01010" "gene:AT1G01020" "gene:AT1G01030"
[4] "gene:AT1G01040" "gene:AT1G01050" "gene:AT1G01060"
```

Gene identifiers

Data associated with gene:AT1G61580 ?

We extract the row corresponding to this `Id`:

```
deseq_genes[deseq_genes$Id == "gene:AT1G61580", ]
```

```
                 Id   baseMean  ...   WT  KO FoldChange log2FoldChange
5120 gene:AT1G61580     173.19  ...  218 128      0.588         -0.766
        stat        pvalue         padj  dispGeneEst  dispFit
5120   -4.48  7.465947e-06  0.0001156724            0   0.0311
      dispMAP   dispersion  betaConv  maxCooks
5120   0.0149       0.0149      TRUE    0.0222
```

Data associated with gene:AT1G61580 ?

We extract the row corresponding to this `Id`:

```
deseq_genes[deseq_genes$Id == "gene:AT1G61580", ]
```

```
                 Id   baseMean  ...   WT  KO FoldChange log2FoldChange
5120 gene:AT1G61580     173.19  ...  218 128      0.588         -0.766
        stat        pvalue         padj  dispGeneEst  dispFit
5120   -4.48  7.465947e-06  0.0001156724            0   0.0311
      dispMAP   dispersion  betaConv  maxCooks
5120   0.0149       0.0149      TRUE    0.0222
```

Someone to explain these terms ?

Data associated with gene:AT1G61580 ?

We extract the row corresponding to this `Id`:

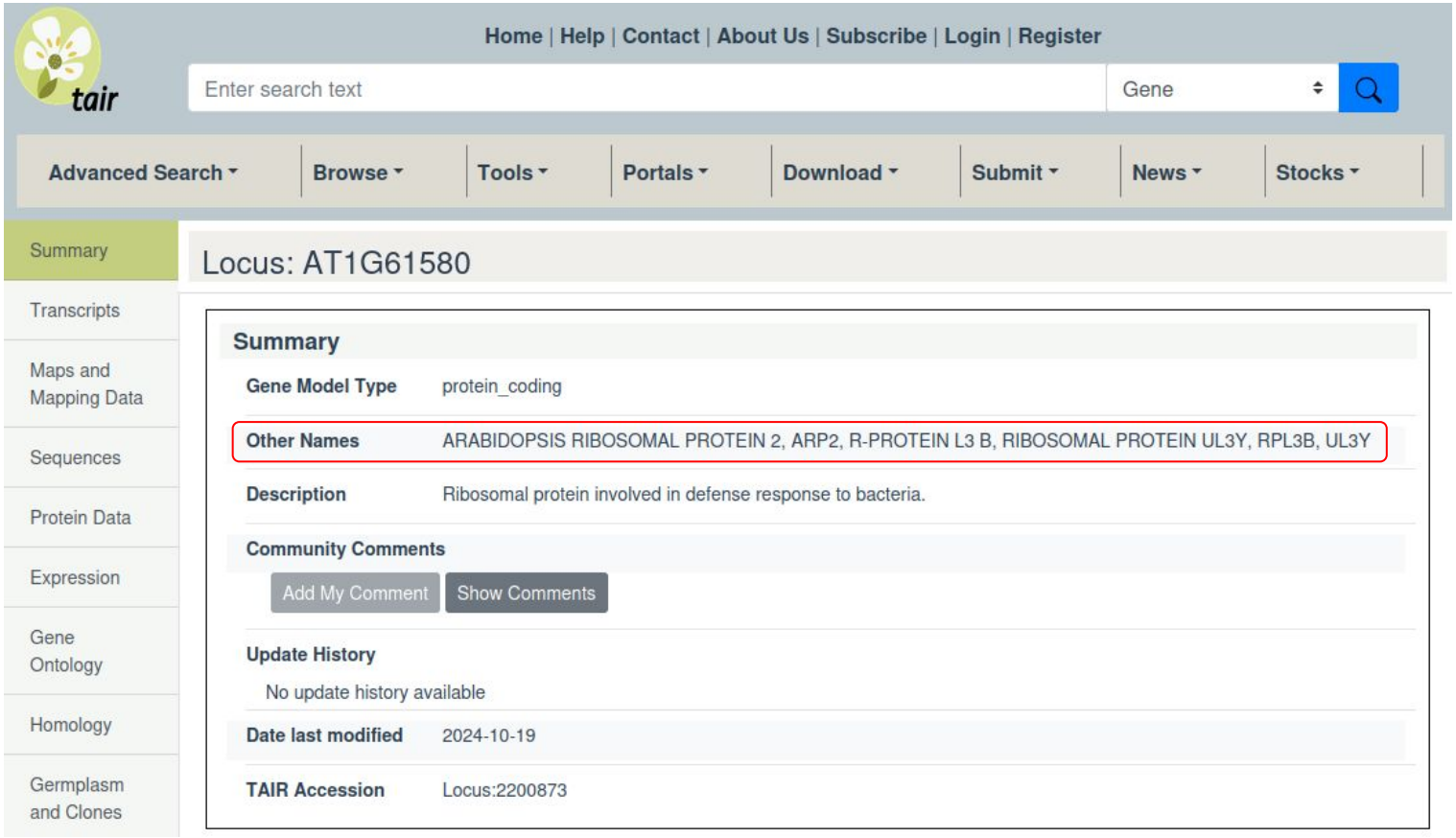
```
deseq_genes[deseq_genes$Id == "gene:AT1G61580", ]
```

```
                 Id   baseMean  ...   WT  KO FoldChange log2FoldChange
5120 gene:AT1G61580     173.19  ...  218 128      0.588         -0.766
        stat        pvalue         padj  dispGeneEst  dispFit
5120   -4.48  7.465947e-06  0.0001156724            0   0.0311
      dispMAP   dispersion  betaConv  maxCooks
5120   0.0149       0.0149      TRUE    0.0222
```

The Id of the gene is `gene:AT1G61580`.
The mean expression in the WT (resp. KO) is `218` (resp. `128`).
The fold change in expression is `0.588` (= `128`/`218`).
The adjusted p-value almost equals to `1e-04`, which means that it is very likely that the difference of expression is related to the KO/WT status.

Name associated with AT1G61580 Id ?

arabidopsis.org



Id associated with ARP2 name ?

arabidopsis.org

Your query for genes where gene name, description, phenotype, locus name, uniprot id or GenBank accession contains the term ARP2 resulted in 16 matches

Displaying 1 - 16 of 16 results

Select All | Clear Selected

No.		Locus	Description
1	☐	AT2G38440	Other Names: ATSCAR2;DIS3;IRREGULAR TRICHOME BRANCH1;ITB1;SCAR HOMOLOG 2;SCAR2;WAVE4 Encodes a subunit of the WAVE complex. The WAVE complex is required for activation of ARP2/3 complex which functions in actin microfilament nucleation and branching. Mutations cause defects in both the actin and microtubule cytoskeletons that result in aberrant epidermal cell expansion. itb1 mutants showed irregularities in trichome branch positioning and expansion. The SHD domain of this protein binds to BRK1 and overexpression of the SHD domain results in a dominant negative phenotype. The mRNA is cell-to-cell mobile.
2	☐	AT5G65274	Other Names: ARP2/3 complex 16 kDa subunit (p16-Arc);(source:Araport11)
3	☐	AT3G27000	Other Names: ACTIN RELATED PROTEIN 2;ARP2;ATARP2;WRM;WURM encodes a protein whose sequence is similar to actin-related proteins (ARPs) in other organisms. its transcript level is down regulated by light and is expressed in very low levels in all organs examined.
4	☐	AT1G61580	Other Names: ARABIDOPSIS RIBOSOMAL PROTEIN 2;ARP2;R-PROTEIN L3 B;RIBOSOMAL PROTEIN UL3Y;RPL3B;UL3Y

chr3 chr1

https://www.ncbi.nlm.nih.gov/gene/?term=ARP2

ARP2 in the world…

ARP2 is not only related to A. thaliana !

Search results

Items: 1 to 20 of 38040

<< First < Prev Page 1 of 1902 Next > Last >>

See also 254 discontinued or replaced items.

Name/Gene ID	Description	Location	Aliases
ARP2 ID: 851532	actin-related protein 2 [*Saccharomyces cerevisiae S288C*]	Chromosome IV, NC_001136.10 (399340..400638)	YDL029W, ACT2
Arp2 ID: 32623	Actin-related protein 2 [*Drosophila melanogaster* (fruit fly)]	Chromosome X, NC_004354.4 (16548290..16553968, complement)	Dmel_CG9901, ARP14D, ARP2, Actr14D, Arp14D, Arp14d, CG9901, Dmel\CG9901, arp2
arp2 ID: 5802965	ARP2/3 actin-organizing complex actin-related protein subunit Arp2 [*Schizosaccharomyces pombe* (fission yeast)]	Chromosome I, NC_003424.3 (4783022..4784765)	SPOM_SPAC11H11.06, SPAC22F8.01
ARP2 ID: 822317	actin related protein 2 [*Arabidopsis thaliana* (thale cress)]	Chromosome 3, NC_003074.8 (9952479..9955982, complement)	AT3G27000, ACTIN RELATED PROTEIN 2, ATARP2, WRM, WURM, actin related protein 2
arp2 ID: 80877320	ARP2/3 actin-organizing complex subunit Arp2 [*Schizosaccharomyces osmophilus*]	Chromosome 2, NC_079239.1 (2976885..2978114)	SOMG_03844

https://www.ncbi.nlm.nih.gov/gene/?term=AT1G61580

AT1G61580 in the world…

However, AT1G61580 is unique.

Search results

Items: 2

Showing Current items.

Name/Gene ID	Description	Location	Aliases
RPL3B ID: 842454	R-protein L3 B [*Arabidopsis thaliana* (thale cress)]	Chromosome 1, NC_003070.9 (22720560..22723152, complement)	AT1G61580, ARABIDOPSIS RIBOSOMAL PROTEIN 2, ARP2, R-protein L3 B, RIBOSOMAL PROTEIN L3, T25B24.7, T25B24_7
RP1 ID: 840916	ribosomal protein 1 [*Arabidopsis thaliana* (thale cress)]	Chromosome 1, NC_003070.9 (16266553..16268945)	AT1G43170, ARP1, F1I21.1, F1I21_1, RPL3A, emb2207, embryo defective 2207, ribosomal protein 1

AT = Arabidopsis Thaliana

1 = Chromosome number

G = Protein coding gene

61580 = Unique gene identifier, given from top to bottom of chromosome

Gene name vs Gene identifier

	Gene name/symbol ARP2	**Gene identifier** AT1G61580
Benefits	human **understandable**	- **unique** in a database - **stable** over the genome versions
Limits	not unique, neither to an organism, nor to a genomic location, nor over time	- not easily readable - each database as its own identifier
Please use for:	- lab meeting - nice-looking graphs	- analysis - interaction with database

Clean gene identifiers

Why cleaning is required ?

The `Id` column is polluted by “gene:”

```
head(deseq_genes$Id)
```

```
[1] "gene:AT1G01010" "gene:AT1G01020" "gene:AT1G01030"
[4] "gene:AT1G01040" "gene:AT1G01050" "gene:AT1G01060"
```

For a computer, **`gene:AT1G01010`** is not **`AT1G01010`**.

Clean gene identifiers

We need a raw gene identifier:

```
deseq_genes$Id = sub(pattern = "gene:",
                     replacement = "",
                     x = deseq_genes$Id)
```

Let's check the output:

```
head(deseq_genes$Id)
```

```
[1] "AT1G01010" "AT1G01020" "AT1G01030" "AT1G01040" "AT1G01050" "AT1G01060"
```

Good !

Conversion

Conversion…

When interacting with databases, you may need TAIR ID, Ensembl ID, ENTREZ ID, UniProt ID…
For instance, we could convert TAIR ID to ENTREZ ID and gene symbol:

```
# Translate TAIR ID to ENTREZ ID
annotation = clusterProfiler::bitr(
  geneID   = deseq_genes$Id,           # Our gene list
  fromType = "TAIR",                   # We have TAIR ID
  toType   = c("ENTREZID", "SYMBOL"), # What we want
  OrgDb    = org.At.tair.db)           # Our annotation

# Add the translation to the result table
deseq_genes_with_symbol = merge(
  x = deseq_genes,
  y = annotation,
  by.x = "Id",         # In deseq_genes, TAIR IDs are stored in the Id column
  by.y = "TAIR")       # In annotation, TAIR IDs are stored in the TAIR column
```

Some IDs correspond to several symbols… (1/2)

Check the size of the merged table and the original one:

```
dim(deseq_genes)
```

```
[1] 27655    27
```

```
dim(deseq_genes_with_symbol)
```

```
[1] 38947    29
```

Why ?

Some IDs correspond to several symbols… (1/2)

Check the size of the merged table and the original one:

```
head(deseq_genes_with_symbol[, c("Id", "SYMBOL", "ENTREZID")])
```

```
         Id  SYMBOL ENTREZID
1 AT1G01010 ANAC001   839580
2 AT1G01010  NAC001   839580
3 AT1G01010   NTL10   839580
4 AT1G01020    ARV1   839569
5 AT1G01030    NGA3   839321
6 AT1G01040    ASU1   839574
```

Database

https://browser.planteome.org/amigo/search/annotation?q=root

Why database ?

We are studying plants. Which genes are expressed in the roots ?

Planteome Home Search Browse Tools & Resources About Quick se

Information about **Annotations** search

Filter results

Total annotation(s): 27947

root

User filters

+ taxon_label: Arabidopsis thaliana

Total annotation(s): 27947; showing: 1-10

Results count 10

«First <Prev Next> Last»

Bookmark

Object	Object name	Object Type	Direct annotation	Ontology (aspect)	Annotation extension	Taxon
ATCSLB05		protein	*root* hair elongation	Bio process (P)		Arabidopsis thaliana
AT4G35720	AT4G35720	protein	*root* development	Bio process (P)		Arabidopsis thaliana

We cannot look individually at all these genes.

But ! We can look at **gene sets**, which are annotated to represent *something*.

What is a gene set ?

A gene set is nothing more than a group of genes belonging to the same…

FUNCTION

members of same biochemical pathway

REGULATION

targets of the same regulatory elements

WHATEVER

used defined relevant classification

LOCATION

proteins expressed in the same cellular compartment

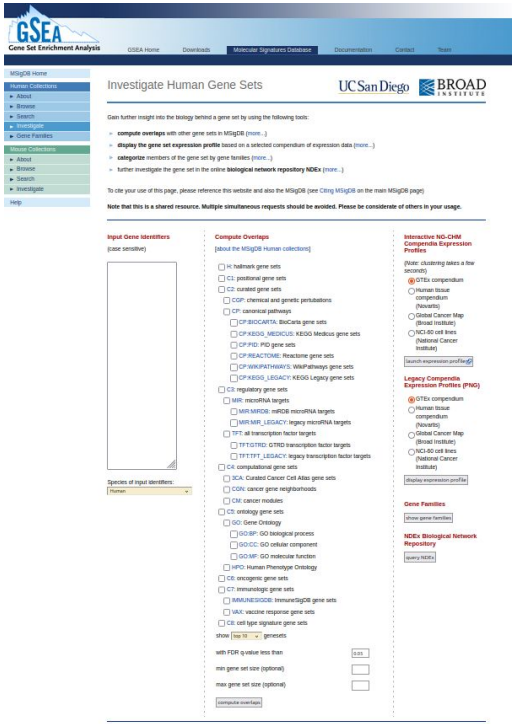
PHENOTYPE

proteins co-expressed under certain conditions

https://www.gsea-msigdb.org/gsea/msigdb/human/annotate.jsp

Which databases ?

There are many many (many) databases. Some are accessible in the **M**olecular **Sig**nature **D**ata**b**ase (MSigDB).



- KEGG https://www.genome.jp/ke
- PID no link
- Reactome https://reactome.org/
- WikiPathways https://www.wikipathways.org/
- Gene Ontology (GO) https://geneontology.org/
  - Molecular Functions (MF)
  - Cellular Components (CC)
  - Biological Processes (BP)
- …

These database (may) store redundant informations.

Which databases ?

https://igordot.github.io/msigdbr/

MSigDB also exists as a R package: msigdbr, which is useful for versioning.

msigdbr 7.5.1.9001 | Search for | Get started | Reference | Changelog

msigdbr: MSigDB Gene Sets for Multiple Organisms in a Tidy Data Format

Overview

The msigdbr R package provides Molecular Signatures Database (MSigDB) gene sets typically used with the Gene Set Enrichment Analysis (GSEA) software:

- in an R-friendly "tidy" format with one gene pair per row
- for multiple frequently studied model organisms, such as mouse, rat, pig, zebrafish, fly, and yeast, in addition to the original human genes
- as gene symbols as well as NCBI Entrez and Ensembl IDs
- without accessing external resources and requiring an active internet connection

Installation

The package can be installed from CRAN.

```
install.packages("msigdbr")
```

Usage

The package data can be accessed using the `msigdbr()` function, which returns a data

Links

View on CRAN

Browse source code

Report a bug

License

Full license

MIT + file LICENSE

Citation

Citing msigdbr

Developers

Igor Dolgalev

Author, maintainer

Dev status

CRAN 7.5.1

downloads 6612/month

R-CMD-check no status

codecov 100%

😔 MSigDB is centered on *Homo sapiens*, with orthologs mapped for *Mus musculus* only.

And for *A. thaliana* ?

https://bioconductor.org/packages/devel/BiocViews.html#Organism

Organisme database: From BioConductor, you may find a lot of organism annotations.

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

About Learn Packages Developers Search Get Started

Home > BiocViews

Bioconductor version 3.20 (Development)

☐ Developers: check this box to toggle the visibility of childless biocViews.

Find biocViews:

- Software (2250)
- AnnotationData (926)
  - ChipManufacturer (400)
  - ChipName (197)
    - CustomArray (2)
  - CustomDBSchema (10)
    - FunctionalAnnotation (32)
    - Organism (664)
      - Anopheles_gambiae (4)
      - Apis_mellifera (4)
      - Arabidopsis_thaliana (15)
      - Asparagus_officinalis (1)

Packages found under Organism:

Rank based on number of downloads: lower numbers are more frequently downloaded.

Show All entries

Search table:

Package	Maintainer	Title	Rank
org.At.tair.db	Bioconductor Package Maintainer	Genome wide annotation for Arabidopsis	34
BSgenome.Athaliana.TAIR.TAIR9	Bioconductor Package Maintainer	Full genome sequences for Arabidopsis thaliana (TAIR9)	180
arabidopsis.db0	Bioconductor Package Maintainer	Base Level Annotation databases for arabidopsis	278
BSgenome.Athaliana.TAIR.04232008	Bioconductor Package Maintainer	Full genome sequences for Arabidopsis thaliana (TAIR version from April 23, 2008)	406

Showing 1 to 4 of 4 entries (filtered from 664 total entries)

Previous Next

Install a package from BioConductor

If the package is not yet installed, you can install it:

```
# If needed, install (once) BiocManager
if (!require("BiocManager", quietly = TRUE)) {
    install.packages("BiocManager")
    }
BiocManager::install(version = "3.19")

# Install package from BioConductor
BiocManager::install("org.At.tair.db")
```

For this session, the package has already been installed (and we already load it):

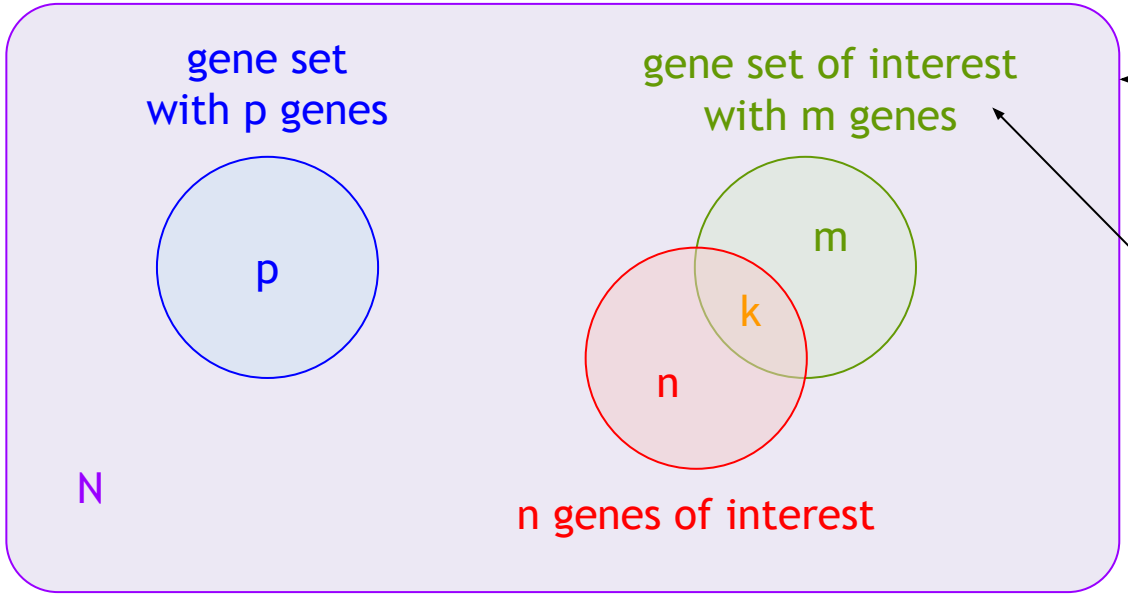
```
library("org.At.tair.db")
```

Over Representation Analysis

Over Representation Analysis

ORA stands for ***O**ver **R**epresentation **A**nalysis*.

Given a list of differentially expressed genes, search the gene sets containing these genes, and run an enrichment test on each of them.



all possible genes
(universe/background)

the gene sets are stored in
the org.At.tair.db database

https://www.gsea-msigdb.org/gsea/msigdb/human/annotate.jsp

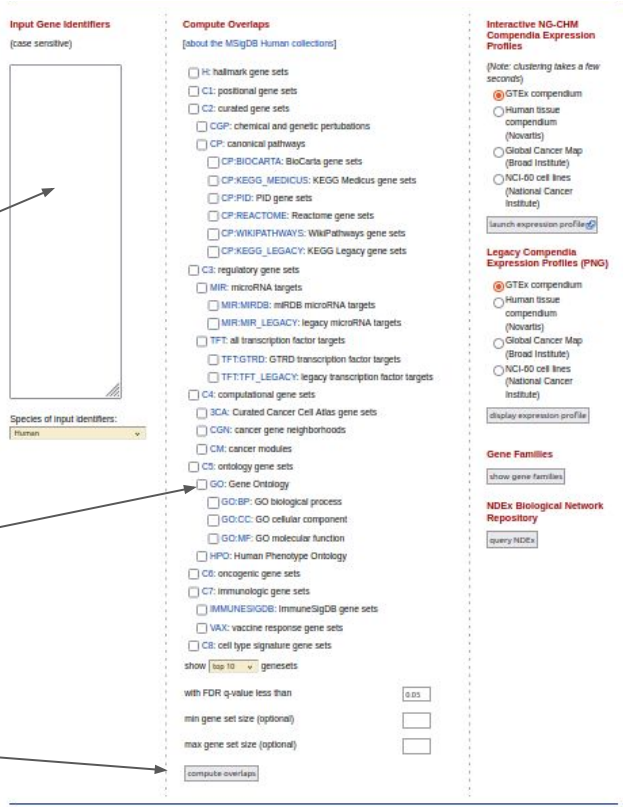
Make ORA **without** programming ?

This is possible if you work with MSigDB, for *H. sapiens* or *M. musculus*.

1) Copy-paste your genes of interest:

2) Select databases

3) Compute



but we are going to use R…

Genes of interest

How many genes are in the table ? This is N, the number of genes in the universe.

```
dim(deseq_genes)
```

```
[1] 27655    29
```

We select differentially expressed genes. There are n genes of interest.

```
de_genes = deseq_genes[deseq_genes[, "padj"] <= 0.001, ]
de_genes = de_genes[!is.na(de_genes[, "log2FoldChange"]), ]
dim(de_genes)
```

```
[1] 1807     27
```

Enrichment analysis using the GO:CC database

We would like to perform the ORA against the gene set in the **Gene Ontology, Cellular Components** gene sets database, which is stored in the org.At.tair.db database.

```
ego = clusterProfiler::enrichGO(
    gene = de_genes$Id,             # gene list
    universe = deseq_genes$Id,      # all genes
    OrgDb = org.At.tair.db,         # annotation
    keyType = "TAIR",               # nature of the genes ID
    ont = "CC",                     # Cellular Components
    pvalueCutoff = 1,               # significance threshold (take all)
    pAdjustMethod = "BH",           # p-value adjustment method
    readable = TRUE                 # For human beings
    )
```

Enrichment analysis using the GO:CC database

What is stored in the ego object ?

```
View(ego)
```

```
head(ego@result, 3)
```

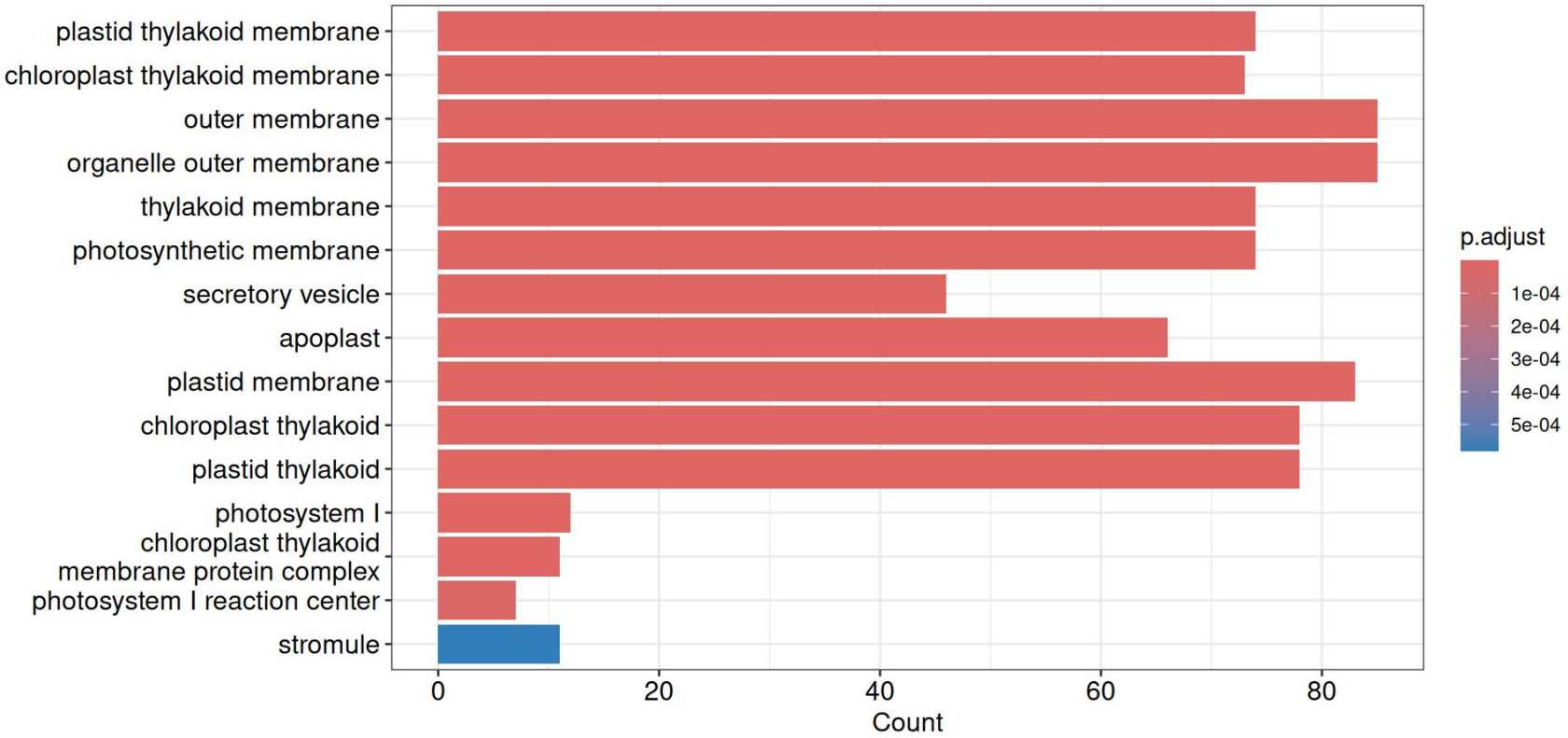
```
                   ID                    Description GeneRatio   BgRatio  RichFactor FoldEnrichment     zScore        pvalue
GO:0055035 GO:0055035     plastid thylakoid membrane   74/1785 357/26909   0.2072829       3.124804  10.772584  1.033262e-18
GO:0009535 GO:0009535 chloroplast thylakoid membrane   73/1785 349/26909   0.2091691       3.153238  10.792105  1.042123e-18
GO:0019867 GO:0019867                 outer membrane   85/1785 477/26909   0.1781971       2.686333   9.904966  5.207312e-17
               p.adjust       qvalue     geneID     Count
GO:0055035 1.443340e-16 1.310881e-16        ...        74
GO:0009535 1.443340e-16 1.310881e-16        ...        73
GO:0019867 3.606064e-15 3.275125e-15        ...        85
```

Visualization

We want to visualize these results. Let's try two visualization methods.

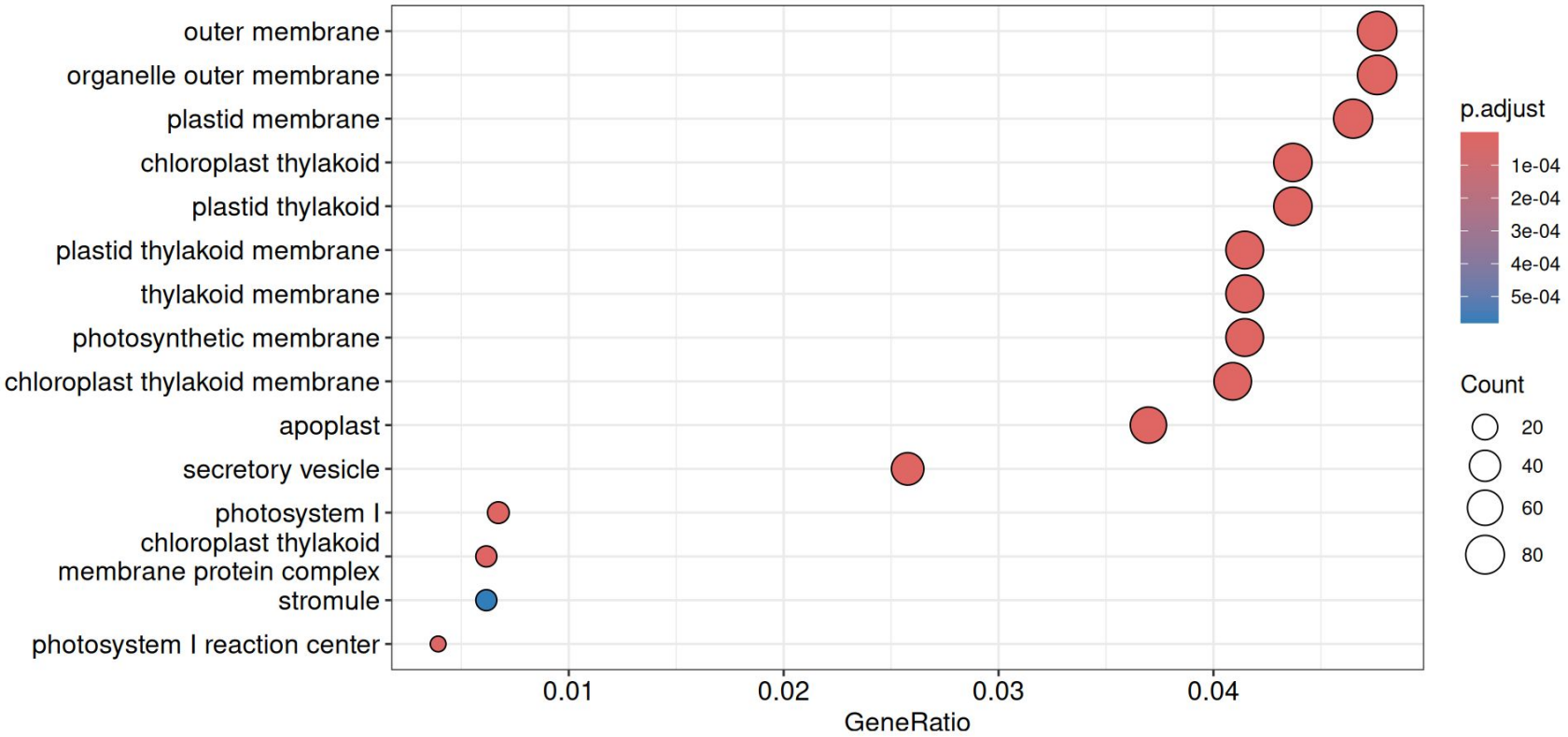
```
graphics::barplot(ego,
                  showCategory = 15)

enrichplot::dotplot(ego,
                    showCategory = 15)
```

Visualization : Barplot



The results may change depending on the packages version.

Visualization : Dotplot



The results may change depending on the packages version.

What about roots ?

We are looking for enrichment in “root” terms. Are they in the output ?

```
grep(ego@result$Description,
     pattern = "root",
     value = TRUE)
```

```
[1] "root hair"
```

```
ego@result[ego@result$Description == "root hair", ]
```

```
                   ID Description GeneRatio    BgRatio RichFactor FoldEnrichment   zScore
GO:0035618 GO:0035618   root hair      5/1785  23/26909  0.2173913       3.277189  2.912161
                pvalue   p.adjust      qvalue                                geneID  Count
GO:0035618 0.01573318   0.1562631   0.1419224     MATE/ATCNGC6/PRX44/AtSFH1/PRX73      5
```

Enrichment analysis using the GO:BP database

We would like to perform the ORA against the gene set in the **Gene Ontology, Biological Processes** gene sets database, which is stored in the org.At.tair.db database.

```
ego = clusterProfiler::enrichGO(
    gene = de_genes$Id,                 # gene list
    universe = deseq_genes$Id,          # all genes
    OrgDb = org.At.tair.db,             # annotation
    keyType = "TAIR",                   # nature of the genes ID
    ont = "BP",                         # Biological Processes
    pvalueCutoff = 1,                   # significance threshold (take all)
    pAdjustMethod = "BH",               # p-value adjustment method
    readable = TRUE                     # For human beings
    )
```

Roots are there !

We are looking for enrichment in “root” terms. Are they in the output ?

```
root_names = grep(ego@result$Description,
                  pattern = "root",
                  value = TRUE)
root_names
```

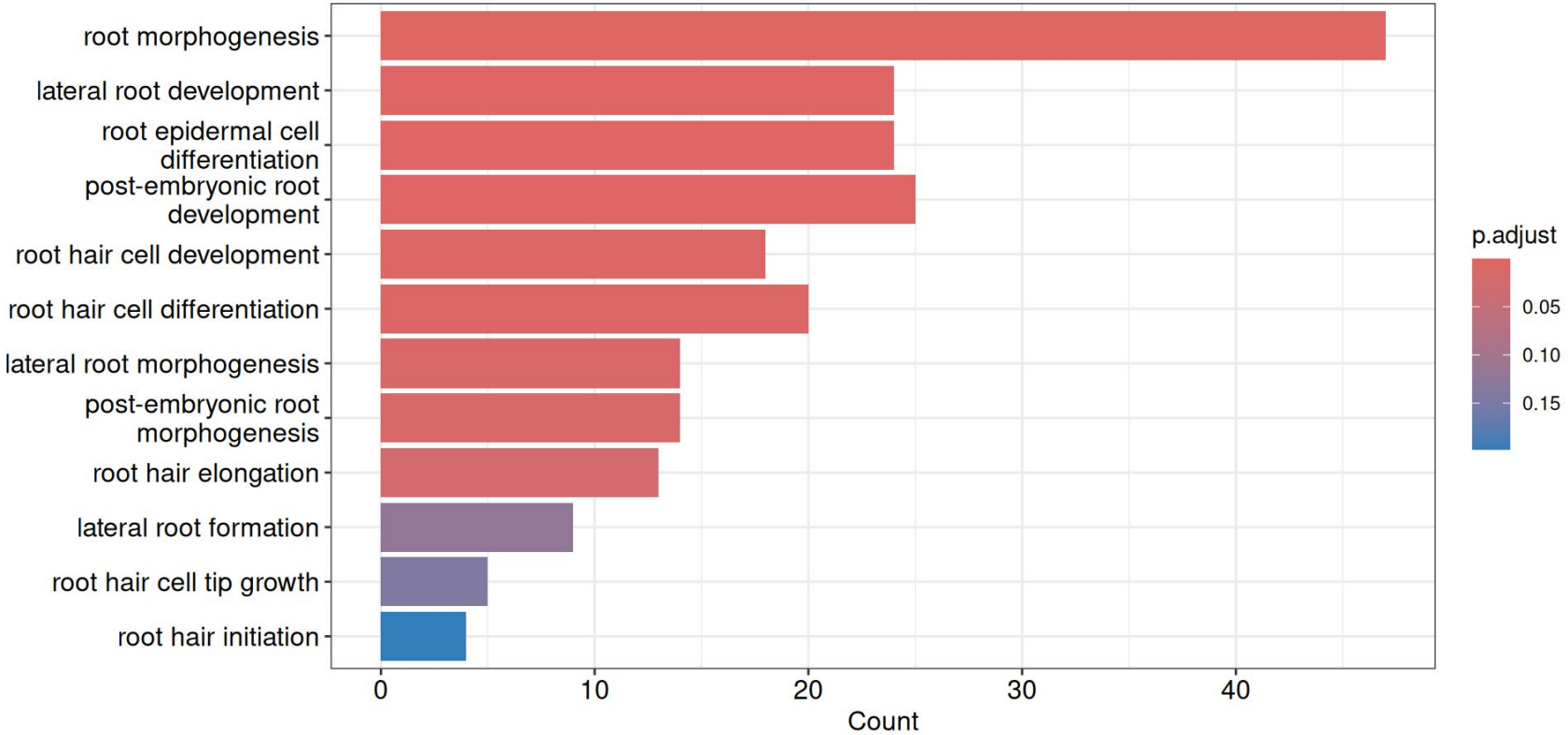
```
 [1] "root morphogenesis"                        "lateral root development"
 [3] "root epidermal cell differentiation"       "post-embryonic root development"
 [5] "root hair cell development"                "root hair cell differentiation"
 [7] "lateral root morphogenesis"                "post-embryonic root morphogenesis"
 [9] "root hair elongation"                      "lateral root formation"
[11] "root hair cell tip growth"                 "root hair initiation"
[13] "regulation of root meristem growth"        "root meristem growth"
[15] "primary root development"                  "regulation of lateral root development"
[17] "regulation of root development"            "root cap development"
[19] "regulation of post-embryonic root development" "regulation of root morphogenesis"
[21] "maintenance of root meristem identity"
```

Make the graphs for root-related terms

The `showCategory` can be either a number of gene sets to display or the specific names of gene sets of interest.

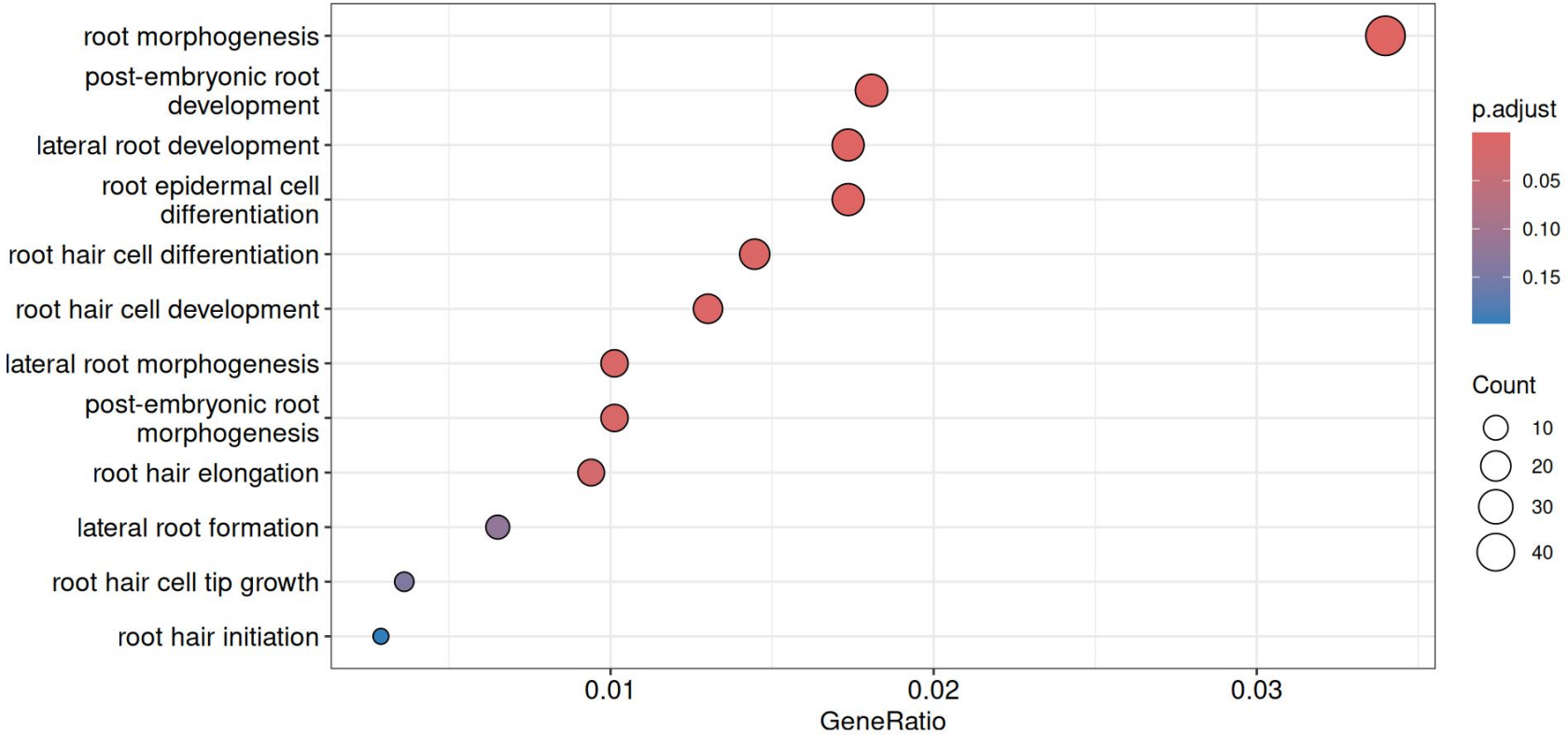
```
graphics::barplot(ego,
                  showCategory = root_names)

enrichplot::dotplot(ego,
                    showCategory = root_names)
```

Visualization : Barplot



The results may change depending on the packages version.

Visualization : Dotplot



The results may change depending on the packages version.

Summary

We used our genes of interest (differentially expressed) and gene sets from a database.

However, we do not know:

- if the gene sets are enriched in the WT or in the KO
- if the gene sets contain highly (or lowly) differentially expressed genes

In the next analysis, the genes will be ranked by order of *importance*.

Gene Set Enrichment Analysis

Gene Set Enrichment Analysis

To perform a Gene Set Enrichment Analysis (GSEA), we need to give "a list of weighted ranked genes in order to compute a running enrichment score."

```
colnames(deseq_genes)
```

```
[1] "Id"        "WT1"         "WT2"         "WT3"         "KO1"         "KO2"
[7] "KO3"       "norm.WT1"    "norm.WT2"    "norm.WT3"    "norm.KO1"    "norm.KO2"
[13] "norm.KO3"   "baseMean"    "WT"        "KO"        "FoldChange" "log2FoldChange"
[19] "stat"       "pvalue"      "padj"      "dispGeneEst"     "dispFit"     "dispMAP"
[25] "dispersion"     "betaConv"        "maxCooks"
```

Using `KO` and `WT` as weights

We have to weight each genes.

We could use the columns `WT` and `KO`, running twice the GSEA, and comparing the enrichment scores. It works, it is used in current publications. Highly expressed genes have a very very very high impact on the enrichment score.

By doing so, we could conclude something like:

> "Root morphogenesis has a higher/lower enrichment score in WT rather than in KO."

Using `log2FoldChange` as weights

We have to weight each genes.

We could use the column `log2FoldChange` and look at the enrichment score.

By doing so, we could conclude something like:

> "Root morphogenesis has up/down regulated genes with an enrichment score of XXX." or
> "Genes in Root morphogenesis are usually up/down regulated in KO plants."

Note: We <u>do not use `FoldChange`</u> to perform a GSEA because they are all > 0 and we will always see an enrichment.

Using `pvalue` as weights

NO ! NO ! USE ADJUSTED P-VALUES !

Using `padj` as weights

We have to weight each genes.

We could use the column `padj` and look at the enrichment score.

It works, but almost never published since it answers the very same questions as ORA:

“Does Root morphogenesis contains differentially expressed genes in an unusual quantity ?”

Using `stat` as weights

We have to weight each genes.

We could use the column `stat` and look at the enrichment score.

Briefly, `stat` considers both the `log2FoldChange` and the `padj`. It answers the very same question as `log2FoldChange` weights, but includes:

- the confidence we have in the differential expression between KO and WT, and,
- the change of expression between conditions.

This is almost never done, but fellow bio-statisticians tell me it is better than `log2FoldChange` alone.

We are going to use stat today, because we trust bio-statisticians.

A ranked list of genes of interest

We prepare the data:

```
# Get the weights
geneList = as.numeric(de_genes$stat)

# Get genes identifiers
names(geneList) = de_genes$Id

# Sort the list
geneList = sort(geneList, decreasing = TRUE)

head(geneList)
```

```
AT2G17820 AT5G19600 AT2G25760 AT3G19670 AT3G48110 AT5G11800
   18.377    16.078    16.002    15.616    15.249    14.443
```

GSEA using the GO:BP database

We would like to perform the GSEA against the gene set in the **Gene Ontology, Biological Processes** gene sets database, which is stored in the org.At.tair.db database.

```
gsea = clusterProfiler::gseGO(
    geneList = geneList,                # ranked gene list
    ont = "BP",                         # Biological Processes
    OrgDb = org.At.tair.db,             # annotation
    keyType = "TAIR",                   # nature of the genes ID
    pAdjustMethod = "BH",               # p-value adjustment method
    pvalueCutoff = 1,                   # significance threshold (take all)
    seed = 1                            # fix randomness for permutations
    )
```

⚠️ Very very (very) important to set a seed if you want replicable results.

GSEA using the GO:BP database

What is stored in the gsea object ?

```
View(gsea)
```

```
head(gsea@result, 3)
```

```
                   ID                                        Description  setSize  enrichmentScore      NES  pvalue     p.adjust
GO:0090304 GO:0090304                     nucleic acid metabolic process      269        0.3245700 3.171488   1e-10  7.6125e-09
GO:0006139 GO:0006139  nucleobase-containing compound metabolic process      313        0.3082655 3.092426   1e-10  7.6125e-09
GO:0016070 GO:0016070                             RNA metabolic process      240        0.3140519 3.004704   1e-10  7.6125e-09
                  qvalue  rank                        leading_edge   core_enrichment
GO:0090304 4.565789e-09   710     tags=62%, list=39%, signal=45%               ...
GO:0006139 4.565789e-09   710     tags=60%, list=39%, signal=44%               ...
GO:0016070 4.565789e-09   635     tags=57%, list=35%, signal=42%               ...
```

The values may change due to distinct parameter values.

Exploration of over-represented gene sets

Let's see the top 8 of the over-represented gene sets:

```
gsea@result %>%
  dplyr::filter(p.adjust < 0.05) %>%
  dplyr::top_n(., n = 8, wt = abs(NES)) %>%
  dplyr::select(Description, NES, p.adjust, setSize)
```

	Description	NES	p.adjust	setSize
GO:0090304	nucleic acid metabolic process	3.171488	7.612500e-09	269
GO:0006396	RNA processing	3.347367	2.342322e-08	49
GO:0016071	mRNA metabolic process	3.384434	2.939140e-08	44
GO:0008380	RNA splicing	3.256452	5.343321e-07	25
GO:0000375	RNA splicing, via ...	3.174464	1.686242e-06	23
GO:0000377	RNA splicing, via ...	3.174464	1.686242e-06	23
GO:0000398	mRNA splicing, via spliceosome	3.174464	1.686242e-06	23
GO:0006397	mRNA processing	3.121489	2.064175e-06	29

The values may change due to distinct parameter values.

Gene sets related to roots ?

We filter the results for gene sets containing “root”:

```
root_names = grep(gsea@result$Description,
                  pattern = "root",
                  value = TRUE)

gsea@result %>%
  dplyr::filter(p.adjust < 0.05) %>%
  dplyr::filter(Description %in% root_names) %>%
  dplyr::top_n(., n = 8, wt = abs(NES)) %>%
  dplyr::select(Description, NES, p.adjust, setSize)
```

```
                                   Description        NES    p.adjust   setSize
GO:0010053 root epidermal cell differentiation -1.992772  0.01912029        24
GO:0048765      root hair cell differentiation -1.801858  0.04573408        20
```

The values may change due to distinct parameter values.

Visualization

We visualize the GSEA curve using the function `gseaplot2` from the package `enrichplot`.

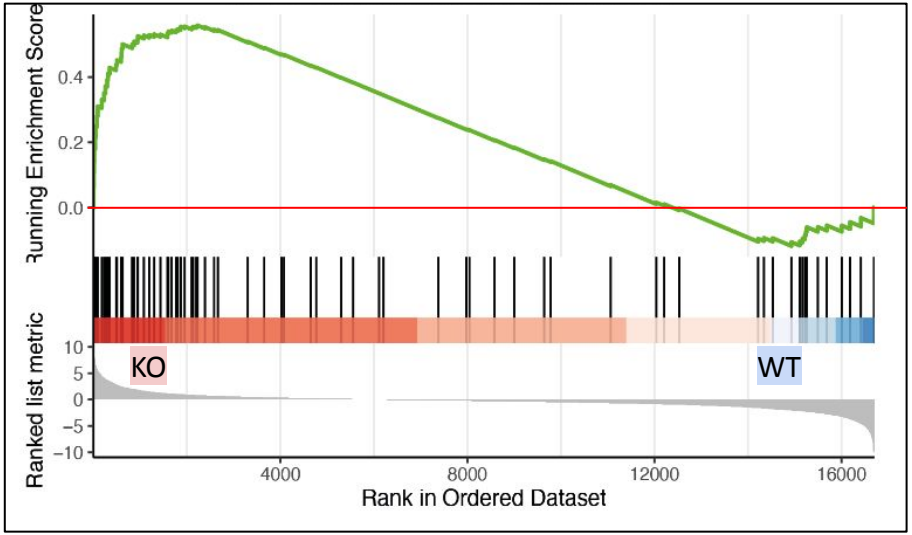
```
gene_set_name = "root hair cell differentiation"
gene_set_id = which(gsea@result$Description == gene_set_name)
gene_set_id
```

```
[1] 207
```

```
enrichplot::gseaplot2(
    x = gsea,
    geneSetID = gene_set_id,
    title = gene_set_name
)
```

The values may change due to distinct parameter values.

Understand the GSEA plot



Understand the GSEA plot

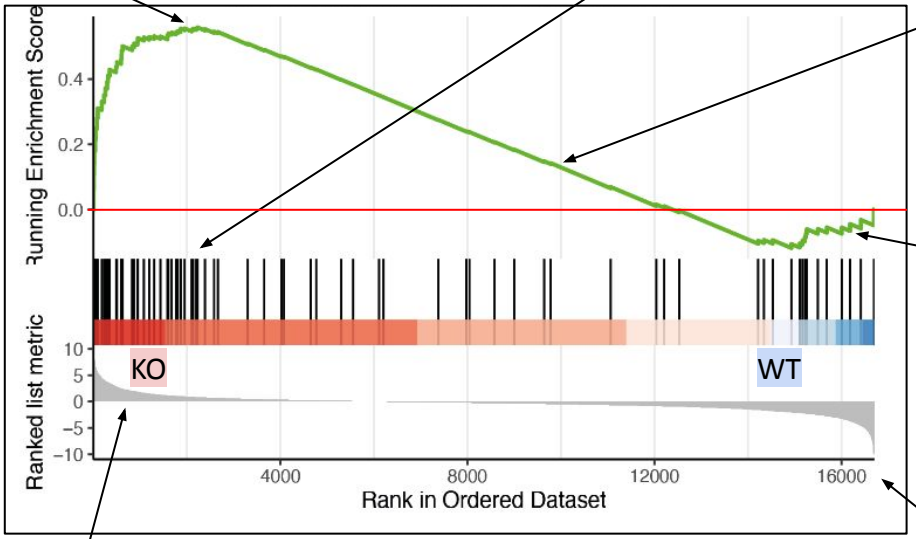
The maximum of the curve defines the **enrichment score** (**ES**) of the gene list ordered in the gene set.
Here, ES ≈ 0.5.

The algorithm computes the ES for 1000 other ordered lists. These ordered lists are obtained by performing **permutations** in the input list.

An **enrichment p-value** is calculated by comparing the ES value to the distribution of the other 1000 ES obtained from the permutations.

The ES is normalized to a **normalized ES** (**NES**), by dividing it by the average of the ES obtained after the permutations.

The algorithm goes through the list, in order of values. Each time it encounters a gene belonging to the gene set, the (green) **enrichment curve** rises.
The gene is marked with a **black line** at its rank. Otherwise, the curve goes down.



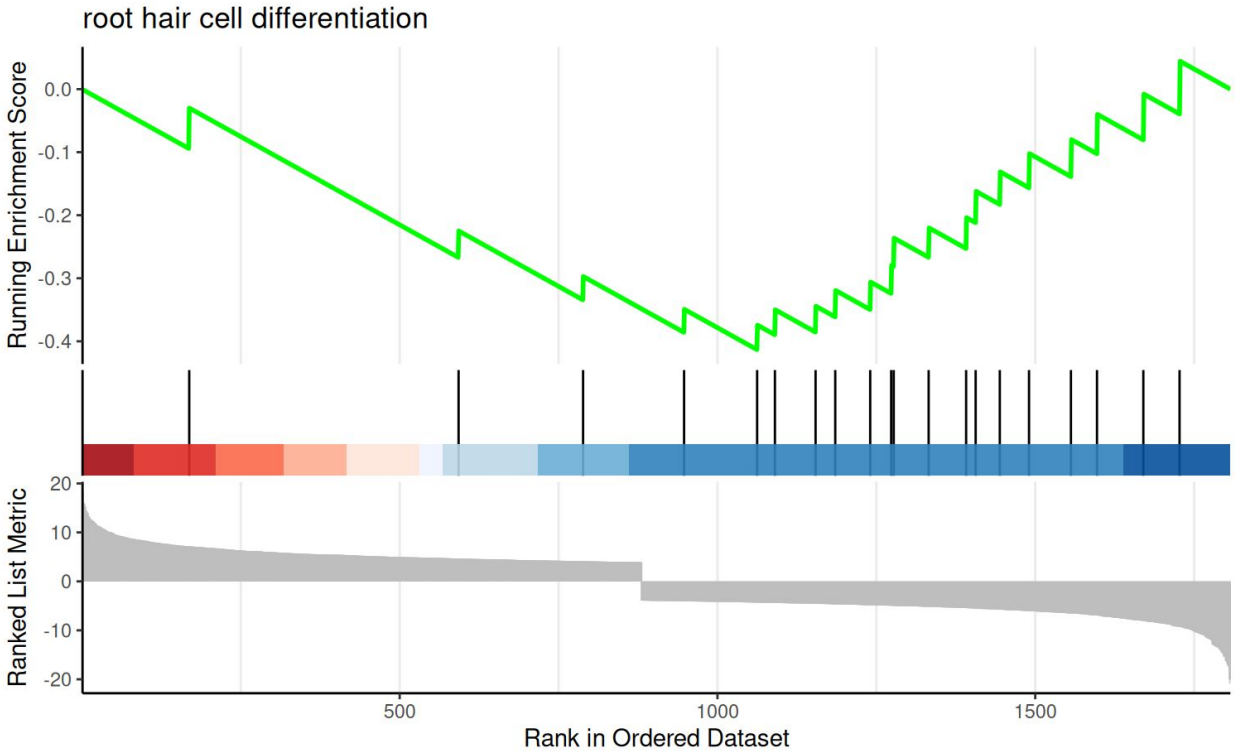
The curve goes up. It means that some genes of the set of genes are enriched in the second condition (WT here).

The genes from the input list are ordered by a numerical value of interest. Here, the log fold change was chosen.

The list contains over 16,000 genes.

Visualization

We visualize the GSEA curve associated with our gene set of interest.



Nota bene

With GSEA, you dot not test if a pathway is up or down regulated.

A pathway contains both enhancers and suppressors genes. An up-regulation of enhancer genes and a down-regulation of suppressor genes will lead to a “bad” enrichment score. However, this will lead to a strong change in your pathway activity !

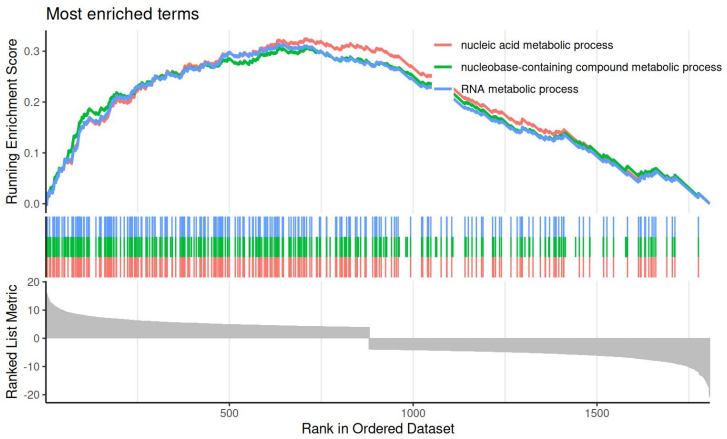
If your favorite pathway does not have a “good enrichment score”, it does not mean that pathway is not affected.

Bonus

Multiple GSEA curves on the same graph

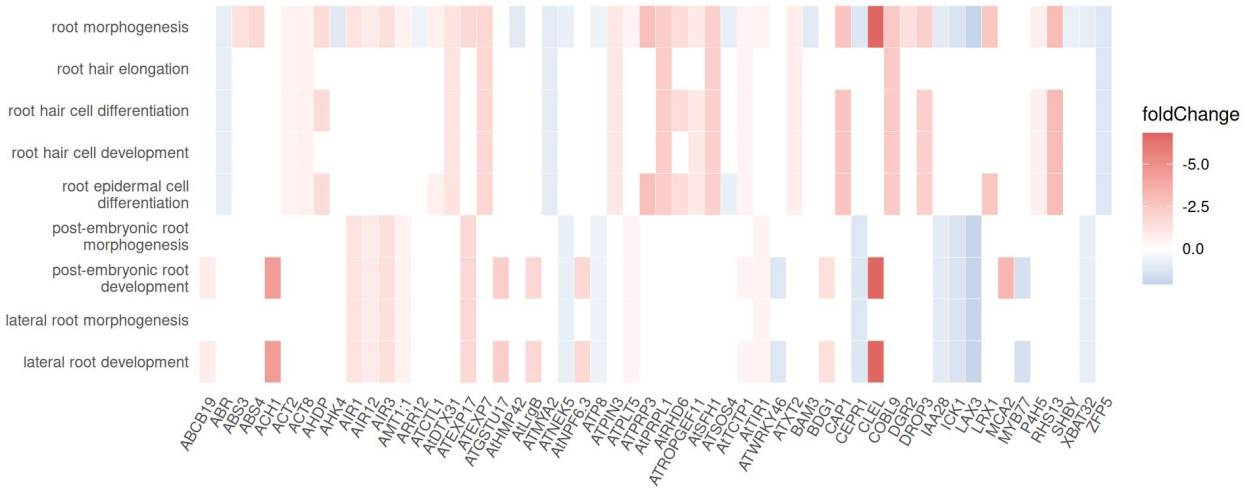
We can visualize (but not read ?) multiple results on the same graph.

```
enrichplot::gseaplot2(
    x = gsea,
    geneSetID = c(1:3),
    title = "Most enriched terms"
)
```



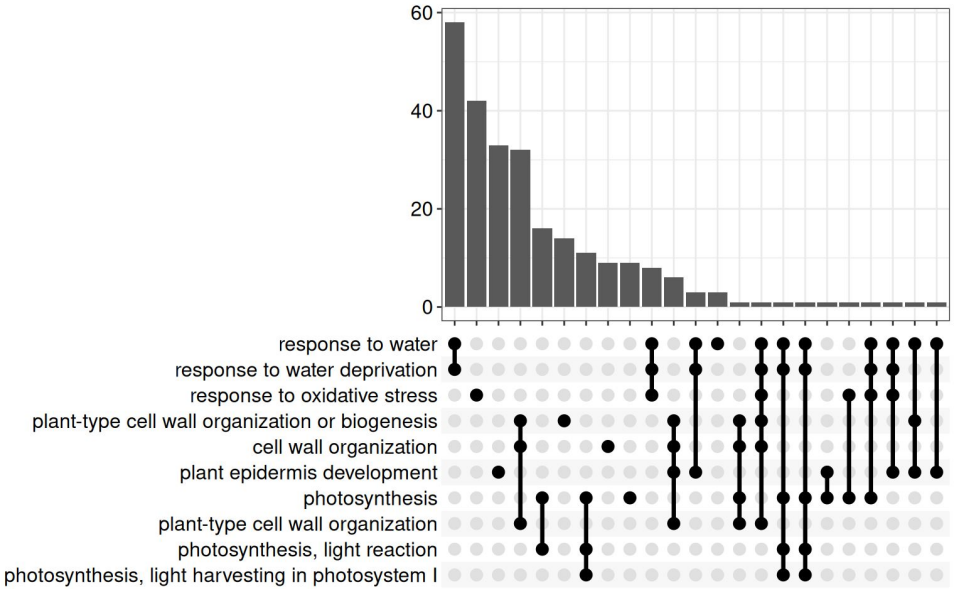
Oncoplot / Heatmap

```
enrichplot::heatplot(
  x = ego,                                          # Our ORA
  showCategory = root_names,                        # Gene sets of interest
  foldChange = setNames(nm = de_genes$Id,
                        de_genes$log2FoldChange)# Our fold changes
)
```



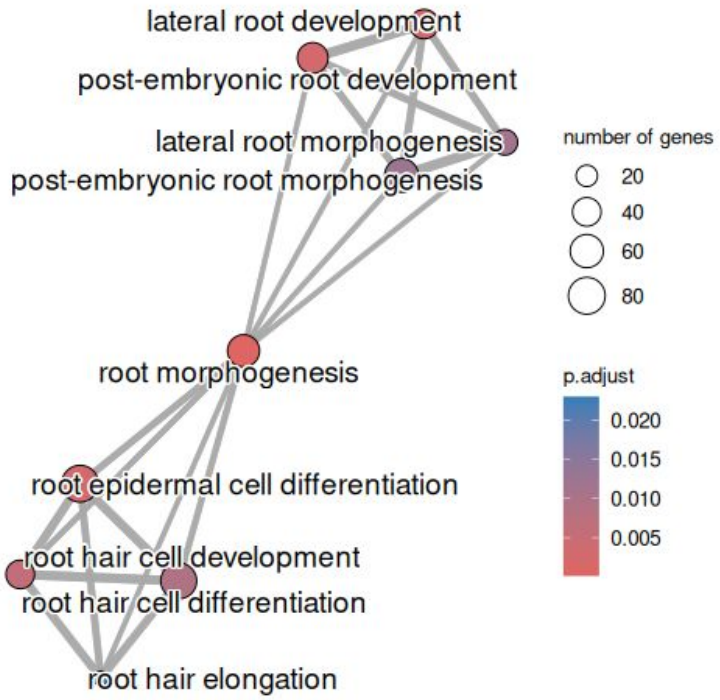
Upset plot

```
ego = enrichplot::pairwise_termsim(ego)

enrichplot::upsetplot(x = ego,      # Our ORA
                      n = 10)       # Nb of terms to display
```



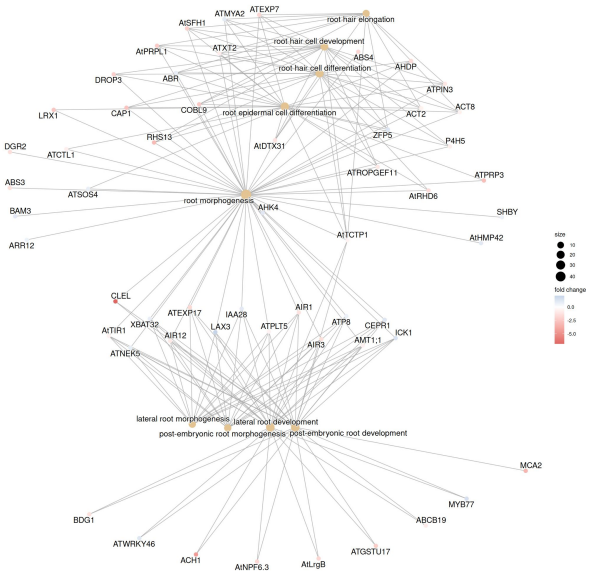
Enrichment map

```
enrichplot::emapplot(x = ego, showCategory = root_names)
```



Gene-concept network

```
enrichplot::cnetplot(ego,
                     showCategory = root_names,
                     foldChange = setNames(nm = de_genes$Id,
                                           de_genes$log2FoldChange))
```



And more…

Look at:

https://yulab-smu.top/biomedical-knowledge-mining-book/enrichplot.html