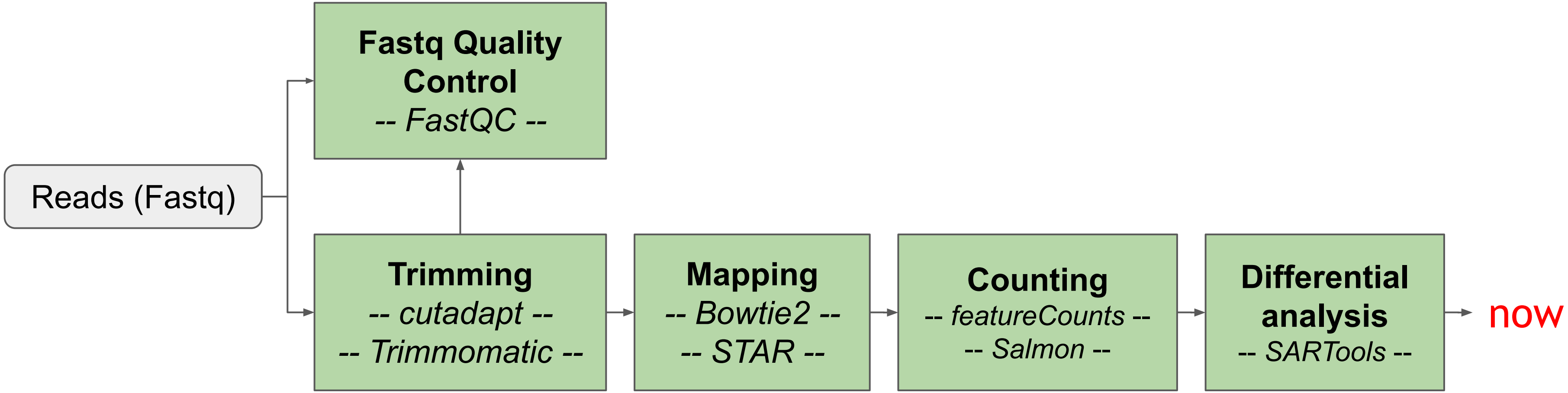
Gene Set Analysis

Thibault Dayris, Jean-Pascal Meneboo, Audrey Onfroy

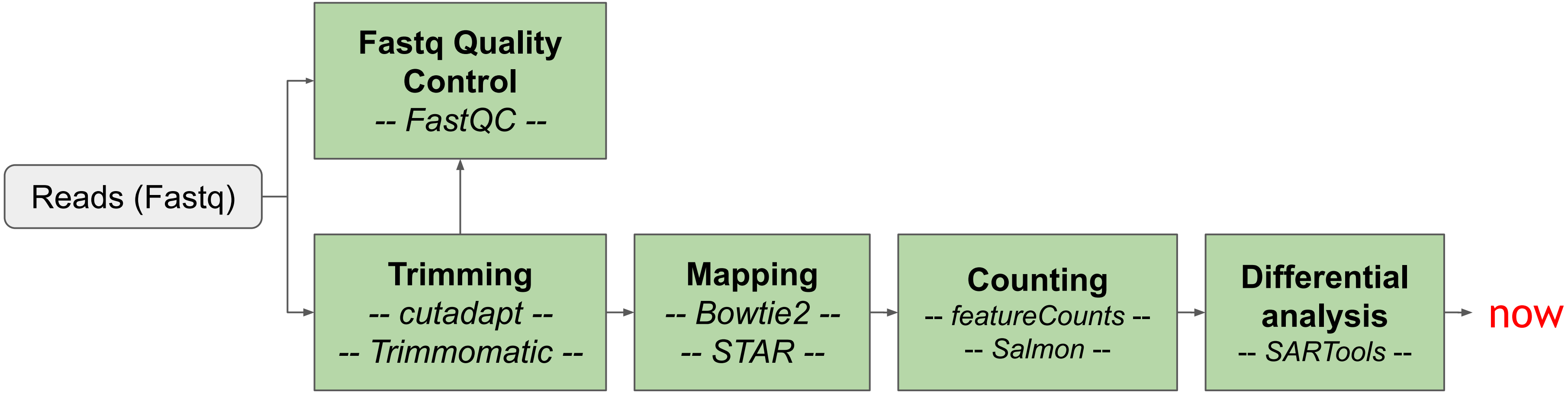
École de Bioinformatique IFB-INSERM-INRAE 2025

So far...



What is the **biology** behind the **differentially expressed genes** ?

So far...



What is the **biology** behind the **differentially expressed genes** ?

May I review the literature for each of the 250 genes ??

Database

Prerequisite - a **database** of gene sets

a **gene set** = a group of genes sharing a common feature:

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

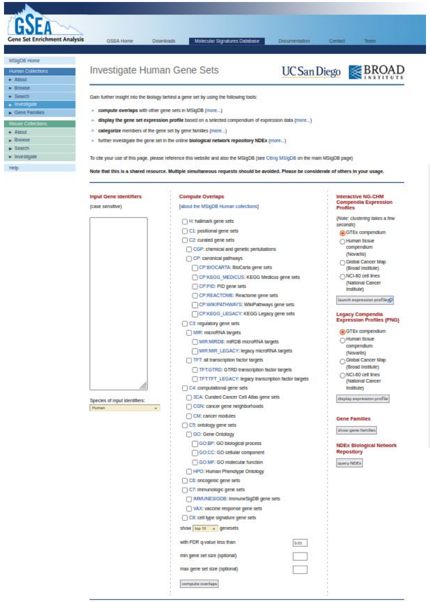
We never look at everything: subset the **database** based on the **biological questions** !

Which databases ?

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

There are many many (*many*) databases - always related to an **organism**

Eg. the **M**olecular **Sig**nature **D**ata**b**ase (MSigDB)
(for *H. sapiens* and *M. musculus*)



- KEGG
- Reactome
- WikiPathways
- Gene Ontology (GO)
- Molecular Functions (MF)
  - Cellular Components (CC)
  - Biological Processes (BP)
- …

They (may) store redundant informations.

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

Which databases ?

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

Organism 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: arab

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)

PreviousNext

Practical session

Copy the support to your folder

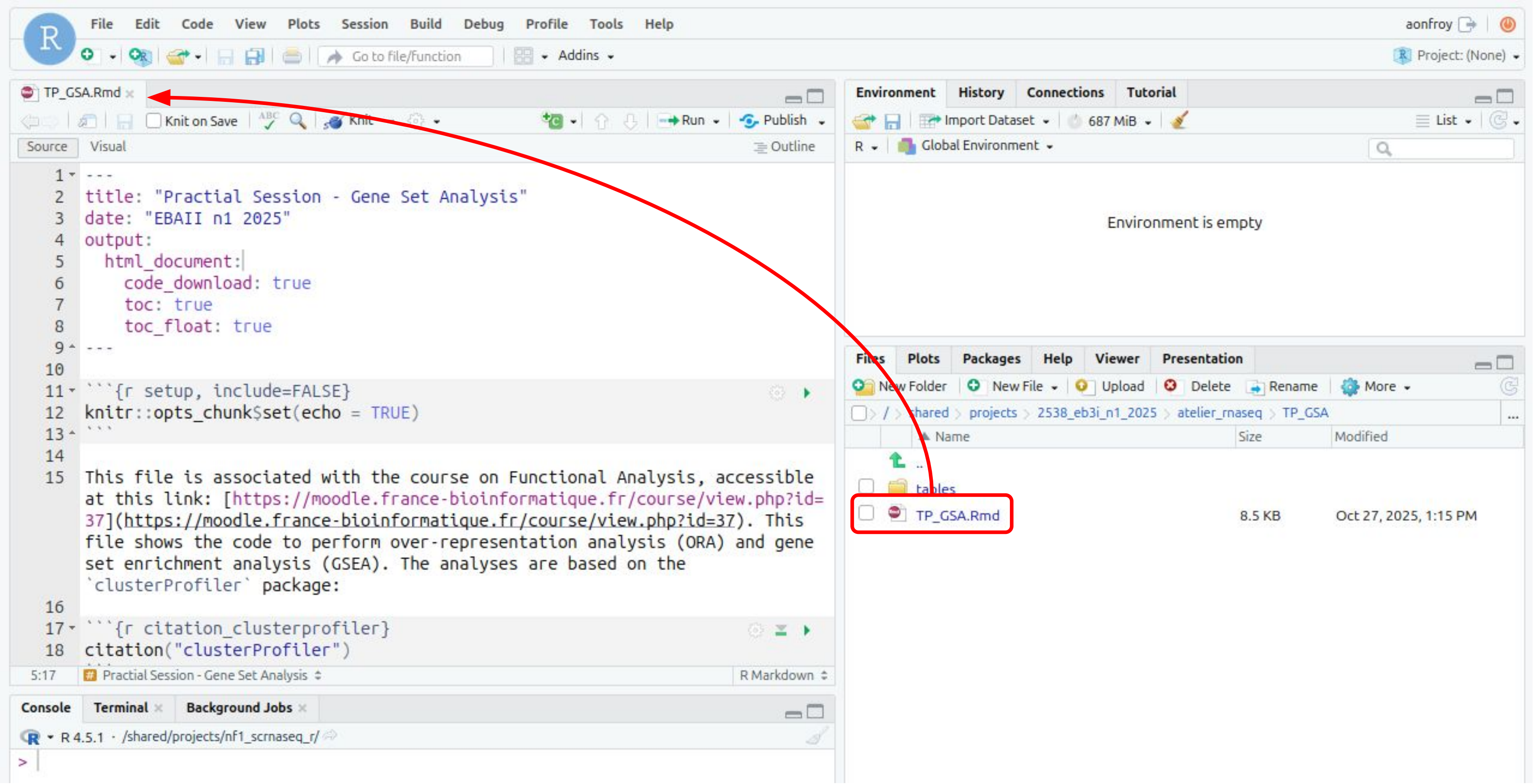
Execute the commands below in the **terminal**, to:

1) make a dedicated directory in your own project
2) copy-paste the course materials

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

cp -r /shared/projects/2538_eb3i_n1_2025/atelier_rnaseq/TP_GSA/*
      /shared/projects/<YOUR_PROJECT>/TP_GSA
```

Open RStudio and the TP_GSA.Rmd file



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"
```

Understand the data

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`).

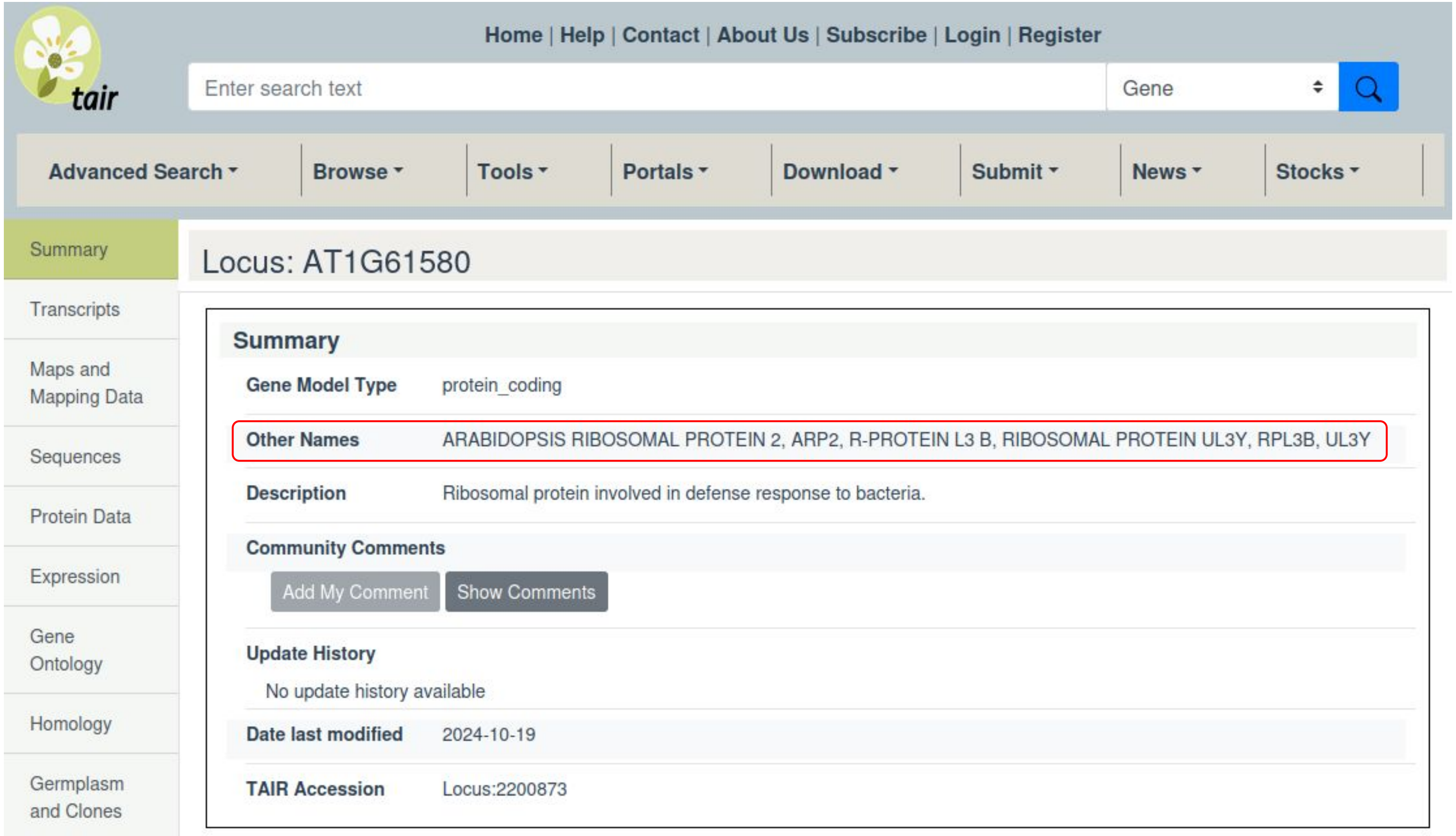
The **fold change** in expression is `0.588` (= `128`/`218`).

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.

Gene names and identifiers

Name associated with AT1G61580 Id ?

arabidopsis.org



arabidopsis.org

Id associated with ARP2 name ?

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=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** across 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*
Usage	- lab meeting - nice-looking graphs	- analysis - interaction with database

*more information about the conversion in the supplementary slides

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"
```

Data is **ready** for gene set analysis !

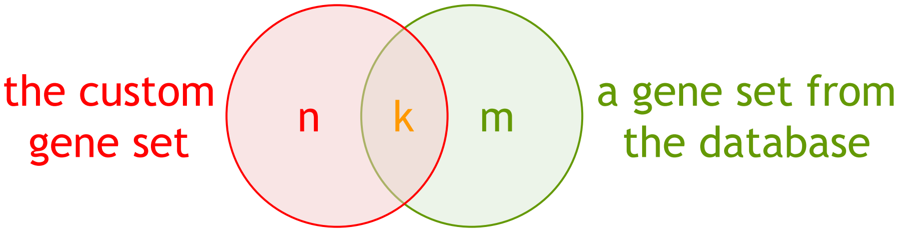
Two types of gene set analysis

Over-Representation Analysis (ORA)

Input: database + a custom gene set

- up <u>OR</u> down-regulated genes (filtered !)
- gene identifiers only

Principle: Assess if the custom gene set contains a lot of genes (or not) defined by each gene set from the database



Gene Set Enrichment Analysis (GSEA)

Input: database + a custom **ranked** gene set

- up <u>AND</u> down-regulated genes (all !)
- gene identifiers and gene weights

Principle: Assess if a gene set from the database is more represented at the top or bottom of the custom ranked gene set list.

Over **R**epresentation **A**nalysis

Genes of interest

How many genes are in our data ?

```
nrow(deseq_genes)
```

```
[1] 27655
```

We select differentially expressed genes.

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

```
[1] 1807
```

Genes of interest

The 1807 genes correspond to up- or down-regulated genes.

```
summary(de_genes$log2FoldChange)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-8.4190 -1.5990 -0.5370 -0.3679  1.0510  6.6990
```

We keep the up-regulated genes only:

```
de_genes = de_genes[de_genes[, "log2FoldChange"] > 0, ]

nrow(de_genes)
```

```
[1] 880
```

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
    )
```

Enrichment analysis using the GO:BP database

What is stored in the ego object ?

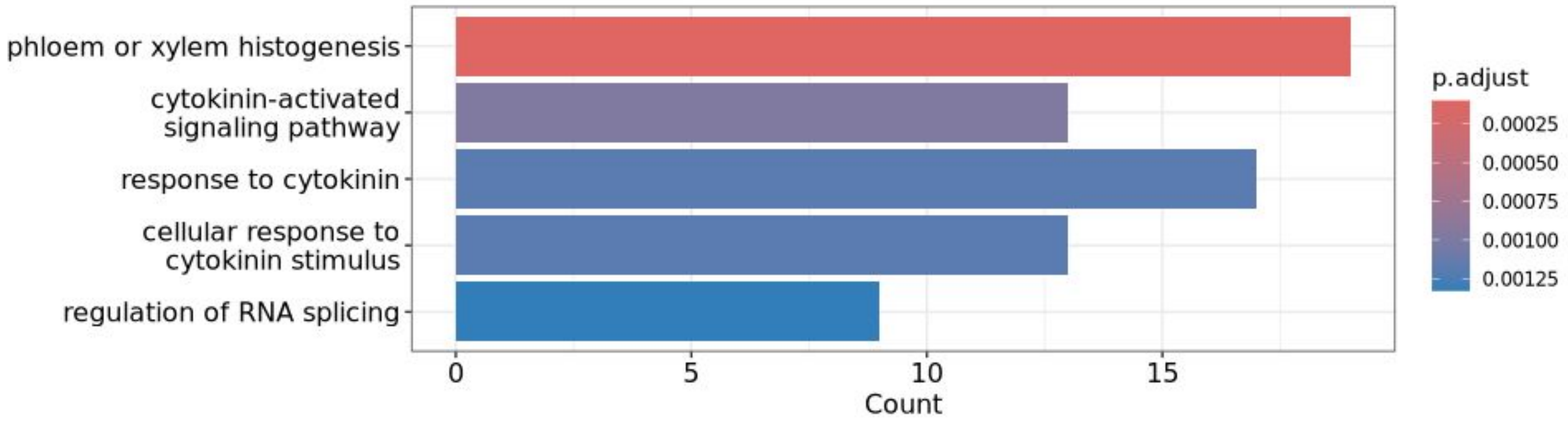
```
View(ego)
```

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

```
                   ID                          Description GeneRatio   BgRatio RichFactor
GO:0010087 GO:0010087         phloem or xylem histogenesis    19/711 130/21364  0.1461538
GO:0009736 GO:0009736 cytokinin-activated signaling pathway    13/711  76/21364  0.1710526
GO:0009735 GO:0009735                 response to cytokinin    17/711 134/21364  0.1268657
           FoldEnrichment   zScore       pvalue     p.adjust       qvalue       geneID     Count
GO:0010087       4.391604 7.196732 6.233419e-08 0.0001004827 8.733348e-05          ...        19
GO:0009736       5.139759 6.707933 1.219050e-06 0.0009825546 8.539769e-04          ...        13
GO:0009735       3.812037 6.058607 2.330987e-06 0.0011947380 1.038394e-03          ...        17
```

Visualization : Barplot

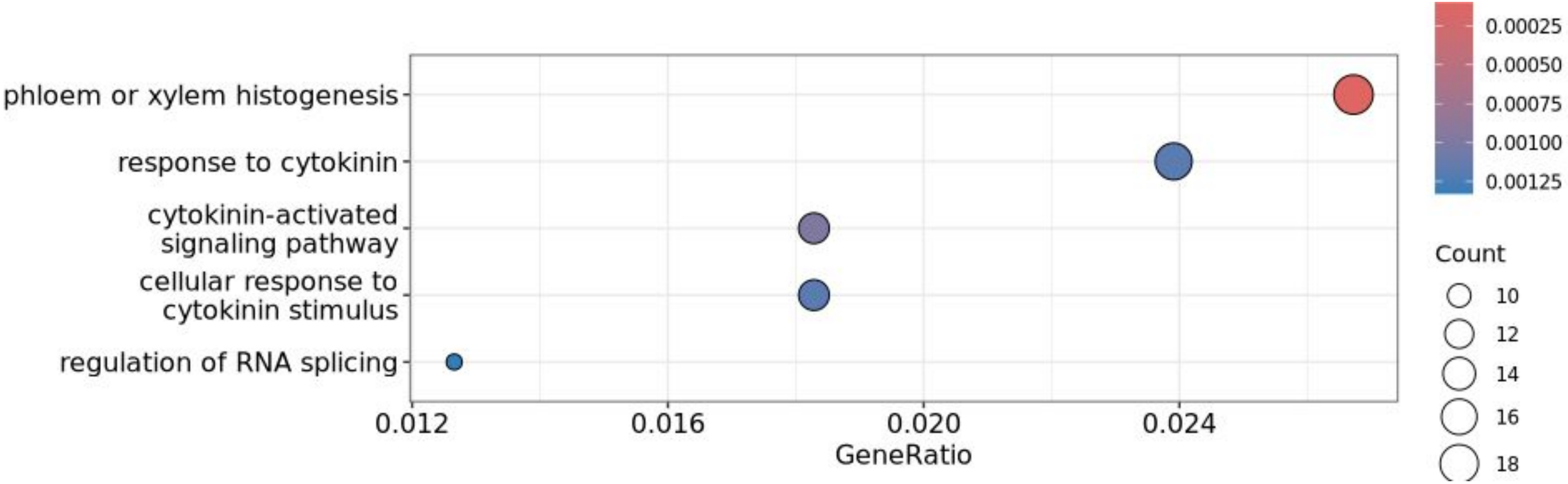
```
graphics::barplot(ego, showCategory = 5)
```



This figure does not inform on the gene set sizes.

Visualization : Dotplot

```
enrichplot::dotplot(ego, showCategory = 5)
```



What about phloem related-terms ?

We are looking for enrichment in “phloem” term. Are they in the output ?

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

```
[1] "phloem or xylem histogenesis"       "phloem transport"
[3] "xylem and phloem pattern formation" "phloem development"
```

We can subset the `ego@result` object these gene sets only, and make the associated graphics.

Summary

We used our genes of interest (up-regulated genes) and gene sets from a database.

We do not know if the gene sets are associated with the highly up-regulated genes or not.

In the GSEA, genes are **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."

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. It includes:

- the confidence (**p-value**) we have in the differential expression between KO and WT,
- the change of expression between conditions (**log2FoldChange**).

A ranked list of genes of interest

We prepare the data:

```r
# Get the weights (here, "stat")
geneList = as.numeric(de_genes$stat)

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

# Sort the list according to the weights
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) important to **set a seed** if you want replicable results !

GSEA using the GO:BP database

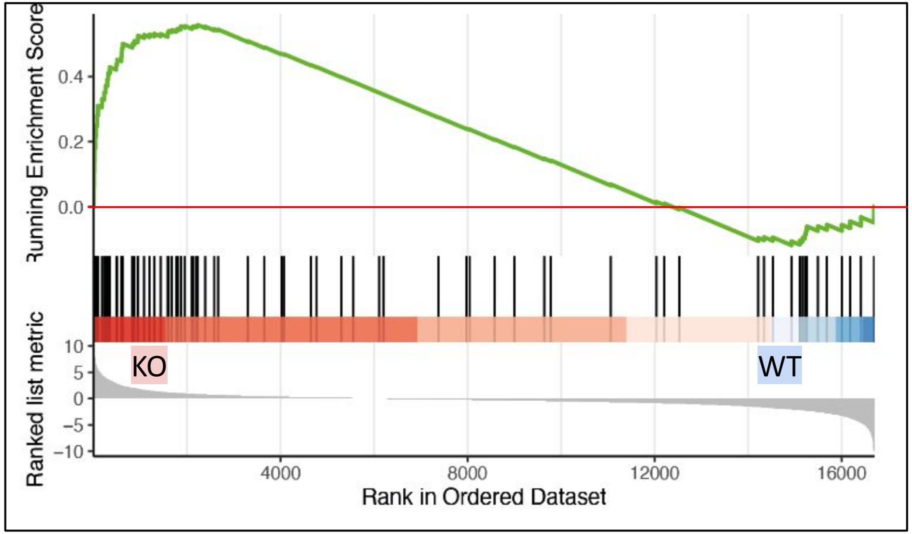
What is stored in the `gsea` object ?

```
View(gsea)
```

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

```
                   ID                                    Description setSize enrichmentScore
GO:0072522 GO:0072522 purine-containing compound biosynthetic process      11       0.6545202
GO:1901293 GO:1901293       nucleoside phosphate biosynthetic process      15       0.5816372
GO:0000375 GO:0000375 RNA splicing, via transesterification reactions      20       0.5279490
                NES       pvalue   p.adjust     qvalue rank                   leading_edge    core_enrichment
GO:0072522 2.234595 0.0002822140 0.04609509 0.03609856  115 tags=64%, list=13%, signal=56%                ...
GO:1901293 2.162722 0.0006942031 0.04609509 0.03609856  115 tags=53%, list=13%, signal=47%                ...
GO:0000375 2.158984 0.0006641180 0.04609509 0.03609856  105 tags=50%, list=12%, signal=45%                ...
```

Understand the GSEA plot



Understand the GSEA plot

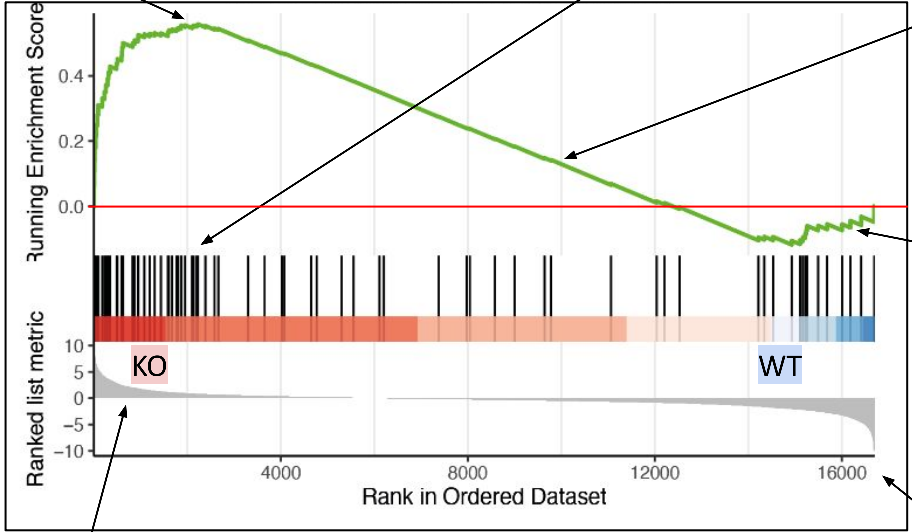
The maximum of the curve defines the **enrichment score** (**ES**) of the gene list ordered in the gene set.
Here, ES ≈ 0.5.

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.

Nota bene

With GSEA, you do 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.

GSEA plot for the 'best' gene set (1/2)

Which is the best gene set ?

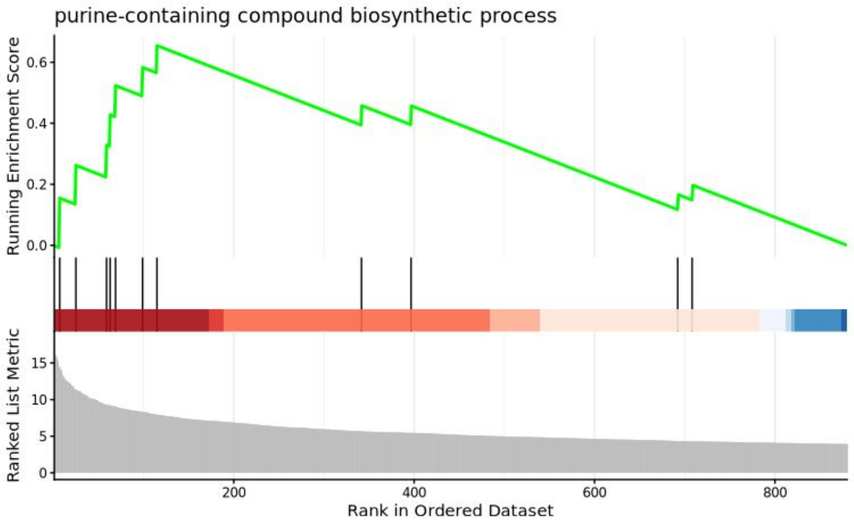
```r
top1_gsea = gsea@result %>%
  dplyr::filter(p.adjust < 0.05) %>%
  dplyr::filter(NES == max(NES)) %>%
  dplyr::select(ID, Description, NES, p.adjust, setSize)
top1_gsea
```

```
                   ID                                   Description      NES   p.adjust setSize
GO:0072522 GO:0072522 purine-containing compound biosynthetic process 2.234595 0.04609509      11
```

GSEA plot for the ‘best’ gene set (2/2)

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

```
enrichplot::gseaplot2(
  x = gsea,
  geneSetID = top1_gsea$ID,
  title = top1_gsea$Description)
```



What about gene sets related to phloem ?

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

```
phloem_names = grep(gsea@result$Description,
                    pattern = "phloem",
                    value = TRUE)

gsea@result %>%
  dplyr::filter(Description %in% phloem_names) %>%
  dplyr::select(ID, Description, NES, p.adjust, setSize)
```

```
                   ID                        Description      NES  p.adjust setSize
GO:0010051 GO:0010051 xylem and phloem pattern formation 1.389843 0.3285184      10
GO:0010087 GO:0010087       phloem or xylem histogenesis 1.121725 0.5207402      19
```

They are not significant. Why ? Let’s draw the curve.

Visualization (1/2)

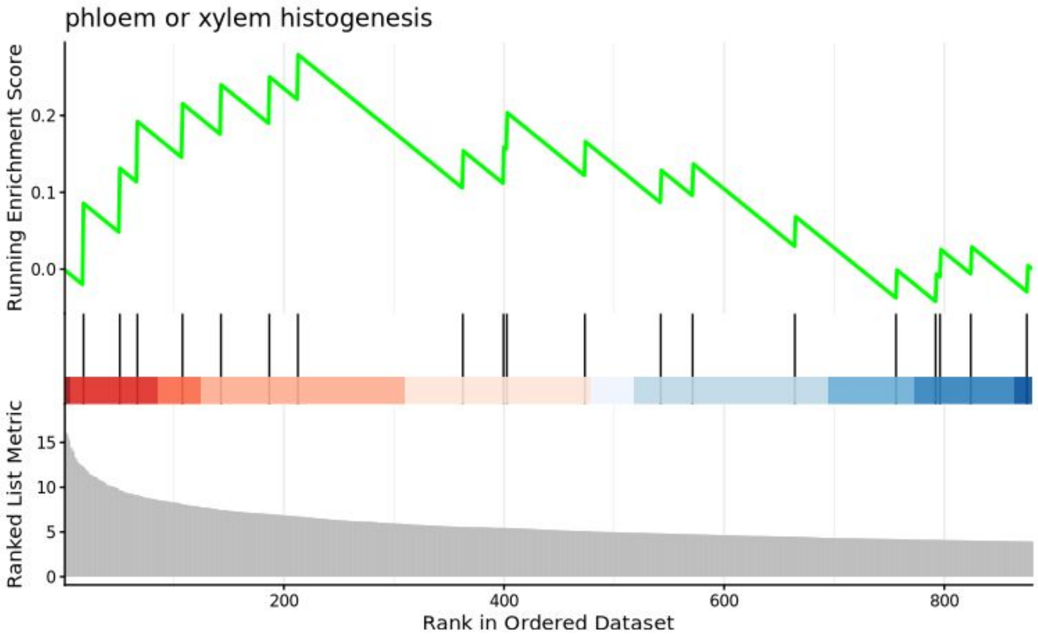
We visualize the GSEA curve using the function `gseaplot2` from the package `enrichplot`.

```
gene_set_id = "GO:0010087"
gene_set_name = gsea@result$Description[which(gsea@result$ID == gene_set_id)]

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

Visualization (2/2)

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



The GSEA results nuance the ORA results.

Conclusion

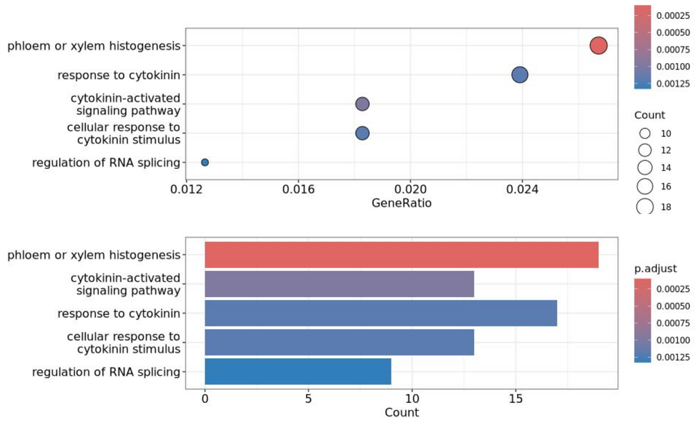
Conclusion

Over-Representation Analysis (ORA)

Input: database + a custom gene set

- up OR down-regulated genes (filtered !)
- gene identifiers only

(possible) **Output:**

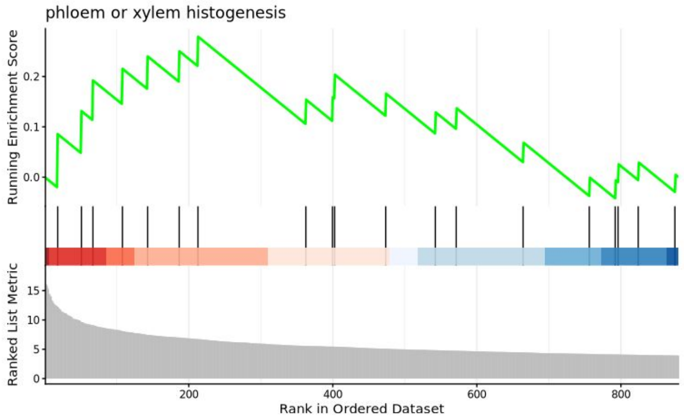


Gene Set Enrichment Analysis (GSEA)

Input: database + a custom **ranked** gene set

- up AND down-regulated genes (all !)
- gene identifiers and gene weights

(possible) **Output:**



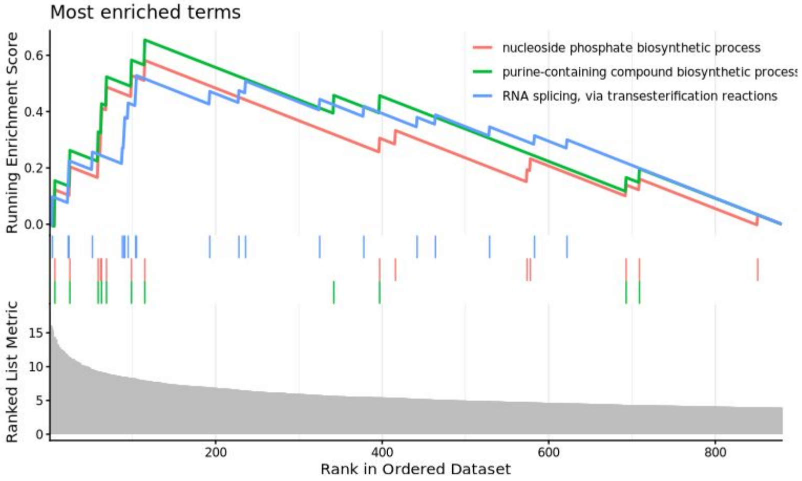
Bonus (1/2)

Code to produce specific figures you may have seen in publications.

Multiple GSEA curves on the same graph

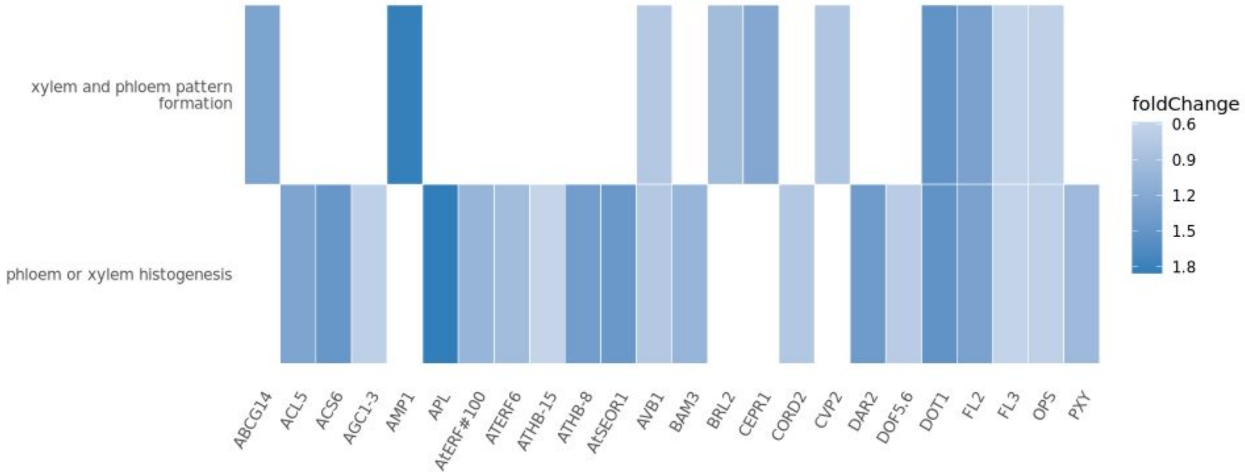
We can visualize (but not read ?) multiple results on the same graph.

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



Oncoplot / Heatmap

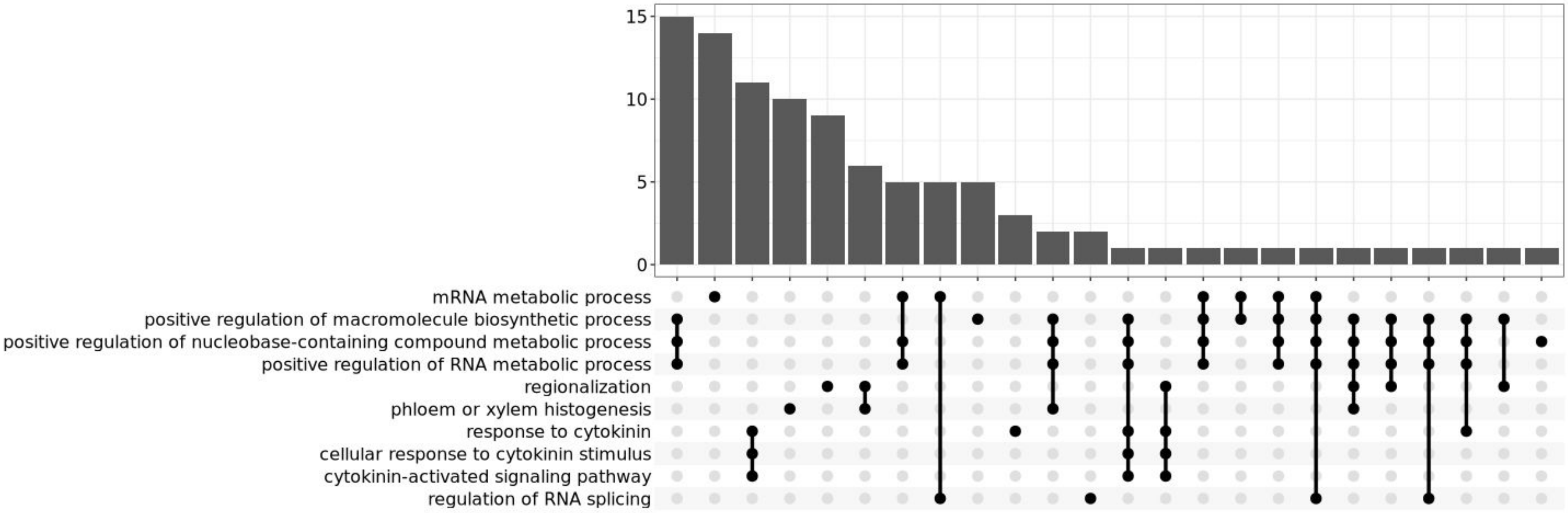
```
enrichplot::heatplot(
  x = ego,                                                   # Our ORA
  showCategory = root_names,                                 # Gene sets of interest
  foldChange = setNames(nm = de_genes$Id,
                        de_genes$log2FoldChange)# Our fold changes
)
```



Upset plot

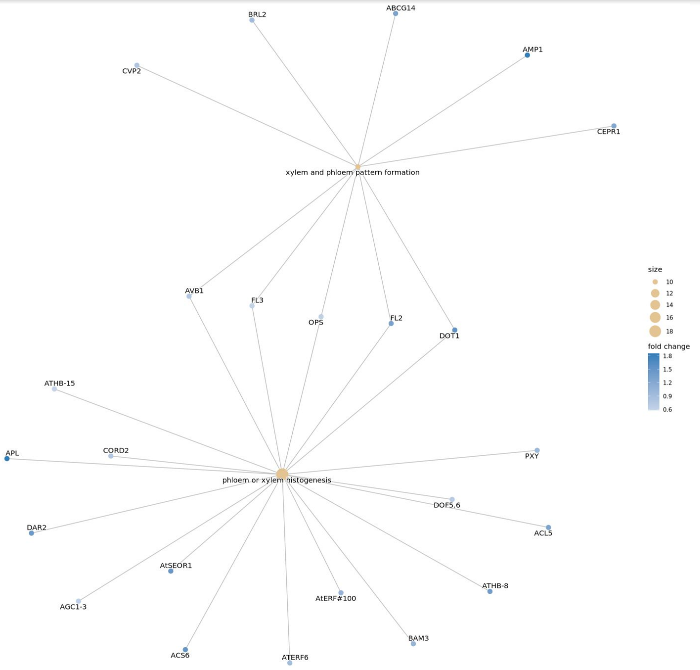
```
ego = enrichplot::pairwise_termsim(ego)

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



Gene-concept network

```r
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

Bonus (2/2)

Conversion of gene identifiers for inter-database compatibility.

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] 38169    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
```

Beware of the duplicates.

Some SYMBOL correspondence are missing

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

```
table(is.na(deseq_genes_with_symbol$SYMBOL))
```

```
FALSE  TRUE
26440 11729
```

```
table(is.na(deseq_genes_with_symbol$ENTREZID))
```

```
FALSE
38169
```

Beware if you use SYMBOL in downstream analyses. Not Available values won’t be considered.

Bonus (3/2)

Additional slides that do not fall in the 3 previous bonus sections.

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")
```