Gene Set Analysis (GSA)

RNA-Seq group, IFB-AVIESAN 2021

2021-11-20

Intro

For this session, we use the following packages:

```
library("clusterProfiler")  # Make enrichment analysis
library("limma")    # A lots of math-related operations
library("DOSE")     # Disease Ontology
library("enrichplot") # Awesome graphs
library("pathview")   # Nice pathway plot
library("org.At.tair.db")  # A. Thaliana annotation
```

So far ...

From monday we have :

1. Cleaned FastQ files (FastQC + Trimmomatic)
2. Mapped FastQ reads to the genome (STAR)
3. Estimated mapped-reads counts over the genome (FeatureCounts)
4. Analyzed differnetially expressed genes over the genome (SARTools)

We have a large table with many columns

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

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

We have a large table with many lines

```r
print(head(deseq_genes$Id))
```

```
                Id
1 gene:AT1G01010
2 gene:AT1G01020
3 gene:AT1G01030
4 gene:AT1G01040
5 gene:AT1G01050
6 gene:AT1G01060
```

Gene Identifiers

Let's search: "gene:AT1G61580"

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

```
                 Id WT1 WT2 WT3 KO1 KO2 KO3 norm.WT1
5120 gene:AT1G61580 248 231 205 119 131 125      229
     norm.WT2 norm.WT3 norm.KO1 norm.KO2 norm.KO3 baseMean
5120      210      215      131      131      123   173.19
      WT  KO FoldChange log2FoldChange  stat       pvalue
5120 218 128      0.588         -0.766 -4.48 7.465947e-06
             padj dispGeneEst dispFit dispMAP dispersion
5120 0.0001156724           0  0.0311  0.0149     0.0149
     betaConv maxCooks
5120     TRUE   0.0222
```

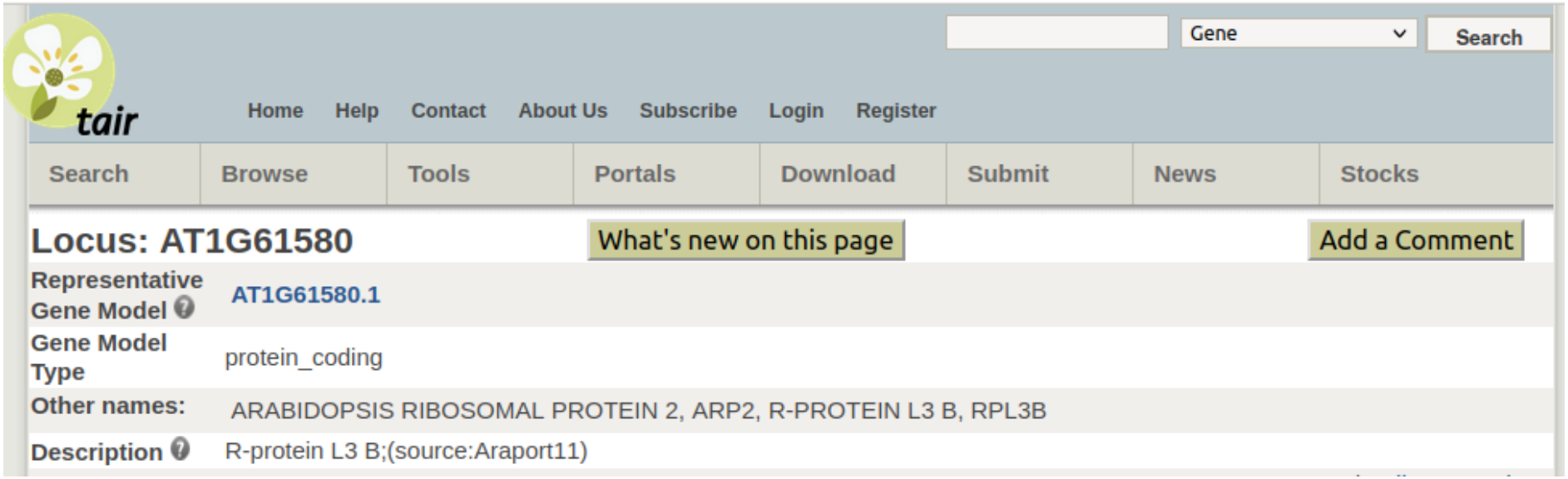
Details about this gene

Its name is `gene:AT1G61580`, it has a mean expression in the KO equal to `128`, a mean expression in the WT equal to `218`, a fold change of `0.588`.

This means that this gene is less expressed in the KO, in comparison to the WT.

The adjusted p-value almost equals to `7.5e-06`, which means that it is very likely that the difference of expression is related to the KO/WT status.

What is that gene name, not in computer gibberish ?



AT1G61580 vs ARP2

To whom does ARP2 refer? Both genes down here does not belong to the same genomic location: Chr1:3A22720431-22723281 or Chr3:3A9952305-9956158

	Locus	Description
1	AT1G61580	Other names: ARABIDOPSIS RIBOSOMAL PROTEIN 2, ARP2, R-PROTEIN L3 B, RPL3B R-protein L3 B;(source:Araport11)
2	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

AT1G61580 vs ARP2

ARP2 is not only related to A. Thaliana!

Search results

Items: 1 to 20 of 6078

<< First < Prev Page 1 of 304 Next > Last >>

See also 195 discontinued or replaced items.

Name/Gene ID	Description	Location	Aliases	MIM
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, CG9901, Dmel\CG9901, arp2	
arp2 ID: 5802965	ARP2/3 actin-organizing complex subunit Arp2 [*Schizosaccharomyces pombe* (fission yeast)]	Chromosome I, NC_003424.3 (4783007..4784765)	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: 30037080	actin-related protein 2 [*Sugiyamaella lignohabitans*]	Chromosome D, NC_031673.1 (877948..878958)	AWJ20_4899	
arp2 ID: 9626912	actin-related protein Arp2 [*Volvox carteri f. nagariensis*]		VOLCADRAFT_107669	

AT1G61580 vs ARP2

AT = Arabidopsis Thaliana

1 = Chromosome number

G = Protein coding gene

61580 = Unique gene identifier, given from top/north to bottom/south of chromosome.

Gene name vs Gene identifier

A *Gene name* is human understandable. If is not unique, neither to an organism, nor to a genomic location. A gene name is also called *Symbol*.

A *Gene identifier* is not designed for human. It is unique to both organism and genomic location.

You must use gene identifiers as much as possible. Keep gene names for meetings, and nice-looking graphs. You're right, gene identifiers are horrible on a daily use !

Fix gene identifiers (1/2)

In our table, the genes identifiers begin with "gene:". This going to break further analysis!

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

```
[1] gene:AT1G01010 gene:AT1G01020 gene:AT1G01030
[4] gene:AT1G01040 gene:AT1G01050 gene:AT1G01060
27655 Levels: gene:AT1G01010 gene:AT1G01020 ... gene:ATMG0141
```

For a computer: "`gene:AT1G01010`" is not "`AT1G01010`"

Fix gene identifiers (2/2)

We need a raw gene identifier:

```
# Replace the names in the ID column
deseq_genes$Id <- sub("gene:", "", deseq_genes$Id)
```

And we can check our genes identifiers with the function "head":

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

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

Translate Gene Identifiers with bitr

```r
annotation <- bitr(
  geneID   = deseq_genes$Id,              # Our gene list
  fromType = "TAIR",                  # We have TAIR ID
  toType   = c("ENTREZID", "SYMBOL"),# Other ID list
  OrgDb    = org.At.tair.db           # Our annotation
)
print(head(annotation))
```

```
       TAIR ENTREZID  SYMBOL
1 AT1G01010   839580 ANAC001
2 AT1G01010   839580  NAC001
3 AT1G01020   839569    ARV1
4 AT1G01030   839321    NGA3
5 AT1G01040   839574    ASU1
6 AT1G01040   839574  ATDCL1
```

Merge the translation and the original table

```r
deseq_genes <- merge(
  x = deseq_genes,  y = annotation,
  by.x = "Id",      by.y = "TAIR"
)
print(head(deseq_genes, 1))
```

```
         Id WT1 WT2 WT3 KO1  KO2  KO3 norm.WT1 norm.WT2
1 AT1G01010 533 541 473 931 1052 1124      493      492
  norm.WT3 norm.KO1 norm.KO2 norm.KO3 baseMean  WT   KO
1      496     1023     1050     1108   777.09 494 1060
  FoldChange log2FoldChange  stat      pvalue         padj
1      2.149          1.104 9.276 1.76535e-20 2.582102e-18
  dispGeneEst dispFit dispMAP dispersion betaConv maxCooks
1           0   0.021  0.0087     0.0087     TRUE   0.0187
  ENTREZID  SYMBOL
1   839580 ANAC001
```

Conclusion

1. We know to read differential gene expression results
2. We know how to read gene identifiers and how to translate them
3. We know that human-readable gene names are source of mistakes/confusions
4. We agree that computer-readable gene identifiers are horrible on Monday morning meetings.

Gene sets

Which genes are expressed in the roots

Go to planteome.org, search for roots ... 19065 genes !

Object	Object name	Object Type	Direct annotation	Ontology (aspect)
ELIP2	AT4G14690	protein	root	Anatomy (A)
APS3	AT4G14680	protein	root	Anatomy (A)
BIA1	AT4G15400	protein	root	Anatomy (A)
RIP2	AT2G37080	protein	root	Anatomy (A)

Definition of gene sets

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

Genes annotations: database expectations (1/2)

- **Gene Ontology (GO)**: which hosts a controlled vocabulary (fixed terms) for annotating genes
  - *Molecular Functions*: Molecular-level activities performed by gene products
  - *Cellular Components*: Locations relative to cell compartments and structures
  - *Biological Process*: Larger processes accomplished by multiple molecular activities

  http://geneontology.org/

Genes annotations: database expectations (2/2)

- **KEGG**: Kyoto Encyclopedia of Genes and Genomes
  - *Pathways*: Larger processes accomplished by multiple molecular activities
  - ...

  https://www.genome.jp/

- **MSigDB**: Molecular Signatures Database
  - Multiple collections of genes sets (human centered)

  http://software.broadinstitute.org/gsea/msigdb/index.jsp

Within R: OrgDb

- **OrgDB**: From bioconductor, you may find a lot of organism annotations

Bioconductor version 3.12 (Release)

Autocomplete biocViews search:

- CustomArray (2)
- CustomDBSchema (6)
- FunctionalAnnotation (31)
- Organism (634)
- PackageType (681)
  - AnnotationHub (11)
  - BSgenome (100)
  - cdf (118)
  - ChipDb (167)
  - db0 (19)
  - EuPathDB (1)
  - FRMA (11)
  - InparanoidDb (8)
  - MeSHDb (78)
  - OrganismDb (3)
  - OrgDb (20)
  - probe (97)

Packages found under OrgDb:

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

Show All entries Search table:

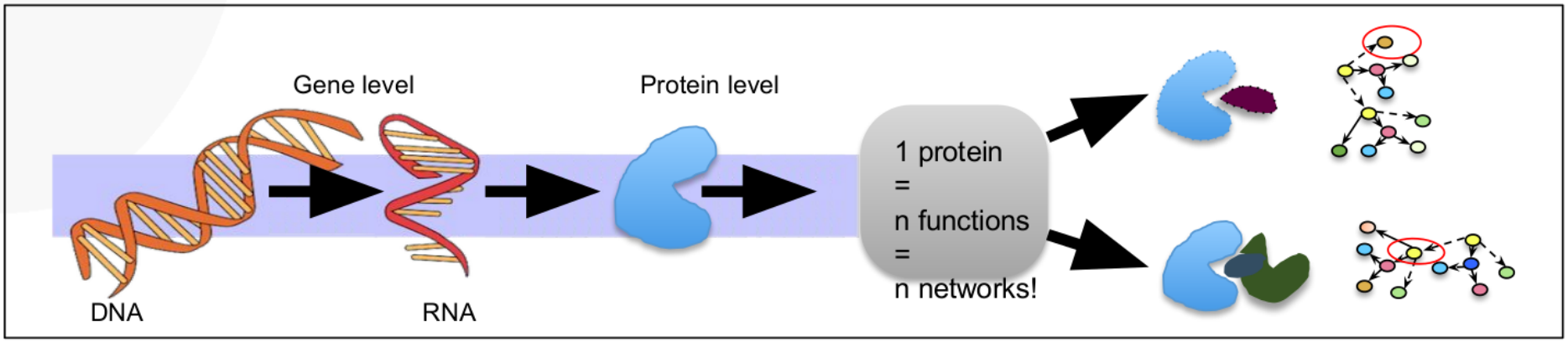
Package	Maintainer	Title	Rank
org.Hs.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Human	3
org.Mm.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Mouse	5
org.Rn.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Rat	18
org.Dm.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Fly	29
org.Sc.sgd.db	Bioconductor Package Maintainer	Genome wide annotation for Yeast	30
org.At.tair.db	Bioconductor Package Maintainer	Genome wide annotation for Arabidopsis	32
org.Dr.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Zebrafish	38
org.Ce.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Worm	50
org.Bt.eg.db	Bioconductor Package Maintainer	Genome wide annotation for Bovine	60

Within R: Many others

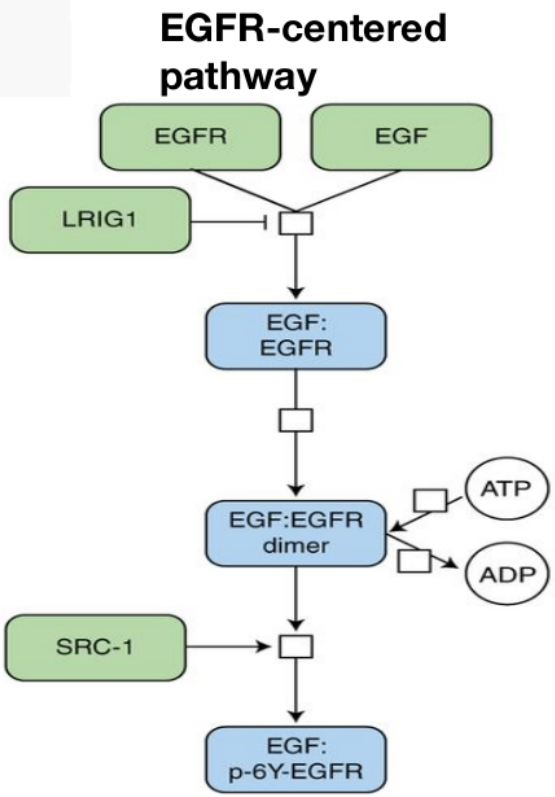
- Software (1975)
- AnnotationData (971)
  - ChipManufacturer (388)
  - ChipName (196)
  - CustomArray (2)
  - CustomDBSchema (6)
  - FunctionalAnnotation (31)
  - Organism (634)
    - Anopheles_gambiae (4)
    - Apis_mellifera (4)
    - Arabidopsis_thaliana (14)
    - Asparagus_officinalis (1)
    - Bacillus_subtilis (2)
    - Bos_taurus (15)
    - Caenorhabditis_elegans (13)
    - Callithrix_jacchus (1)

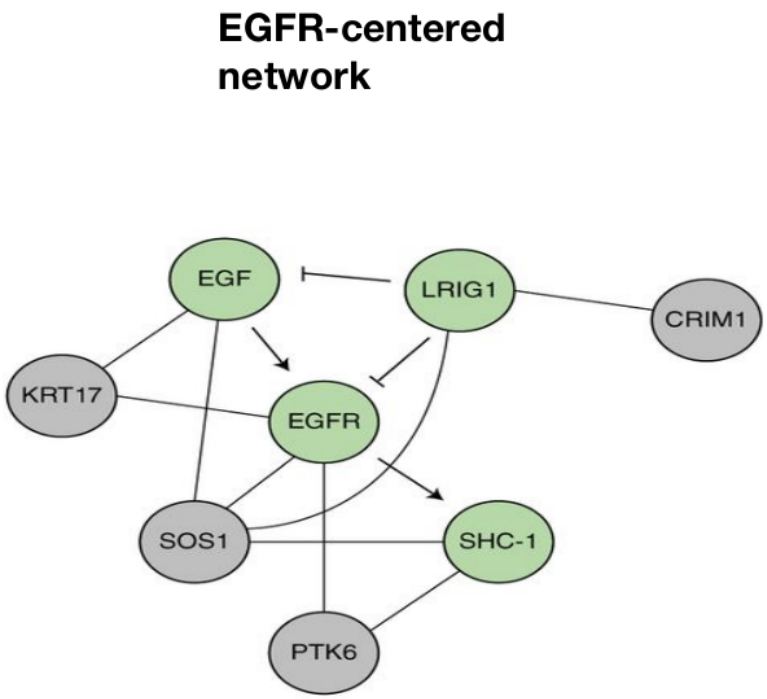
Protein - Protein Interactions (PPIs)

PPIs are useful for understanding functional relationships between proteins and the biology of the cell



Pathways vs Network





Adapted from: **Nature Methods**. Pathway and network analysis of cancer genomes (2015)

Institut Pasteur

Conclusion

1. A gene set is a group of genes that have a function, location, treatment response, or anything else in common.
2. There are a lot of gene set databases, one must choose them wisely.
3. You can relate genes, proteins, other kind of molecules together in a (gene) set.

ORA

Over Representation Analysis

ORA stands for *Over Representation Analysis*. It is almost what we did 5 minutes earlier!

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

Select differentially expressed genes

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

```
[1] 35684    29
```

```
dim(de_genes)
```

```
[1] 2743   29
```

Cluster Profiler Enrichment on GO: Cellular Components

We would like to perform Gene Set Enrichment analysis against the Gene Ontology's Cellular Components:

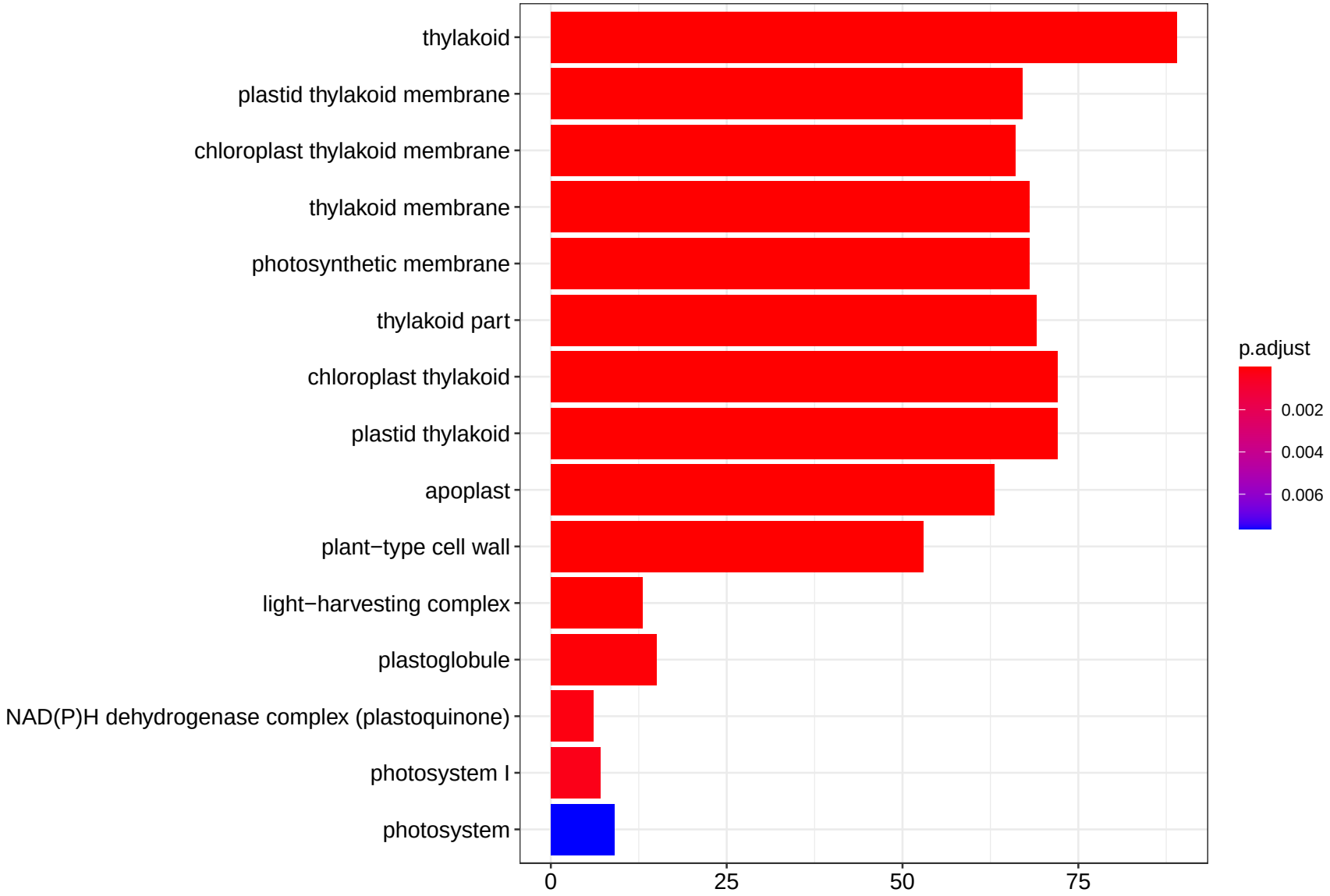
```
ego <- enrichGO(
  gene     = de_genes$ENTREZID,  # Ranked gene list
  universe= deseq_genes$ENTREZID,#All genes
  OrgDb    = org.At.tair.db,     # Annotation
  keyType = "ENTREZID",          # The genes ID
  ont      = "CC",               # Cellular Components
  pvalueCutoff  = 1,             # Significance Threshold
  pAdjustMethod = "BH",          # Adjustment method
  readable = TRUE                # For human beings
)
```

Cluster Profiler: Plots (1/3)

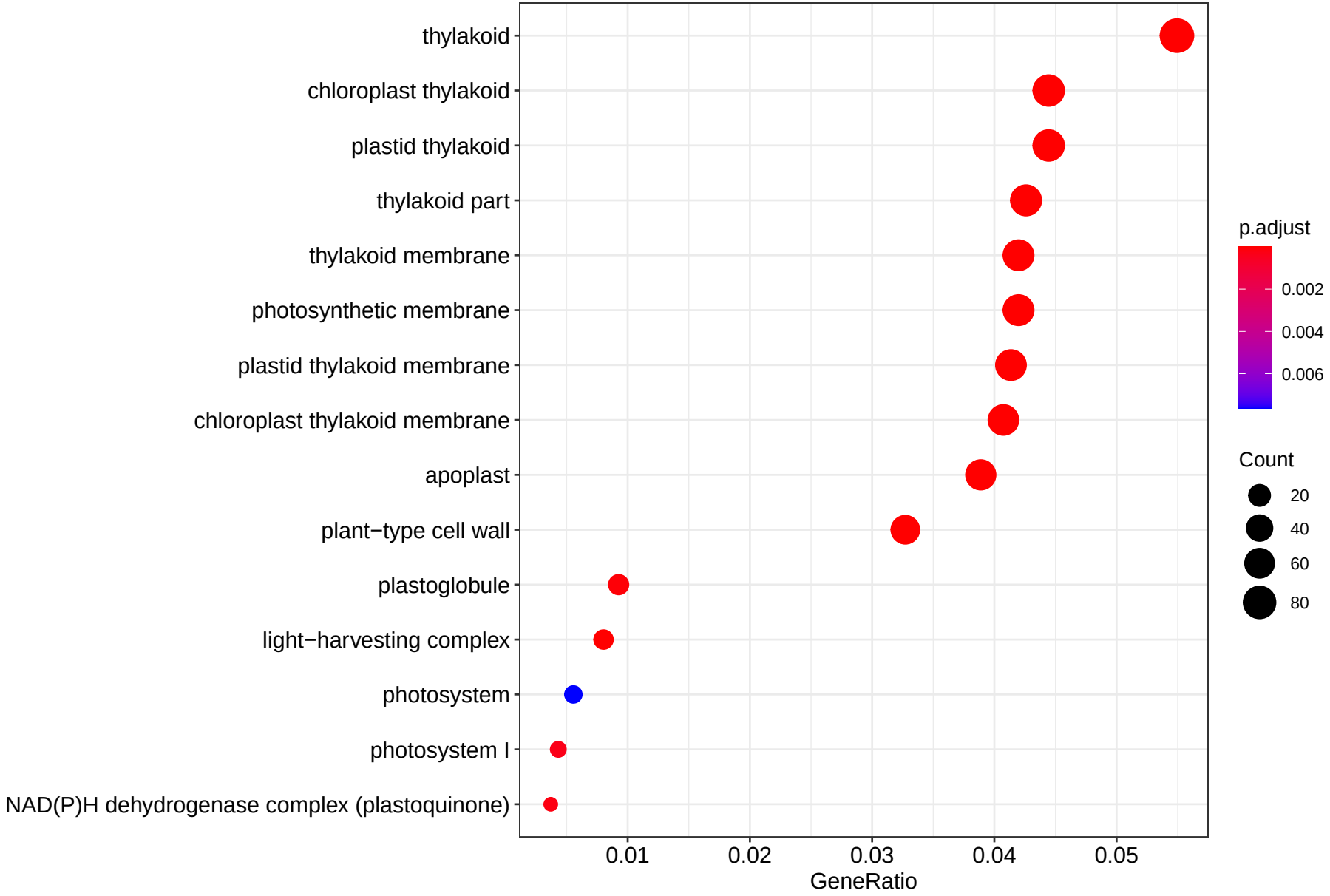
We want to visualize these results. To do so, let's use the function "barplot" and the function "dotplot" from the "enrichplot" package:

```
barplot(ego, showCategory=15)
dotplot(object = ego, showCategory=15)
```

Cluster Profiler: Plots (2/3)



Cluster Profiler: Plots (3/3)



What about roots ? (1/4)

```
res_ego <- ego@result
print(head(res_ego, 3))
```

```
                   ID                        Description
GO:0009579 GO:0009579                          thylakoid
GO:0055035 GO:0055035     plastid thylakoid membrane
GO:0009535 GO:0009535 chloroplast thylakoid membrane
           GeneRatio   BgRatio       pvalue     p.adjust
GO:0009579   89/1620 470/24412 1.519540e-19 2.873124e-17
GO:0055035   67/1620 295/24412 2.789440e-19 2.873124e-17
GO:0009535   66/1620 293/24412 8.176875e-19 5.382279e-17
                 qvalue
GO:0009579 2.554540e-17
GO:0055035 2.554540e-17
GO:0009535 4.785471e-17

GO:0009579 PSAD-2/EMB2784/PnsL2/NA/NDF1/NDF6/ATLFNR2/DRT112/NA
```

What about roots ? (2/4)

Nothing about roots ? Really ?

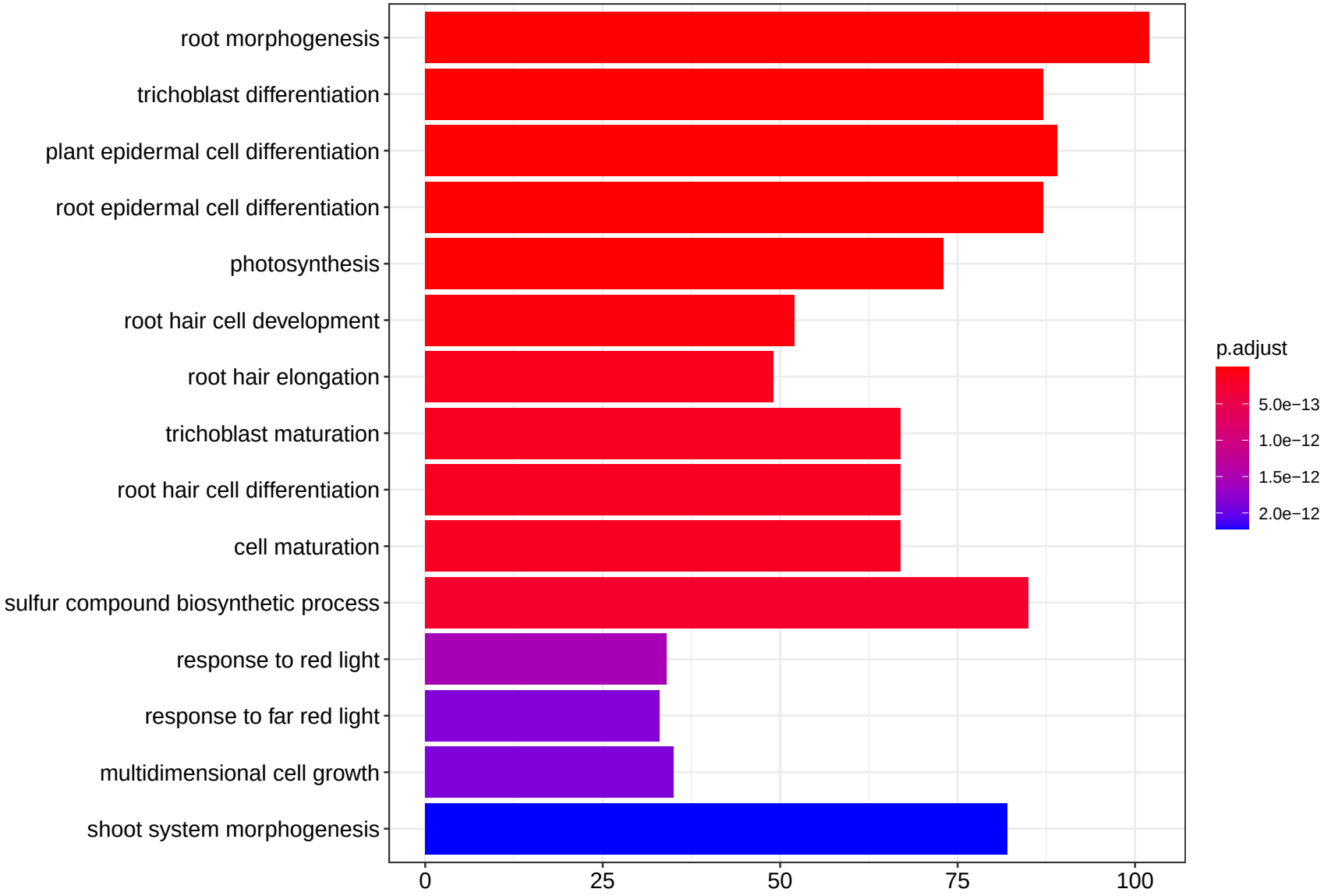
```
roots <- res_ego[with(res_ego, grepl("root", Description)), ]
print(head(roots))
```

```
[1] ID          Description GeneRatio   BgRatio
[5] pvalue      p.adjust    qvalue      geneID
[9] Count
<0 rows> (or 0-length row.names)
```

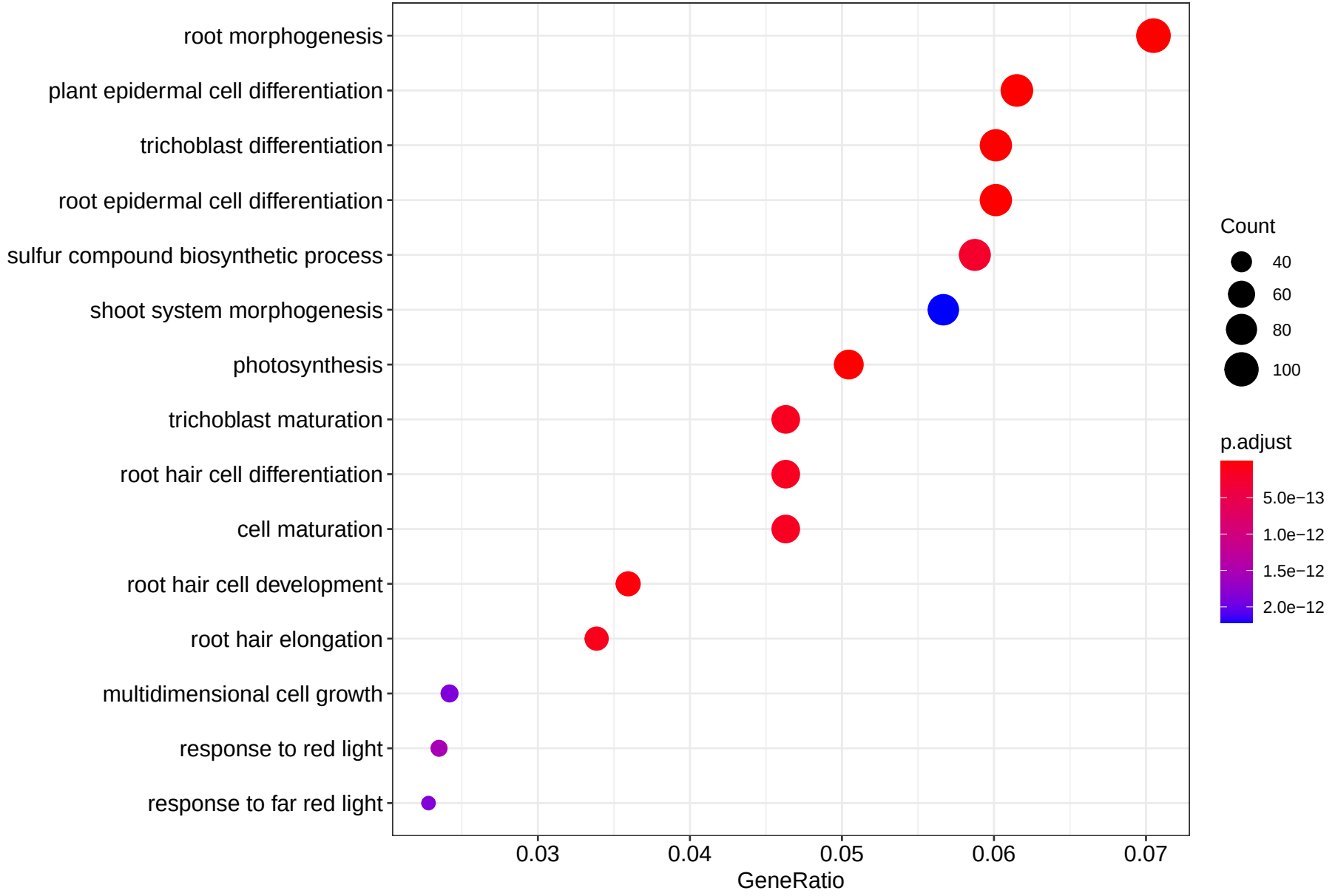
What about roots ? (3/4)

```
ego <- enrichGO(
  gene     = de_genes$ENTREZID,  # Ranked gene list
  universe= deseq_genes$ENTREZID,#All genes
  OrgDb    = org.At.tair.db,     # Annotation
  keyType = "ENTREZID",          # The genes ID
  ont      = "BP",               # Biological Process
  pvalueCutoff  = 1,             # Significance Threshold
  pAdjustMethod = "BH",          # Adjustment method
  readable = TRUE                # For human beings
)
```

Cluster Profiler: Roots (1/2)



Cluster Profiler: Roots (2/2)



What about roots ? (4/4)

```
res_ego <- ego@result
roots <- res_ego[with(res_ego, grepl("root", Description)), ]
print(head(roots))
```

```
                    ID                                Description
GO:0010015 GO:0010015                          root morphogenesis
GO:0010053 GO:0010053 root epidermal cell differentiation
GO:0080147 GO:0080147                  root hair cell development
GO:0048767 GO:0048767                        root hair elongation
GO:0048765 GO:0048765              root hair cell differentiation
GO:0048527 GO:0048527                    lateral root development
           GeneRatio   BgRatio       pvalue     p.adjust
GO:0010015  102/1447 458/20443 6.349710e-26 9.549964e-23
GO:0010053   87/1447 365/20443 2.118610e-24 7.965974e-22
GO:0080147   52/1447 203/20443 1.694001e-16 4.246296e-14
GO:0048767   49/1447 188/20443 5.962711e-16 1.281131e-13
GO:0048765   67/1447 323/20443 9.195374e-16 1.536649e-13
```

GSEA

Enrichment VS Gense Set Enrichment Analysis (1/2)

Note that, up to now, we used genes identifiers, and only genes identifiers. We have guessed which pathways had more differentially expressed genes than the others.

However:

1. we do not have any idea whether these pathways are up/donw regulated.
2. we do not know whether these pathway have up/down-regulated genes.

Enrichment VS Gense Set Enrichment Analysis (2/2)

Most of the time, we do not need to know that.

We saw our roots and plant organs being smaller, having morphology issues. Look at the plant!

However, we like numbers and statistics. Let's have fun!

Back to the data

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."

```
print(colnames(deseq_genes))
```

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

Using TAIR, ENTREZID or SYMBOL

We need a list of genes. What kind of name/identifier should we use ?

1. TAIR identifiers (<- Good)
2. ENTREZ identifiers (<- Good)
3. Gene Symbols (<- not this one)

Using WT/KO 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 FoldChange as weights

We have to weight each genes. We could use the column `FoldChange`, and look at the enrichment score.

It works, it is used in current publications. Highly differentially expressed genes have a very very very high impact on 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 log2FoldChange as weights

The very same conclusions are being done with log2FoldChange or FoldChange, however there will be no bias related to the initial gene expression.

This is, imho, the most published way to do. I almost always see this in current publications.

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 almose 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

To make short, `stat` is FoldChange weighted by adjusted pvalue.

It answers the very same question as log2FoldChange/FoldChange weights, but includes the confidence we have in the differential expression between KO and WT in addition to the change of expression between conditions.

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

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

Prepare data

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

# Get genes identifiers
names(geneList) <- de_genes$ENTREZID

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

We now have a sorted list of weighted genes.

Run analysis

Dear statisticians, please look aside for a minute.

```
gsea <- gseGO(
  geneList = geneList,          # Ranked gene list
  ont      = "BP",              # Biological Process
  OrgDb    = org.At.tair.db,    # Annotation
  keyType  = "ENTREZID",        # Identifiers
  pAdjustMethod = "BH",         # Pvalue Adjustment
  pvalueCutoff = 1              # Significance Threshold
)
```

GSEA plot (1/6)

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

```
columns_of_interest <- c(
  "Description",
  "enrichmentScore",
  "p.adjust"
)
head(
  x = gsea[, columns_of_interest], # Pathway ID
  8                                # lines to display
)
```

GSEA plot (2/6)

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

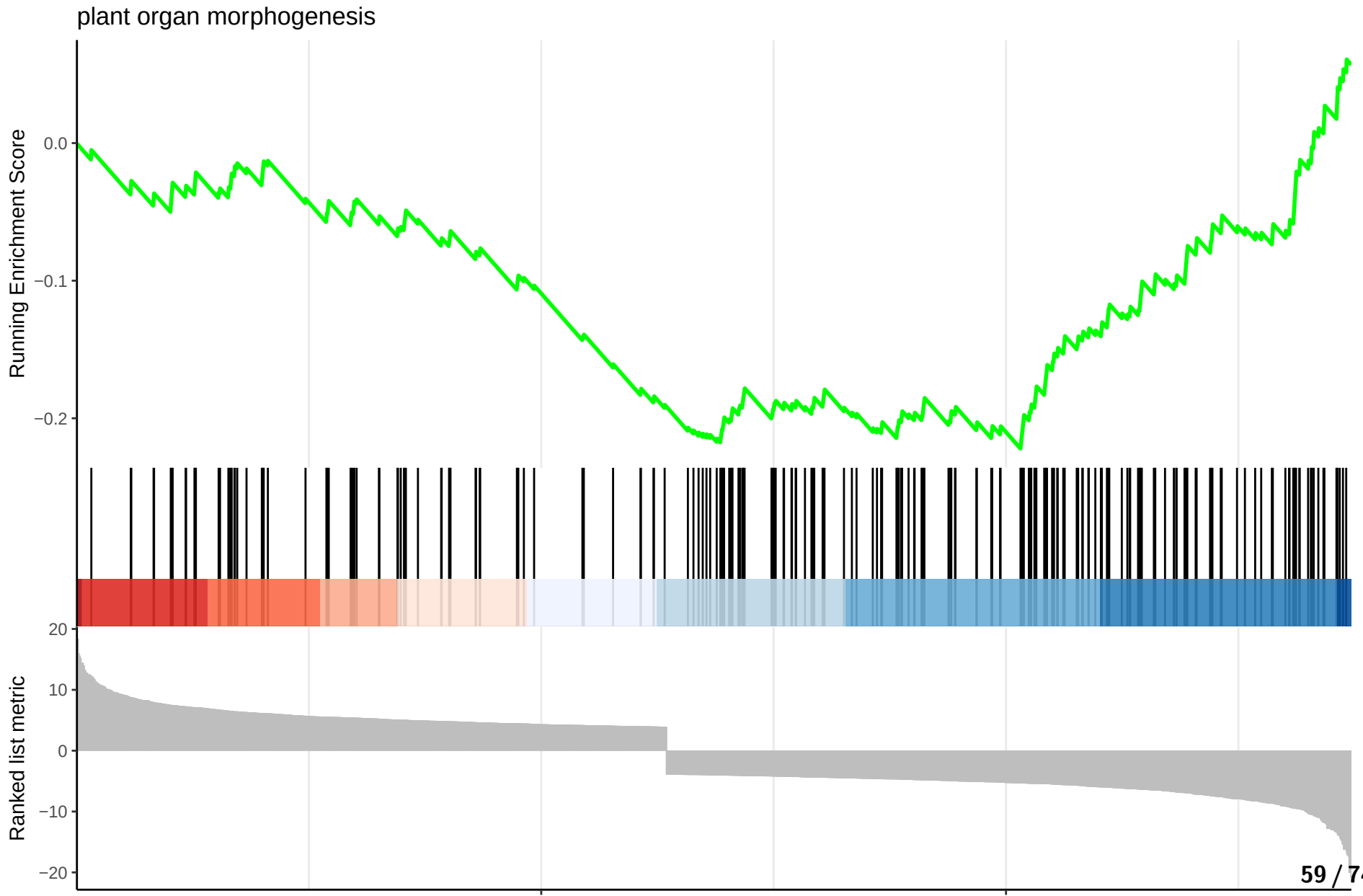
Description	enrichmentScore	p.adjust
response to stress	-0.1824687	0.0147129
response to chemical	-0.2084632	0.0147129
localization	-0.2263491	0.0147129
establishment of localization	-0.2376088	0.0147129
transport	-0.2477163	0.0147129
response to oxygen-containing compound	-0.2456208	0.0147129
carboxylic acid metabolic process	-0.2396659	0.0147129
organic acid metabolic process	-0.2459200	0.0147129

GSEA plot (3/6)

Finally, building the GSEA plot is being done with the function "gseaplot2" from "clusterProfiler":

```
# We need the number of the line
# Containing our pathway of interest
gsea_line <- match(
  "plant organ morphogenesis",
  gsea$Description
)
gseaplot2(
  x          = gsea,                    # Our analysis
  geneSetID = gsea$ID[gsea_line],  # Pathway ID
  title      = "plant organ morphogenesis" # Its name
)
```

GSEA plot (4/6)

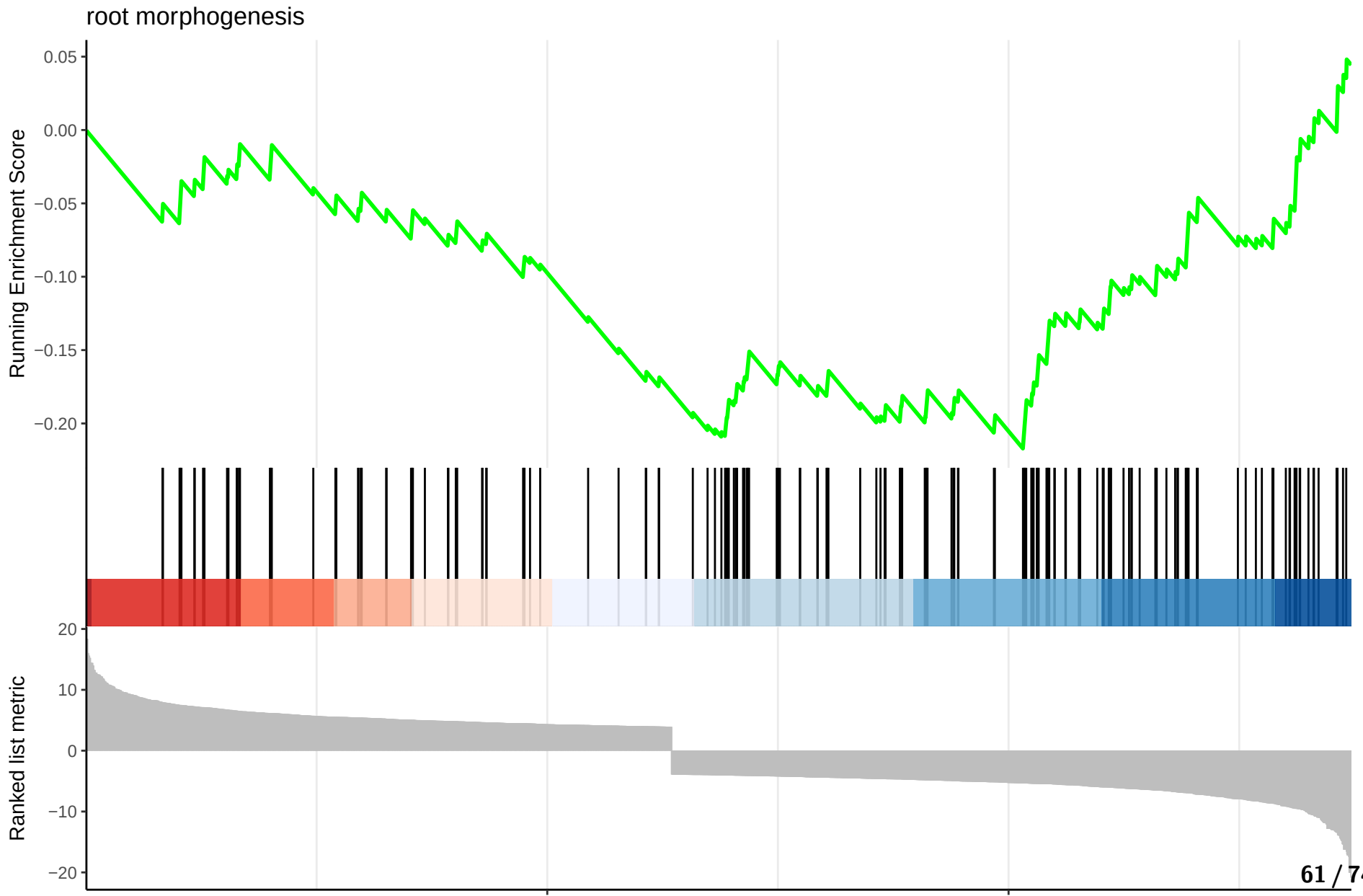

plant organ morphogenesis
Running Enrichment Score
0.0
−0.1
−0.2
20
10
0
−10
−20
Ranked list metric

GSEA plot (5/6)

Finally, building the GSEA plot is being done with the function "gseaplot2" from "clusterProfiler":

```
# We need the number of the line
# Containing our pathway of interest
gsea_line <- match(
  "root morphogenesis",
  gsea$Description
)
gseaplot2(
  x          = gsea,                  # Our analysis
  geneSetID = gsea$ID[gsea_line],  # Pathway ID
  title      = "root morphogenesis" # Its name
)
```

GSEA plot (6/6)

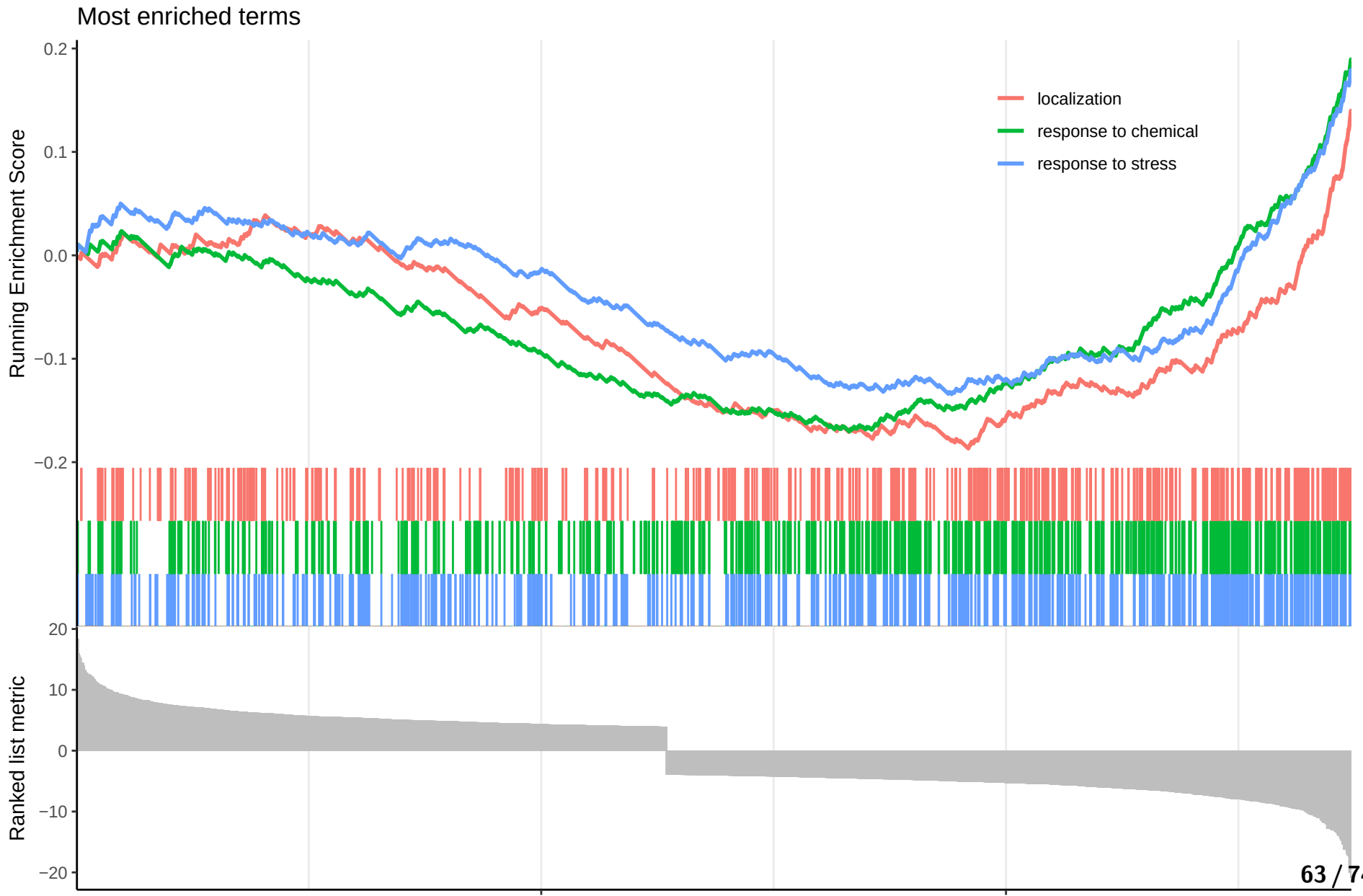

root morphogenesis
Running Enrichment Score
0.05
0.00
−0.05
−0.10
−0.15
−0.20
Ranked list metric
20
10
0
−10
−20

Multiple GSEA on the same graph (1/2)

... because we can!

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

Multiple GSEA on the same graph (2/2)



Conclusion on GSEA

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.

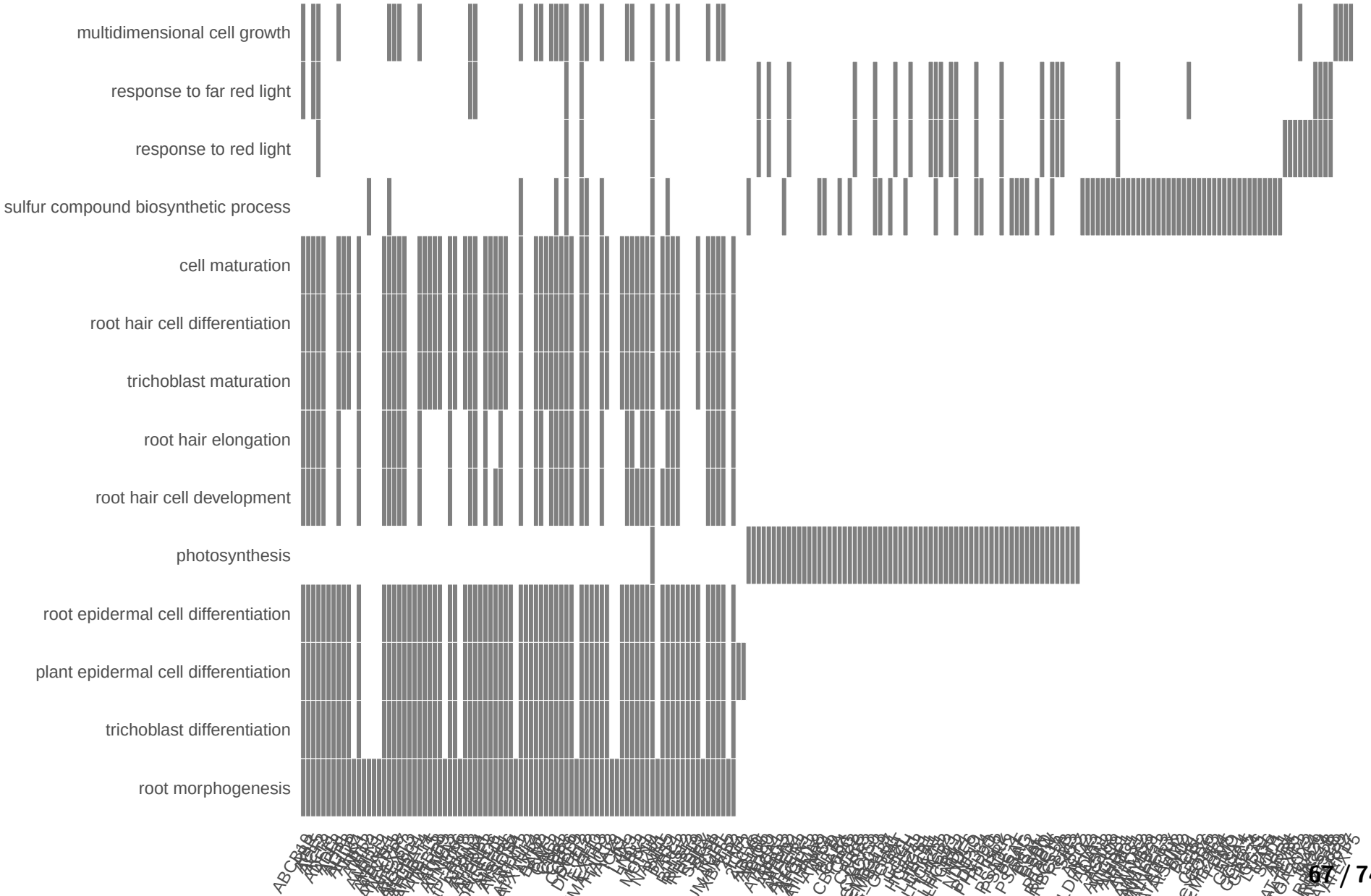
Sets

Heatmap (1/2)

Very common in publications

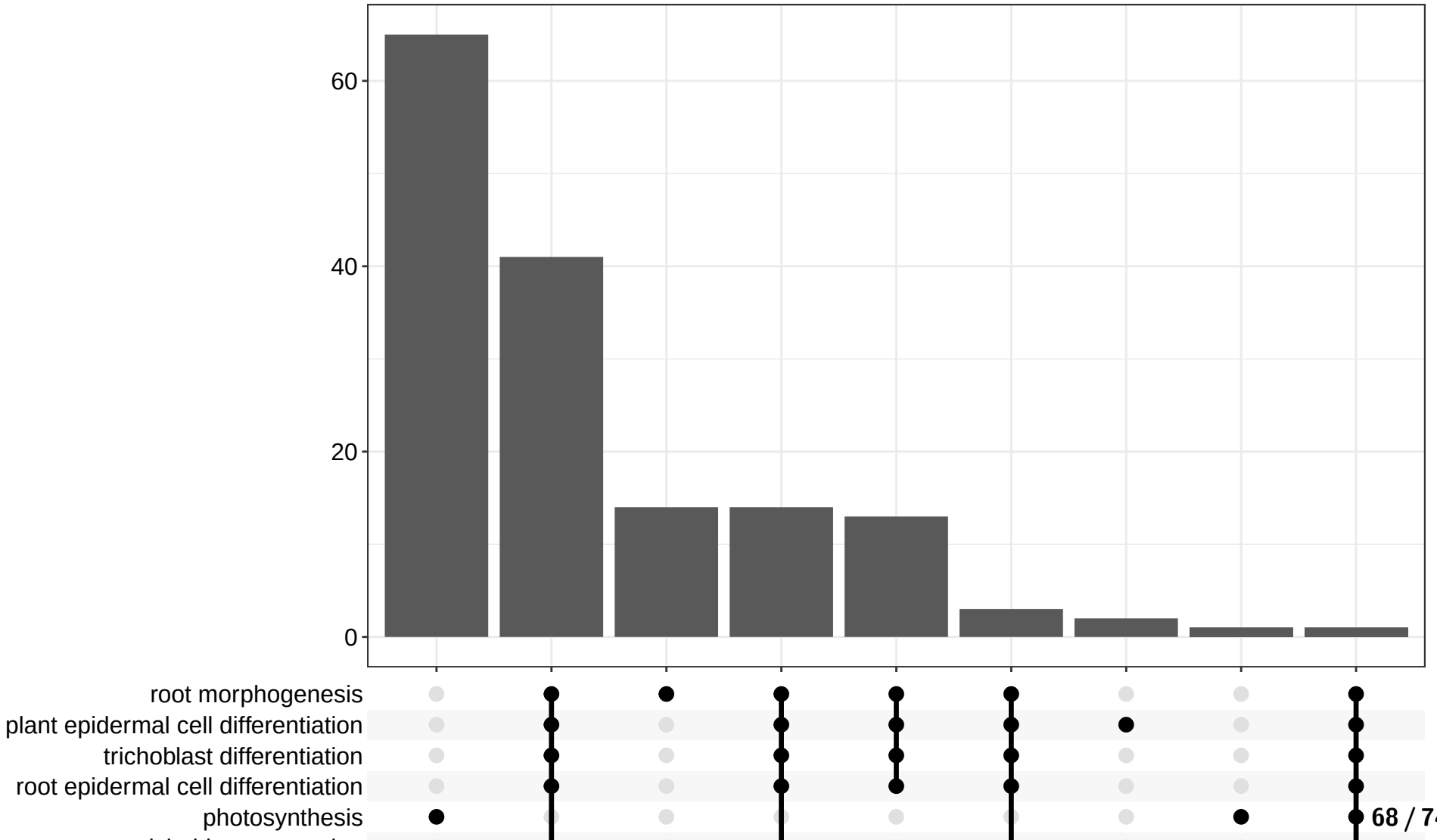
```
heatplot(
  x =  ego,                        # Our enrichment
  showCategory = 15,               # Nb of terms to display
  foldChange = geneList[1:10] # Our fold changes
)
```

Heatmap (2/2)


multidimensional cell growth
response to far red light
response to red light
sulfur compound biosynthetic process
cell maturation
root hair cell differentiation
trichoblast maturation
root hair elongation
root hair cell development
photosynthesis
root epidermal cell differentiation
plant epidermal cell differentiation
trichoblast differentiation
root morphogenesis

UpSet plot

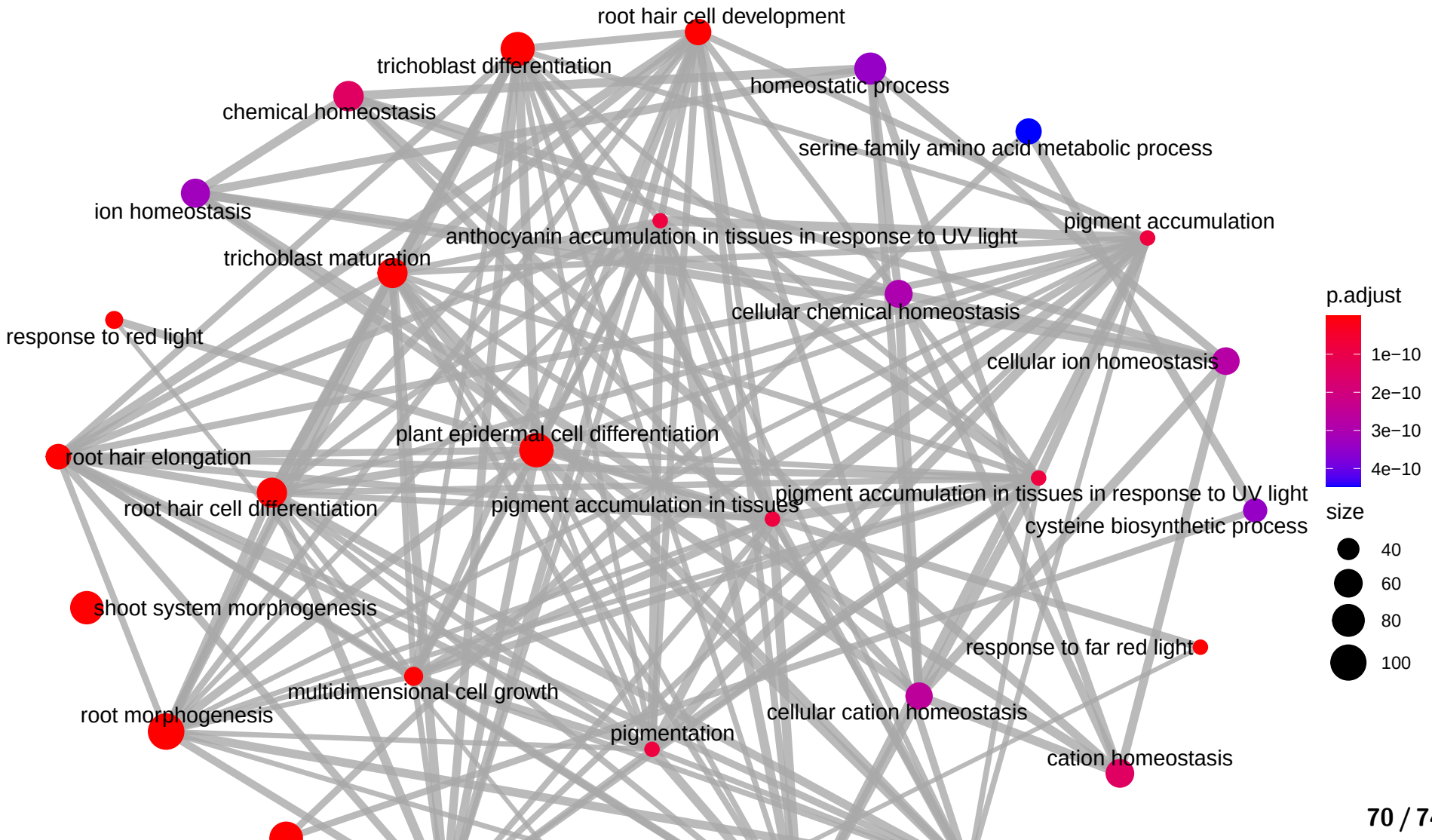
```
upsetplot(x = ego) # From our enrichment analysis
```



Networks

Enrichment map

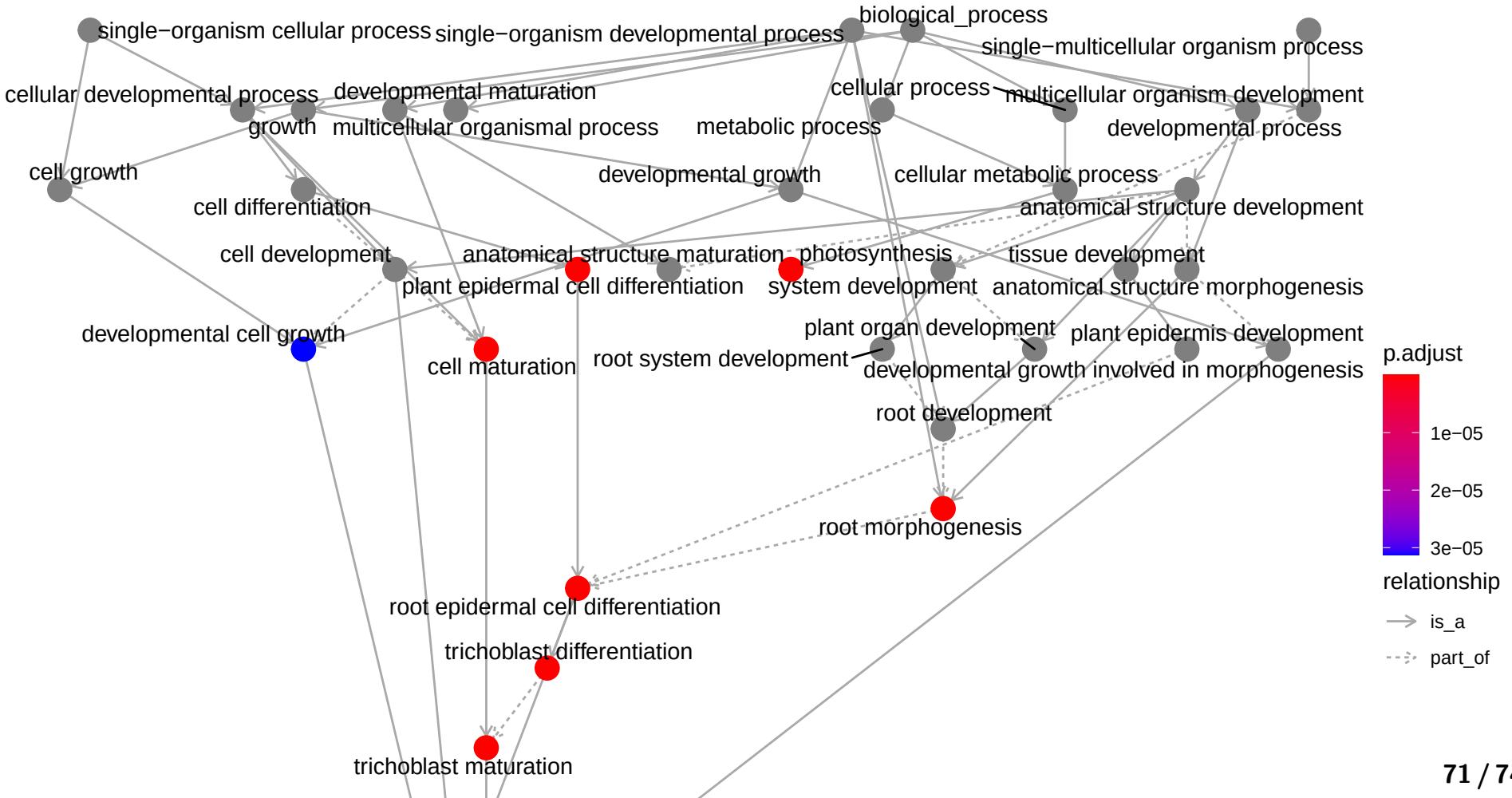
```
emapplot(ego) # From our enrichment analysis
```



GO plot

Relate enriched terms with each others:

```
goplot(ego) # From our enrichment analysis
```



Kegg (1/2)

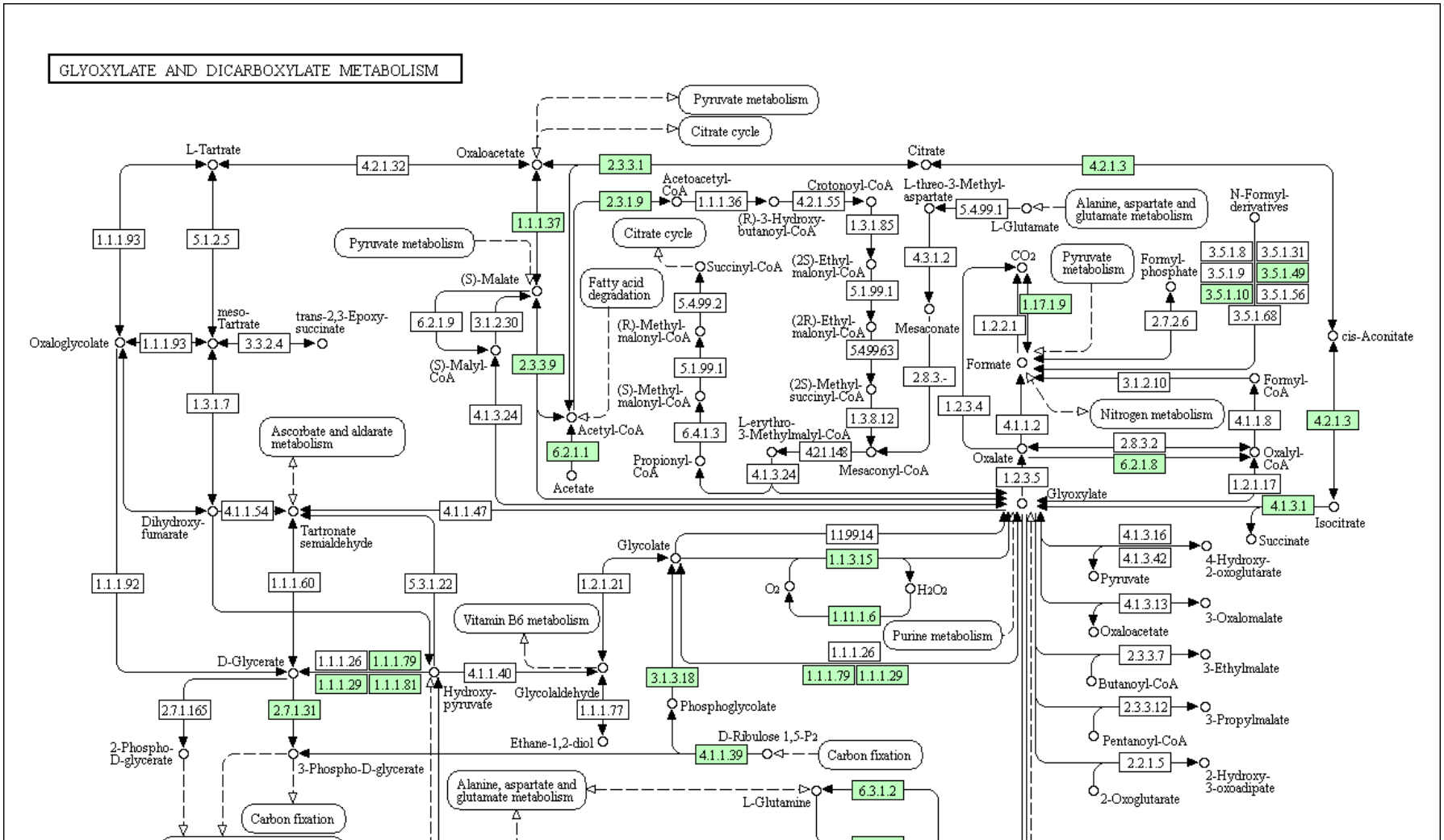
The Kegg analysis is done with the "pathview" package and this eponymous function:

```
names(geneList) <- de_genes$TAIR  # Use TAIR id

pv.out <- pathview(
  gene.data = geneList,      # Our gene list
  pathway.id = "ath00630",   # Our pathway
  species = "ath",           # Our organism
  # The color limits
  limit = list(gene=max(abs(geneList))),
  gene.idtype = "TAIR"       # The genes identifiers
)
```

KEGG (2/2)

There is the representation of our pathway, with differentially expressed genes colored!



Thanks

Thanks to the rest of the team for their reviews and advises.