Lab 4 – Visualization of Gene Ontology Enrichment Analyses

[Software needed: web access]

The results of "-omic" investigations are often long lists of genes or gene products. How can we characterize these gene lists? One approach might be called the "cherry-picking" approach, where we focus on certain genes that we're interested in. This might result in us missing important trends – are the genes that are increased in expression in response to a particular perturbation all involved in the biosynthesis of a particular hormone, for instance? Fortunately, Michael Ashburner, who was involved in the sequencing of the *Drosophila melanogaster* genome in the late 1990s, proposed the "Gene Ontology" at the 1998 annual meeting for bioinformatics and computational biology researchers, Intelligent Systems for Molecular Biology (ISMB), in Montréal, Canada. Although it was not immediately well-received, researchers from AstraZeneca who were privately sequencing the human genome provided some funding to further develop GO for annotating human genes[1]. After that, the U.S. National Institutes of Health provided and continues to provide funding to develop and maintain GO, which now has thousands of GO terms encompassing millions of genes/gene products from many different organisms. It is one of the key innovations in bioinformatics and allows us to ask exactly those kinds of questions about lists of genes or gene products to help us make sense of the oftentimes overwhelming amount of data produced by an "-omics" experiment.

Table 1: Tools covered in this lab (see end of lab for citations)

Tool	Description
AgriGO	GO term enrichment analysis tool for agricultural plants (and many other species). Offers acyclic graph visualization of enriched GO terms.
GOrilla	A less species-specific GO enrichment analysis tool with similar graph visualizations as AgriGO for enriched GO terms.
g:Profiler	GO term enrichment analysis tool for over 200 species. Displays the source of evidence for GO terms. Offers visualization by a colour-coded matrix.
R	Creating overview bubble charts of the results of GO enrichment tests from many comparisons.

AgriGO

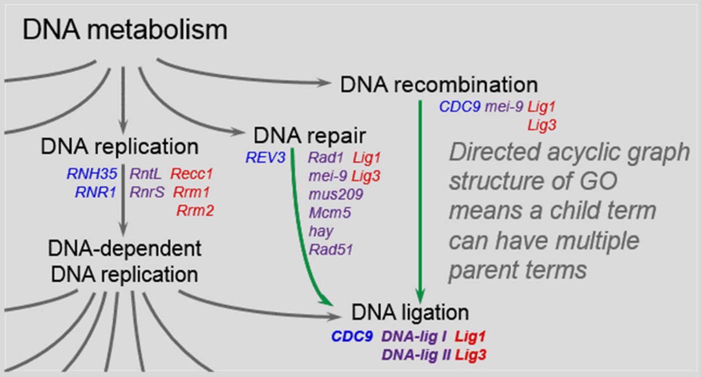
AgriGO from Zhen Su's laboratory at the Chinese Agricultural University is a user-friendly tool for analyzing whether any particular GO terms are enriched in a given gene list from Arabidopsis (or from many other agriculturally important species). It provides a nice visualization in the same directed acyclic graph format that the GO system was uses.

1. Go to http://systemsbiology.cau.edu.cn/agriGOv2/ (or use the mirror site at https://systemsbiology.cpolar.cn/agriGOv2/) and in the "Analysis" tab along the top, select Plant then Others.

[1] https://www.genomeweb.com/informatics/making-go-it-michael-ashburner-reflects-little-ontology-project-could

Box 1. The Gene Ontology framework, evidence codes etc.

At the core of the Gene Ontology framework are 3 main categories – Biological Process (BP), Molecular Function (MF), and Cellular Component (CC), described in the seminal GO paper from 2000 by Ashburner *et al*. (Nature Genet., https://doi.org/10.1038/75556). Each category has a collection of "child" terms beneath it, such a "biological adhesion" or "DNA metabolism" in the case of the GO Biological Process category. In turn, the child terms will have more specific child terms of their own associated with them. Any term can have genes *from any species* associated with it, as in the snapshot from part of the GO graph network for Biological Process – DNA Metabolism below, where genes from yeast are in blue, genes from Drosophila are in purple, and genes from mouse are in red. The figure also shows that the GO network's structure is a **directed acyclic graph**, meaning that a child term can have multiple parent terms, as is the case for "DNA ligation" being associated both with DNA recombination and DNA-dependent DNA replication.



Part of the Biological Process Gene Ontology; figure redrawn from Ashburner *et al*. (2000; Nature Genetics)

Part of the reason GO was developed was the observation shortly after their genome sequences were published that about 12% of *C. elegans* genes (a commonly researched worm) encode proteins whose biological roles can be predicted based on their similarity to their putative orthologs in yeast. GO is "a tool for the unification of biology" providing a "common language for annotation". When genes from a newly sequenced genome are compared to those in sequence databases like GenBank, annotations, including GO annotations, can be automatically transferred to them from the probable orthologs in sequenced species.

When GO curators are entering information about a gene's function, they use several **evidence codes** to denote why a given GO term was associated with a given gene. There are 26 evidence codes, see http://www.geneontology.org/page/guide-go-evidence-codes for the full list. If a given GO term has been inferred from an experiment, then it will receive an EXP (Inferred from Experiment), IDA (Inferred from Direct Assay) or other experimental evidence code. Sometimes computationally derived codes, such as IEA (Inferred from Electronic Annotation) are not used when doing GO enrichment analyses.

2. Go to http://systemsbiology.cau.edu.cn/agriGOv2/ (or use the backup site at https://systemsbiology.cpolar.cn/agriGOv2/ if the original is not available) and in the "Analysis" tab along the top, select Plant then Others.

3. Select the "Arabidopsis thaliana" link on the table.

4. In the first section for selecting the analysis tool, select "Singular Enrichment Analysis (SEA)".

5. **Input a gene list** as AGI IDs, gene aliases (e.g. *ABI3*), GenBank IDs etc. A large number of different identifiers are supported.

   *For parts of this lab, we will use the top 50 coexpressed genes for ABI3 across a "Developmental Map" as identified with the BAR's Expression Angler tool – download the "**ABI3_devmap_coexpressed_top50.txt**" file from the Coursera website (from the same section where you retrieved this manual), open it in a text editor, copy the AGI IDs and paste them into the appropriate box(es) for the next steps.*

6. **Select reference** – if the list comes from a microarray experiment, then you would choose the appropriate microarray platform, otherwise if the list comes from an experiment where it is possible to identify *any* of the AGI IDs present in the TAIR genome annotation (such as the case with a proteomics experiment or an mRNA-seq experiment) then you would choose the one of the TAIR options – this aspect is a nice feature of AgriGO.

   The data used to obtain the co-expressed genes come from the Affymetrix ATH1 platform, so we'll use "Affymetrix ATH1 Genome Array (blast)" as our reference (GPL198) – only around 22,000 genes out of 27,000 genes are on this platform.

7. Under "**Advanced Options – optional**" one can select one of three methods for statistical enrichment (Hypergeometric distribution, Fisher, or Chi-square) as well as one of seven multiple hypothesis testing correction methods.

   We recommend the use of the Hypergeometric test and (Benjamini &) Hochberg FDR multiple testing adjustment method, but the Fisher test and Yekutieli (FDR under dependency) multiple testing adjustment method will give similar results.

8. Then click submit (leave other settings as their defaults).

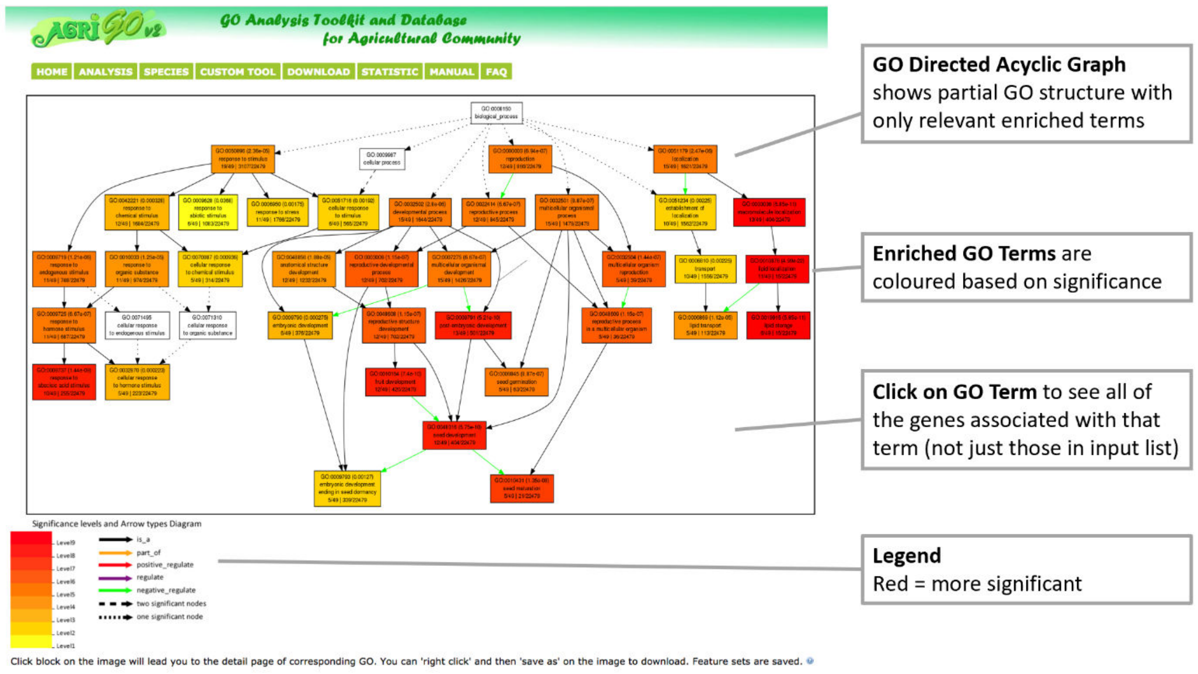
9. In the output, a table of enriched GO categories for our list of 50 genes is displayed among which four GO Biological Process terms (lipid localization, response to abscisic acid stimulus, seed development, post-embryonic development) and two GO Molecular Function terms (nutrient reservoir activity, lipid binding) are included.

   Examining these, they seem to "make sense" in the context of the later stages of seed development, when *ABI3* and these genes are expressed, insofar as this is the time when lipid reserves are being accumulated and the seed begins to desiccate, etc. – we saw that the *ABI3* gene is expressed in maturing seeds last lab, with the eFP Browser.

   There is also the possibility of creating "Graphical Results" or a "GO Flash Chart". If we click on the Generate Image button, the following output is generated for enriched Biological Processes (see **Figure 1**).

   What is the most significant GO biological process? How many genes are associated with this term? Identify the level of significance of the "seed development" GO term. Identify the level of significance of the "lipid storage" GO term.

Lab Quiz Question 1
*Answer lab quiz questions while doing lab!


AgriGO v2
GO Analysis Toolkit and Database for Agricultural Community
HOME
ANALYSIS
SPECIES
CUSTOM TOOL
DOWNLOAD
STATISTIC
MANUAL
FAQ
GO Directed Acyclic Graph shows partial GO structure with only relevant enriched terms
Enriched GO Terms are coloured based on significance
Click on GO Term to see all of the genes associated with that term (not just those in input list)
Legend
Red = more significant
Significance levels and Arrow types Diagram
Level9
Level8
Level7
Level6
Level5
Level4
Level3
Level2
Level1
is_a
part_of
positive_regulate
regulate
negative_regulate
two significant nodes
one significant node

Click block on the image will lead you to the detail page of corresponding GO. You can 'right click' and then 'save as' on the image to download. Feature sets are saved.

Figure 1: Graphical output from AgriGO for the top 50 A*BI3*-coexpressed genes in the AtGenExpress Tissue Set as provided on the Coursera course page. "Lipid localization" is the most enriched term, in red.

You can download the results of an enrichment analysis in text-based format for use in other plotting program, such as the R part later on!

<u>GOrilla</u>

GOrilla is another useful tool for such analyses, and permits the ability to upload a ranked list of genes for enrichment analysis, or a list of target genes (i.e. those that are differentially expressed or coexpressed) and background genes. It offers similar visualization of enriched categories as AgriGO.

1. Go to http://cbl-gorilla.cs.technion.ac.il/ to access the GOrilla. Choose "Arabidopsis thaliana" as our Organism.

2. GOrilla does not provide a background data set, unlike AgriGO. We can paste our top 50 ABI3-coexpressed genes into Box 1 of the "Two unranked lists of genes (target and background)" option and then upload all the genes present on the GPL198 platform as the background (provided as GPL198.txt from the course website; we obtained the AGI IDs of all genes on the GPL198 platform from TAIR in a file called affy_ATH1_array_elements-2010-12-20.txt and filtered it to remove duplicate IDs).

3. Leave the Ontology as "Process" then click "Search Enriched GO terms". The results will be similar to those shown in **Figure 2**.

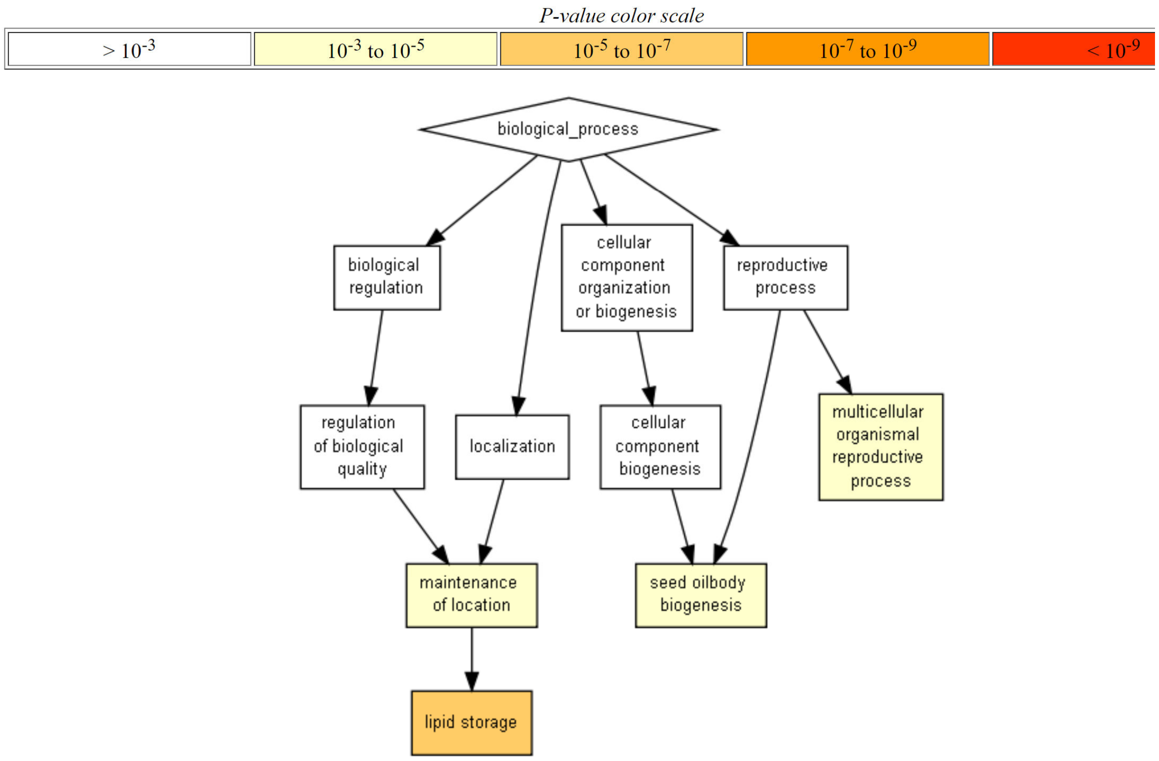
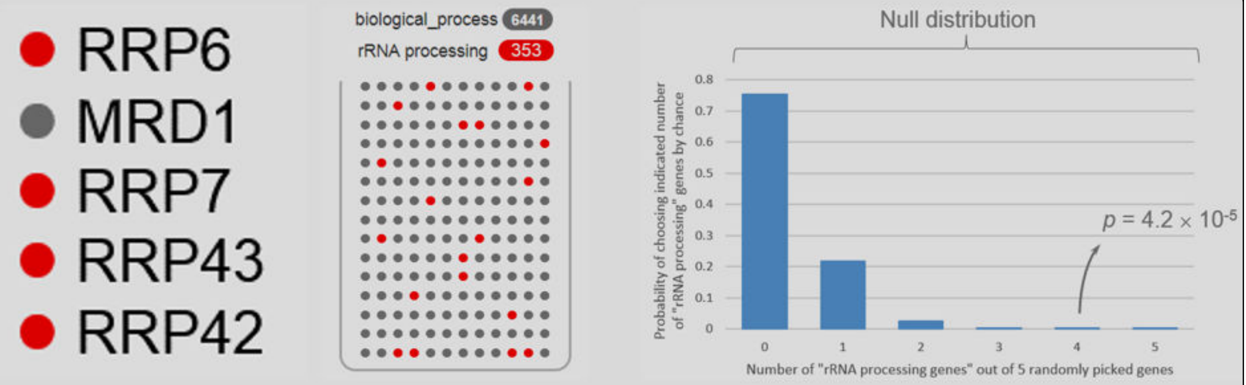


Figure 2: Output (partial) from GOrilla shows the enriched GO terms in the data set.

How does this output differ from the AgriGO output?

Box 2. GO term enrichment significance testing and correcting for multiple tests.

Let's say we have identified 5 genes as being significantly increased in expression in response to some kind of environmental stress, and 4 of those genes are associated with the GO term "rRNA processing". It's immediately clear that the main response is activating rRNA processing, correct? Not necessarily! If 80% of the genes in the organism's genome were involved in rRNA processing, then that proportion is just what we'd expect by randomly sampling genes. But if only a small fraction of the genes are involved in rRNA processing, then we'd be fairly confident that our list is enriched.



If we randomly sampled the 6,441 yeast genes annotated with some sort of biological process GO term, of which just 353 are in the "rRNA processing" category (as of mid-October, 2018), around three quarters of the time we'd produce 5 member lists with no "rRNA processing" genes in them. We can use a **hypergeometric test** to compute a *p*-value for randomly generating a 5-member list with 4 rRNA processing genes from the background set: 4.2×10^{-5}, or quite unlikely! Thus, a list of 5 genes with 4 in that GO category seems highly significant.

Often, there are thousands of GO terms that we can test against, and thus because we're doing so many tests, if we set a commonly used *p*-value of < 0.05 as our significance cut-off, one time in 20 we'd call a test as significant when in fact it occurred just by chance. So, we can adjust our resultant *p*-values by multiplying them by the number of tests we're performing – this is the **Bonferroni correction**. The strictness of Bonferroni correction means that many categories that might be considered significant will now not be: that is, they become "false negatives".

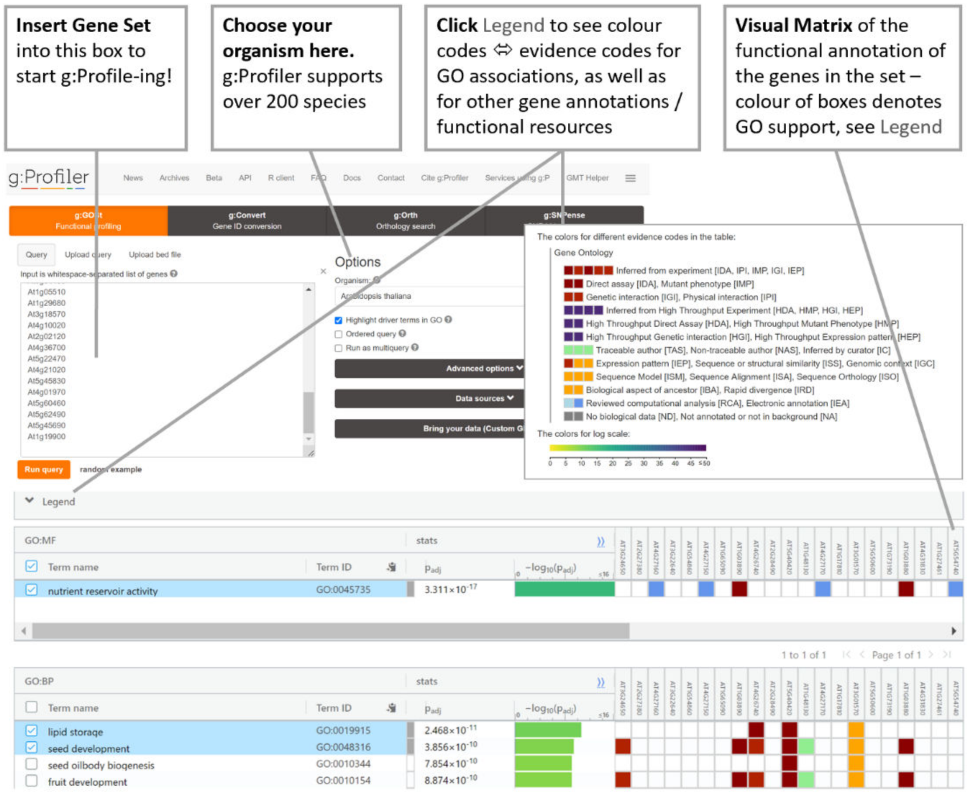
In order to be less strict, but still address the issue of multiple testing, we can attempt to control the false discovery rate (FDR) using the **Benjamini-Hochberg** or other procedures. In this case, we adjust our *p*-values by ordering the *p*-values for all GO term tests from smallest to largest. Starting with the largest *p*-value, the adjusted *p*-value for that category is set to be that *p*-value. The adjusted *p*-value for the second largest result is set to be either the previous adjusted *p*-value or the current (unadjusted) *p*-value multiplied by the total number of *p*-values divided by the *p*-value rank, whichever is smaller. We proceed to adjust all *p*-values all the way down to the smallest one. In many cases, the adjusted FDR *p*-value is not as drastically modified as under Bonferroni correction, thus resulting in fewer "false negatives" (but more false positives). Note that the Benjamini-Hochberg FDR adjustment assumes independence of the tests. Because GO terms are often child terms of others, this assumption isn't always met. Thus, other adjustments (e.g. **Benjamini-Hochberg-Yekutieli**), which take dependence into account, are sometimes used for correcting *p*-values in GO analyses.

g:Profiler

1. Go to https://biit.cs.ut.ee/gprofiler/
2. Paste the 50 *ABI3*-coexpressed gene set (in "***ABI3_devmap_coexpressed_top50.txt***") into the query text box in the g:GOSt tab, and select the organism as *Arabidopsis thaliana.* We'll use the default settings.
3. Submit by selecting the 'Run Query' button. The output shown in **Figure 3** displays the enriched pathways for the set of 50 coexpressed genes. *What is the p-value for the lipid storage biological process? For which gene/genes is there direct assay evidence?*

Lab Quiz Question 2

A couple of nice features of g:Profiler are that the GO evidence codes (as described in **Box 1**) are presented as part of the output. Enrichment for Plant Reactome Pathways (or Reactome Pathways, for other species) – not covered in this lab but a useful resource to investigate – is also provided, with enrichment significance for specific pathways calculated in a similar manner as described in **Box 2**.



Insert Gene Set into this box to start g:Profile-ing!

Choose your organism here. g:Profiler supports over 200 species

Click Legend to see colour codes ⇔ evidence codes for GO associations, as well as for other gene annotations / functional resources

Visual Matrix of the functional annotation of the genes in the set – colour of boxes denotes GO support, see Legend

The colors for different evidence codes in the table:

Gene Ontology

- 🟥🟥🟥🟥🟥 Inferred from experiment [IDA, IPI, IMP, IGI, IEP]
- 🟥🟥 Direct assay [IDA], Mutant phenotype [IMP]
- 🟥🟥 Genetic interaction [IGI], Physical interaction [IPI]
- 🟪🟪🟪🟪 Inferred from High Throughput Experiment [HDA, HMP, HGI, HEP]
- 🟪🟪 High Throughput Direct Assay [HDA], High Throughput Mutant Phenotype [HMP]
- 🟪🟪 High Throughput Genetic interaction [HGI], High Throughput Expression pattern [HEP]
- 🟩🟩🟩 Traceable author [TAS], Non-traceable author [NAS], Inferred by curator [IC]
- 🟥🟧🟧 Expression pattern [IEP], Sequence or structural similarity [ISS], Genomic context [IGC]
- 🟧🟧🟧 Sequence Model [ISM], Sequence Alignment [ISA], Sequence Orthology [ISO]
- 🟧🟧 Biological aspect of ancestor [IBA], Rapid divergence [IRD]
- 🟦🟦 Reviewed computational analysis [RCA], Electronic annotation [IEA]
- ⬛⬛ No biological data [ND], Not annotated or not in background [NA]

The colors for log scale:

GO:MF Term name	Term ID	p_{adj}
✓ nutrient reservoir activity	GO:0045735	3.311×10^{-17}

GO:BP Term name	Term ID	p_{adj}
✓ lipid storage	GO:0019915	2.468×10^{-11}
✓ seed development	GO:0048316	3.856×10^{-10}
seed oilbody biogenesis	GO:0010344	7.854×10^{-10}
fruit development	GO:0010154	8.874×10^{-10}

Figure 3: Partial output of g:Profiler using the top 50 *ABI3*-coexpressed genes. Results may differ slightly.

R

One fairly common way of visualizing the results of multiple Gene Ontology enrichment analyses is to plot them side-by-side using bubble plots. As with other small multiples, it helps to have the axes scaled so that they all have the same range (or the same labels in this case), and additionally, circle sizes and colours should encode the same information in terms of what they represent (for example, enrichment relative to background, or corresponding FDRs).

We're going to try to replicate Figure 6 of Van Weringh et al., (2021; https://www.biorxiv.org/content/10.1101/2021.04.15.439991v1.abstract), summarizing the results of Gene Ontology enrichment analyses for differential gene expression responses in both guard cells – a pair of kidney-shaped cells that create a pore and whose swelling during drought can limit water loss via transpiration through the pore, but at the expense of a restriction of CO_2 uptake, necessary for photosynthesis – and whole leaves over the course of a developing drought, as measured by decreasing soil water content. See **Figure 4**.

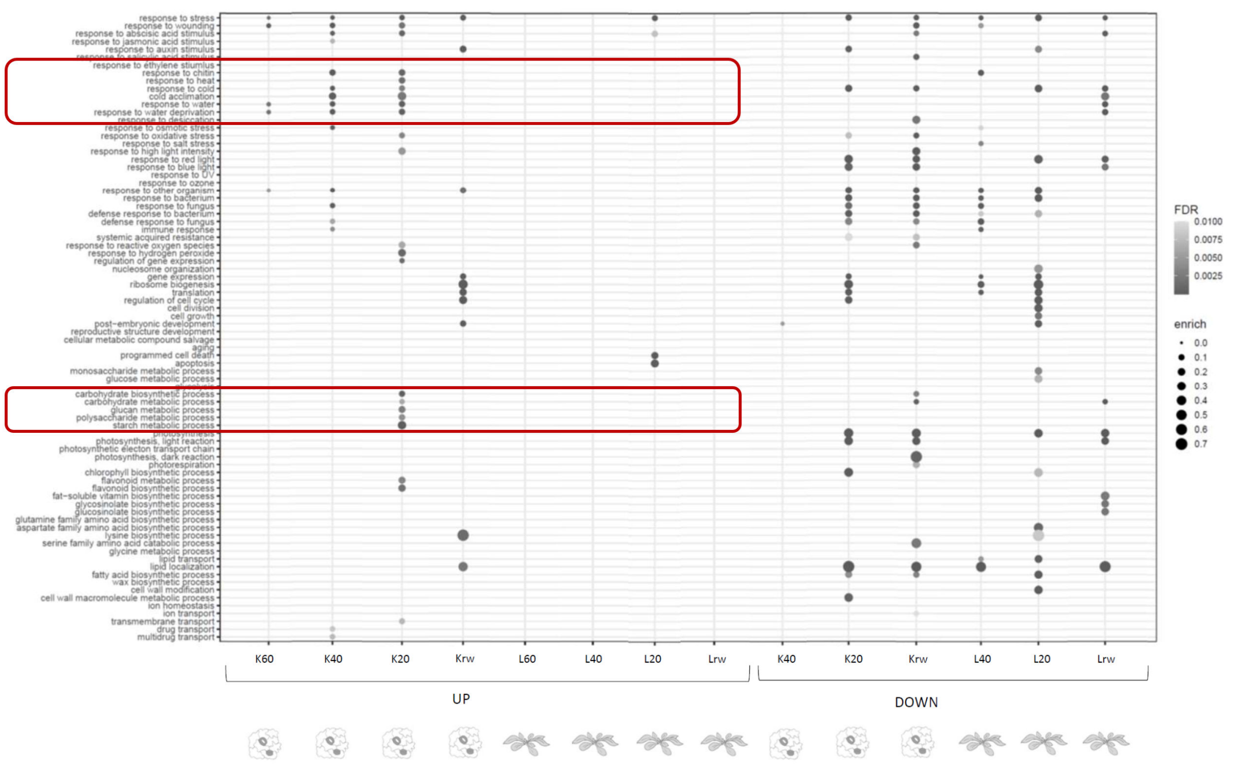


Figure 4. Reproduction of Gene Ontology enrichment analysis results in guard cell RNA-seq data from Van Weringh et al. (2021). "Up" and "Down" DEGs (i.e., those genes exhibiting increases or decreases, respectively, in transcript abundance compared to their age- and tissue-matched well-watered controls) were submitted for GO term enrichment analysis with AgriGO. Samples are labelled to show the genotype time point and comparison: *K* represents GCs sampled using KAT1:INTACT, *L* represents whole shoots sampled using 35S:INTACT; *60*, *40* or *20* represents the time point referring to the % SWC (soil water content) of the paired drought sample. Red highlighted areas denote terms enriched specifically for GC "up" DEGs as compared to whole leaf samples.

1. Upload the "K[60|40|20|rw]dvsw_up_GO.txt" and "L[60|40|20|rw]dvsw_up_GO.txt" to your Jupyter notebook running on Coursera as per Step 3 of the R part of Lab 1. Upload the two rewatered samples too, "Krwvsw_up_GO.txt" and "Lrwvsw_up_GO.txt". Also, upload the "keep_formakinggofigures.txt" file. Alternatively, use the coded notebook, which already has these files. *Examine these files in Excel or Google Sheets to see what kinds of data they contain.*

   See the Lab 4 R (code).ipynb for the steps to generate the "UP" side of the figure above, with options to use the Viridis colour scheme.

2. As for last week's lab, we'll load some packages that will help us with our goal of creating some plots this week.

```
# Install these packages onto JupyterHub
library(tidyverse)
library(repr)
library(viridis)
getwd()# check our working directory
```

You will receive several messages back about these packages having been attached successfully.

3. Load the datasets from your working directory, which should be `'/home/jovyan/work'` if you're using Coursera's Jupyter Hub.

```
# Create a vector with your file names
fileNames <- c("K60dvsw", "K40dvsw", "K20dvsw", "Krwvsw",
"L60dvsw", "L40dvsw", "L20dvsw", "Lrwvsw")
# While we import the files, we'll add a column to denote
the dataset that it comes from.
# This will help us put the data into a long format
data.list <- lapply(fileNames, FUN = function(x)
mutate(read_tsv(paste0(x, "_up_GO.txt")), dataSet = x))
names(data.list) <- fileNames
# What is the structure of the data set
str(data.list, give.attr = FALSE)
# Note that L60dvsw_up_GO.txt has no data in it.
# How many rows are there across all data sets?
sum(unlist(lapply(data.list, FUN = function(x) dim(x)[1])))
```

```
Parsed with column specification:
cols(
  GO_acc = col_character(),
  term_type = col_character(),
  Term = col_character(),
  queryitem = col_double(),
  querytotal = col_double(),
  bgitem = col_double(),
  bgtotal = col_double(),
  pvalue = col_double(),
  FDR = col_double(),
  entries = col_character()
)
…
List of 8
 $ K60dvsw: tbl_df [105 × 11] (S3: tbl_df/tbl/data.frame)
  ..$ GO_acc    : chr [1:105] "GO:0050896" "GO:0009611" "GO:0042221" "GO:0006950" ...
  ..$ term_type : chr [1:105] "P" "P" "P" "P" ...
  ..$ Term      : chr [1:105] "response to stimulus" "response to wounding" ...
  ..$ queryitem : num [1:105] 30 7 18 18 6 6 9 7 8 9 ...
  ..$ querytotal: num [1:105] 94 94 94 94 94 94 94 94 94 94 ...
  ..$ bgitem    : num [1:105] 4057 197 2085 2320 229 ...
  ..$ bgtotal   : num [1:105] 37767 37767 37767 37767 37767 ...
  ..$ pvalue    : num [1:105] 2.7e-08 8.2e-07 3.7e-06 1.6e-05 2.8e-05 3.6e-05 ...
  ..$ FDR       : num [1:105] 8.8e-06 1.3e-04 4.0e-04 1.3e-03 1.7e-03 1.7e-03 ...
  ..$ entries   : chr [1:105] "// AT3G17790 // AT2G17840 // AT5G65140 //..."
  ..$ dataSet   : chr [1:105] "K60dvsw" "K60dvsw" "K60dvsw" ...
2916
```

4. We're going to combine our lists into one data frame using the **`rbind()`** function. We are essentially creating a long format data set, in terms of how R will handle the data.

```
# Now that all the data is in a list, we can combine it
# into a single data frame
allData.df <-
    do.call(rbind, data.list) %>%
    mutate(enrich = queryitem/bgitem)

# What does the dataframe look like?
str(allData.df)
```

```
tbl_df [2,916 × 12] (S3: tbl_df/tbl/data.frame)
 $ GO_acc    : chr [1:2916] "GO:0050896" "GO:0009611" "GO:0042221" "GO:0006950" ...
 $ term_type : chr [1:2916] "P" "P" "P" "P" ...
 $ Term      : chr [1:2916] "response to stimulus" "response to wounding" ...
 $ queryitem : num [1:2916] 30 7 18 18 6 6 9 7 8 9 ...
 $ querytotal: num [1:2916] 94 94 94 94 94 94 94 94 94 94 ...
 $ bgitem    : num [1:2916] 4057 197 2085 2320 229 ...
 $ bgtotal   : num [1:2916] 37767 37767 37767 37767 37767 ...
 $ pvalue    : num [1:2916] 2.7e-08 8.2e-07 3.7e-06 1.6e-05 2.8e-05 3.6e-05 3.4e-05 ...
 $ FDR       : num [1:2916] 8.8e-06 1.3e-04 4.0e-04 1.3e-03 1.7e-03 1.7e-03 1.7e-03 ...
 $ entries   : chr [1:2916] "// AT3G17790 // AT2G17840 // AT5G65140 // AT2G33380 // ...
 $ dataSet   : chr [1:2916] "K60dvsw" "K60dvsw" "K60dvsw" "K60dvsw" ...
 $ enrich    : num [1:2916] 0.00739 0.03553 0.00863 0.00776 0.0262 ...
```

5. The code provided in the R notebook shows how to figure out how many different enriched GO terms there are across all of the samples (there are 118). We'll actually use a predefined list of 80 terms that focuses on GO terms of interest.

```
# Import the specific term list

goTermKeep <- read_file("keep_formakinggofigures.txt") %>%
str_split(pattern = "\r\n") %>% unlist()

# What does it look like
goTermKeep

# What are the attributes
str(goTermKeep)
```

```
chr [1:80] "multidrug transport" "drug transport" ...
```

6. We'll now filter our data by GO terms into a new set called **`sigData.df`**. However, we'll mutate a couple of variables first to make them into factors.

```
# What happens when we filter by these terms?
sigData.df <-
allData.df %>%
    # This mutation step will return the file names back
    # as x-axis values
    mutate(dataSet = factor(dataSet, levels = fileNames)) %>%
    filter(Term %in% goTermKeep) %>%
    mutate(Term = factor(Term, levels = goTermKeep)) %>%
    # Filter the data again based on the lab notes
    filter(FDR <= 0.01, term_type == "P")

str(sigData.df)
```

```
tbl_df [57 × 12] (S3: tbl_df/tbl/data.frame)
 $ GO_acc    : chr [1:57] "GO:0009611" "GO:0006950" "GO:0009414" "GO:0009415" ...
 $ term_type : chr [1:57] "P" "P" "P" "P" ...
 $ Term      : Factor w/ 80 levels "multidrug transport",..: 79 80 68 69 58 80 73 79 ...
 $ queryitem : num [1:57] 7 18 6 6 8 55 16 16 16 13 ...
 $ querytotal: num [1:57] 94 94 94 94 94 275 275 275 275 275 ...
 $ bgitem    : num [1:57] 197 2320 229 240 599 2320 151 229 240 197 ...
 $ bgtotal   : num [1:57] 37767 37767 37767 37767 37767 ...
 $ pvalue    : num [1:57] 8.2e-07 1.6e-05 2.8e-05 3.6e-05 1.4e-04 ...
 $ FDR       : num [1:57] 0.00013 0.0013 0.0017 0.0017 0.0049 ...
 $ entries   : chr [1:57] "// AT1G76650 // AT1G01470 // AT2G22330 // AT2G30020 // …" ...
 $ dataSet   : Factor w/ 8 levels "K60dvsw","K40dvsw",..: 1 1 1 1 1 2 2 2 2 2 ...
 $ enrich    : num [1:57] 0.03553 0.00776 0.0262 0.025 0.01336 ...
```

7. We are now all set to generate our plot! There are some variables at the top of the code cell for **`max_dotsize`**: this sets how big our bubbles in the visualization will get; **`enrichMax`**: this sets the upper limit on our enrich values when setting the bubble size scale; **`min_FDR`**: this sets the lower limit of our FDR values when setting the colour scale.

```
options(repr.plot.width = 16, repr.plot.height = 10)
# Set a maximum dot size
max_dotsize <-6
enrichMax <- max(sigData.df$enrich)
min_FDR = min(sigData.df$FDR)

# The hardest part of printing this is you want to maintain
# (as much of) the original goTermKeep values
# On the other hand you want to drop any data where FDR >
# 0.01 or the term_type != "P"

# If you filter the dataset this way, you'll lose y-axis
# values, even if converted to a factor with levels!
# To retain those missing values/levels from the factor,
# set scale_y_discrete(drop = FALSE)

ggplot(sigData.df) +
   # 2. Aesthetics and Theme
   aes(x = dataSet, y = Term, colour = FDR, size = enrich) + #xy axes

   labs(x = NULL, y = NULL) +
   theme(axis.title.x = element_text(size = rel(0.7))) +

   # 3. Scaling
   scale_size_continuous(range = c(1, max_dotsize),
                         limits = c(0, enrichMax),
                         breaks = seq(0, enrichMax, 0.1),
                         guide = guide_legend(order=2)
                        ) +
   scale_y_discrete(drop = FALSE) +
   scale_x_discrete(drop = FALSE) +
#  scale_colour_gradient(limits = c(min_FDR, 0.01),
#                          low="gray36",
#                          high="gray87",
#                          guide = guide_colorbar(order=1)
#                         ) +
    scale_colour_viridis(guide = guide_colorbar(order=1)) +

    # 4. Geoms
    geom_point(aes(size = enrich, color = FDR))
```

See **Figure 5** for an example output. You can save the output with the **`ggsave("bubble_plot.pdf", width = 25,  height = 30,  units = "cm")`** command. The **`ggsave()`** function will automatically try to adjust the spacing of e.g. columns depending on what height and width you choose.

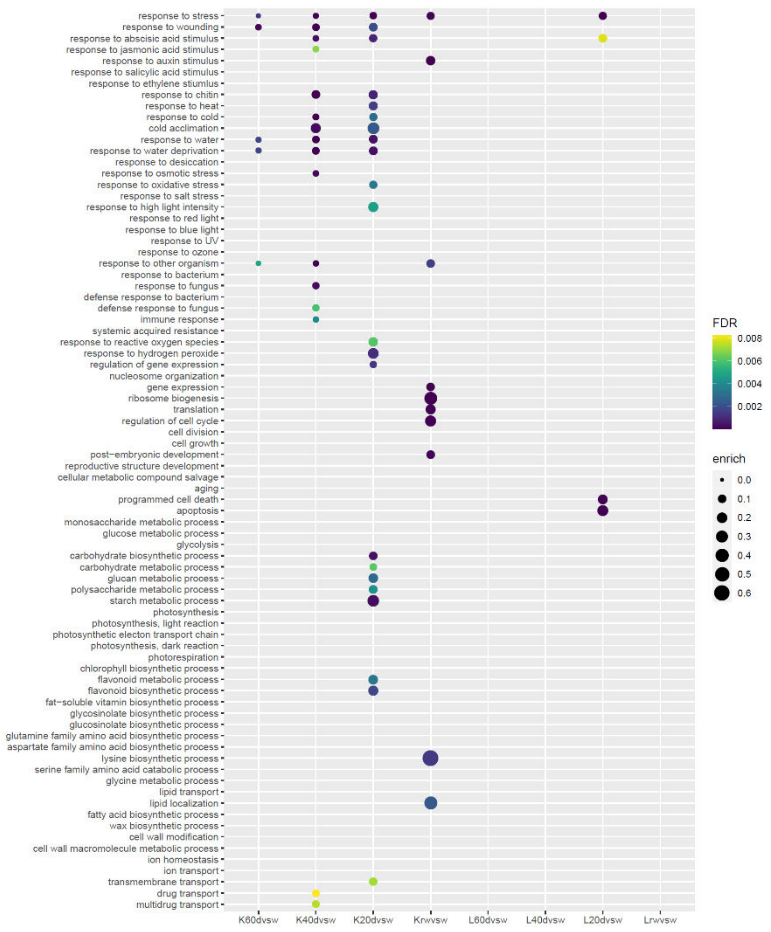


Figure 5. Output of the comparison for guard cell and whole leaf drought versus well-watered samples for our specified GO categories.

Lab Quiz Question 3

Note that for all of these tools, the enrichment outputs are only as good as the GO annotations themselves – only about 50% of Arabidopsis genes have a GO term associated with them, and this is similar in other model organisms. Further, different resources use different versions of GO annotation files, which can sometimes be quite out of date. Changes in GO annotations made over time can alter the results/interpretation of term enrichment results so be aware of this when performing your analyses. *Are you able to find out which version of GO data sets the tools in this lab use?*

End of lab!

Lab 4 Objectives

By the end of Lab 4 (comprising the lab including its boxes, and the lecture), you should be able to do the following:

- from a list of query genes, identify enriched classification terms (e.g. GO terms) and their exact significance levels in AgriGO, GOrilla, and g:Profiler;
- visualize the enrichment and significance of these terms;
- identify the evidence used to assign terms to genes;
- create a bubble chart of significantly enriched GO terms with R;

Do not hesitate to check with the Forums for this course on Coursera if you do not understand any of the above after reading the relevant material.

Additional Resources

AgriGO: https://doi.org/10.1093/nar/gkq310

Du, Z., Zhou, X., Ling, Y., Zhang, Z. & Su, Z. (2010) "AgriGO: a GO analysis toolkit for the agricultural community", Nucleic Acids Research,vol. 38, no. suppl_2, pp. W70.

GOrilla: https://dx.doi.org/10.1186/1471-2105-10-48

Eden, E., Navon, R., Steinfeld, I., Lipson, D. and Yakhini, Z. (2009) "GOrilla: A Tool For Discovery And Visualization of Enriched GO Terms in Ranked Gene Lists", BMC Bioinformatics 10:48.

g:Profiler: https://doi.org/10.1093/nar/gkw199

Reimand, J., Arak, T., Adler, P., Kolberg, L., Reisberg, S., Peterson, H. & Vilo, J. (2016) "g:Profiler—a web server for functional interpretation of gene lists (2016 update)", Nucleic Acids Research, vol. 44, no. W1, pp. W89.

A nice overview of enrichment analysis is available from this BioRXiv paper by Riemand et al., https://www.biorxiv.org/content/biorxiv/early/2017/12/12/232835.full.pdf