Differential expression module-guide

Introduction

Different genes are activated and transcribed into RNA transcripts at different times to respond to environmental cues. For example, when yeast is growing in media containing glucose, genes involved in metabolizing glucose are active; however, if yeast are put into media with galactose, genes involved in metabolizing glucose become inactive, and genes involved in metabolizing galactose become active. We can use RNA sequencing methods to quantify RNA transcript levels for both genes to get a sense of their activity in different stages of growth or in different conditions. If we looked at the expression of genes in glucose compared to galactose conditions, we would find that both the glucose and galactose genes have different activity levels, or are differentially expressed, in these two conditions. Seeing that a gene is differentially expressed in different conditions suggests it is transcriptionally regulated and potentially involved in pathways related to that condition.

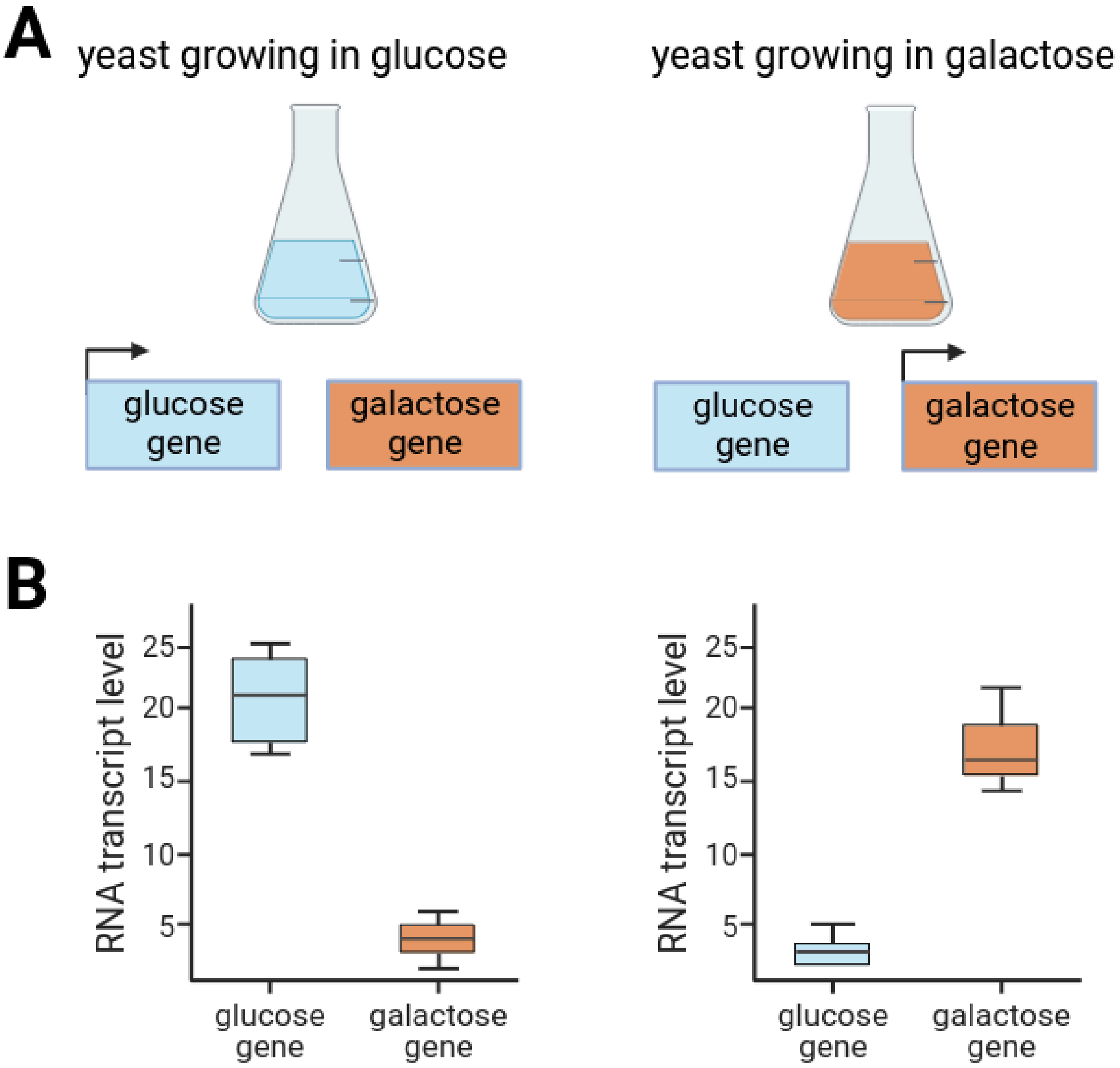


Figure 1: **A**) Genes involved in glucose metabolism are active (expressed) when yeast is grown in glucose, and genes involved in galactose metabolism are not active. The converse is true when yeast is grown in galactose. **B**) A hypothetical expression profile graph showing the RNA expression level for the glucose and galactose gene when grown in glucose (left) and galactose (right). Figure made with BioRender.com

This guide will walk through how to identify genes and proto-genes that are differentially expressed between two experimental conditions using the online web tool GEO2R. Follow along and answer the questions in the accompanying worksheet.

Goals: In this module, participants will learn:

1) Learn how to search databases for publicly available datasets of interest based on species, experiment, and condition types
2) How to identify differentially expressed genes using expression data such as microarray and RNA seq datasets with online GEO2R tool
3) Learn how to do QC checks and interpret results of differential expression using RNA seq data
4) Learn how to filter and query results using Excel or R

*Note to instructors:
1. We recommend using R with Google Colab to query the results (the second half of this module). To use Google Colab, you need to be logged into a Gmail account. If this is not feasible, we have added instructions in the appendix on how to perform these queries in Excel. However, we caution that the commands and functions may differ between versions of Excel.
2. If technical errors occur, the output files from this module (list of DE proto-genes and list of non-DE proto-genes) can be found in the following Google drive folder: https://tinyurl.com/DifferentialExpression
There are different folders corresponding to the GEO dataset you are using.

Activity

Query GEO & perform differential expression in GEO2R

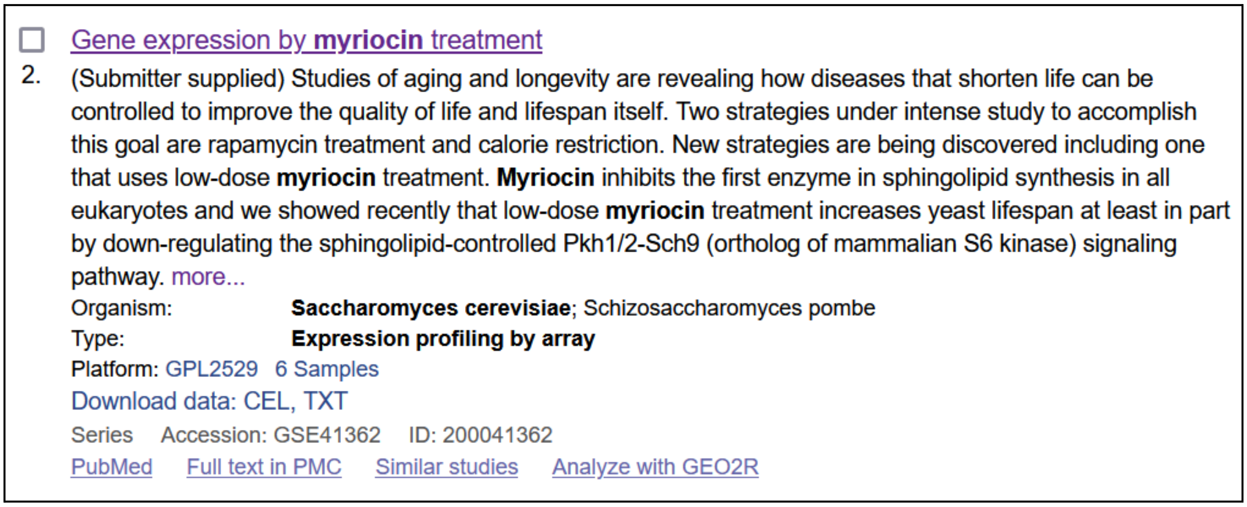
1. Go to the Gene Expression Omnibus (GEO) website: https://www.ncbi.nlm.nih.gov/gds
GEO is a database for microarray or RNA seq data managed by the NCBI

2. In the search bar type in your GEO study or construct a query to look for a dataset of interest. Below we detail how to construct a GEO query.
In this example we will be looking for RNA expression datasets that are: compatible with the GEO2R tool, are in the yeast species *S. cerevisiae,* and include the keyword "myriocin". The following query will search for datasets meeting this criteria:

```
(geo2r) AND Saccharomyces cerevisiae[Organism] AND (“Expression
profiling by array”[DataSet Type] OR “Expression profiling by high
throughput sequencing”[DataSet Type]) AND “myriocin”
```

More info on how to construct a query in GEO can be found here: https://www.ncbi.nlm.nih.gov/geo/info/qqtutorial.html This includes things such as searching for specific experimental conditions or specific types of experimental assays.

For this tutorial, we will be using the GEO dataset GSE41362, which looks at the effect of gene expression with and without myriocin treatment, which is a compound that inhibits sphingolipid synthesis. Follow along with your GEO dataset.



☐ Gene expression by **myriocin** treatment

2. (Submitter supplied) Studies of aging and longevity are revealing how diseases that shorten life can be controlled to improve the quality of life and lifespan itself. Two strategies under intense study to accomplish this goal are rapamycin treatment and calorie restriction. New strategies are being discovered including one that uses low-dose **myriocin** treatment. **Myriocin** inhibits the first enzyme in sphingolipid synthesis in all eukaryotes and we showed recently that low-dose **myriocin** treatment increases yeast lifespan at least in part by down-regulating the sphingolipid-controlled Pkh1/2-Sch9 (ortholog of mammalian S6 kinase) signaling pathway. more...

Organism:	**Saccharomyces cerevisiae**; Schizosaccharomyces pombe
Type:	**Expression profiling by array**

Platform: GPL2529 6 Samples

Download data: CEL, TXT

Series Accession: GSE41362 ID: 200041362

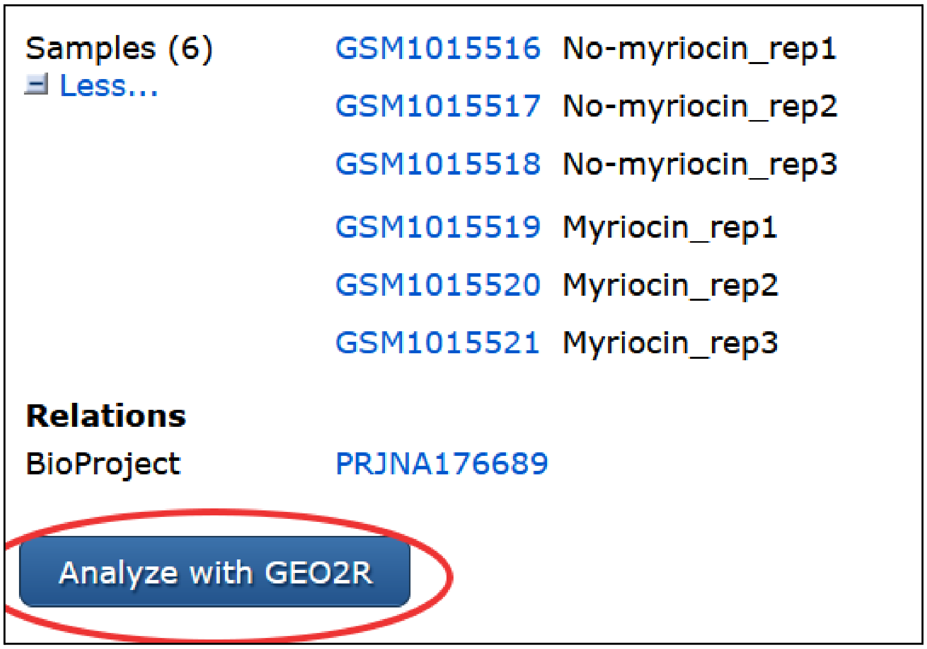
PubMed Full text in PMC Similar studies Analyze with GEO2R

3. Click on the dataset title link. This will bring you to a page that contains more information about the study, including things like what was the researcher's experimental question, link to the paper (if published), how many samples and replicates the experiment includes, etc.

   Answer question 1 in the worksheet: What is your GEO study? Give some detail about your study (i.e what was the research hypothesis, what treatment or conditions are being used?)

   Answer question 2 in the worksheet: How many samples does the dataset include? How many replicates?

4. Now we would like to identify which genes are expressed differently when the organism is exposed to the drug myriocin (treatment) compared to no drug (control). Scroll down towards the bottom of the page and click on the ***Analyze with GEO2R button***.

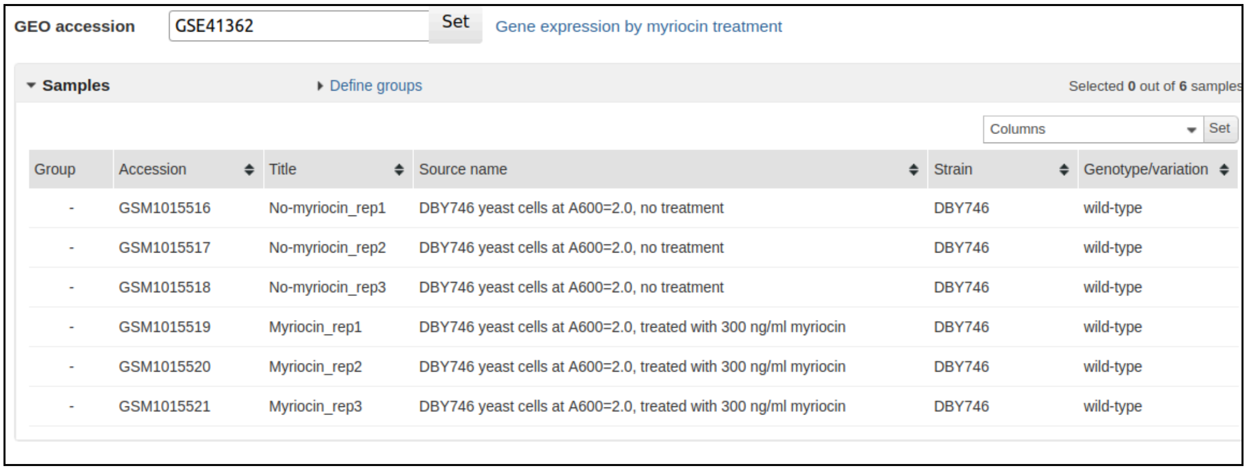


Samples (6)	GSM1015516	No-myriocin_rep1
	GSM1015517	No-myriocin_rep2
	GSM1015518	No-myriocin_rep3
	GSM1015519	Myriocin_rep1
	GSM1015520	Myriocin_rep2
	GSM1015521	Myriocin_rep3

Relations

BioProject PRJNA176689

5. Up will come a list of the samples in this dataset.



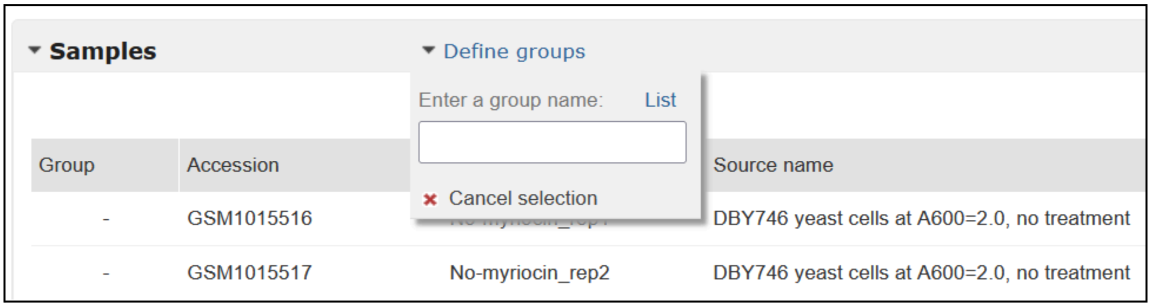
GEO accession GSE41362 Gene expression by myriocin treatment

Samples

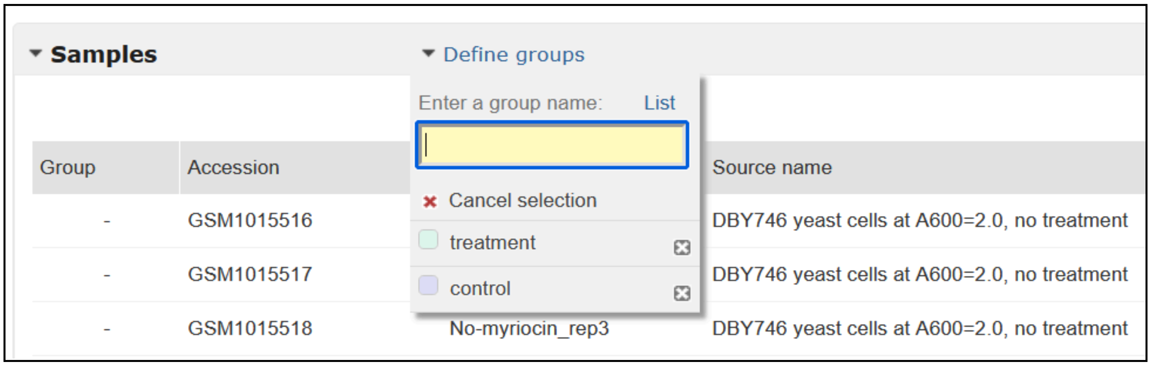
Selected 0 out of 6 samples

Group	Accession	Title	Source name	Strain	Genotype/variation
-	GSM1015516	No-myriocin_rep1	DBY746 yeast cells at A600=2.0, no treatment	DBY746	wild-type
-	GSM1015517	No-myriocin_rep2	DBY746 yeast cells at A600=2.0, no treatment	DBY746	wild-type
-	GSM1015518	No-myriocin_rep3	DBY746 yeast cells at A600=2.0, no treatment	DBY746	wild-type
-	GSM1015519	Myriocin_rep1	DBY746 yeast cells at A600=2.0, treated with 300 ng/ml myriocin	DBY746	wild-type
-	GSM1015520	Myriocin_rep2	DBY746 yeast cells at A600=2.0, treated with 300 ng/ml myriocin	DBY746	wild-type
-	GSM1015521	Myriocin_rep3	DBY746 yeast cells at A600=2.0, treated with 300 ng/ml myriocin	DBY746	wild-type

Next, we need to define the sample groups: i.e. which samples are part of the control group and which are part of the treatment group. To do this, click on the ***define groups*** link. For this example, I will use the group names “treatment” and “control”. Type in *treatment* into the text box and hit enter followed by *control* and hit enter.
*if your dataset has multiple time points and conditions, choose one time point and one condition to compare.



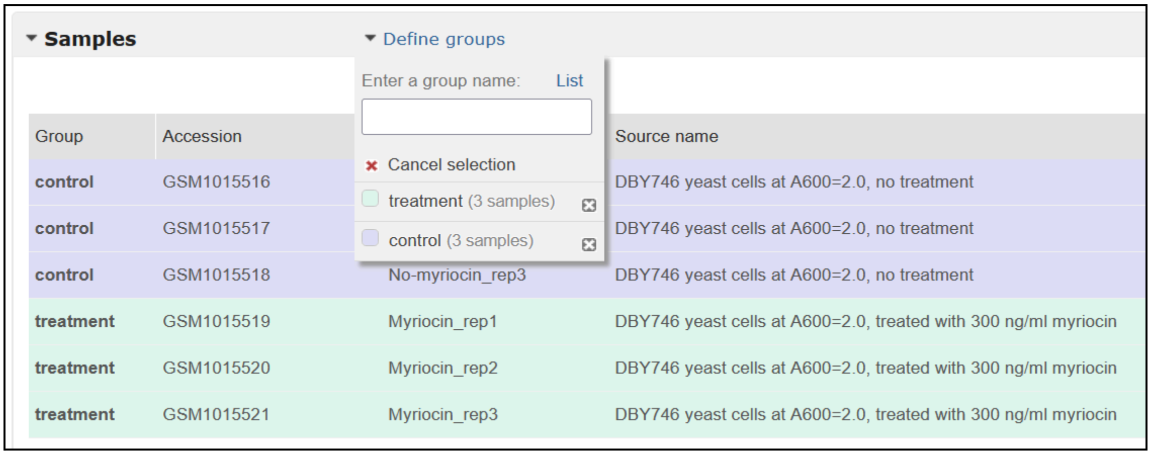
Group	Accession		Source name
-	GSM1015516		DBY746 yeast cells at A600=2.0, no treatment
-	GSM1015517	No-myriocin_rep2	DBY746 yeast cells at A600=2.0, no treatment



Group	Accession		Source name
-	GSM1015516		DBY746 yeast cells at A600=2.0, no treatment
-	GSM1015517		DBY746 yeast cells at A600=2.0, no treatment
-	GSM1015518	No-myriocin_rep3	DBY746 yeast cells at A600=2.0, no treatment

While holding shift, click on the three treatment samples to select them. Then click the ***treatment group*** button in the group tab. Repeat for the control samples.

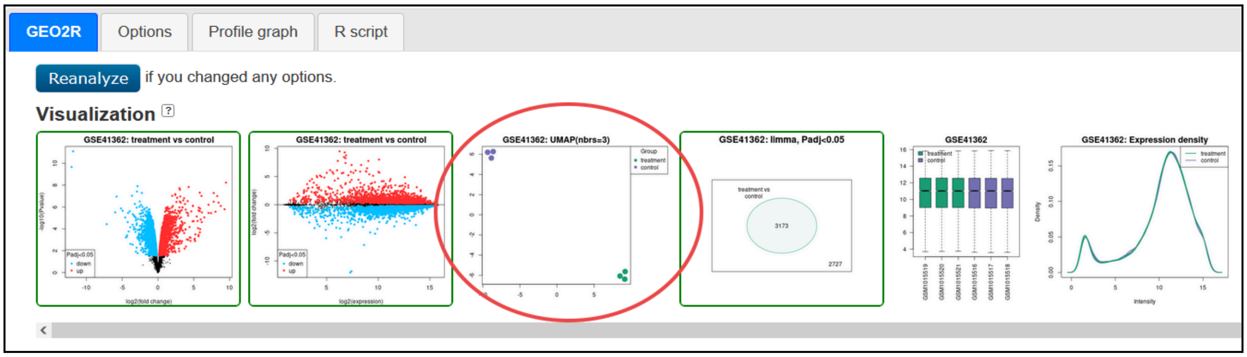
The order in which you define the groups matters for interpreting the results. The group you define second will be the baseline group, such that increases or decreases in expression will be relative to this group. In this example, be sure to define the treatment group first, then the control group. More explanation is provided in step 7.



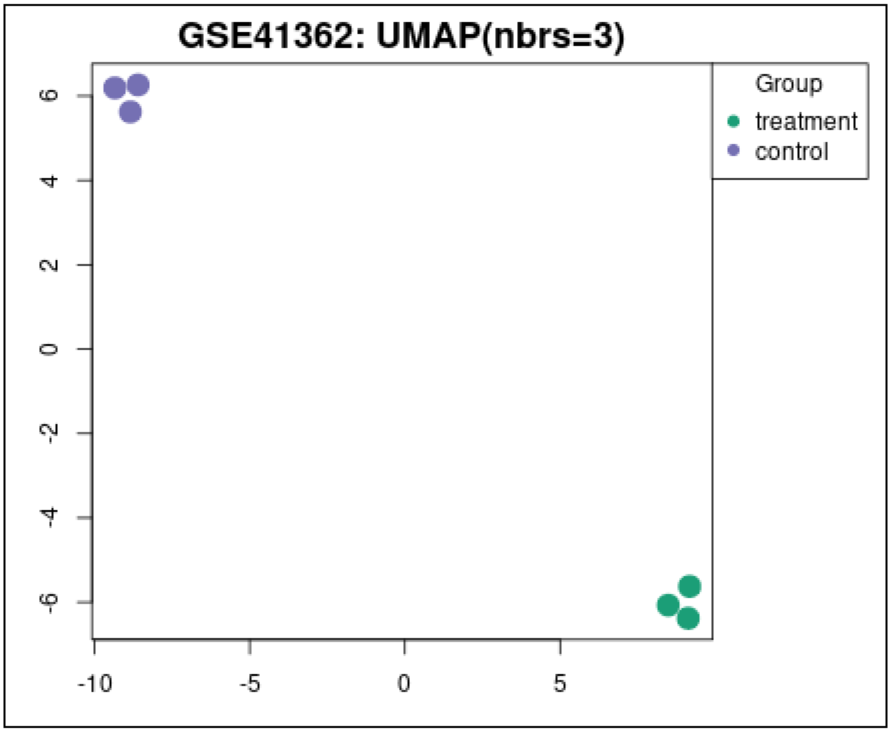
Group	Accession		Source name
control	GSM1015516		DBY746 yeast cells at A600=2.0, no treatment
control	GSM1015517		DBY746 yeast cells at A600=2.0, no treatment
control	GSM1015518	No-myriocin_rep3	DBY746 yeast cells at A600=2.0, no treatment
treatment	GSM1015519	Myriocin_rep1	DBY746 yeast cells at A600=2.0, treated with 300 ng/ml myriocin
treatment	GSM1015520	Myriocin_rep2	DBY746 yeast cells at A600=2.0, treated with 300 ng/ml myriocin
treatment	GSM1015521	Myriocin_rep3	DBY746 yeast cells at A600=2.0, treated with 300 ng/ml myriocin

6. After selecting the samples, click the ***Analyze*** button. Up will come many visualization graphs and a list of the top differentially expressed genes. Before we start looking into the differentially expressed genes we want to do some quality control to make sure our samples and data looks good.

   First we will look at the UMAP plot to see if our samples cluster by groups.

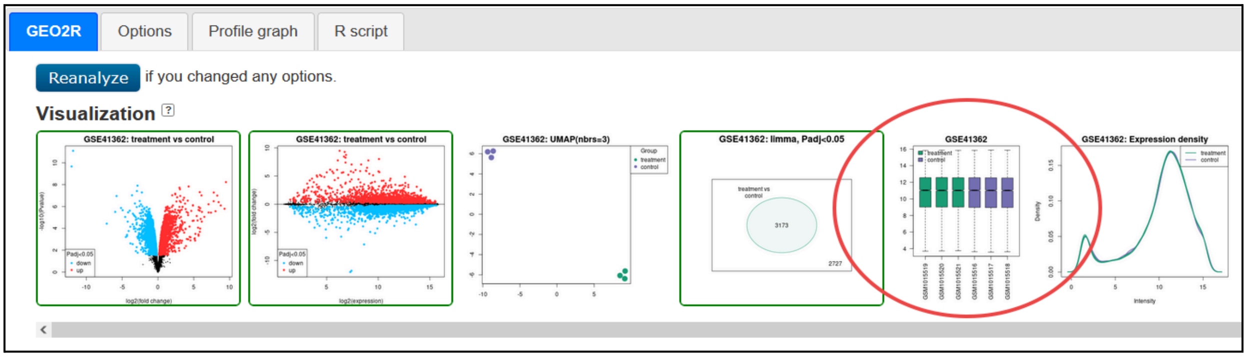


UMAP is a method that collapses the information about many variables, in this case the expression of genes, into only two dimensions, allowing it to be easily plotted and visualized. This means that samples that are closer together in the plot are more similar in terms of their gene expression, and samples that are further apart are less similar. In general, having samples that cluster, or group together, by experimental condition suggests that there is difference in the gene expression between the two groups that can be investigated and that samples were not mislabeled.



<u>Answer question 3 in the worksheet</u>: Add a picture of the UMAP plot for the samples in your expression set. Do the samples cluster by group?

Next we will check the boxplot to see the expression distribution for each sample to see if the values are properly normalized.


GEO2R
Options
Profile graph
R script
Reanalyze
if you changed any options.
Visualization
GSE41362: treatment vs control
GSE41362: treatment vs control
GSE41362: UMAP(nbrs=3)
GSE41362: limma, Padj<0.05
GSE41362
GSE41362: Expression density

Normalization is a mathematical way to put the expression levels of genes in different samples on a similar scale, which allows for the expression of a gene to be compared across the samples. Normalization is important because it can reduce biases that can be caused by technical (non-biological) reasons such as accounting for the total amount of RNA sequenced in each sample. For example, if we find twice as much RNA for a gene in sample 1 compared to sample 2 but also sequenced twice as much RNA in total for sample 1, then if we account for the total amount sequenced, we would say the two samples have a similar expression for this gene.

A graph that shows the boxes with different distributions across samples, especially different medians, such as the graph in Figure 2a, suggests the values are not normalized and cannot be properly compared. In contrast, a graph with similar distributions and similar medians, such as Figure 2b, suggests that the values are normalized and can be compared.

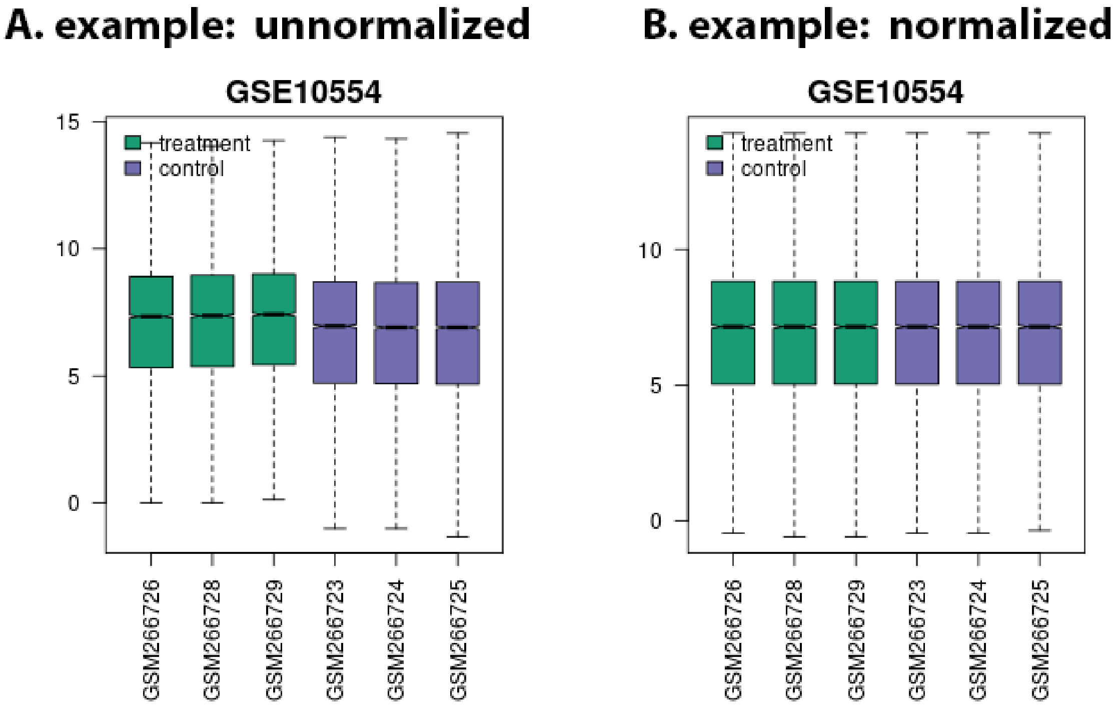
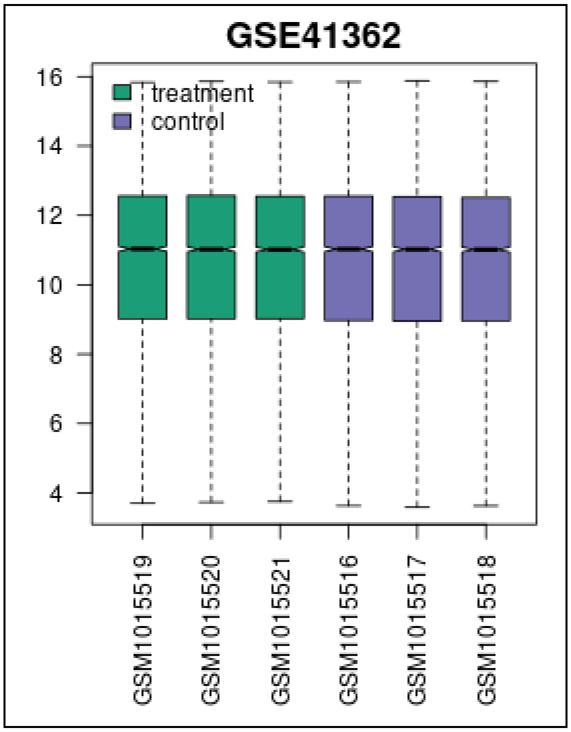
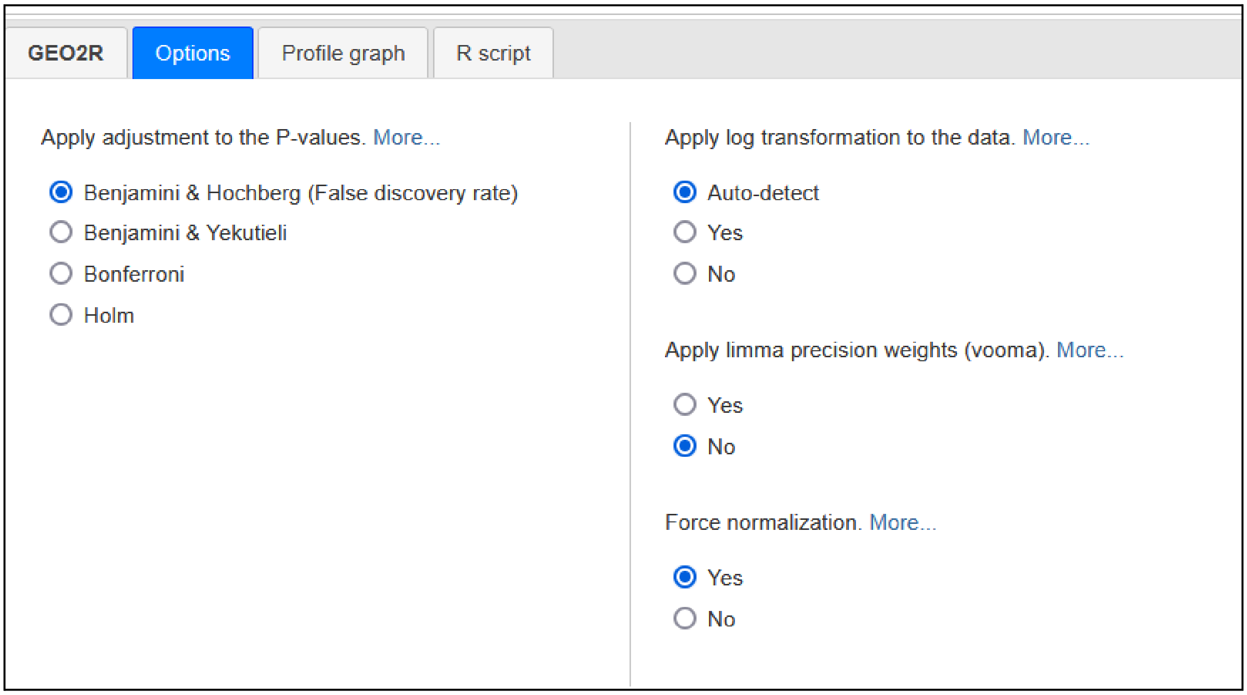


Figure 2: Example expression data **A**) before normalization and **B**) after normalization.

The samples in this dataset, GSE41362, appear to be well normalized and no further action is needed.



However, if your samples do not have similar medians, you can normalize the data by clicking on the ***Options*** tab and select ‘Yes’ under the ***Force normalization*** option. Then click the ‘***Reanalyze***’ button to rerun the analysis.



Apply adjustment to the P-values. More...

- ● Benjamini & Hochberg (False discovery rate)
- ○ Benjamini & Yekutieli
- ○ Bonferroni
- ○ Holm

Apply log transformation to the data. More...

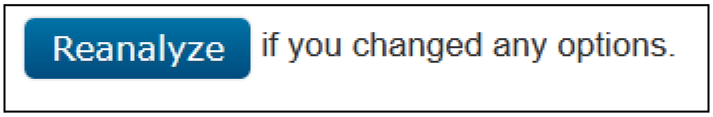
- ● Auto-detect
- ○ Yes
- ○ No

Apply limma precision weights (vooma). More...

- ○ Yes
- ● No

Force normalization. More...

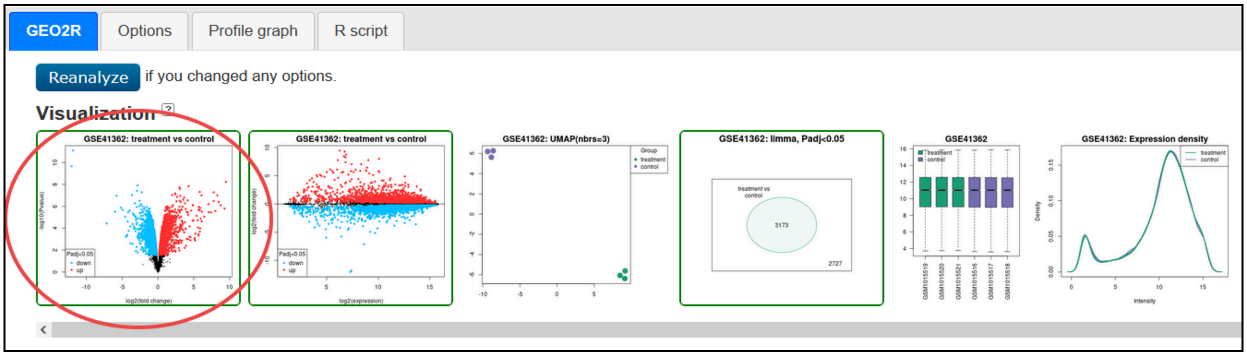
- ● Yes
- ○ No



if you changed any options.

<u>Answer question 4 in the worksheet</u>: Add a picture of the expression boxplot for the samples in your expression set. Do the samples appear to be normalized?

7. Next we will look at the volcano plot to explore the most upregulated and downregulated genes in the treatment group compared to the control group.



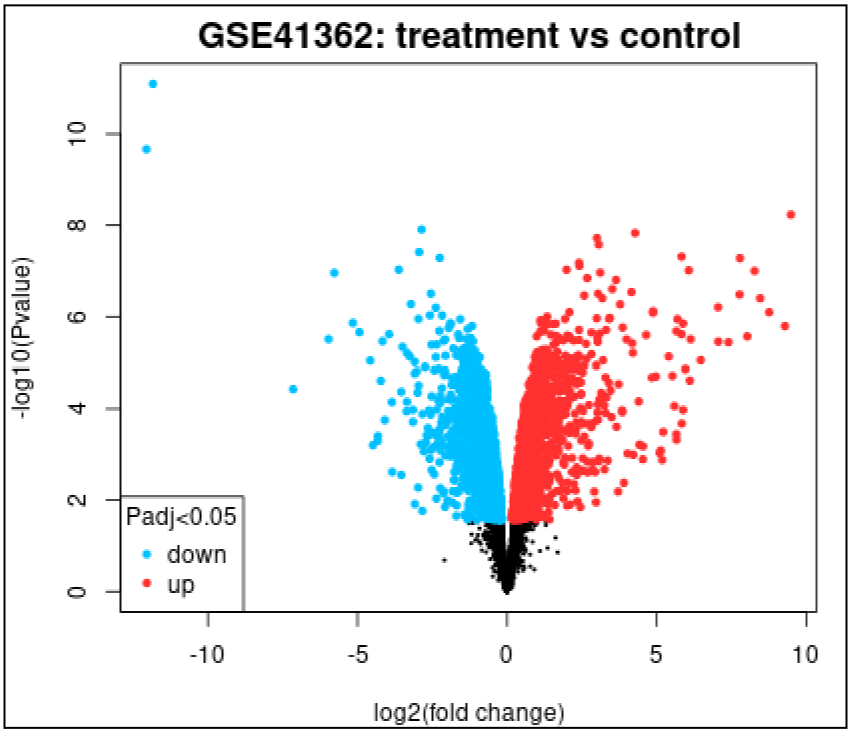
Changes in expression are measured as fold changes, which is the expression of $gene_1$ in the treatment condition divided by the expression of $gene_1$ in the control. This value tells you how much more or less the gene is expressed. Fold change values are often transformed by using log_2 transformation, which tells you how many times the expression doubled (or halved). For example, a log_2 (fold change) of 1 means the expression in the treatment condition is 2^1 times greater than in the control, and a $\log_2$ (fold change) of -1 means the expression in the treatment is 2^{-1} times (ie ½ as much) as in the control.

In general, the formula for how many times more (or less) a gene is expressed is:

$$2^{\text{log2(fold change)}}$$

The other metric we are interested in looking at when determining if a gene is differentially expressed is the p-value. This tells us how confident we can be in this observation, where the lower the p-value the more confident we are and the higher the p-value the less confident we are. For some plots, like the volcano plot below, it is useful to transform the p-values by $-\log_{10}$. This means that lower p-values, i.e. ones we are more confident in, become larger values, and larger p-values, i.e. ones we are less confident in, become smaller values.

A volcano plot has the change in expression, $\log_2$(fold change), on the x-axis and how significant/confident we are in this change, $-\log_{10}$(p-value), on the y-axis. Therefore genes that have large increases in expression in the treatment group and we are confident it is not due to chance are located on the upper right hand side of the graph. Genes that have large decreases in expression in the treatment group and we are confident it is not due to random chance are located on the upper left hand side of the graph.



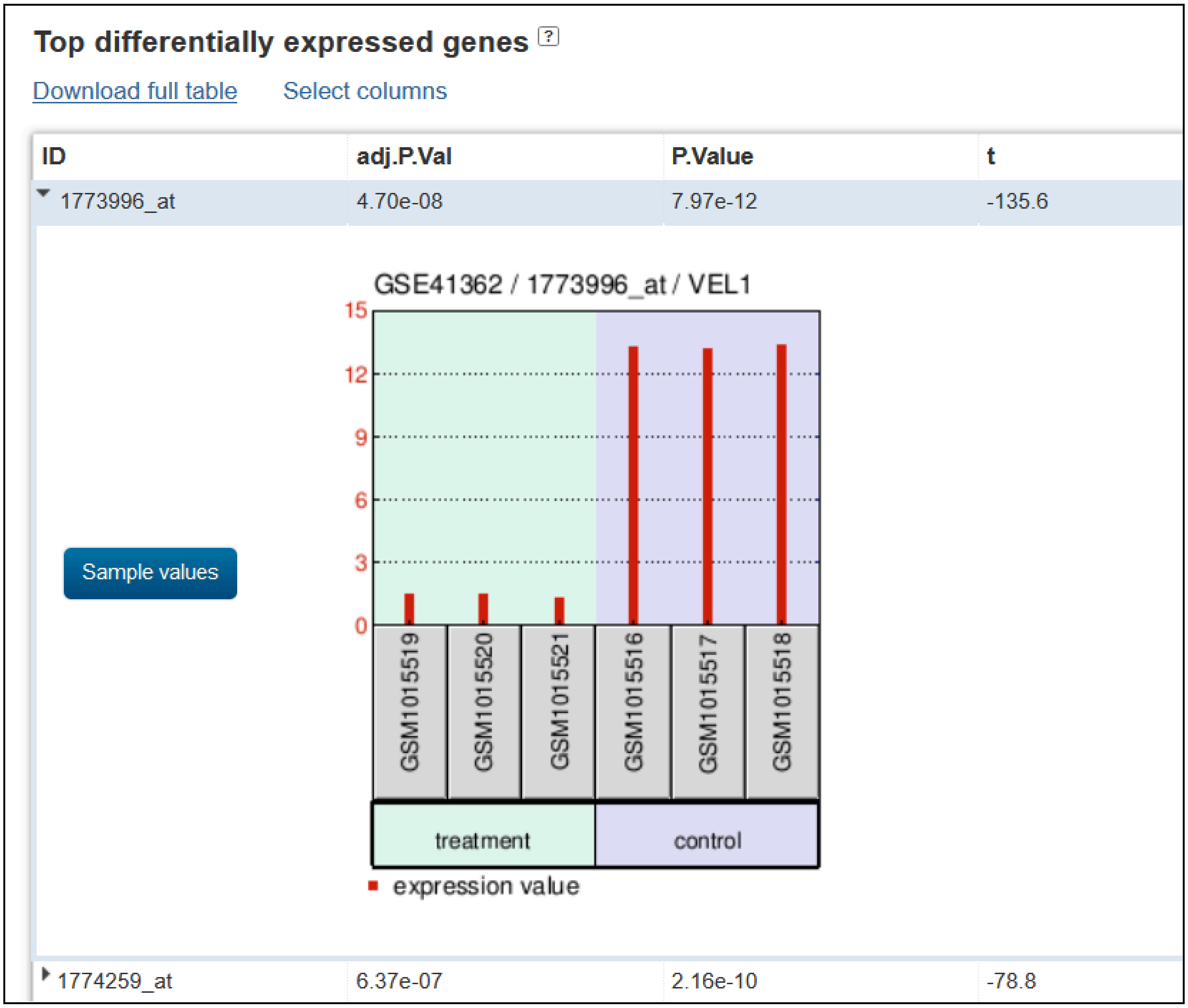
Now click on the ***Explore and download*** button to make the graph interactive. Hover over the points to see the names of the genes.

Answer question 5 in the worksheet: What is the most upregulated gene? What is its $\log_2$FC and p-value?

Answer question 6 in the worksheet: What is the most downregulated gene? What is its $\log_2$FC and p-value?

8. Check to make sure that the upregulated genes have higher expression in the treatment samples compared to the control samples and that downregulated genes have lower expression in the treatment compared to the control samples. You can check this by making a profile graph. A profile graph has the samples on the x-axis and plots the expression level of a given gene for each sample on the y-axis.

   To view the profile graph for a given gene, click on the gene's ID in the '***Top differentially expressed genes***' table. Let's look at the profile graph for a highly downregulated gene in this dataset, VEL1.

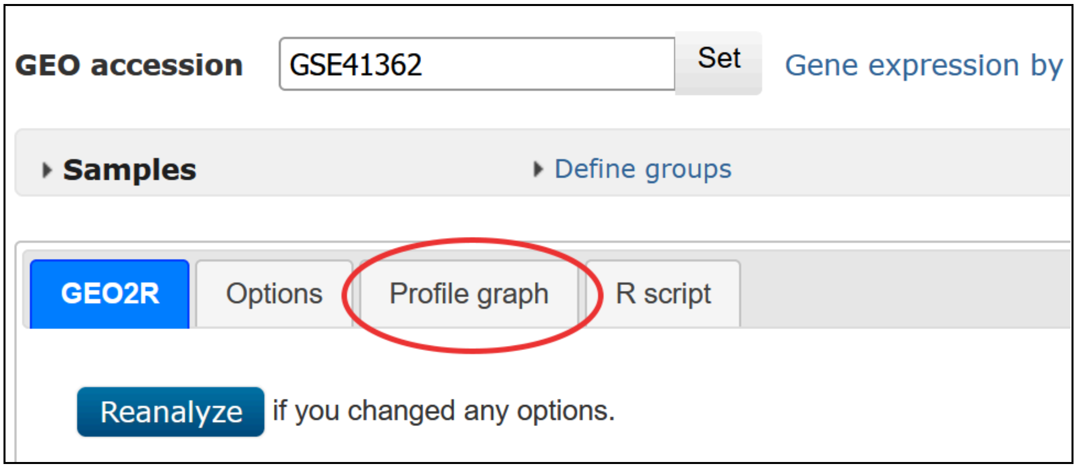


Top differentially expressed genes

Download full table   Select columns

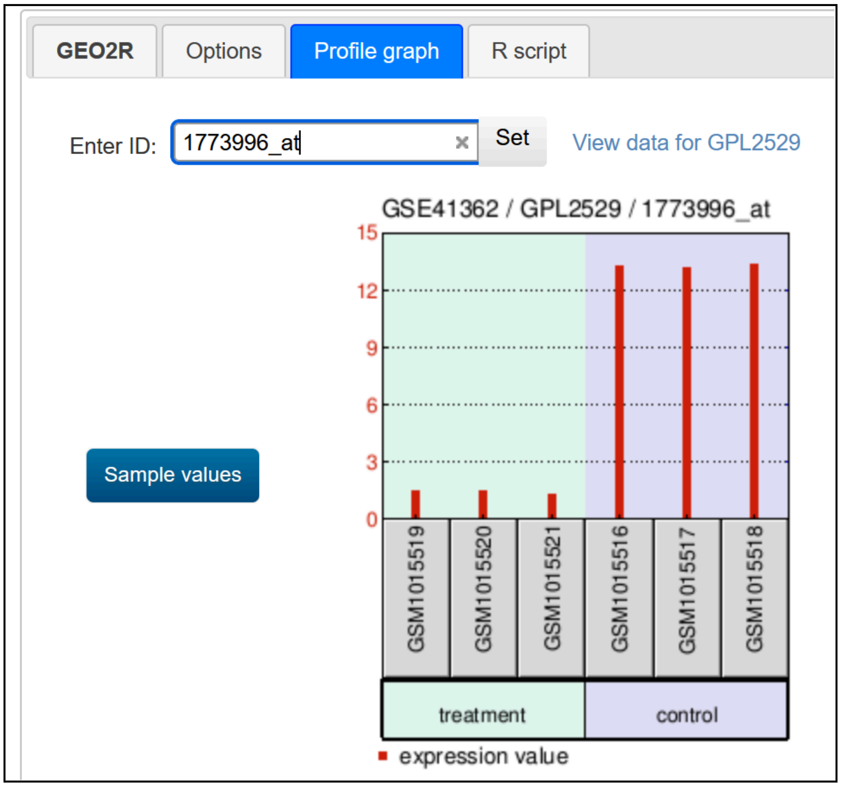
ID	adj.P.Val	P.Value	t
1773996_at	4.70e-08	7.97e-12	-135.6
1774259_at	6.37e-07	2.16e-10	-78.8

You can also get the profile graph by searching for the gene using its ID in the ***Profile graph*** tab. You can get the gene ID from the interactive volcano plot or from the '*Top differentially expressed genes'* table. The gene ID will be a sequence of numbers.



GEO accession GSE41362

Reanalyze if you changed any options.



Enter ID: 1773996_at

Figure 3: Profile graph for the gene VEL1 (ID 1773996_at). The x-axis is the samples, and the y-axis is the RNA expression level. This gene has much lower expression in the treatment samples compared to the control samples.

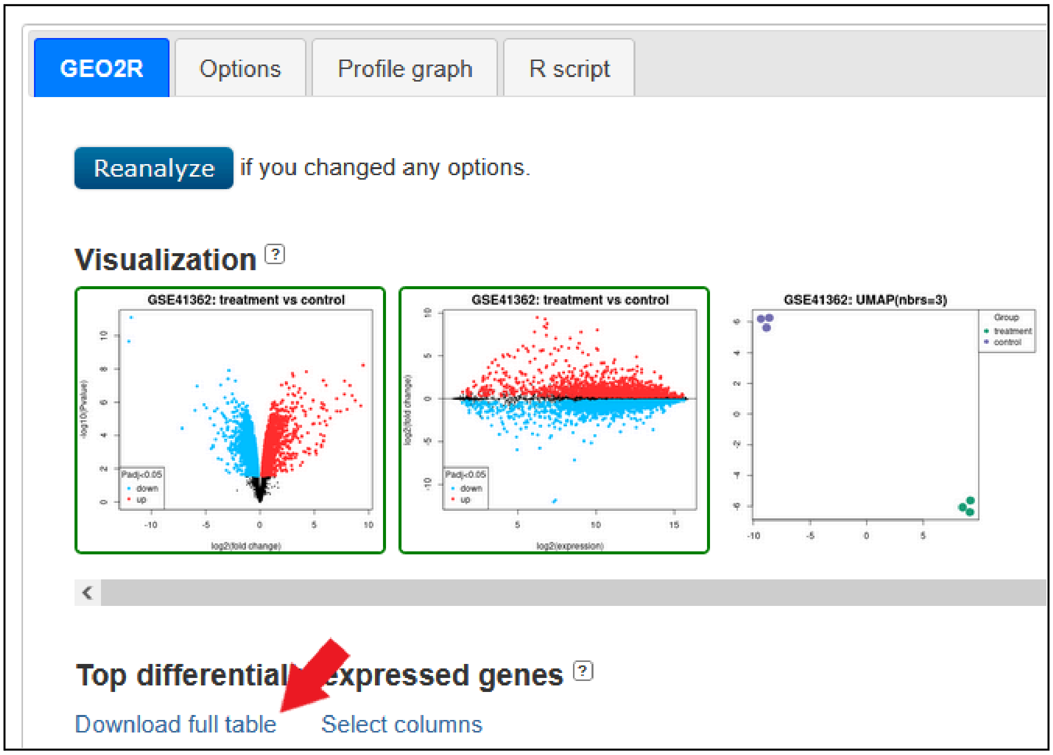
For the most downregulated gene we can see from the profile graph in Figure 3 that the expression of VEL1 (gene ID 1773996_at) is higher in the control samples compared to the treatment samples.

If this is not the case, this means you defined the control group first in step 5, which means the signs are just flipped and the opposite is true, a $\log_2$(fold change) of 1 means expression in the control groups is two times greater than in the treatment group and a $\log_2$(fold change) of -1 means expression is the control groups is two times less in the treatment group.

Answer question 7 in the worksheet: Add the profile graph for the most upregulated gene. How many times more is the gene expressed in the treatment samples compared to the control samples?

Answer question 8 in the worksheet: Add the profile graph for the most downregulated gene. How many times less is the gene expressed in the treatment samples compared to the control samples?

9. Now we will look for differentially expressed proto-genes. Click on the ***Download full table*** link



Reanalyze if you changed any options.

Visualization

Top differentially expressed genes

A tab-separated value (.tsv) file will be downloaded to your computer. In this example, the file is entitled GSE41362.top.table.tsv

Next, we will query our results using R. To use Excel instead, see the appendix section of this module.

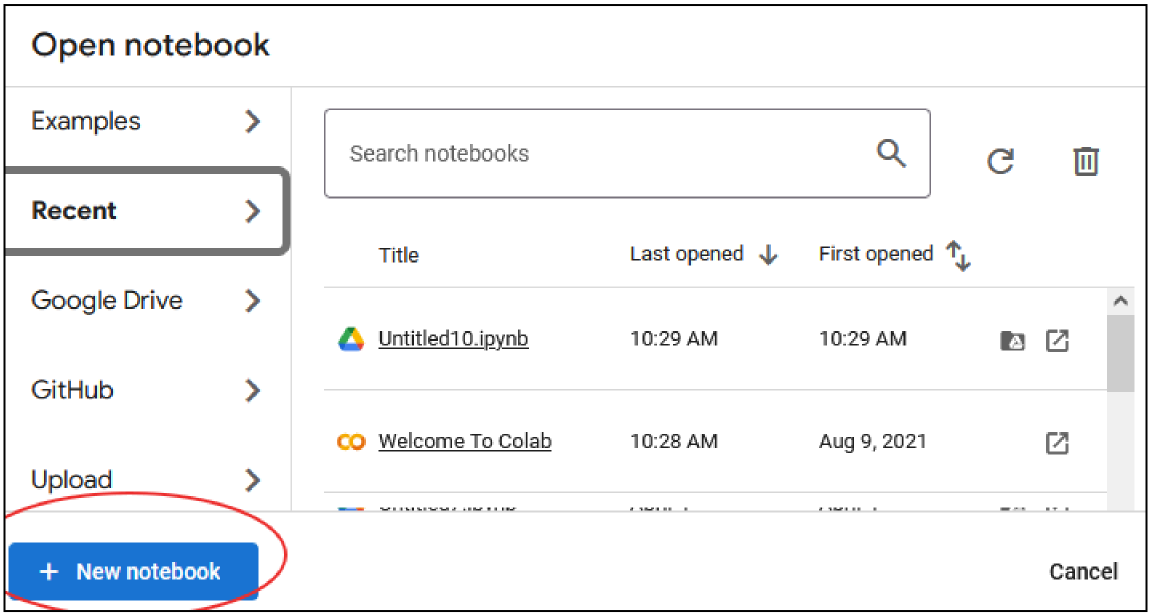
Use R to query results for DE proto-genes

An R Google Colab file with all the code can be accessed here:
https://colab.research.google.com/drive/1B_irLZvtL6ninglkw0gZu4sJwKGaqMGV?usp=sharing

1. Download a list of proto-genes as a comma-separated value (csv) file from the following link:
   https://docs.google.com/spreadsheets/d/1gvtiX-MWGHYL_Xv7SKyQ-IdbM63uG3qKwz8gyHkq9Kw/edit?usp=sharing

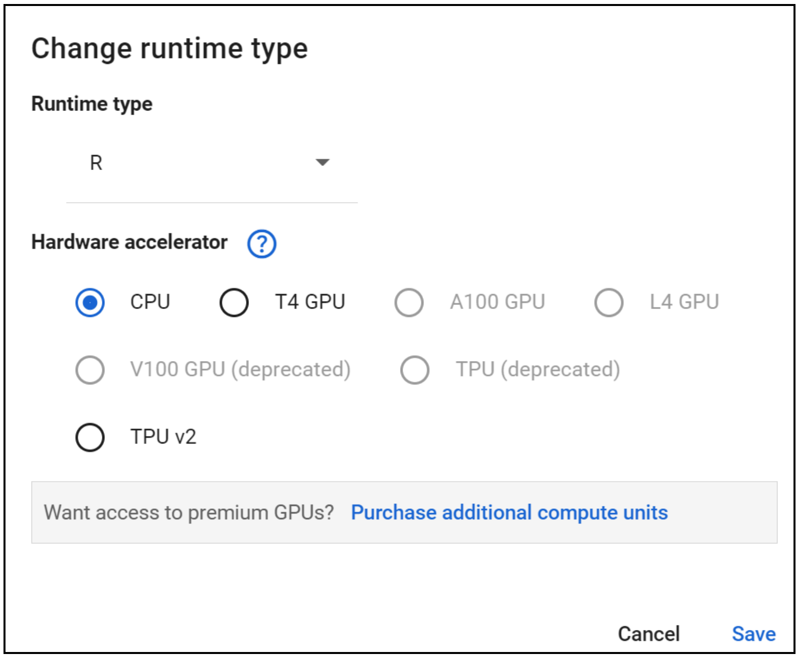
   Click ***File***, and select ***Download*** > ***Comma-separated values*** (.csv)

2. Open a new Google Colab notebook by clicking on the following link
   https://colab.research.google.com/notebook
   And in the window that pops up click the ***New notebook*** button. If there is no popup window, select ***File > New notebook*** in the top toolbar menu.



Title	Last opened	First opened
Untitled10.ipynb	10:29 AM	10:29 AM
Welcome To Colab	10:28 AM	Aug 9, 2021

3. Change the coding language to be R (default is python) by going to the ***Runtime tab*** in the top toolbar menu select ***Change runtime type*** and in the ***Runtime type*** drop down menu select ***R*** and click ***save***



Change runtime type

Runtime type

R

Hardware accelerator

● CPU ○ T4 GPU ○ A100 GPU ○ L4 GPU

○ V100 GPU (deprecated) ○ TPU (deprecated)

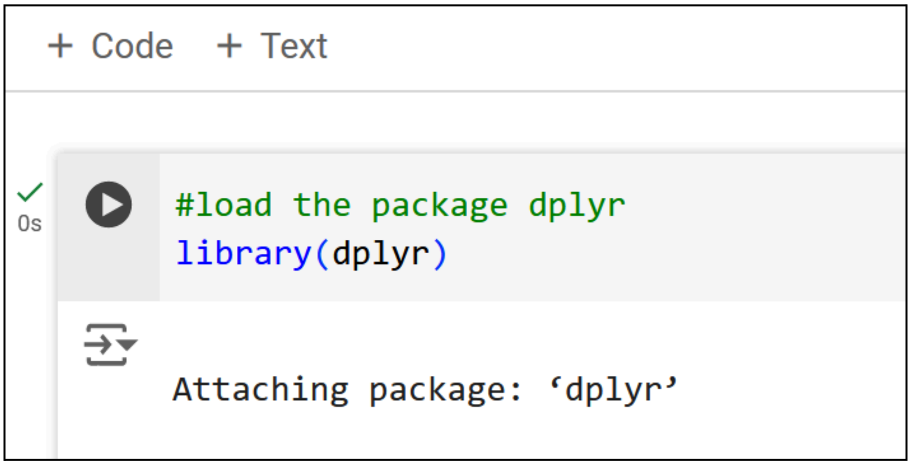
○ TPU v2

Want access to premium GPUs? Purchase additional compute units

4. To begin typing code click the **+ *Code*** button at the top. This will create a new code box for you to type in. In this first box we will type in the following command:

```
library(dplyr)
```

To run the command press the shift and enter key at the same time or press the play button beside the code box

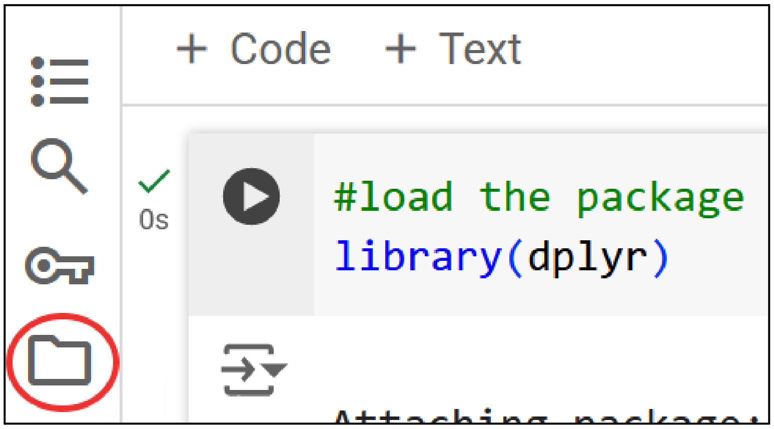


```
#load the package dplyr
library(dplyr)
```

```
Attaching package: ‘dplyr’
```

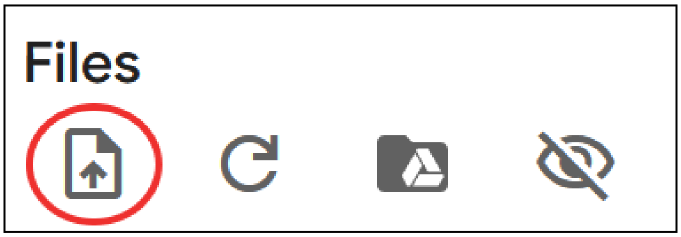
This command loads the dplyr package, which contains many useful functions for querying and analyzing data.

5. Next load the proto-genes file (“annotated_protogenes_list_workshop2024.csv”), and the DE results file (GSE41362.top.table.tsv“) into the collab environment by clicking the ***folder*** icon on the left menu and clicking the ‘***upload to session storage***’ icon.

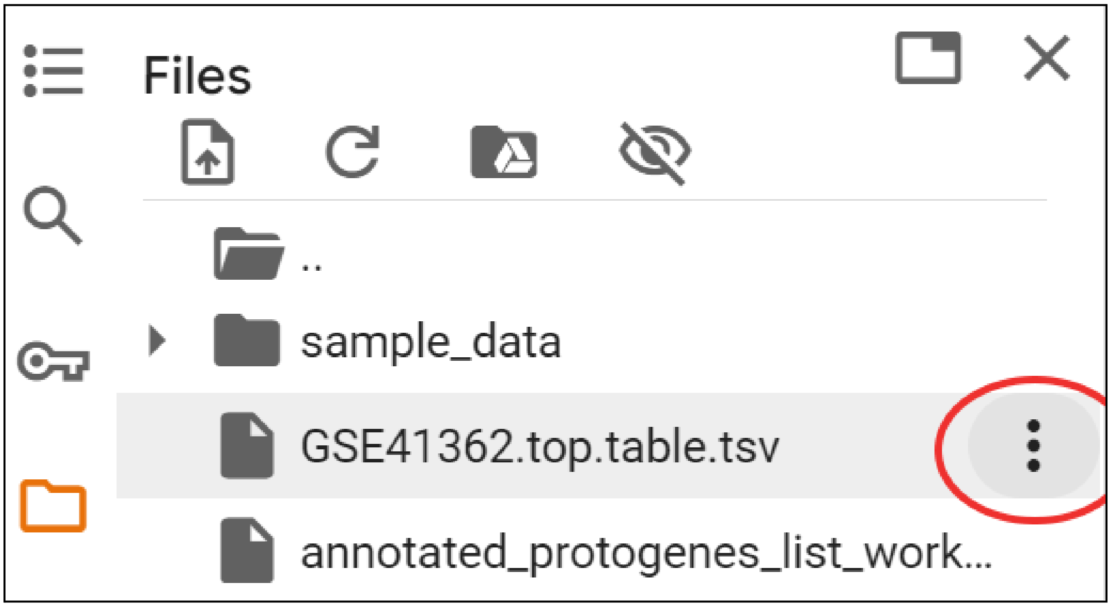


```
#load the package
library(dplyr)
```

```
Attaching package:
```

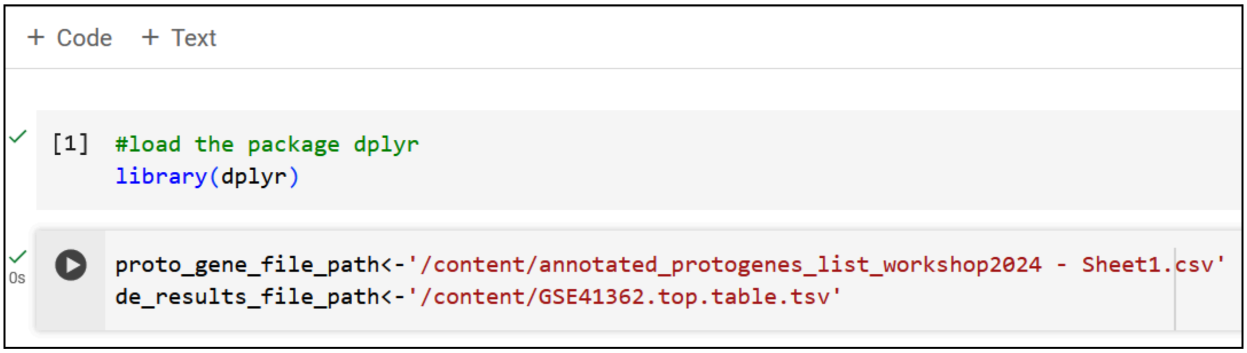


6. Get the file paths for both files by clicking on the 3 vertical dots beside the file names and clicking ***copy path***.



Save these paths into variables by adding another code box (**+ *Code*** button) and typing the commands below. Be sure to replace the text in quotes with the file paths you copied

```
proto_gene_file_path<-'/content/annotated_protogenes_list_workshop2024 - Sheet1.csv'
de_results_file_path<-'/content/GSE41362.top.table.tsv'
```



```
[1] #load the package dplyr
    library(dplyr)
```

```
proto_gene_file_path<-'/content/annotated_protogenes_list_workshop2024 - Sheet1.csv'
de_results_file_path<-'/content/GSE41362.top.table.tsv'
```

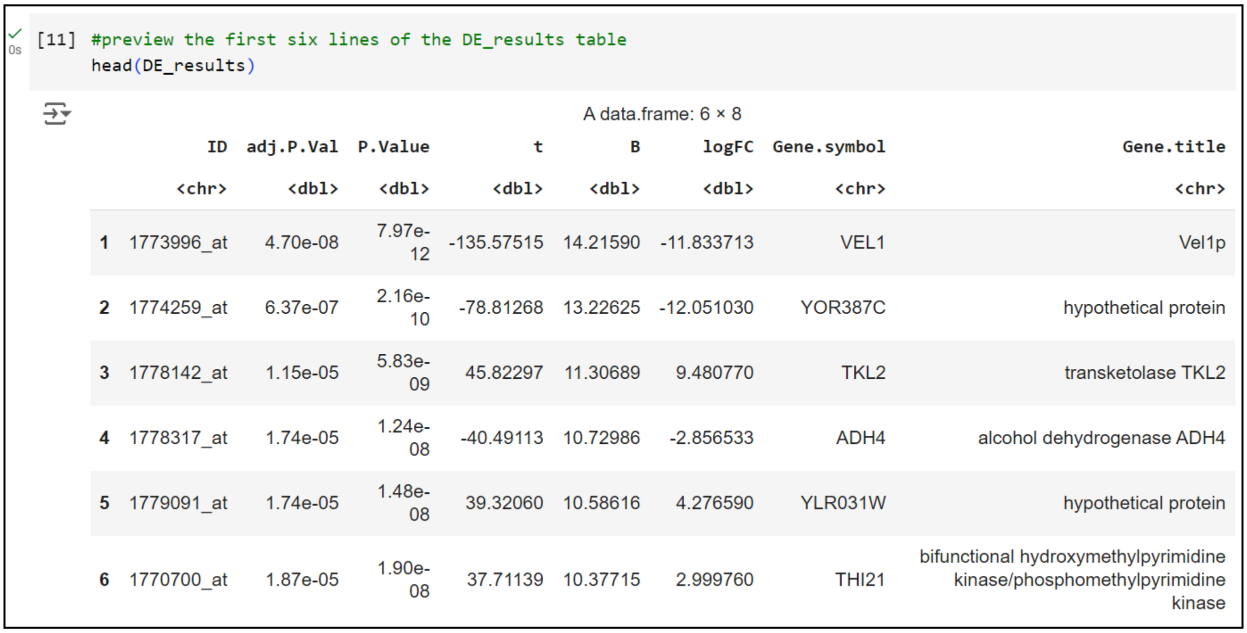
Hit the play arrow beside the code to run those lines or press the shift and enter key at the same time

7. In a new code box, read in the files using the function `read.delim()`

```
proto_genes<-read.delim(proto_gene_file_path,sep=',')
DE_results<-read.delim(de_results_file_path,sep='\t')
```

8. To preview the first six lines of the data use the `head()` function

```
head(DE_results)
```

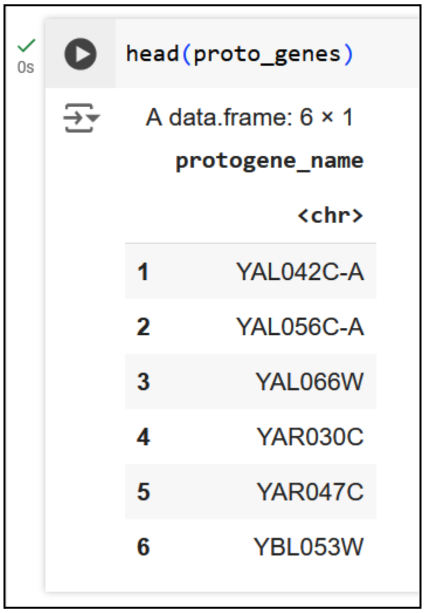


```
[11] #preview the first six lines of the DE_results table
head(DE_results)
```

A data.frame: 6 × 8

	ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
1	1773996_at	4.70e-08	7.97e-12	-135.57515	14.21590	-11.833713	VEL1	Vel1p
2	1774259_at	6.37e-07	2.16e-10	-78.81268	13.22625	-12.051030	YOR387C	hypothetical protein
3	1778142_at	1.15e-05	5.83e-09	45.82297	11.30689	9.480770	TKL2	transketolase TKL2
4	1778317_at	1.74e-05	1.24e-08	-40.49113	10.72986	-2.856533	ADH4	alcohol dehydrogenase ADH4
5	1779091_at	1.74e-05	1.48e-08	39.32060	10.58616	4.276590	YLR031W	hypothetical protein
6	1770700_at	1.87e-05	1.90e-08	37.71139	10.37715	2.999760	THI21	bifunctional hydroxymethylpyrimidine kinase/phosphomethylpyrimidine kinase

```
head(proto_genes)
```



```
head(proto_genes)
```

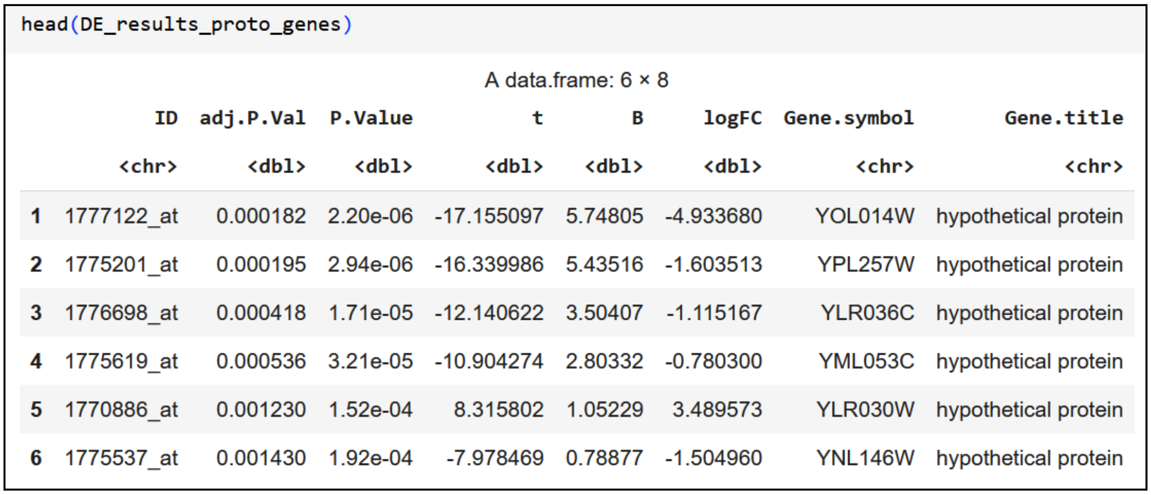
A data.frame: 6 × 1

	protogene_name
	<chr>
1	YAL042C-A
2	YAL056C-A
3	YAL066W
4	YAR030C
5	YAR047C
6	YBL053W

9. Now we will search for the rows in the `DE_results` table that contain one of the proto-gene names by searching in the `DE_results` table in the `Gene.symbol` column using a function from the dplyr package called `filter()`. Save the filtered table into a new variable called `DE_results_proto_genes` which contains the DE information for the proto-genes only.

*Note: in different datasets the column names may be different.

```
DE_results_proto_genes<-filter(DE_results, Gene.symbol %in%
proto_genes$protogene_name)
head(DE_results_proto_genes)
```



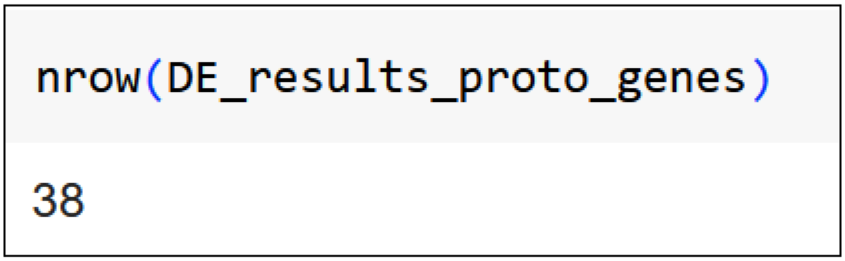
```
head(DE_results_proto_genes)
```

A data.frame: 6 × 8

	ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
1	1777122_at	0.000182	2.20e-06	-17.155097	5.74805	-4.933680	YOL014W	hypothetical protein
2	1775201_at	0.000195	2.94e-06	-16.339986	5.43516	-1.603513	YPL257W	hypothetical protein
3	1776698_at	0.000418	1.71e-05	-12.140622	3.50407	-1.115167	YLR036C	hypothetical protein
4	1775619_at	0.000536	3.21e-05	-10.904274	2.80332	-0.780300	YML053C	hypothetical protein
5	1770886_at	0.001230	1.52e-04	8.315802	1.05229	3.489573	YLR030W	hypothetical protein
6	1775537_at	0.001430	1.92e-04	-7.978469	0.78877	-1.504960	YNL146W	hypothetical protein

To see how many proto-genes were detected in this study, we can count the number of rows using the following line:

```
nrow(DE_results_proto_genes)
```



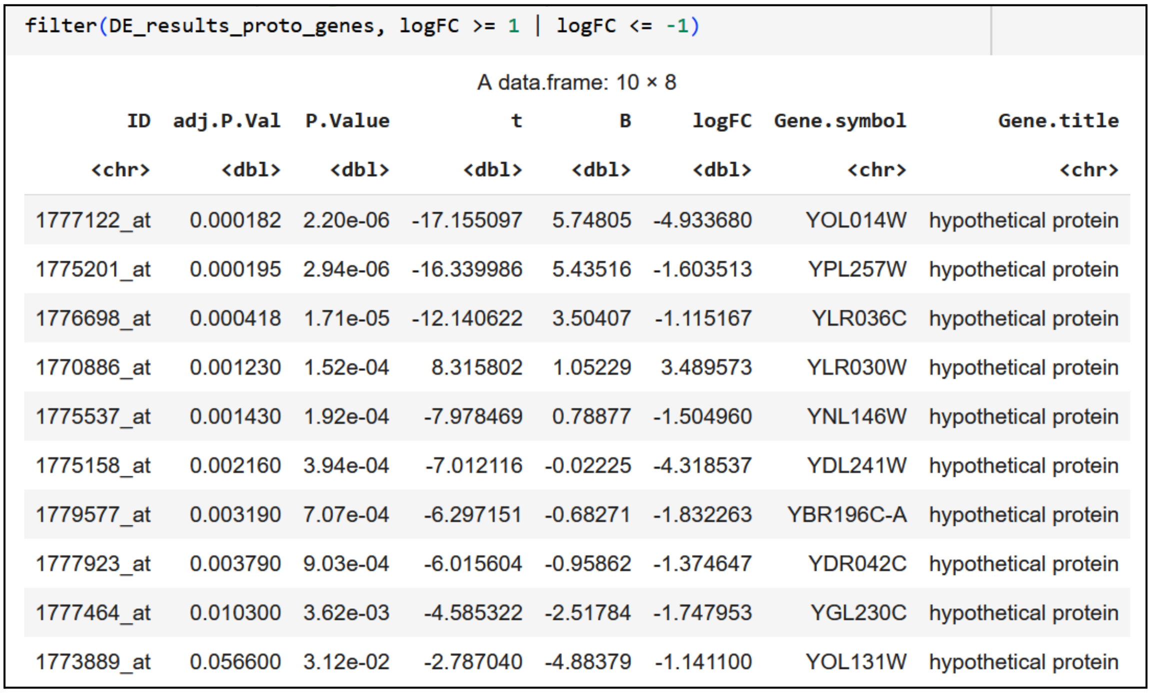
```
nrow(DE_results_proto_genes)
```

```
38
```

10. Now let's search for any differentially expressed proto-genes.

    We will define a proto-gene to be upregulated, or more expressed in the treatment compared to the control, when the logFC is greater than or equal to 1. And we will define a proto-gene to be downregulated, or less expressed in the treatment compared to the control when the logFC is less than or equal to -1.

```
filter(DE_results_proto_genes, logFC >= 1 | logFC <= -1)
```



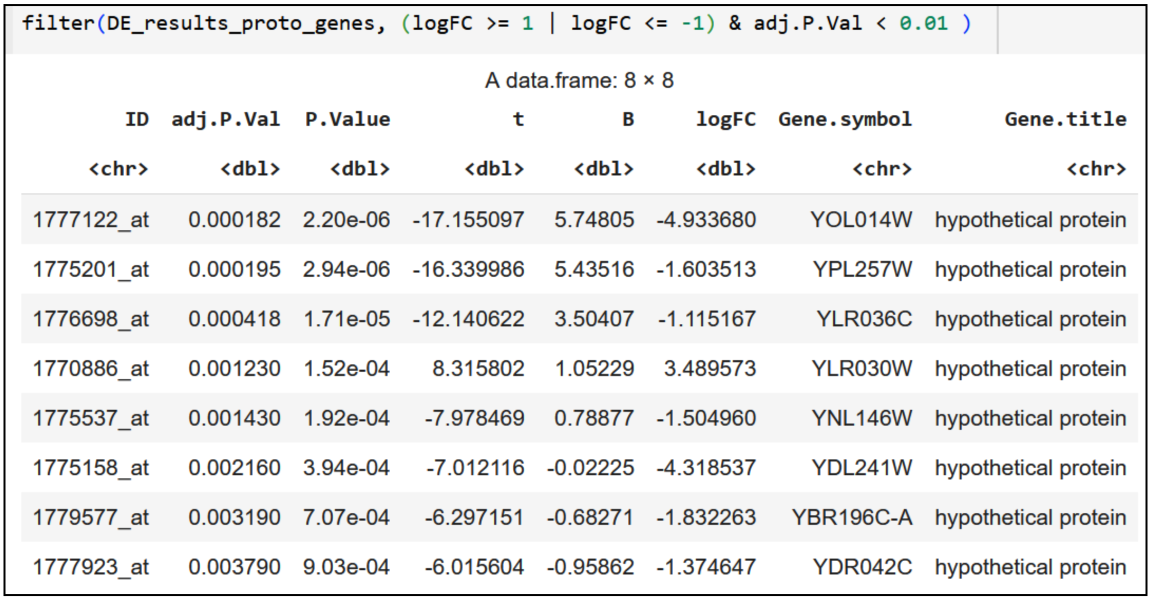
```
filter(DE_results_proto_genes, logFC >= 1 | logFC <= -1)
```

A data.frame: 10 × 8

ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
1777122_at	0.000182	2.20e-06	-17.155097	5.74805	-4.933680	YOL014W	hypothetical protein
1775201_at	0.000195	2.94e-06	-16.339986	5.43516	-1.603513	YPL257W	hypothetical protein
1776698_at	0.000418	1.71e-05	-12.140622	3.50407	-1.115167	YLR036C	hypothetical protein
1770886_at	0.001230	1.52e-04	8.315802	1.05229	3.489573	YLR030W	hypothetical protein
1775537_at	0.001430	1.92e-04	-7.978469	0.78877	-1.504960	YNL146W	hypothetical protein
1775158_at	0.002160	3.94e-04	-7.012116	-0.02225	-4.318537	YDL241W	hypothetical protein
1779577_at	0.003190	7.07e-04	-6.297151	-0.68271	-1.832263	YBR196C-A	hypothetical protein
1777923_at	0.003790	9.03e-04	-6.015604	-0.95862	-1.374647	YDR042C	hypothetical protein
1777464_at	0.010300	3.62e-03	-4.585322	-2.51784	-1.747953	YGL230C	hypothetical protein
1773889_at	0.056600	3.12e-02	-2.787040	-4.88379	-1.141100	YOL131W	hypothetical protein

This will give us rows where the expression of that proto-gene in the treatment samples is either at least two times greater or at least two times less than the control.

11. We also want to check that these results are significant and not just due to chance, so we will also check the adjusted p-value (shown in column `adj.P.Val`) and will only consider rows where the adjusted p-value is less than 0.01.



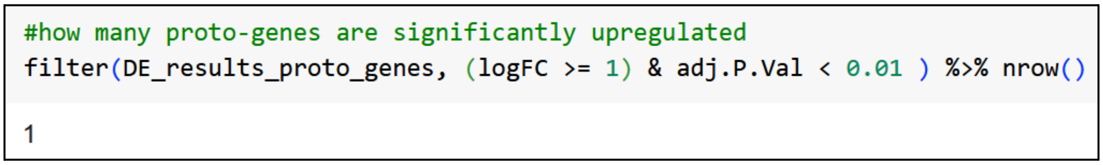
```
filter(DE_results_proto_genes, (logFC >= 1 | logFC <= -1) & adj.P.Val < 0.01 )
```

A data.frame: 8 × 8

ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
1777122_at	0.000182	2.20e-06	-17.155097	5.74805	-4.933680	YOL014W	hypothetical protein
1775201_at	0.000195	2.94e-06	-16.339986	5.43516	-1.603513	YPL257W	hypothetical protein
1776698_at	0.000418	1.71e-05	-12.140622	3.50407	-1.115167	YLR036C	hypothetical protein
1770886_at	0.001230	1.52e-04	8.315802	1.05229	3.489573	YLR030W	hypothetical protein
1775537_at	0.001430	1.92e-04	-7.978469	0.78877	-1.504960	YNL146W	hypothetical protein
1775158_at	0.002160	3.94e-04	-7.012116	-0.02225	-4.318537	YDL241W	hypothetical protein
1779577_at	0.003190	7.07e-04	-6.297151	-0.68271	-1.832263	YBR196C-A	hypothetical protein
1777923_at	0.003790	9.03e-04	-6.015604	-0.95862	-1.374647	YDR042C	hypothetical protein

If we are interested in how many proto-genes are significantly upregulated we can use the following line, which says filter to find proto-genes that are significantly upregulated and then count how many:

```
filter(DE_results_proto_genes, (logFC >= 1) & adj.P.Val < 0.01 ) %>%
nrow()
```

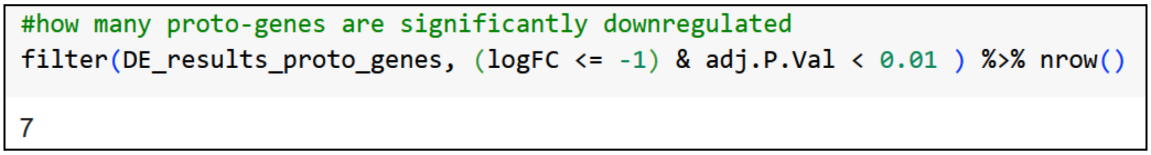


```
#how many proto-genes are significantly upregulated
filter(DE_results_proto_genes, (logFC >= 1) & adj.P.Val < 0.01 ) %>% nrow()
```

```
1
```

If we are interested in how many proto-genes are significantly downregulated we can use the following line, which says filter to find proto-genes that are significantly downregulated and then count how many:

```
filter(DE_results_proto_genes, (logFC <= -1) & adj.P.Val < 0.01 ) %>%
nrow()
```



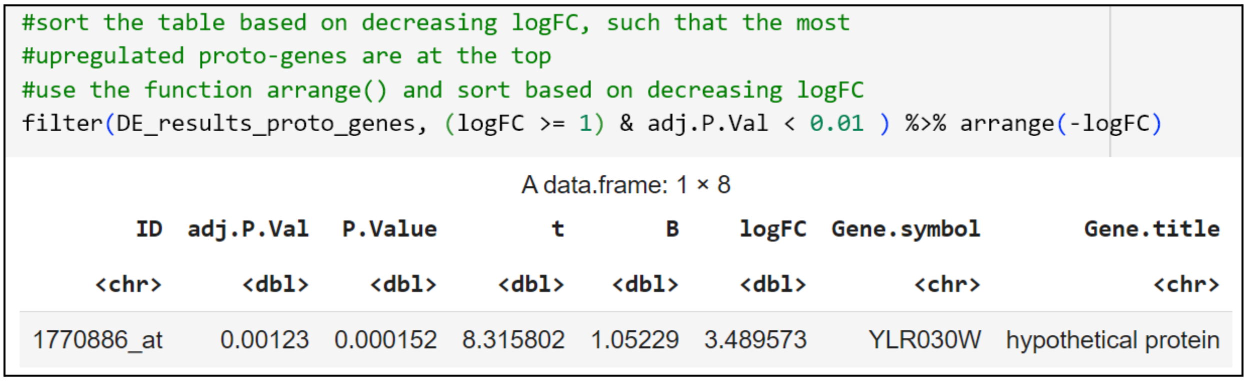
```
#how many proto-genes are significantly downregulated
filter(DE_results_proto_genes, (logFC <= -1) & adj.P.Val < 0.01 ) %>% nrow()

7
```

Answer question 9 in the worksheet: How many proto-genes are significantly upregulated in the treatment samples?

Answer question 10 in the worksheet: How many proto-genes are significantly downregulated in the treatment samples?

12. To sort the results based on the most upregulated proto-gene (ie largest `logFC`) we can use the following line:

```
filter(DE_results_proto_genes, (logFC >= 1) & adj.P.Val < 0.01 ) %>%
arrange(-logFC)
```



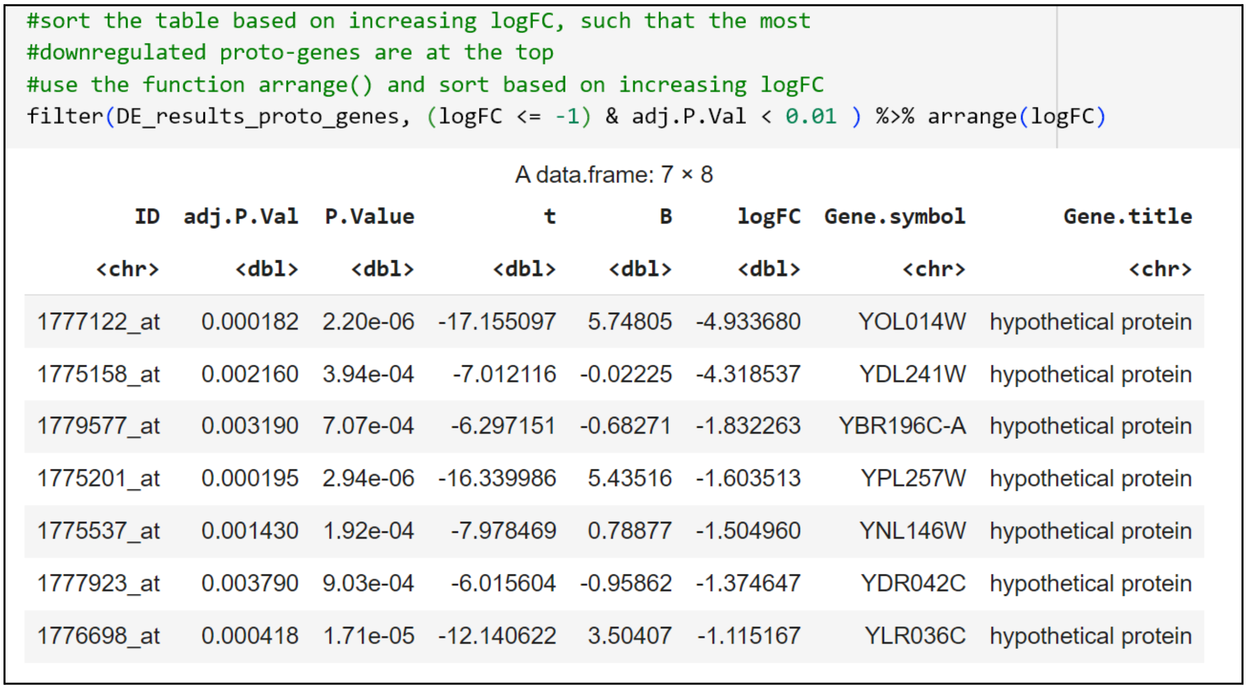
```
#sort the table based on decreasing logFC, such that the most
#upregulated proto-genes are at the top
#use the function arrange() and sort based on decreasing logFC
filter(DE_results_proto_genes, (logFC >= 1) & adj.P.Val < 0.01 ) %>% arrange(-logFC)
```

A data.frame: 1 × 8

ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
1770886_at	0.00123	0.000152	8.315802	1.05229	3.489573	YLR030W	hypothetical protein

or based on the most downregulated proto-gene (ie smallest `logFC`):

```
filter(DE_results_proto_genes, (logFC <= -1) & adj.P.Val < 0.01 ) %>%
arrange(logFC)
```



```
#sort the table based on increasing logFC, such that the most
#downregulated proto-genes are at the top
#use the function arrange() and sort based on increasing logFC
filter(DE_results_proto_genes, (logFC <= -1) & adj.P.Val < 0.01 ) %>% arrange(logFC)
```

A data.frame: 7 × 8

ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<chr>	<chr>
1777122_at	0.000182	2.20e-06	-17.155097	5.74805	-4.933680	YOL014W	hypothetical protein
1775158_at	0.002160	3.94e-04	-7.012116	-0.02225	-4.318537	YDL241W	hypothetical protein
1779577_at	0.003190	7.07e-04	-6.297151	-0.68271	-1.832263	YBR196C-A	hypothetical protein
1775201_at	0.000195	2.94e-06	-16.339986	5.43516	-1.603513	YPL257W	hypothetical protein
1775537_at	0.001430	1.92e-04	-7.978469	0.78877	-1.504960	YNL146W	hypothetical protein
1777923_at	0.003790	9.03e-04	-6.015604	-0.95862	-1.374647	YDR042C	hypothetical protein
1776698_at	0.000418	1.71e-05	-12.140622	3.50407	-1.115167	YLR036C	hypothetical protein

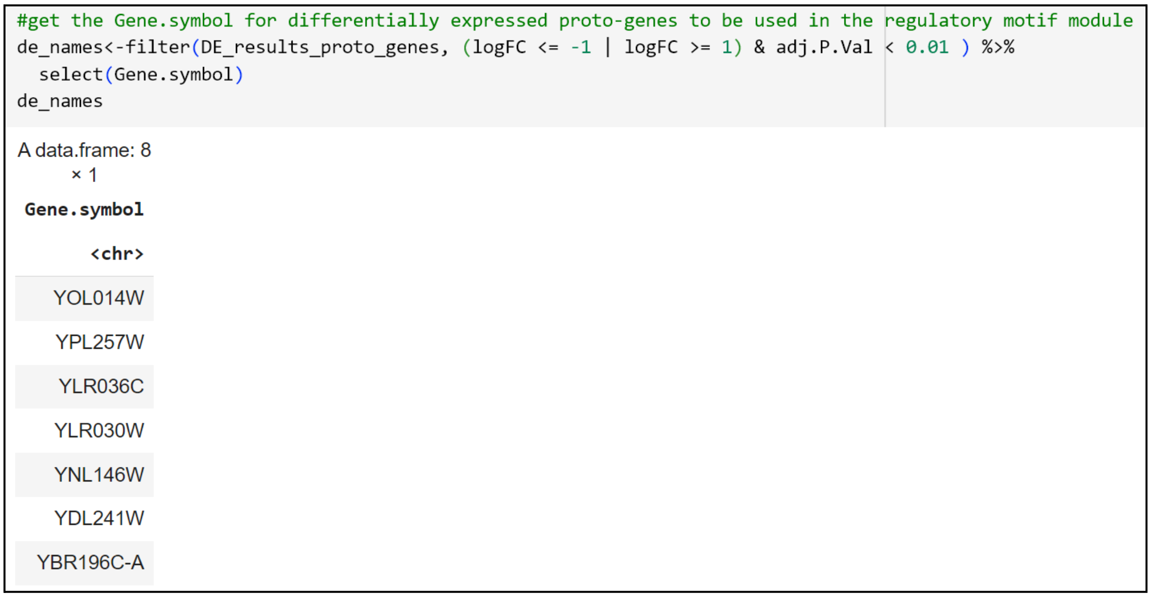
Answer question 11 in the worksheet: What is the most upregulated proto-gene?

Answer question 12 in the worksheet: What is the most downregulated proto-gene?

13. For the next module we will be looking at the regulatory sequences upstream of these differentially expressed proto-genes. To do this we need to gather a list of proto-genes that are differentially expressed and a list of proto-genes that are not differentially expressed.

    To get the names of differentially expressed proto-genes, we again will search for proto-genes that have a logFC >=1 or logFC <= -1 and have an adjusted p-value < 0.01. After filtering the table to contain only those proto-genes we will then select only the `Gene.symbol` to get our DE proto-genes list. We will use the `filter()` function to get only rows that correspond to DE proto-genes and then will use the `select()` function to return only the `Gene.symbol` column. Save this list into the variable `de_names`

```
de_names<-filter(DE_results_proto_genes, (logFC <= -1 | logFC >= 1) &
adj.P.Val < 0.01 ) %>% select(Gene.symbol)
de_names
```



```
#get the Gene.symbol for differentially expressed proto-genes to be used in the regulatory motif module
de_names<-filter(DE_results_proto_genes, (logFC <= -1 | logFC >= 1) & adj.P.Val < 0.01 ) %>%
  select(Gene.symbol)
de_names
```

```
A data.frame: 8
      × 1
 Gene.symbol
       <chr>
    YOL014W
    YPL257W
    YLR036C
    YLR030W
    YNL146W
    YDL241W
  YBR196C-A
```

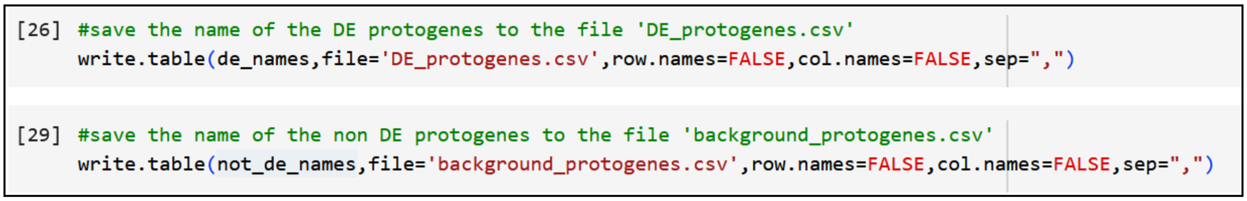
14. To get the names of non-differentially expressed proto-genes, we will search for proto-genes with a `logFC` > -1 and `logFC` < 1. After filtering the table to contain only those proto-genes, we will select only the `Gene.symbol` to get our list of non-differentially expressed proto-genes. Save this list into the variable `not_de_names`

```
not_de_names<-filter(DE_results_proto_genes, logFC > -1 & logFC < 1 )
%>% select(Gene.symbol)
not_de_names
```

15. Save these lists because they will be used in the next module (the Regulatory Motif module). You can save the lists either by keeping this Google Colab notebook open or by saving the list to a file using the following commands

```
write.table(de_names,file='DE_protogenes.csv',row.names=FALSE,col.name
s=FALSE,sep=",")
```

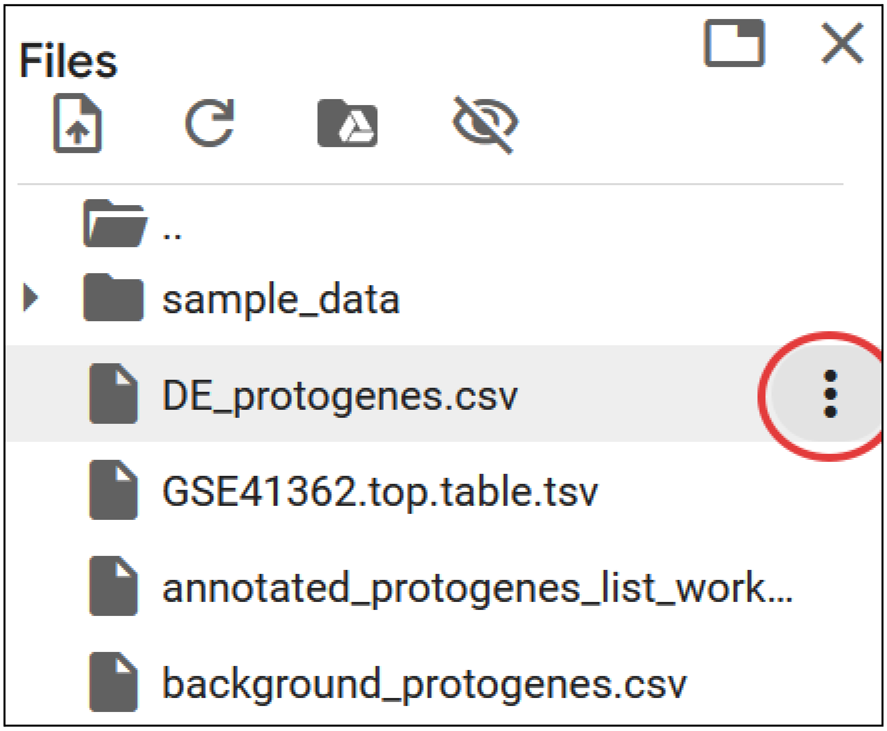
```
write.table(not_de_names,file='background_protogenes.csv',row.names=FA
LSE,col.names=FALSE,sep=",")
```



```
[26] #save the name of the DE protogenes to the file 'DE_protogenes.csv'
     write.table(de_names,file='DE_protogenes.csv',row.names=FALSE,col.names=FALSE,sep=",")

[29] #save the name of the non DE protogenes to the file 'background_protogenes.csv'
     write.table(not_de_names,file='background_protogenes.csv',row.names=FALSE,col.names=FALSE,sep=",")
```

This will write the two files into the Google Colab notebook environment. You will need to download these files to your computer or Google drive otherwise they will be removed when the notebook times out.

To download these files click on the ***three dots*** beside the file name and click ***Download*.**
Download the `DE_protogenes.csv` and `background_protogenes.csv` files



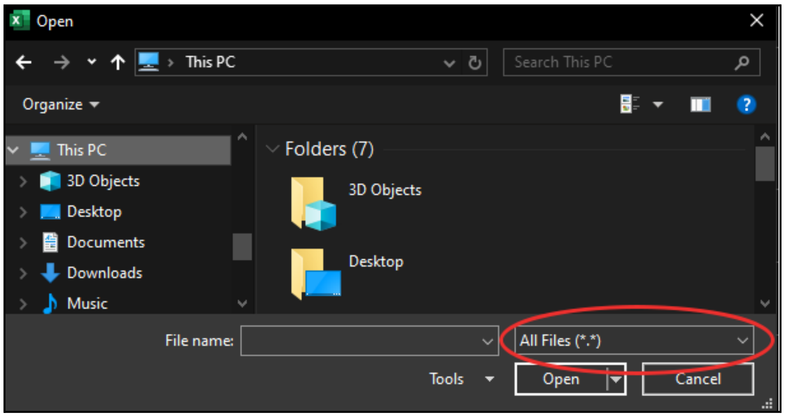
Sources

- Barrett et al. (2012). NCBI GEO: archive for functional genomics data sets—update, *Nucleic Acids Research*. https://doi.org/10.1093/nar/gks1193
- Liu et al. (2013). Reducing sphingolipid synthesis orchestrates global changes to extend yeast lifespan, *Aging Cell*. https://doi.org/10.1111/acel.12107

Appendix

Use Excel to query results for DE proto-genes

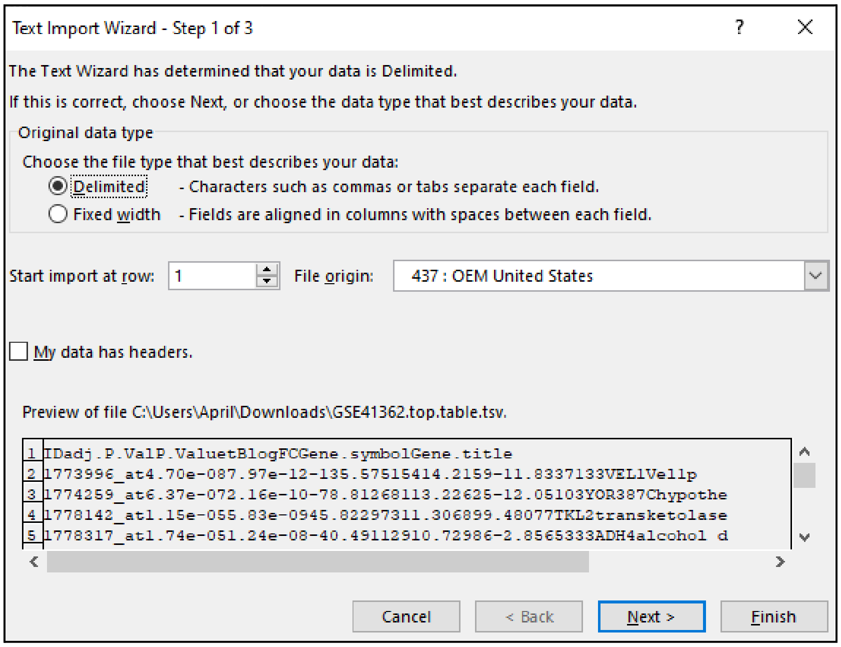
1. Download a list of proto-genes from the following link: https://docs.google.com/spreadsheets/d/1gvtiX-MWGHYL_Xv7SKyQ-IdbM63uG3qKwz8gyHkq9Kw/edit?usp=sharing
2. Open the list of proto-genes file in excel.
3. Open the DE results file in excel. If you are not seeing the file be sure to select the option ***All files (*.*)***



File name:

All Files (*.*)

4. The ‘Text Import Wizard’ window will pop up. Make sure ‘***Delimited***’ is selected and click ***Finish***.



Text Import Wizard - Step 1 of 3

The Text Wizard has determined that your data is Delimited.

If this is correct, choose Next, or choose the data type that best describes your data.

Original data type

Choose the file type that best describes your data:

● Delimited - Characters such as commas or tabs separate each field.

○ Fixed width - Fields are aligned in columns with spaces between each field.

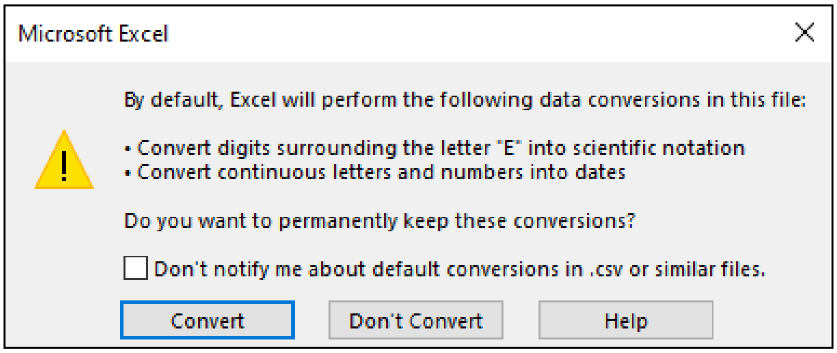
Start import at row: 1 File origin: 437 : OEM United States

□ My data has headers.

Preview of file C:\Users\April\Downloads\GSE41362.top.table.tsv.

```
IDadj.P.ValP.ValuetBlogFCGene.symbolGene.title
1773996_at4.70e-087.97e-12-135.57515414.2159-11.8337133VEL1Vellp
1774259_at6.37e-072.16e-10-78.81268113.22625-12.05103YOR387Chypothe
1778142_at1.15e-055.83e-0945.82297311.306899.48077TKL2transketolase
1778317_at1.74e-051.24e-08-40.49112910.72986-2.8565333ADH4alcohol d
```

5. Next a window about data conversions will pop up. This is asking if you want to convert the numbers that are in scientific notation into numbers. Click the ***Convert*** button



Microsoft Excel

By default, Excel will perform the following data conversions in this file:

- Convert digits surrounding the letter "E" into scientific notation
- Convert continuous letters and numbers into dates

Do you want to permanently keep these conversions?

□ Don't notify me about default conversions in .csv or similar files.

6. We want to filter the DE results to retrieve only the rows that correspond to proto-genes, to do this we will use the excel functions XMATCH and FILTER.

   Begin converting the data into a table by going to ***Home*** > ***Format as Table*** and selecting any of the styles. Check to make sure all the data is selected and click ***ok***.

7. Next name the table so it can be referenced by name. Go to the ***Table Design*** tab in the top menu and in the far left corner type in DE_results in the ***Table Name*** box.

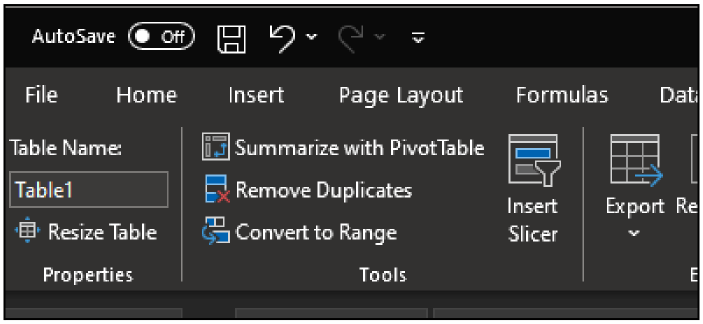


Table Name: Table1

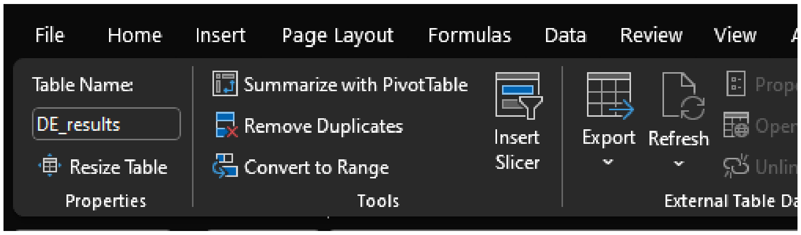
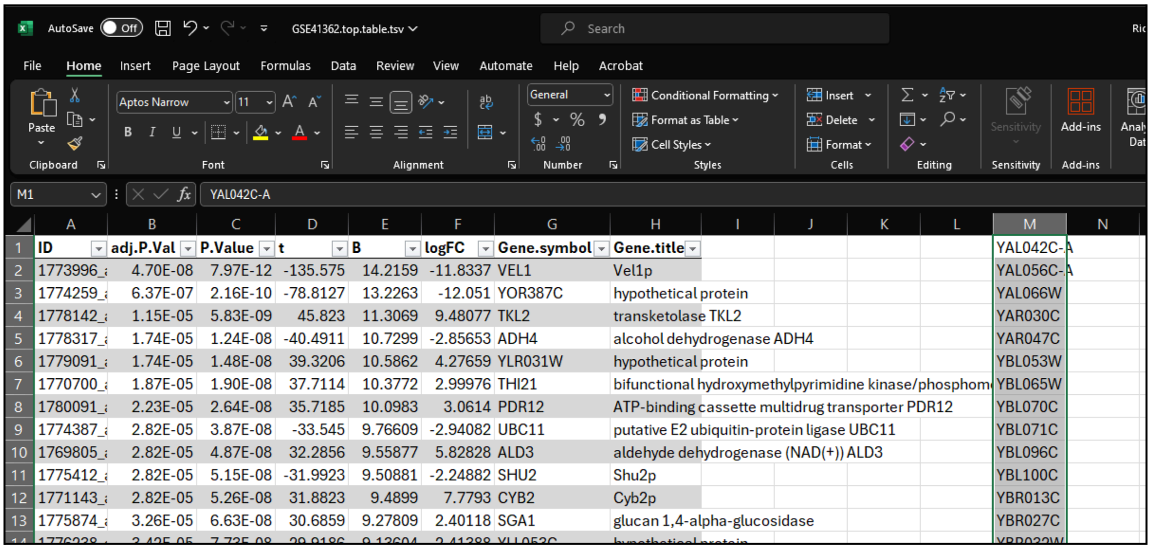


Table Name: DE_results

8. In the proto-genes file in excel and copy the values (but not the column name) into another column in the DE results file, in this example we will copy them into column M, starting at M1.



M1 | YAL042C-A

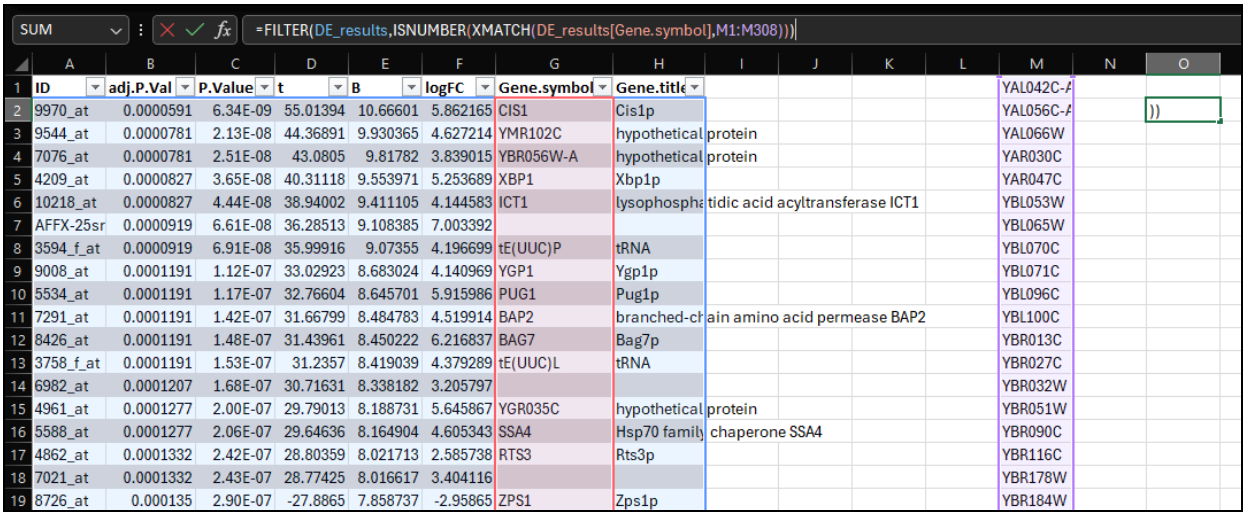
	A	B	C	D	E	F	G	H	M
1	ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title	YAL042C-A
2	1773996_	4.70E-08	7.97E-12	-135.575	14.2159	-11.8337	VEL1	Vel1p	YAL056C-A
3	1774259_	6.37E-07	2.16E-10	-78.8127	13.2263	-12.051	YOR387C	hypothetical protein	YAL066W
4	1778142_	1.15E-05	5.83E-09	45.823	11.3069	9.48077	TKL2	transketolase TKL2	YAR030C
5	1778317_	1.74E-05	1.24E-08	-40.4911	10.7299	-2.85653	ADH4	alcohol dehydrogenase ADH4	YAR047C
6	1779091_	1.74E-05	1.48E-08	39.3206	10.5862	4.27659	YLR031W	hypothetical protein	YBL053W
7	1770700_	1.87E-05	1.90E-08	37.7114	10.3772	2.99976	THI21	bifunctional hydroxymethylpyrimidine kinase/phosphom	YBL065W
8	1780091_	2.23E-05	2.64E-08	35.7185	10.0983	3.0614	PDR12	ATP-binding cassette multidrug transporter PDR12	YBL070C
9	1774387_	2.82E-05	3.87E-08	-33.545	9.76609	-2.94082	UBC11	putative E2 ubiquitin-protein ligase UBC11	YBL071C
10	1769805_	2.82E-05	4.87E-08	32.2856	9.55877	5.82828	ALD3	aldehyde dehydrogenase (NAD(+)) ALD3	YBL096C
11	1775412_	2.82E-05	5.15E-08	-31.9923	9.50881	-2.24882	SHU2	Shu2p	YBL100C
12	1771143_	2.82E-05	5.26E-08	31.8823	9.4899	7.7793	CYB2	Cyb2p	YBR013C
13	1775874_	3.26E-05	6.63E-08	30.6859	9.27809	2.40118	SGA1	glucan 1,4-alpha-glucosidase	YBR027C
14	1776238_	3.42E-05	7.73E-08	29.9186	9.13604	2.41388	YLL053C	hypothetical protein	YBR032W

9. Now we will search for the rows in the `DE_results` table that contain one of the proto-gene names in column M by searching in the `Gene.Symbol` column.

   Click on an empty cell where you would like the filtered table to go, in this example we will use the cell O2, and type in the following formula:

```
=FILTER(DE_results,ISNUMBER(XMATCH(DE_results[Gene.symbol],M1:M308)))
```

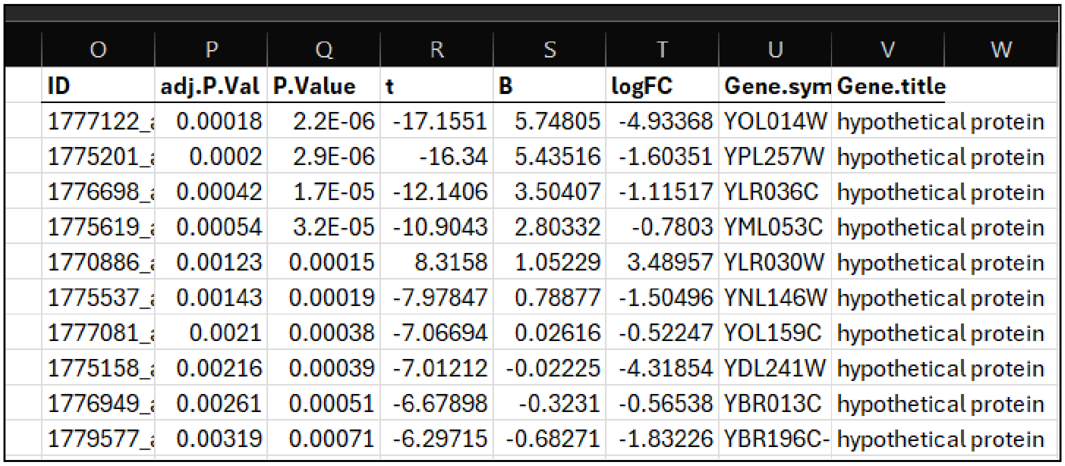
   Where `Gene.symbol` is the name of the column that contains the gene names in the `DE_results` table and the M1:M308 is the column and rows where the proto-gene list is.



SUM | =FILTER(DE_results,ISNUMBER(XMATCH(DE_results[Gene.symbol],M1:M308)))

	A	B	C	D	E	F	G	H	M	O
1	ID	adj.P.Val	P.Value	t	B	logFC	Gene.symbol	Gene.title	YAL042C-A	
2	9970_at	0.0000591	6.34E-09	55.01394	10.66601	5.862165	CIS1	Cis1p	YAL056C-A	))
3	9544_at	0.0000781	2.13E-08	44.36891	9.930365	4.627214	YMR102C	hypothetical protein	YAL066W	
4	7076_at	0.0000781	2.51E-08	43.0805	9.81782	3.839015	YBR056W-A	hypothetical protein	YAR030C	
5	4209_at	0.0000827	3.65E-08	40.31118	9.553971	5.253689	XBP1	Xbp1p	YAR047C	
6	10218_at	0.0000827	4.44E-08	38.94002	9.411105	4.144583	ICT1	lysophosphatidic acid acyltransferase ICT1	YBL053W	
7	AFFX-25sr	0.0000919	6.61E-08	36.28513	9.108385	7.003392			YBL065W	
8	3594_f_at	0.0000919	6.91E-08	35.99916	9.07355	4.196699	tE(UUC)P	tRNA	YBL070C	
9	9008_at	0.0001191	1.12E-07	33.02923	8.683024	4.140969	YGP1	Ygp1p	YBL071C	
10	5534_at	0.0001191	1.17E-07	32.76604	8.645701	5.915986	PUG1	Pug1p	YBL096C	
11	7291_at	0.0001191	1.42E-07	31.66799	8.484783	4.519914	BAP2	branched-chain amino acid permease BAP2	YBL100C	
12	8426_at	0.0001191	1.48E-07	31.43961	8.450222	6.216837	BAG7	Bag7p	YBR013C	
13	3758_f_at	0.0001191	1.53E-07	31.2357	8.419039	4.379289	tE(UUC)L	tRNA	YBR027C	
14	6982_at	0.0001207	1.68E-07	30.71631	8.338182	3.205797			YBR032W	
15	4961_at	0.0001277	2.00E-07	29.79013	8.188731	5.645867	YGR035C	hypothetical protein	YBR051W	
16	5588_at	0.0001277	2.06E-07	29.64636	8.164904	4.605343	SSA4	Hsp70 family chaperone SSA4	YBR090C	
17	4862_at	0.0001332	2.42E-07	28.80359	8.021713	2.585738	RTS3	Rts3p	YBR116C	
18	7021_at	0.0001332	2.43E-07	28.77425	8.016617	3.404116			YBR178W	
19	8726_at	0.000135	2.90E-07	-27.8865	7.858737	-2.95865	ZPS1	Zps1p	YBR184W	

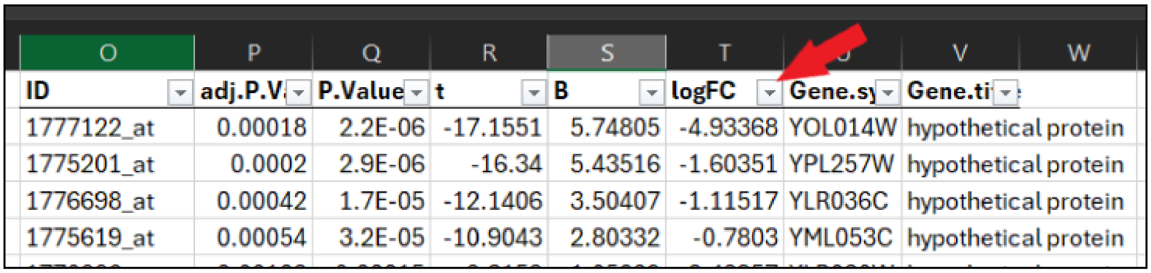
10. Hit enter and you will have the filtered table that contains the expression information about proto-genes in treatment versus control samples. Copy the column headers from the `DE_results` table and paste them above the new filtered table



O	P	Q	R	S	T	U	V	W
ID	adj.P.Val	P.Value	t	B	logFC	Gene.sym	Gene.title	
1777122_a	0.00018	2.2E-06	-17.1551	5.74805	-4.93368	YOL014W	hypothetical protein	
1775201_a	0.0002	2.9E-06	-16.34	5.43516	-1.60351	YPL257W	hypothetical protein	
1776698_a	0.00042	1.7E-05	-12.1406	3.50407	-1.11517	YLR036C	hypothetical protein	
1775619_a	0.00054	3.2E-05	-10.9043	2.80332	-0.7803	YML053C	hypothetical protein	
1770886_a	0.00123	0.00015	8.3158	1.05229	3.48957	YLR030W	hypothetical protein	
1775537_a	0.00143	0.00019	-7.97847	0.78877	-1.50496	YNL146W	hypothetical protein	
1777081_a	0.0021	0.00038	-7.06694	0.02616	-0.52247	YOL159C	hypothetical protein	
1775158_a	0.00216	0.00039	-7.01212	-0.02225	-4.31854	YDL241W	hypothetical protein	
1776949_a	0.00261	0.00051	-6.67898	-0.3231	-0.56538	YBR013C	hypothetical protein	
1779577_a	0.00319	0.00071	-6.29715	-0.68271	-1.83226	YBR196C-	hypothetical protein	

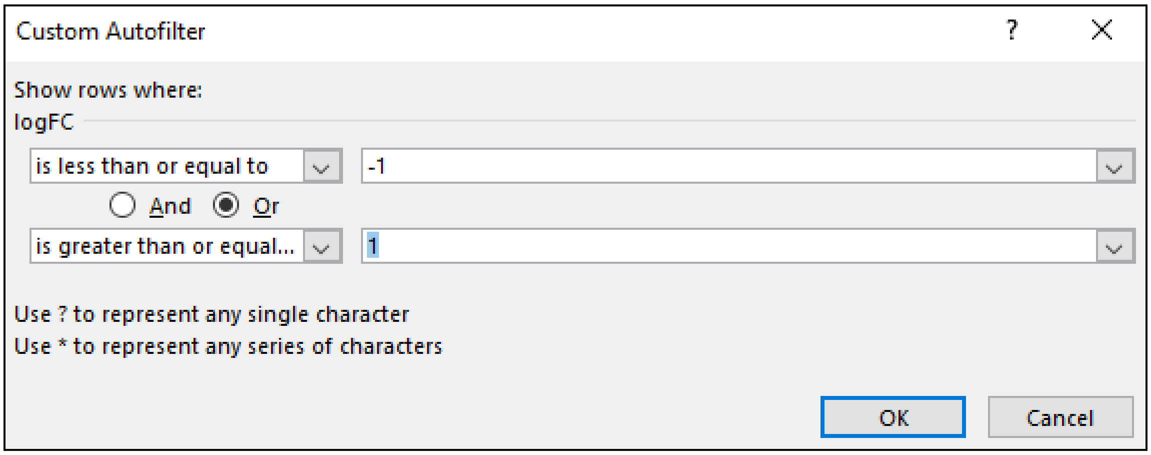
11. Now let's search for any differentially expressed proto-genes.

    We will define a proto-gene to be upregulated, or more expressed in the treatment compared to the control, when the logFC is greater than or equal to 1. And we will define a proto-gene to be downregulated, or less expressed in the treatment compared to the control, when the logFC is less than or equal to -1.



O	P	Q	R	S	T	U	V	W
ID	adj.P.V	P.Value	t	B	logFC	Gene.sy	Gene.ti	
1777122_at	0.00018	2.2E-06	-17.1551	5.74805	-4.93368	YOL014W	hypothetical protein	
1775201_at	0.0002	2.9E-06	-16.34	5.43516	-1.60351	YPL257W	hypothetical protein	
1776698_at	0.00042	1.7E-05	-12.1406	3.50407	-1.11517	YLR036C	hypothetical protein	
1775619_at	0.00054	3.2E-05	-10.9043	2.80332	-0.7803	YML053C	hypothetical protein	

To do this in excel, click on the `LogFC` column, then go to ***Home*** > ***Sort & Filter*** > ***Filter***. Then click on the ***drop down arrow*** in the `LogFC` column. Go to ***Number Filters*** > ***Custom Filters***. And select '***less than or equal to***' and type in -1 and select the "***Or***" button and select "***greater than or equal to***" and type in 1. Click ***OK***



Custom Autofilter

Show rows where:

logFC

is less than or equal to: -1

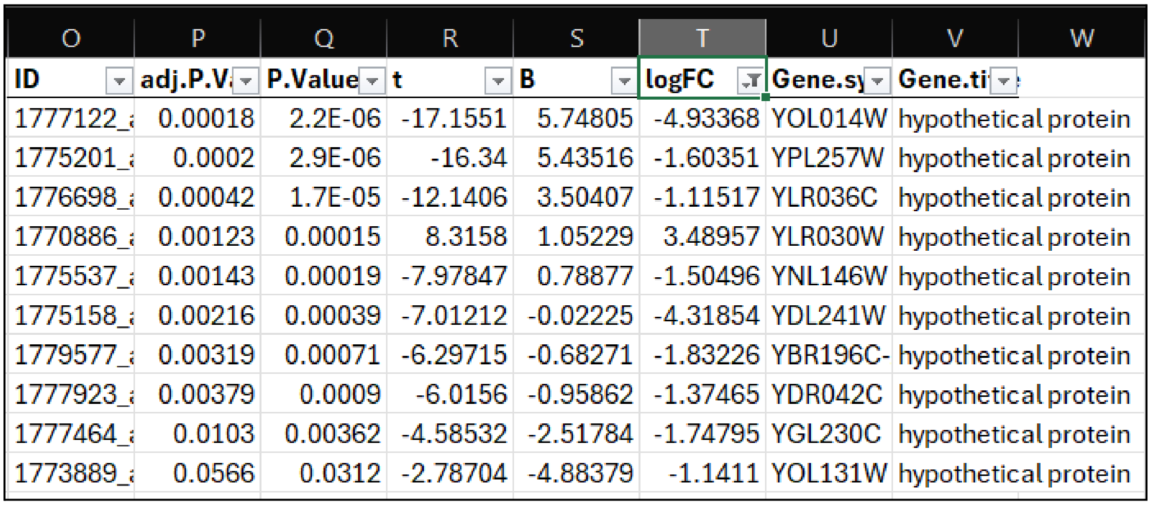
○ And ● Or

is greater than or equal...: 1

Use ? to represent any single character

Use * to represent any series of characters

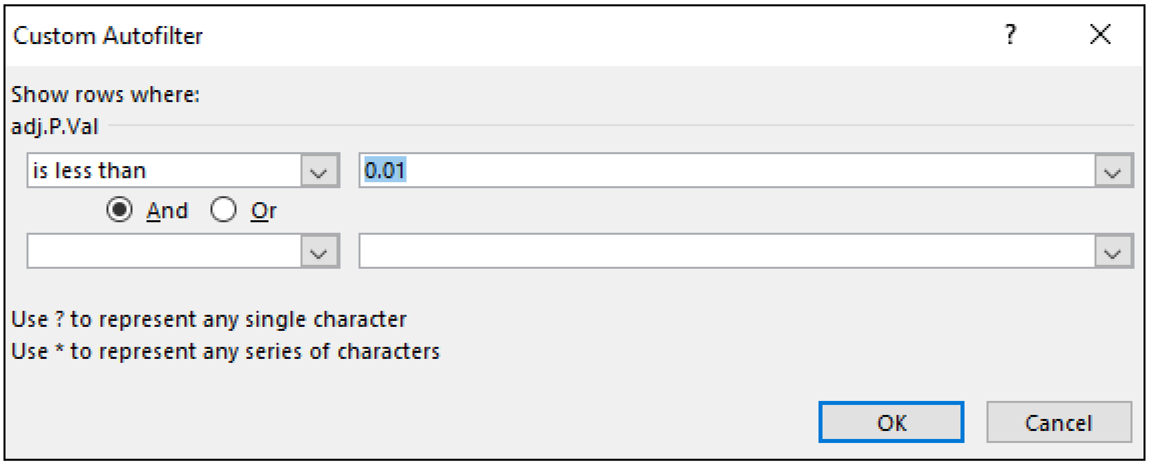
This will give us rows where the expression of that proto-gene is either at least 2 times greater or at least 2 times less expression than the control.



O	P	Q	R	S	T	U	V	W
ID	adj.P.V	P.Value	t	B	logFC	Gene.sy	Gene.ti	
1777122_	0.00018	2.2E-06	-17.1551	5.74805	-4.93368	YOL014W	hypothetical protein	
1775201_	0.0002	2.9E-06	-16.34	5.43516	-1.60351	YPL257W	hypothetical protein	
1776698_	0.00042	1.7E-05	-12.1406	3.50407	-1.11517	YLR036C	hypothetical protein	
1770886_	0.00123	0.00015	8.3158	1.05229	3.48957	YLR030W	hypothetical protein	
1775537_	0.00143	0.00019	-7.97847	0.78877	-1.50496	YNL146W	hypothetical protein	
1775158_	0.00216	0.00039	-7.01212	-0.02225	-4.31854	YDL241W	hypothetical protein	
1779577_	0.00319	0.00071	-6.29715	-0.68271	-1.83226	YBR196C-	hypothetical protein	
1777923_	0.00379	0.0009	-6.0156	-0.95862	-1.37465	YDR042C	hypothetical protein	
1777464_	0.0103	0.00362	-4.58532	-2.51784	-1.74795	YGL230C	hypothetical protein	
1773889_	0.0566	0.0312	-2.78704	-4.88379	-1.1411	YOL131W	hypothetical protein	

12. We also want to check that these results are significant and not just due to chance so we will also check the adjusted p-value, in this case the column `adj.P.Val` and will only consider rows where the adjusted p-value is less than 0.01

    Click on the ***down arrow*** beside the `adj.P.Val` column , click ***Number Filters*** > ***Less than…***
    In the box type 0.01 and click ***OK***



Custom Autofilter

Show rows where:

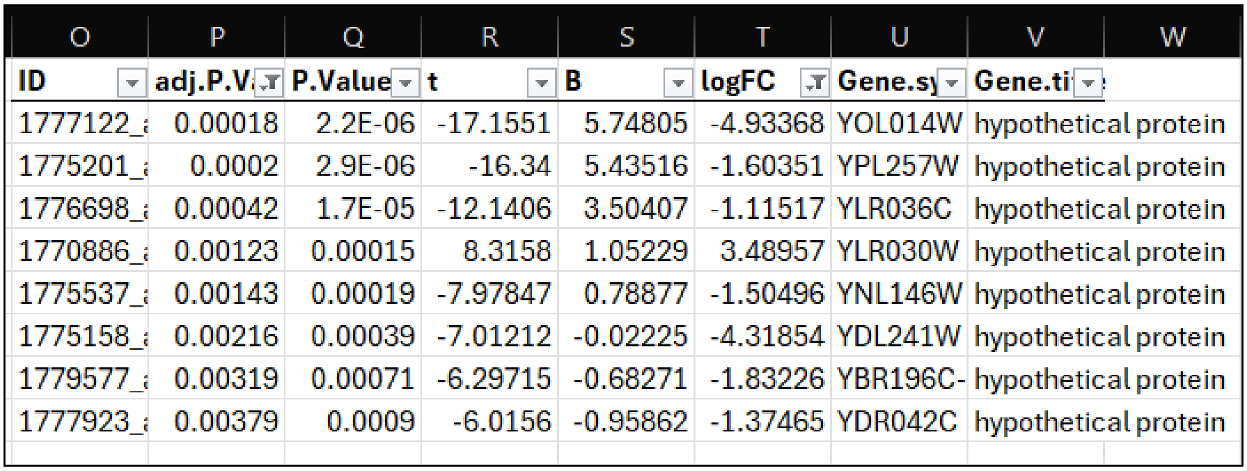
adj.P.Val

is less than 0.01

● And ○ Or

Use ? to represent any single character

Use * to represent any series of characters

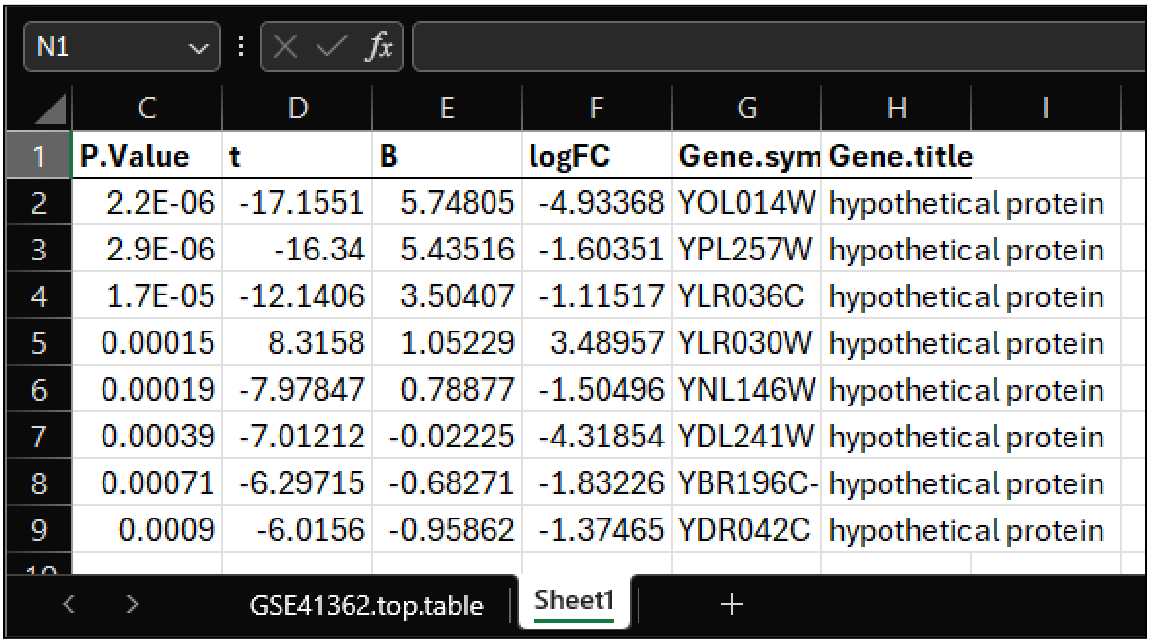


O	P	Q	R	S	T	U	V	W
ID	adj.P.V	P.Value	t	B	logFC	Gene.sy	Gene.ti	
1777122_a	0.00018	2.2E-06	-17.1551	5.74805	-4.93368	YOL014W	hypothetical protein	
1775201_a	0.0002	2.9E-06	-16.34	5.43516	-1.60351	YPL257W	hypothetical protein	
1776698_a	0.00042	1.7E-05	-12.1406	3.50407	-1.11517	YLR036C	hypothetical protein	
1770886_a	0.00123	0.00015	8.3158	1.05229	3.48957	YLR030W	hypothetical protein	
1775537_a	0.00143	0.00019	-7.97847	0.78877	-1.50496	YNL146W	hypothetical protein	
1775158_a	0.00216	0.00039	-7.01212	-0.02225	-4.31854	YDL241W	hypothetical protein	
1779577_a	0.00319	0.00071	-6.29715	-0.68271	-1.83226	YBR196C-	hypothetical protein	
1777923_a	0.00379	0.0009	-6.0156	-0.95862	-1.37465	YDR042C	hypothetical protein	

__Answer question 9 in the worksheet__: How many proto-genes are significantly upregulated in the treatment samples?

__Answer question 10 in the worksheet__: How many proto-genes are significantly downregulated in the treatment samples?

13. To be able to sort the table we need to copy it and put it on a new tab. Select the proto-gene filtered table, copy it and create a new sheet at the bottom, paste this table into the new sheet



	C	D	E	F	G	H	I
1	P.Value	t	B	logFC	Gene.sym	Gene.title	
2	2.2E-06	-17.1551	5.74805	-4.93368	YOL014W	hypothetical protein	
3	2.9E-06	-16.34	5.43516	-1.60351	YPL257W	hypothetical protein	
4	1.7E-05	-12.1406	3.50407	-1.11517	YLR036C	hypothetical protein	
5	0.00015	8.3158	1.05229	3.48957	YLR030W	hypothetical protein	
6	0.00019	-7.97847	0.78877	-1.50496	YNL146W	hypothetical protein	
7	0.00039	-7.01212	-0.02225	-4.31854	YDL241W	hypothetical protein	
8	0.00071	-6.29715	-0.68271	-1.83226	YBR196C-	hypothetical protein	
9	0.0009	-6.0156	-0.95862	-1.37465	YDR042C	hypothetical protein	

14. To sort the results based on the most upregulated proto-gene (i.e. largest logFC) click on the `logFC` column. Click the ***Sort & Filter*** button in the Home menu then click ***Sort Z to A***

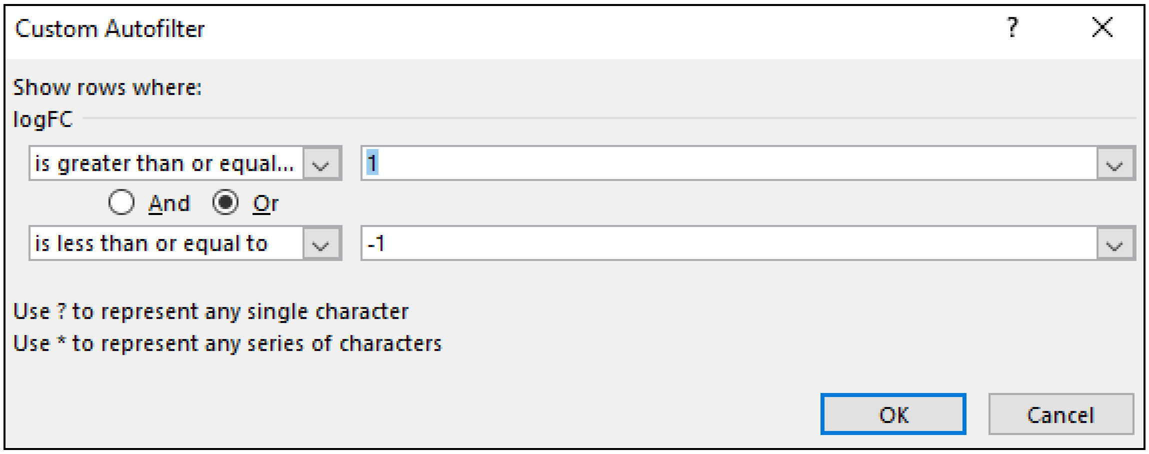
15. To sort the results based on the most downregulated proto-gene (i.e. smallest logFC) click on the `logFC` column. Click the ***Sort & Filter*** button in the Home menu then click ***Sort A to Z***

    Answer question 11 in the worksheet: What is the most upregulated proto-gene?

    Answer question 12 in the worksheet: What is the most downregulated proto-gene?

16. For the next module we will be looking at the regulatory sequences upstream of these differentially expressed proto-genes. To do this we need to gather a list of proto-genes that are differentially expressed and a list of proto-genes that are not differentially expressed.

    To get the names of differentially expressed proto-genes, go back to the ***GSE41362.top.table*** tab. Make sure our table shows proto-genes that have a `logFC` >=1 or `logFC` <= -1 and have an `adj.P.Val` < 0.01. After filtering the table to contain only those proto-genes we will then select only the `Gene.symbol` to get our DE proto-genes list.



Custom Autofilter

Show rows where:

logFC

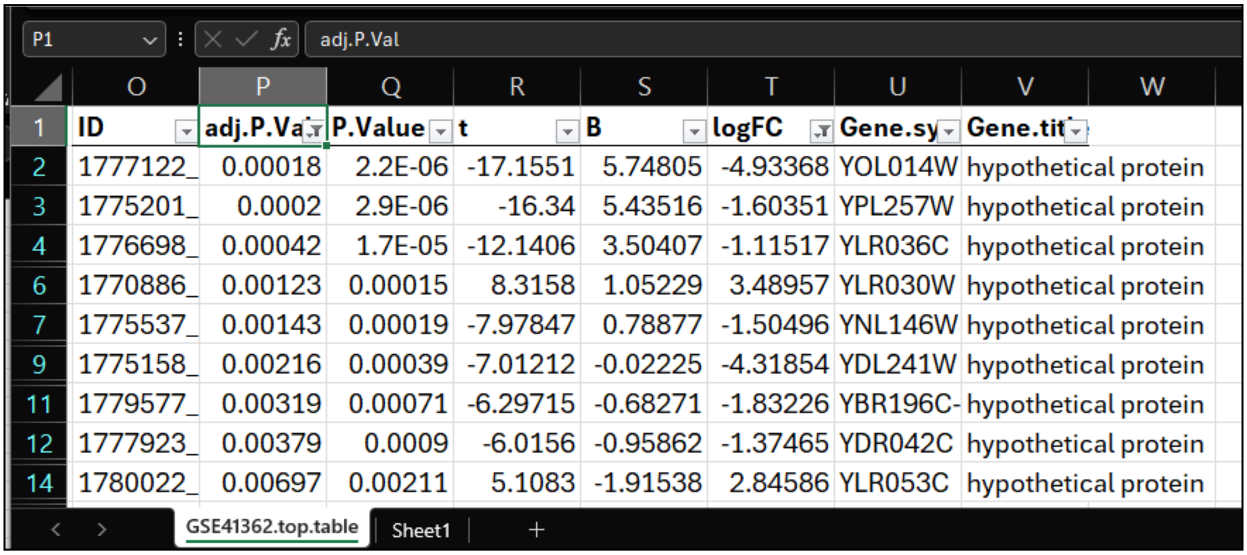
is greater than or equal... 1

○ And ● Or

is less than or equal to -1

Use ? to represent any single character

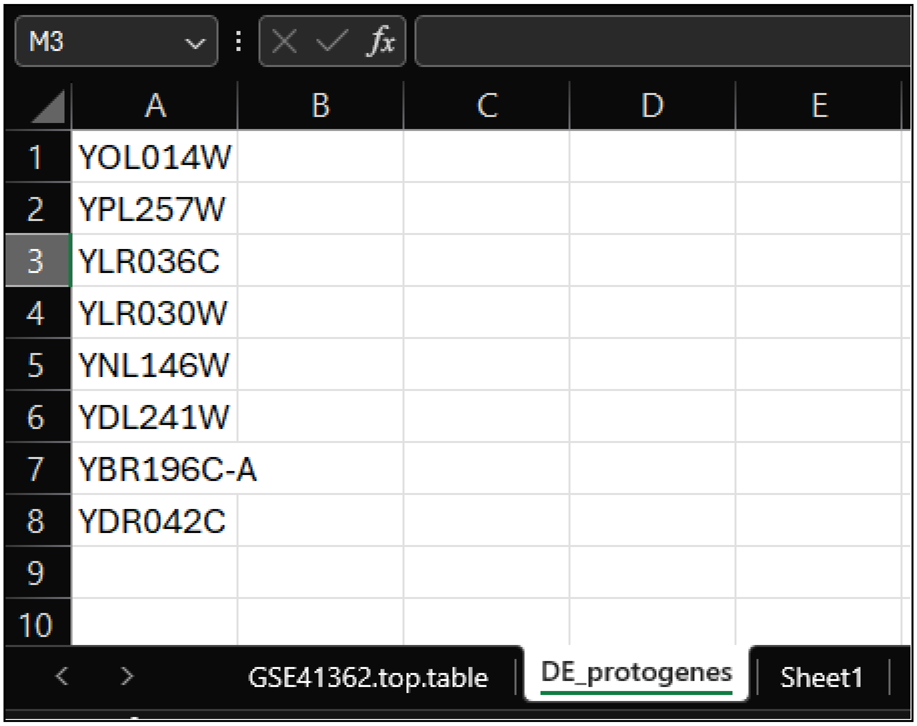
Use * to represent any series of characters



P1 adj.P.Val

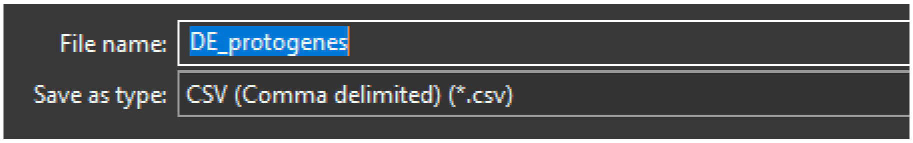
	O	P	Q	R	S	T	U	V	W
1	ID	adj.P.Va	P.Value	t	B	logFC	Gene.sy	Gene.tit	
2	1777122_	0.00018	2.2E-06	-17.1551	5.74805	-4.93368	YOL014W	hypothetical protein	
3	1775201_	0.0002	2.9E-06	-16.34	5.43516	-1.60351	YPL257W	hypothetical protein	
4	1776698_	0.00042	1.7E-05	-12.1406	3.50407	-1.11517	YLR036C	hypothetical protein	
6	1770886_	0.00123	0.00015	8.3158	1.05229	3.48957	YLR030W	hypothetical protein	
7	1775537_	0.00143	0.00019	-7.97847	0.78877	-1.50496	YNL146W	hypothetical protein	
9	1775158_	0.00216	0.00039	-7.01212	-0.02225	-4.31854	YDL241W	hypothetical protein	
11	1779577_	0.00319	0.00071	-6.29715	-0.68271	-1.83226	YBR196C-	hypothetical protein	
12	1777923_	0.00379	0.0009	-6.0156	-0.95862	-1.37465	YDR042C	hypothetical protein	
14	1780022_	0.00697	0.00211	5.1083	-1.91538	2.84586	YLR053C	hypothetical protein	

17. Copy the Gene.symbol values and paste them into a new sheet entitled ‘DE_protogenes’



	A	B	C	D	E
1	YOL014W				
2	YPL257W				
3	YLR036C				
4	YLR030W				
5	YNL146W				
6	YDL241W				
7	YBR196C-A				
8	YDR042C				
9					
10					

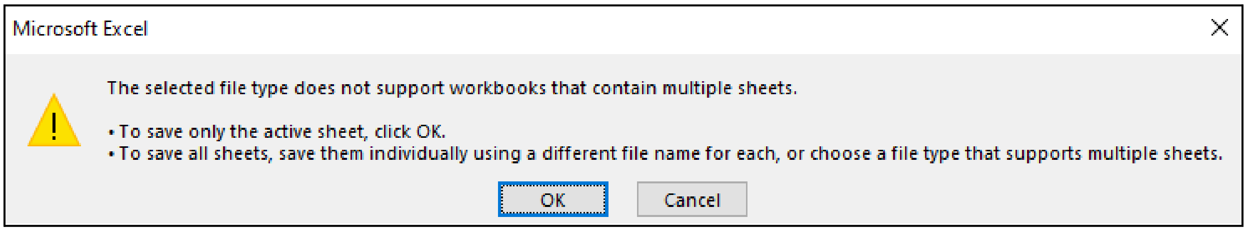
Save this sheet as a .csv by going to ***File*** > ***Save As*** . Type in DE_protogenes as the file name and select ***CSV*** file type in the drop down menu



File name: DE_protogenes

Save as type: CSV (Comma delimited) (*.csv)

If a popup warning comes up. Click ***OK***



Microsoft Excel

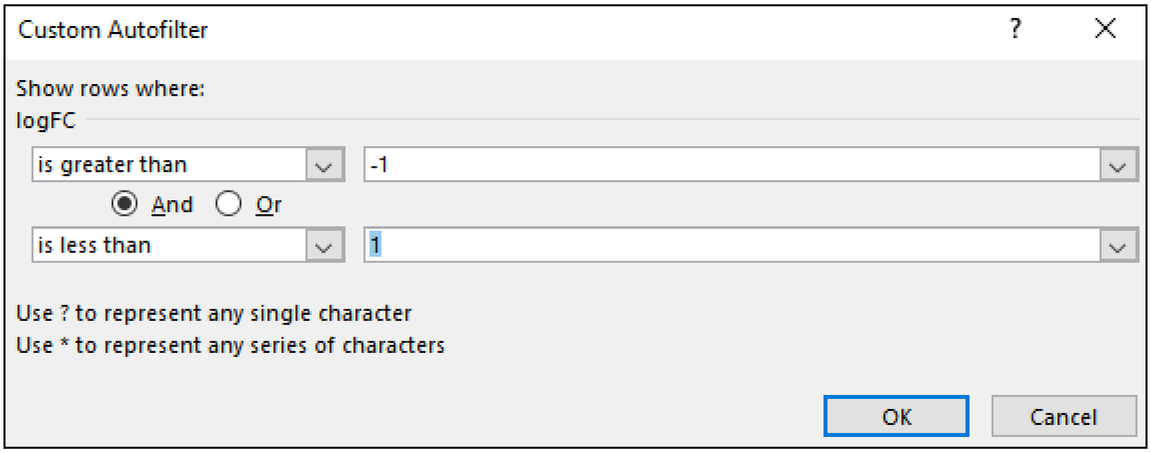
The selected file type does not support workbooks that contain multiple sheets.

- To save only the active sheet, click OK.
- To save all sheets, save them individually using a different file name for each, or choose a file type that supports multiple sheets.

We will use this file in the next module

18. Next we will get the names of non-differentially expressed proto-genes. Go back to the ***GSE41362.top.table*** sheet.
    Clear the `adj.P.Val` filter by clicking on the box beside the column name and select ***Clear Filter from "adj.P.Val".*** Repeat this step for the `logFC` column

Now search for proto-genes that have a logFC > -1 and logFC < 1 by making a new custom filter in the `logFC` column and selecting ‘***is greater than’*** and typing -1 in the box. Click the button beside ‘***And***’ and in the next row select ‘***is less than***’ and type 1 in the box. Click ‘***Ok***’



Custom Autofilter

Show rows where:

logFC

is greater than -1

● And ○ Or

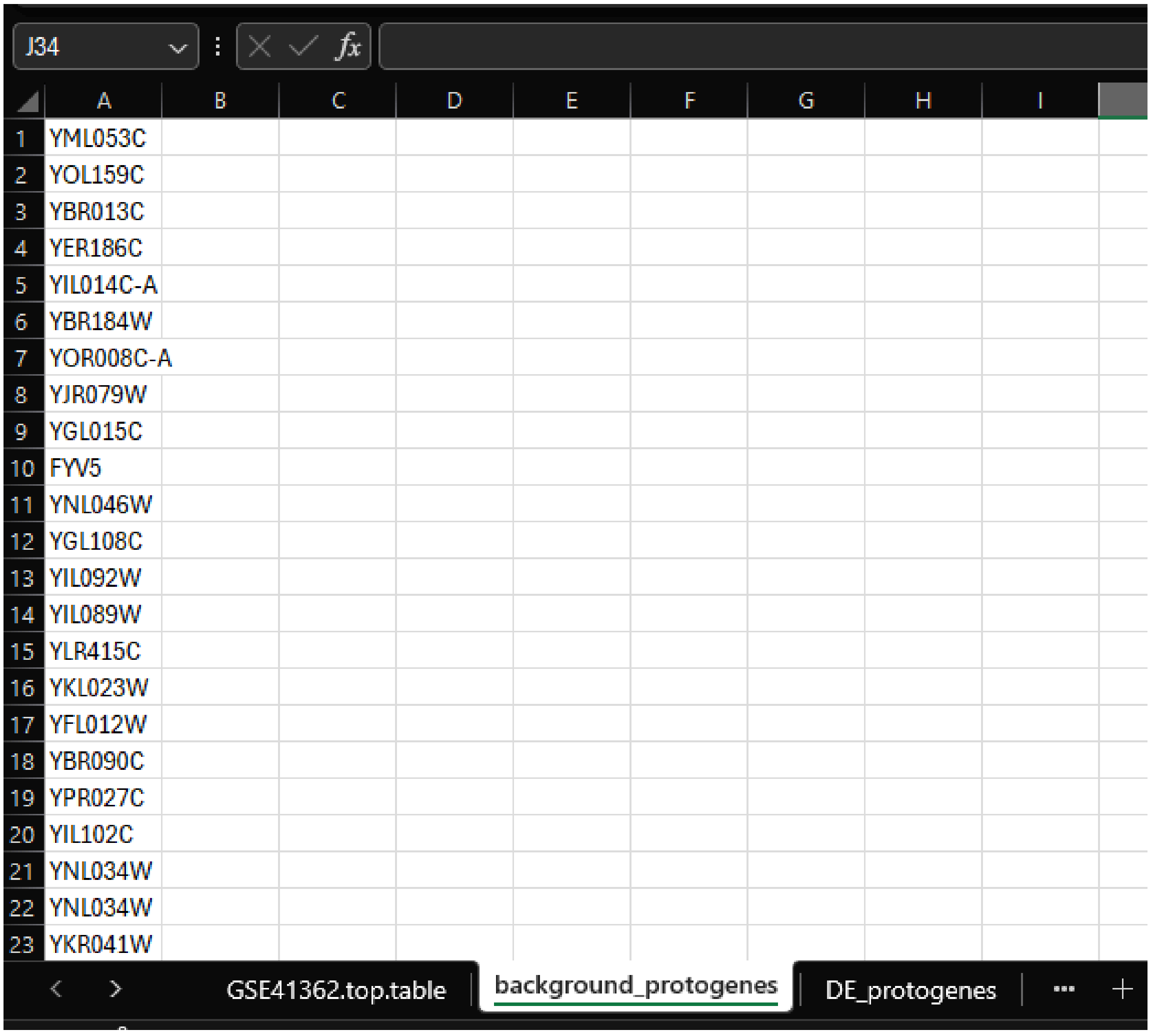
is less than 1

Use ? to represent any single character

Use * to represent any series of characters

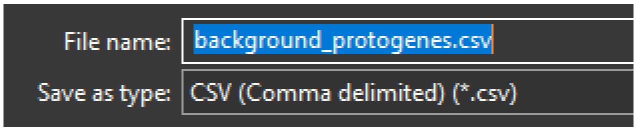
After filtering the table to contain only those proto-genes we will then select only the `Gene.symbol` to get our non-differentially expressed proto-genes list.

Copy this into a separate sheet and label it background proto-genes.



	A	B	C	D	E	F	G	H	I
1	YML053C								
2	YOL159C								
3	YBR013C								
4	YER186C								
5	YIL014C-A								
6	YBR184W								
7	YOR008C-A								
8	YJR079W								
9	YGL015C								
10	FYV5								
11	YNL046W								
12	YGL108C								
13	YIL092W								
14	YIL089W								
15	YLR415C								
16	YKL023W								
17	YFL012W								
18	YBR090C								
19	YPR027C								
20	YIL102C								
21	YNL034W								
22	YNL034W								
23	YKR041W								

Save this sheet as a .csv by going to ***File*** > ***Save As*** . Type in background_protogenes as the file name and select ***CSV*** file type in the drop down menu



File name: background_protogenes.csv

Save as type: CSV (Comma delimited) (*.csv)

We will use this file in the next module