

# Genomic Data Analysis Course Exercises

Carson Stacy & Jeffrey Lewis

2023-10-27



# Contents

<b>Preface</b>	<b>9</b>
0.1 Usage . . . . .	9
0.2 Key Features . . . . .	9
0.3 Disclaimer . . . . .	10
<b>1 Getting Started in R</b>	<b>11</b>
1.1 Exercise Description . . . . .	12
1.2 Learning outcomes . . . . .	12
1.3 Using R and RStudio . . . . .	12
1.4 Load data directly from the URL . . . . .	13
1.5 Working with data in R . . . . .	14
1.6 Looking at Data in RStudio . . . . .	14
1.7 Exploring the data . . . . .	15
<b>2 Gene Ontology</b>	<b>17</b>
2.1 Description . . . . .	18
2.2 Learning outcomes . . . . .	18
2.3 Analysis Workflow . . . . .	18
2.4 Get DE gene list . . . . .	19
2.5 The Hypergeometric Distribution in practice . . . . .	23
2.6 Now it is your turn . . . . .	24
2.7 Questions . . . . .	28

<b>3 Working with Sequences: Raw Data &amp; Quality Control</b>	<b>31</b>
3.1 Description . . . . .	31
3.2 Learning outcomes . . . . .	32
3.3 Download fastq . . . . .	32
3.4 Examining fastq . . . . .	35
3.5 Trimming . . . . .	40
3.6 Batch file processing . . . . .	42
3.7 QC and adapters . . . . .	44
3.8 Running fastqc . . . . .	46
3.9 Multiqc for QC on mutliple samples . . . . .	62
<b>4 Read Mapping</b>	<b>65</b>
4.1 Alignment . . . . .	66
4.2 Retrieve the genome . . . . .	66
4.3 Build Rsubread Index . . . . .	67
4.4 Pseudomapping with Salmon . . . . .	71
4.5 Questions . . . . .	75
<b>5 Read Counting</b>	<b>79</b>
5.1 featureCounts . . . . .	79
5.2 Salmon . . . . .	98
5.3 Questions . . . . .	134
<b>6 Differential Expression: EdgeR</b>	<b>137</b>
6.1 Description . . . . .	137
6.2 Learning outcomes . . . . .	138
6.3 Loading in the featureCounts object . . . . .	138
6.4 Count loading and Annotation . . . . .	141
6.5 Filtering to remove low counts . . . . .	142
6.6 Normalization for composition bias . . . . .	142
6.7 MDS plots . . . . .	143
6.8 Exploring differences between libraries . . . . .	144

<b>CONTENTS</b>	<b>5</b>
6.9 Estimate Dispersion . . . . .	145
6.10 Testing for differential expression . . . . .	147
6.11 Looking at all contrasts at once . . . . .	151
6.12 Questions . . . . .	153
6.13 A template set of code chunks for doing this is below: . . . . .	153
<b>7 Differential Expression: DESeq2</b>	<b>161</b>
7.1 Description . . . . .	161
7.2 Learning outcomes . . . . .	162
7.3 Loading in the featureCounts object . . . . .	162
7.4 Count loading and Annotation . . . . .	163
7.5 Filtering to remove low counts . . . . .	164
7.6 Testing for differential expression . . . . .	165
7.7 Questions . . . . .	173
<b>8 Differential Expression: limma</b>	<b>179</b>
8.1 Description . . . . .	179
8.2 Learning Objectives . . . . .	180
8.3 Loading in the count data file . . . . .	180
8.4 Count loading and Annotation . . . . .	182
8.5 Filtering to remove low counts . . . . .	183
8.6 Normalization for composition bias . . . . .	184
8.7 Exploring differences between libraries . . . . .	185
8.8 Estimate Dispersion . . . . .	186
8.9 Testing for differential expression . . . . .	188
8.10 Examining a specific contrast . . . . .	193
8.11 Visualization . . . . .	196
8.12 <code>treat()</code> testing . . . . .	197
8.13 Comparing DE analysis softwares . . . . .	201
8.14 Correlation between logFC estimates across softwares . . . . .	203
8.15 Questions . . . . .	206

<b>9 Visualizing Differential Expression Results</b>	<b>209</b>
9.1 Description . . . . .	209
9.2 Learning Outcomes . . . . .	210
9.3 MA-plot . . . . .	211
9.4 Volcano Plot . . . . .	213
9.5 Using Glimma for an interactive visualization . . . . .	215
9.6 Generating bar graph summaries . . . . .	219
9.7 Exercise . . . . .	224
<b>10 Clustering</b>	<b>227</b>
10.1 Description . . . . .	227
10.2 Learning outcomes . . . . .	227
10.3 Cluster 3.0 . . . . .	227
10.4 Visualizing Clusters with Java TreeView . . . . .	245
10.5 Performing clustering on yeast stress data . . . . .	251
10.6 Questions . . . . .	251
<b>11 KEGG Analysis</b>	<b>253</b>
11.1 Description . . . . .	254
11.2 Learning Outcomes . . . . .	254
11.3 Loading in the edgeR DE gene file output. . . . .	254
11.4 KEGG Analysis . . . . .	255
11.5 Visualize on the KEGG website . . . . .	270
11.6 Comparing Paralogs in common pathways . . . . .	270
11.7 Additional KEGG-related analyses . . . . .	273
11.8 Dotplot . . . . .	274
11.9 cnetplot . . . . .	274
11.10heatplot . . . . .	277
11.11upsetplot . . . . .	278
11.12emapplot . . . . .	279
11.13GSEA . . . . .	280
11.14Questions . . . . .	291

<b>12 Motif Analysis: MEME Suite</b>	<b>293</b>
12.1 Description . . . . .	293
12.2 Learning Objectives . . . . .	293
12.3 Install MEME suite . . . . .	294
12.4 Analysis: Motif Discovery for <i>msn2/4</i> vs WT Response to EtOH	295
12.5 Motif Analysis for Genes Downregulated in EtOH Response . . .	298
12.6 <i>skn7</i> exposed to salt. . . . .	312
12.7 Questions . . . . .	313



# Preface

This online resource is a compilation of exercises created for a graduate level course in Genomic Data Analysis at the University of Arkansas, taught by Dr. Jeffrey Lewis. The exercises included have been developed by graduate student Carson Stacy in collaboration with Dr. Jeffrey Lewis.

## 0.1 Usage

Each chapter corresponds to a class exercise, most of which are completed in R. There are .Rmd files available for each of the chapters available here, where you can complete the exercises yourself.

## 0.2 Key Features

- **Real Genomic Datasets:** Explore exercises using genuine genomic datasets. This hands-on experience allows users to bridge theoretical knowledge with practical application, mirroring the challenges encountered in real-world genomics research.
- **Focus on Biological Context:** Beyond coding, the exercises emphasize the biological questions addressed by genomic data analysis. Understanding the context behind the code is crucial for meaningful interpretation of results in genomics research.
- **Self-Paced Learning:** Tailor your learning experience to your pace. The exercises are designed to accommodate a range of skill levels, allowing users to progress gradually and revisit concepts as needed.

Happy Learning!

```
bookdown::serve_book()
```

### 0.3 Disclaimer

The exercises included are a compilation of resources we have worked with through the years. Earnest attempts has been made to give credit where credit is due, but we can provide no guarantee to the origins of every piece of this document.

# Chapter 1

## Getting Started in R

last updated: 2023-10-27

### Installing Packages

First things first: Click the “Visual” button in the top-left corner of the code box. This makes the code look more like a word processor. You can always switch back to Source anytime you prefer.

The following code installs a set of R packages used in this document – if not already installed – and then loads the packages into R. Note that we utilize the US CRAN repository, but other repositories may be more convenient according to geographic location.

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# the p_load function
#   A) installs the package if not installed (like install.packages("package_name")),
#   B) loads the package (equivalent of library(package_name))

p_load("tidyverse", # An ecosystem of packages for making life in R easier
       "here", # For locating files easily
       "knitr", # For generating ("knitting") html or pdf files from .Rmd file
       "readr", # For faster and easier reading in files to R
       "pander", # For session info at the end of the document
       "BiocManager", # For installing Bioconductor R packages
       "dplyr" # A key part of the tidyverse ecosystem, has useful functions
       )
```

## 1.1 Exercise Description

This activity is intended to familiarize you with using RStudio and the R ecosystem to analyze genomic data

## 1.2 Learning outcomes

At the end of this exercise, you should be able to:

- open, modify, and knit an Rmd file to a pdf/html output
- relate Rmarkdown to a traditional lab notebook
- run commands in an Rmarkdown file

## 1.3 Using R and RStudio

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
# print a statement
print("R code in a .Rmd chunk works just like a script")
```

```
## [1] "R code in a .Rmd chunk works just like a script"
```

```
# perform basic calculations
2+2
```

```
## [1] 4
```

R is a useful tool for analyzing data. Let's download a data file from GitHub to work with. First, we will download the file manually and open it. Later, we will download the same file directly from the url.

- Click here to open the file in GitHub and click the download icon to download it to your computer.
- Use the “Import Dataset” in the Environment panel of RStudio to open the file browser and select the downloaded file

- You'll want to use the “From text (readr)...” option
- Adjust settings to make sure the file loads in properly.
- Copy the code that the Import Dataset feature provides for reading in the file and paste it in the code chunk below

```
# insert here the code used to load the file in from your computer
```

## 1.4 Load data directly from the URL

Rather than downloading the file manually and then loading it in from where we downloaded it to, we can just load it directly from the URL, as shown below. A word of caution, this won't work with any URL and you can't guarantee the URL will always work in the future.

```
# assign url to a variable
DE_data_url <- "https://raw.githubusercontent.com/clstacy/GenomicDataAnalysis_Fa23/main/data/etha

# download the data from the web
DE_results_msn24_EtOH <-
  read_tsv(file=DE_data_url)

## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 5756 Columns: 18
## -- Column specification -----
## Delimiter: "\t"
## chr (3): Gene ID, Common Name, Annotation
## dbl (15): logFC: YPS606 (WT) EtOH response, Pvalue: YPS606 (WT) EtOH respons...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Do remember that this function uses the package `readr` (a part of the `tidyverse` package we loaded above). If you don't have that package (1) installed and (2) loaded into your script, it won't work. Thankfully, the `p_load` function takes care of both of these simultaneously.

## 1.5 Working with data in R

To get a quick summary of our data and how it looks

```
# take a quick look at how the data is structured
glimpse(DE_results_msn24_EtOH)
```

```
## Rows: 5,756
## Columns: 18
## $ `Gene ID` <chr> "YMR105C", "YML100W", "YER053~  

## $ `Common Name` <chr> "PGM2", "TSL1", "PIC2", "NCE1~  

## $ Annotation <chr> "Phosphoglucomutase", "Large ~  

## $ `logFC: YPS606 (WT) EtOH response` <dbl> 7.5999973, 7.7618280, 6.69400~  

## $ `Pvalue: YPS606 (WT) EtOH response` <dbl> 9.40e-38, 1.04e-35, 3.03e-39,~  

## $ `FDR: YPS606 (WT) EtOH response` <dbl> 3.26e-35, 1.54e-33, 2.07e-36,~  

## $ `logFC: YPS606 msn2/4ΔΔ EtOH response` <dbl> 0.78481798, 0.60949852, 1.735~  

## $ `Pvalue: YPS606 msn2/4ΔΔ EtOH response` <dbl> 3.430000e-06, 8.401730e-04, 4~  

## $ `FDR: YPS606 msn2/4ΔΔ EtOH response` <dbl> 7.420000e-06, 1.398507e-03, 2~  

## $ `logFC: WT v msn2/4ΔΔ: EtOH response` <dbl> -6.815179, -7.152329, -4.9580~  

## $ `Pvalue: WT v msn2/4ΔΔ: EtOH response` <dbl> 6.34e-32, 2.53e-30, 1.35e-27,~  

## $ `FDR: WT v msn2/4ΔΔ: EtOH response` <dbl> 3.65e-28, 7.28e-27, 2.59e-24,~  

## $ `logFC: WT v msn2/4ΔΔ: unstressed` <dbl> -0.144061475, -0.365016862, --~  

## $ `Pvalue: WT v msn2/4ΔΔ: unstressed` <dbl> 0.350436027, 0.041423492, 0.4~  

## $ `FDR: WT v msn2/4ΔΔ: unstressed` <dbl> 0.998531082, 0.998531082, 0.9~  

## $ `logFC: WT v msn2/4ΔΔ: EtOH absolute` <dbl> -6.959241, -7.517346, -5.0845~  

## $ `Pvalue: WT v msn2/4ΔΔ: EtOH absolute` <dbl> 8.55e-37, 2.04e-35, 3.06e-36,~  

## $ `FDR: WT v msn2/4ΔΔ: EtOH absolute` <dbl> 1.64e-33, 1.96e-32, 3.52e-33,~
```

We see in the output there are 5756 rows and 18 columns in the data. The same information should be available in the environment panel of RStudio

## 1.6 Looking at Data in RStudio

If we want to take a closer look at the data, we have a few options. To see just the first few lines we can run the following command:

```
head(DE_results_msn24_EtOH)
```

```
## # A tibble: 6 x 18
##   `Gene ID` `Common Name` Annotation logFC: YPS606 (WT) E-1
##   <chr>     <chr>       <chr>          <dbl>
## 1 YMR105C   PGM2        Phosphoglucomutase 7.60
```

```

## 2 YML100W    TSL1      Large subunit of trehalose 6-p-          7.76
## 3 YER053C    PIC2      Mitochondrial copper and phospho-        6.69
## 4 YPR149W    NCE102    Protein involved in regulation-        0.714
## 5 YKL035W    UGP1      UDP-glucose pyrophosphorylase ~       4.42
## 6 YLR258W    GSY2      Glycogen synthase                      7.52
## # i abbreviated name: 1: `logFC: YPS606 (WT) EtOH response` 
## # i 14 more variables: `Pvalue: YPS606 (WT) EtOH response` <dbl>,
## #   `FDR: YPS606 (WT) EtOH response` <dbl>,
## #   `logFC: YPS606 msn2/4ΔΔ EtOH response` <dbl>,
## #   `Pvalue: YPS606 msn2/4ΔΔ EtOH response` <dbl>,
## #   `FDR: YPS606 msn2/4ΔΔ EtOH response` <dbl>,
## #   `logFC: WT v msn2/4ΔΔ: EtOH response` <dbl>, ...

```

This can be difficult to look at. For looking at data similar to an Excel file, RStudio allows this by clicking on the name of the data.frame in the top right corner of the IDE. We can also view a file by typing `View(filename)`. To open the data in a new window, click the “pop out” button next to “filter” just above the opened dataset.

## 1.7 Exploring the data

This dataset includes the log fold changes of gene expression in an experiment testing the ethanol stress response for the YPS606 strain of *S. cerevisiae* and an *msn2/4ΔΔ* mutant. There are also additional columns of metadata about each gene. In later classes, we will cover the details included, but we can already start answering questions.

**Using RStudio, answer the following questions:**

1. How many genes are included in this study?
2. Which gene has the highest log fold change in the *msn2/4ΔΔ* mutant EtOH response?
3. How many HSP genes are differentially expressed (FDR < 0.01) in un-stressed conditions for the mutant?
4. Do the genes with the largest magnitude fold changes have the smallest p-values?
5. Which isoform of phosphoglucomutase is upregulated in response to ethanol stress? Do you think *msn2/4* is responsible for this difference?

Be sure to knit this file into a pdf or html file once you’re finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

**R version 4.3.1 (2023-06-16)**

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8||en\_US.UTF-8||C||en\_US.UTF-8|en\_US.UTF-8

**attached base packages:** *stats, graphics, grDevices, utils, datasets, methods* and *base*

**other attached packages:** *BiocManager(v.1.30.22), pander(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3), forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)*

**loaded via a namespace (and not attached):** *utf8(v.1.2.3), generics(v.0.1.3), stringi(v.1.7.12), hms(v.1.1.3), digest(v.0.6.33), magrittr(v.2.0.3), evaluate(v.0.22), grid(v.4.3.1), timechange(v.0.2.0), bookdown(v.0.36), fastmap(v.1.1.1), rprojroot(v.2.0.3), fansi(v.1.0.5), scales(v.1.2.1), codetools(v.0.2-19), cli(v.3.6.1), crayon(v.1.5.2), rlang(v.1.1.1), bit64(v.4.0.5), munsell(v.0.5.0), withr(v.2.5.1), yaml(v.2.3.7), parallel(v.4.3.1), tools(v.4.3.1), tzdb(v.0.4.0), colorspace(v.2.1-0), curl(v.5.1.0), vctrs(v.0.6.4), R6(v.2.5.1), lifecycle(v.1.0.3), bit(v.4.0.5), vroom(v.1.6.4), pkgconfig(v.2.0.3), pillar(v.1.9.0), gtable(v.0.3.4), glue(v.1.6.2), Rcpp(v.1.0.11), xfun(v.0.40), tidyselect(v.1.2.0), rstudioapi(v.0.15.0), htmltools(v.0.5.6.1), rmarkdown(v.2.25) and compiler(v.4.3.1)*

# Chapter 2

## Gene Ontology

last updated: 2023-10-27

### Installing Packages

The following code installs all of the packages used in this document – if not already installed – and then loads the packages into R. We need to install packages specific to our gene ontology bioinformatic analysis. Many of these packages aren’t available on the R CRAN package repository, instead they are hosted on BioConductor repository that is focused on packages used in biological research. Today, we need to install the package clusterProfiler with the code below. The `p_load()` function will check the bioconductor repository if the package isn’t on CRAN

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

p_load("tidyverse", "here", "knitr", "dplyr", # already downloaded last activity
       "readr", "pander", "BiocManager", # also from last activity
       "janitor", # for cleaning column names
       "igraph", "tidytree", # dependencies that require explicit download on latest Mac OS
       "ggVennDiagram", # visualization venn diagram
       "clusterProfiler", # for GO enrichment
       "AnnotationDbi", # database of common genome annotations
       "org.Sc.sgd.db" # annotation database for S. cerevesiae
       )

library(dplyr)
```

## 2.1 Description

This activity is intended to familiarize you with Gene Ontology analysis and some of the unique challenges that come from working with bioinformatic data.

## 2.2 Learning outcomes

At the end of this exercise, you should be able to:

- Understand gene ontology and its significance in functional annotation
- learn to perform a GO enrichment & appropriate statistical methods (hypergeometric & Fisher's exact test) for the enrichment analysis
- interpret & critically evaluate the results of GO enrichment & limitations/challenges

```
# we don't have to run this, but if you install without pacman, we have to do load lib
library(clusterProfiler)
library(org.Sc.sgd.db)
```

## 2.3 Analysis Workflow

Let's use the same file from last class, this time performing GO term enrichment

```
# assign url to a variable
DE_data_url <- "https://raw.githubusercontent.com/clstacy/GenomicDataAnalysis_Fa23/main/MSN24_EtOH.csv"

# download the data from the web
DE_results_msn24_EtOH <-
  read_tsv(file=DE_data_url)

## Warning: One or more parsing issues, call `problems()` on your data frame for details
## e.g.:
##   dat <- vroom(dat)
##   problems(dat)

## Rows: 5756 Columns: 18
## -- Column specification -----
## Delimiter: "\t"
## chr (3): Gene ID, Common Name, Annotation
## dbl (15): logFC: YPS606 (WT) EtOH response, Pvalue: YPS606 (WT) EtOH respons...
##
```

```
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.

msn24_EtOH <- # assign a new object name
DE_results_msn24_EtOH |> # our object with messy names
clean_names() # function from janitor package to make names consistent
```

## 2.4 Get DE gene list

We need a list of differentially expressed genes to test for over or under enrichment of terms. Here we choose genes with significantly ( $FDR < 0.05$ ) higher expression ( $\log_2$ -fold change ( $\log FC$ ) greater than 1) in the *msn2/4ΔΔ* mutant's EtOH response compared to the wild-type strains EtOH response (positive values in the  $\log FC$  column of WT vs *msn2/4ΔΔ*: EtOH response).

```
# subset to just genes with significant fdr & log2FC>1
msn24_EtOH |>
  filter(log_fc_wt_v_msn2_4dd_et_oh_response > 1 & fdr_wt_v_msn2_4dd_et_oh_response < 0.05)

## # A tibble: 94 x 18
##   gene_id common_name annotation log_fc_yps606_wt_et_~1 pvalue_yps606_wt_et_~2
##   <chr>    <chr>      <chr>           <dbl>          <dbl>
## 1 YOR315W  SFG1       Putative ~        -5.52          4.33e-31
## 2 YFL051C  YFL051C    <NA>            -4.54          6.37e-27
## 3 YMR016C  SOK2       Nuclear p~       -3.09          1.32e-32
## 4 YPL061W  ALD6       Cytosolic~      -7.04          4.96e-26
## 5 YER073W  ALD5       Mitochond~     -1.88          6.14e-17
## 6 YBL005W~ YBL005W-B  Retrotran~     1.91           5.46e-13
## 7 YBL039C  URA7       Major CTP~      -6.95          3.62e-41
## 8 YJL050W  MTR4       RNA duple~     -4.59          4.73e-36
## 9 YMR241W  YHM2       Citrate a~     -1.72          1.72e-20
## 10 YIL131C  FKH1      Forkhead ~     -2.20          1.23e-27
## # i 84 more rows
## # i abbreviated names: 1: log_fc_yps606_wt_et_oh_response,
## #   2: pvalue_yps606_wt_et_oh_response
## # i 13 more variables: fdr_yps606_wt_et_oh_response <dbl>,
## #   log_fc_yps606_msn2_4dd_et_oh_response <dbl>,
## #   pvalue_yps606_msn2_4dd_et_oh_response <dbl>,
## #   fdr_yps606_msn2_4dd_et_oh_response <dbl>, ...

# the above command gave us what we want, here it is again but saved to a new variable:
DE_genes_upregulated_msn24_EtOH <-
  msn24_EtOH |>
```

```
filter(log_fc_wt_v_msn2_4dd_et_oh_response > 1 & fdr_wt_v_msn2_4dd_et_oh_response <
      pull(gene_id) # get just the gene names
```

Now we have a list of genes (saved as `DE_genes_upregulated_msn24_EtOH`) that we want to perform GO term enrichment on. Let's do that now, using the `clusterProfiler` package's `enrichGO` function

```
GO_msn24_EtOH_up_results <- enrichGO(
  gene = DE_genes_upregulated_msn24_EtOH,
  OrgDb = "org.Sc.sgd.db",
  universe = msn24_EtOH$gene_id,
  keyType = "ORF",
  ont = "BP"
) |>
  # let's add a 'richFactor' column that gives us the proportion of genes DE in the test
  mutate(richFactor = Count / as.numeric(sub("/\\d+", "", BgRatio)))
```

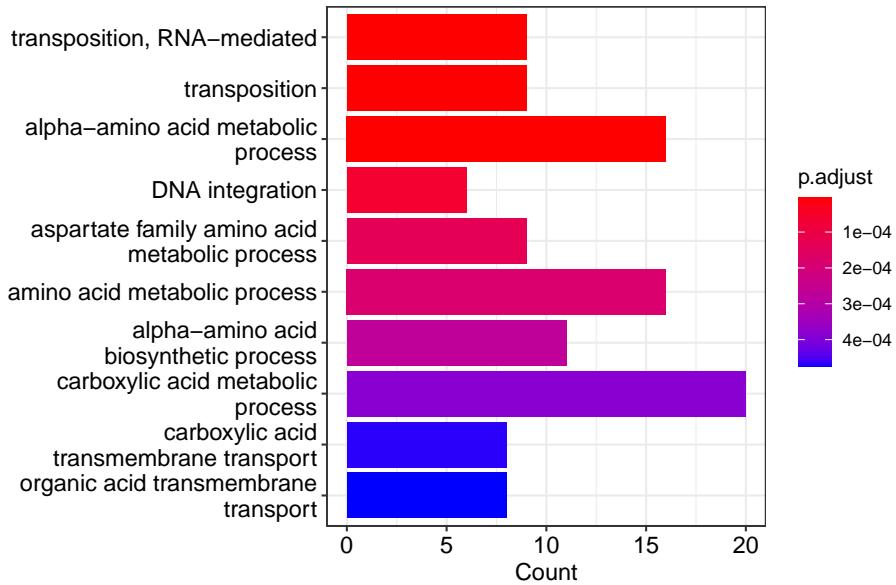
Now, we can look at the results in table form.

```
# open up the results in a data frame to examine
GO_msn24_EtOH_up_results |>
  as_tibble() |>
  View()

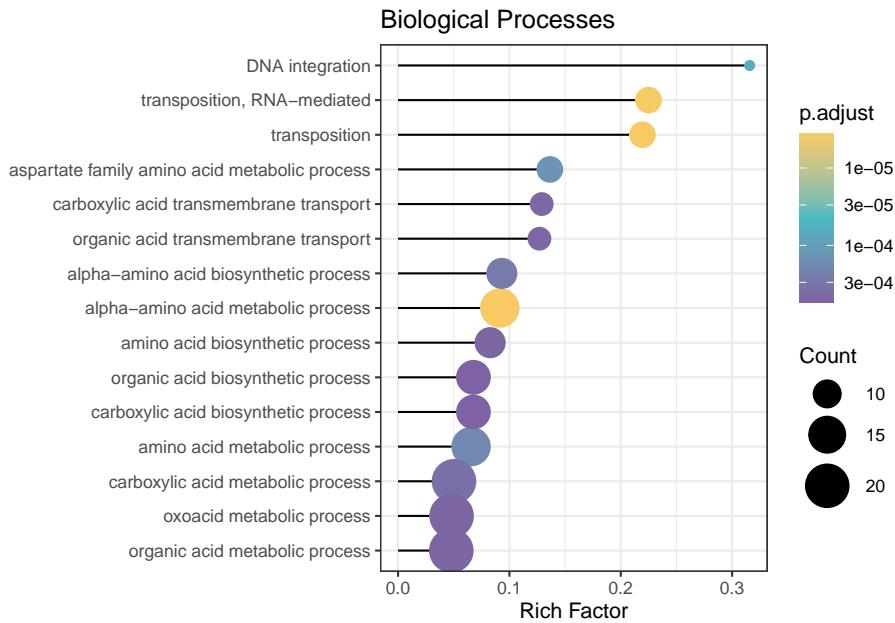
# Here is how we could write this result into a text file:
GO_msn24_EtOH_up_results |>
  as_tibble() |>
  write_tsv(file = "~/Desktop/GO_msn24_EtOH_up_results.tsv")
```

Now we can visualize the enrichment results, which shows us gene ontology categories that are enriched in genes with higher expression (upregulated) in the WT vs  $msn2/4\Delta\Delta$ : EtOH response.

```
# a simple visualization
plot(barplot(GO_msn24_EtOH_up_results, showCategory = 10))
```



```
# a more complicated visualization, with more information density
ggplot(GO_msn24_EtOH_up_results,
       showCategory = 15,
       aes(richFactor, fct_reorder(Description, richFactor))) +
  geom_segment(aes(xend = 0, yend = Description)) +
  geom_point(aes(color = p.adjust, size = Count)) +
  scale_color_gradientn(
    colours = c("#f7ca64", "#46bac2", "#7e62a3"),
    trans = "log10",
    guide = guide_colorbar(reverse = TRUE, order = 1)
  ) +
  scale_size_continuous(range = c(2, 10)) +
  xlab("Rich Factor") +
  ylab(NULL) +
  ggtitle("Biological Processes") +
  theme_bw()
```



You can try adjusting the size of the output figures by clicking the gear icon in the top right of the code chunk and click “use custom figure size”. Note this updates the chunk header so the change is saved.

#### 2.4.1 Saving ggplot output to a file

We usually want to save our visualizations for later. When plotting with the ggplot package, there is an easy way to do this. See below:

```
# First, let's create a folder to save our visualizations
dir_visualization <- path.expand("~/Desktop/Genomic_Data_Analysis/Visualization/")
if (!dir.exists(dir_visualization)) {dir.create(dir_visualization, recursive = TRUE)}

# type ?ggsave in the console for more information via the help page.
ggsave(
  "GO_BP_msn24_EtOH_up_results_lollipopPlot.pdf",
  # if we don't need the image to go to a certain spot, we only need the file name above
  plot = last_plot(), # either the last plot, or name of a ggplot object you've saved.
  device = "pdf", #Can be "png", "eps", "ps", "tex" (pictex), "pdf", "jpeg", "tiff", "svg"
  # note that pdf, eps, svg are vector/line art, so zooming doesn't pixelate.
  path = dir_visualization, # Path of the directory to save plot to. defaults to work directory
  scale = 2, # multiplicative scaling factor
  width = 12,
  height = 8,
```

```

units = "cm", # must be one of: "in", "cm", "mm", "px"
dpi = 300, # adjusting this larger gives higher quality plot, making a larger file.
limitsize = TRUE, # prevents accidentally making it massive, defaults to TRUE
bg = NULL # Background colour. If NULL, uses the plot.background fill value from the plot theme
)

```

Recall that when we knit this Rmarkdown notebook, we keep a copy of the plots/images there as well, in the same place as the code and analysis used to generate it. However, we may want a higher resolution file of just the image, or the image in a different format. In this case, saving the plot is a useful option for us. The journal Science has the following recommendations: “We prefer prefer ai, eps, pdf, layered psd, tif, and jpeg files. ...minimum file resolution of 300 dpi.”

## 2.5 The Hypergeometric Distribution in practice

Notice that the DNA integration process does not have very many genes in the category, but they appear to be highly present in the the upregulated gene list. Specifically, DE genes have this GO term, where in the entire genome, there are only genes. What are the odds that we see this by random chance? let's do the math:

```

# number of genes that have GO:0015074 (DNA integration)
integration_genes = 23
# number of genes that are DE (msn2/4 EtOH response, logFC>1)
DE_genes = 91
# number of genes that are both DE and DNA integration genes
Overlap = 6
# total number of genes in experiment
total = 5538 # number of genes in genome

```

Without doing the math, do you expect these to be underrepresented, overrepresented, or neither?

```

# test for underrepresentation (depletion)
phyper(q = Overlap, # number of integration genes that were DE
       m = DE_genes, # number of DE genes
       n = total-DE_genes, # number of non DE genes
       k = integration_genes, # number of observed DE DNA integration genes
       lower.tail = TRUE) # the probability that X <= x

```

```
## [1] 0.9999999
```

```
# test for overrepresentation (enrichment)
hyper(q = Overlap-1, # number of integration genes that were DE
      # we subtract 1 b/c of lower.tail=FALSE means greater than
      # without equality, so have to do one less
      m = DE_genes, # number of DE genes
      n = total-DE_genes, # number of non DE genes
      k = integration_genes, # number of observed DE integration genes
      lower.tail = FALSE) # the probability that X > x

## [1] 1.344447e-06
```

As we see, there is strong evidence that the number of genes with this GO term is unlikely to be seen due to chance. In layman's terms, this GO term is enriched in upregulated genes in this contrast. The test for underrepresentation shows there is no support for a hypothesis that this gene is underrepresented in the DE gene list.

Interestingly, the hypergeometric distribution is the same thing as the Fisher's Exact test, so we can rerun the same tests above with a different command:

```
#fisher test for underrepresentation
fisher.test(matrix(c(Overlap, DE_genes-Overlap, integration_genes-Overlap, total-DE_gen

## [1] 0.9999999

#fisher test for overrepresentation
fisher.test(matrix(c(Overlap, DE_genes-Overlap, integration_genes-Overlap, total-DE_gen

## [1] 1.344447e-06
```

How does the p-value that we get from this test compare to the results table? They should match.

## 2.6 Now it is your turn

Try running your own GO enrichment with a different gene list. Some options could be:

- Start with the WT vs *msn2/4ΔΔ*: EtOH response again, and this time change to "downregulated" (i.e., genes with higher expression in the wild-type strain compared to the *msn2/4ΔΔ* mutant). These would potentially include genes with defective induction.

- See what happens when you change the FDR threshold from a liberal one (0.05) to a more conservative one (0.01).
- Try different logFC cutoffs.
- Look at different comparisons in the data file (there are 5 total)
- Look at a different GO category (we only looked at BP, not MF or CC)
- Advanced: include multiple filters (e.g., genes upregulated by EtOH stress in the WT strain that ALSO have defective induction during ethanol stress in the *msn2*/4ΔΔ mutant).

The code below is a template for you to modify to complete this activity. The example code below looks at the downregulated genes in response to stress in the WT (choose something else for your gene list)

---

```
# subset to just genes meeting your requirements
DE_genes_GIVE_NAME <- msn24_EtOH |>
  # change the below line for the filters that you want
  filter(log_fc_yps606_wt_et_oh_response < 1 & pvalue_yps606_wt_et_oh_response<0.05) |>
  pull(gene_id) # grabbing just the gene names
```

### 2.6.1 Run Enrichment

```
GO_GIVE_NAME_results <- enrichGO(
  gene = DE_genes_GIVE_NAME,
  OrgDb = "org.Sc.sgd.db",
  universe = msn24_EtOH$gene_id,
  keyType = "ORF",
  ont= "BP"
) |>
  mutate(richFactor = Count / as.numeric(sub("/\\d+", "", BgRatio)))
```

### 2.6.2 see the data

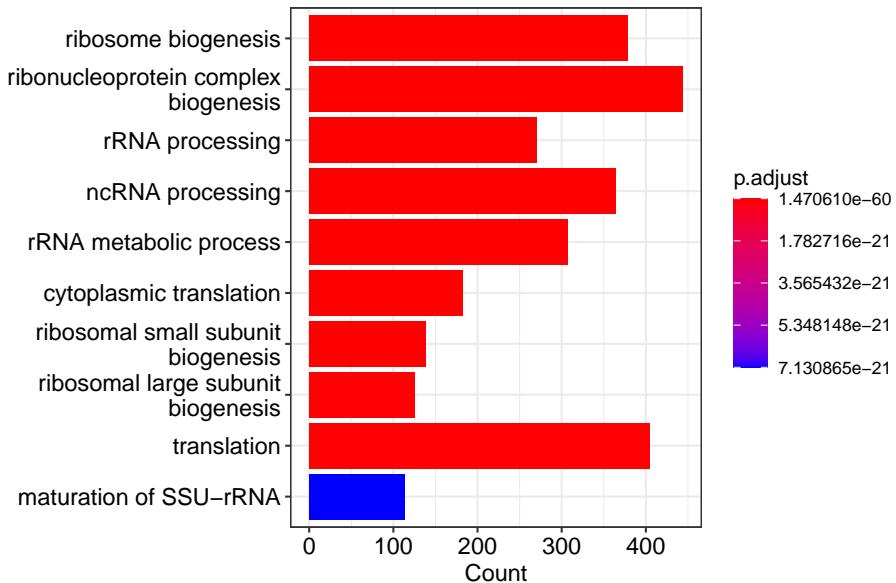
```
# open up the results in a data frame to examine
GO_GIVE_NAME_results |>
  as_tibble() |>
  View()

# write out your results to a text file
```

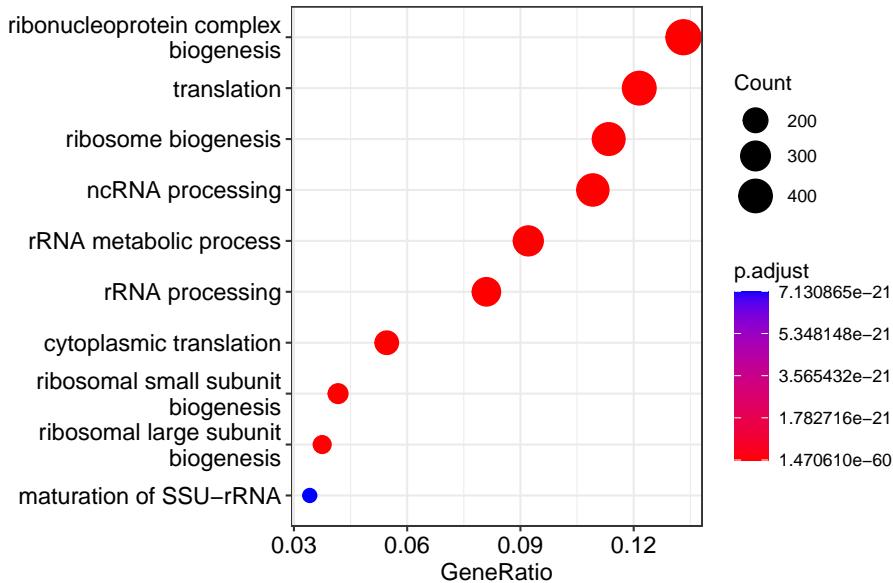
```
GO_GIVE_NAME_results |>
  as_tibble() |>
  write_tsv(file = "~/Desktop/GO_GIVE_NAME_DIRECTION_results.tsv")
```

### 2.6.3 create plots

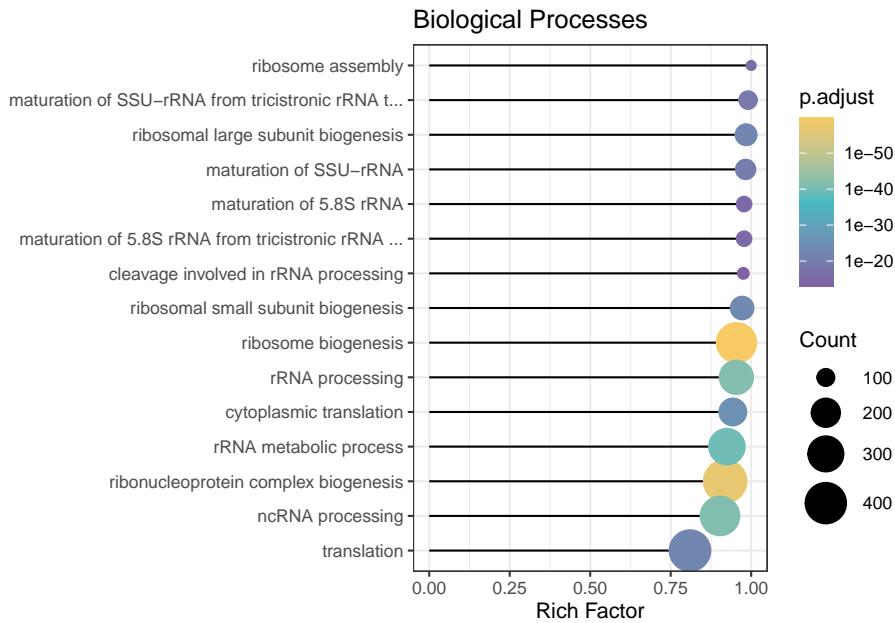
```
# a simple visualization
plot(barplot(GO_GIVE_NAME_results, showCategory = 10))
```



```
# built in visualization with dots instead
dotplot(GO_GIVE_NAME_results, showCategory=10)
```



```
# a more complicated visualization, with more information density
ggplot(GO_GIVE_NAME_results,
       showCategory = 15,
       aes(richFactor, fct_reorder(Description, richFactor))) +
  geom_segment(aes(xend = 0, yend = Description)) +
  geom_point(aes(color = p.adjust, size = Count)) +
  scale_color_gradientn(
    colours = c("#f7ca64", "#46bac2", "#7e62a3"),
    trans = "log10",
    guide = guide_colorbar(reverse = TRUE, order = 1)
  ) +
  scale_size_continuous(range = c(2, 10)) +
  scale_y_discrete(label = function(x) stringr::str_trunc(x, 50)) + # cut off long names
  xlab("Rich Factor") +
  ylab(NULL) +
  ggtitle("Biological Processes") +
  theme_bw()
```



## 2.7 Questions

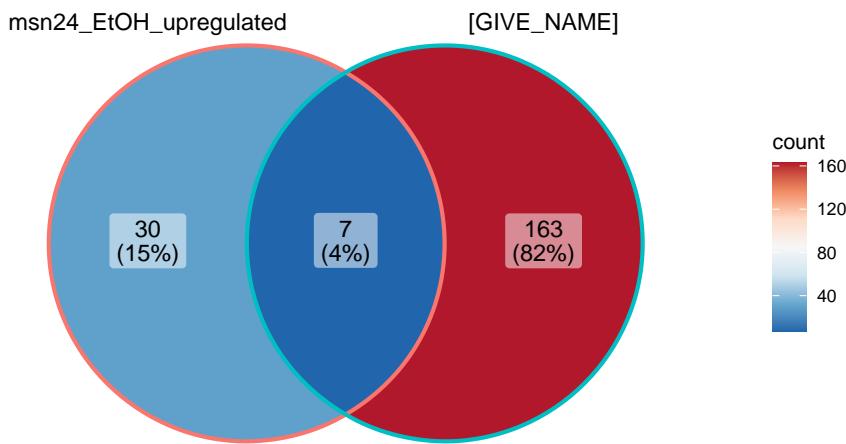
Answer the following questions:

1. Which GO term had the smallest adjusted p-value in the upregulated comparison example that we did together?
2. What percent of the genes would we expect to have that GO term in the DE list under the null hypothesis? What percent of the DE genes actually had that GO term?
3. For the upregulated comparison, what GO terms are enriched for genes with  $pval < 0.01$  but  $fdr > 0.01$  and what is their average/median log fold change?
4. For one of your own novel comparisons, explain what comparison you were interested in, and your rationale for the cutoffs you chose for your gene list.
5. For that novel gene list you chose for yourself, which GO term had the smallest adjusted p-value?
6. In simple terms, how would you describe what the “Rich Factor” tells about a given GO term in the gene list.

7. Challenge: create a venn diagram of the GO terms in the GO analysis you ran comparing to the upregulated comparison example.

```
# create a list of the data we want to compare
GO_results_list <- list(data.frame(GO_msn24_EtOH_up_results)$ID,
                         data.frame(GO_GIVE_NAME_results)$ID)

# visualize the GO results list as a venn diagram
ggVennDiagram(GO_results_list,
               category.names = c("msn24_EtOH_upregulated", "[GIVE_NAME]")) +
  scale_x_continuous(expand = expansion(mult = .2)) +
  scale_fill_distiller(palette = "RdBu"
)
```



Be sure to knit this file into a pdf or html file once you're finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

**R version 4.3.1 (2023-06-16)**

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8||en\_US.UTF-8||en\_US.UTF-8||C||en\_US.UTF-8||en\_US.UTF-8

**attached base packages:** *stats4, stats, graphics, grDevices, utils, datasets, methods* and *base*

**other attached packages:** *org.Sc.sgd.db(v.3.17.0), AnnotationDbi(v.1.62.2), IRanges(v.2.34.1), S4Vectors(v.0.38.2), Biobase(v.2.60.0), BiocGenerics(v.0.46.0), clusterProfiler(v.4.8.2), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pandoc(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3), forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyverse(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)*

**loaded via a namespace (and not attached):** *RColorBrewer(v.1.1-3), rstudiosapi(v.0.15.0), jsonlite(v.1.8.7), magrittr(v.2.0.3), farver(v.2.1.1), rmarkdown(v.2.25), ragg(v.1.2.6), fs(v.1.6.3), zlibbioc(v.1.46.0), vctrs(v.0.6.4), memoise(v.2.0.1), RCurl(v.1.98-1.12), ggtree(v.3.8.2), htmltools(v.0.5.6.1), curl(v.5.1.0), gridGraphics(v.0.5-1), KernSmooth(v.2.23-22), plyr(v.1.8.9), cachem(v.1.0.8), lifecycle(v.1.0.3), pkgconfig(v.2.0.3), Matrix(v.1.6-1.1), R6(v.2.5.1), fastmap(v.1.1.1), gson(v.0.1.0), GenomeInfoDbData(v.1.2.10), snakecase(v.0.11.1), digest(v.0.6.33), aplot(v.0.2.2), enrichplot(v.1.20.0), colorspace(v.2.1-0), patchwork(v.1.1.3), rprojroot(v.2.0.3), textshaping(v.0.3.7), RSQLite(v.2.3.1), labeling(v.0.4.3), fansi(v.1.0.5), timechange(v.0.2.0), httr(v.1.4.7), polyclip(v.1.10-6), compiler(v.4.3.1), proxy(v.0.4-27), bit64(v.4.0.5), withr(v.2.5.1), downloader(v.0.4), BiocParallel(v.1.34.2), viridis(v.0.6.4), DBI(v.1.1.3), ggforce(v.0.4.1), MASS(v.7.3-60), classInt(v.0.4-10), HDO.db(v.0.99.1), units(v.0.8-4), tools(v.4.3.1), ape(v.5.7-1), scatterpie(v.0.2.1), glue(v.1.6.2), nlme(v.3.1-163), GOSemSim(v.2.26.1), sf(v.1.0-14), grid(v.4.3.1), shadowtext(v.0.1.2), reshape2(v.1.4.4), fgsea(v.1.26.0), generics(v.0.1.3), gtable(v.0.3.4), tzdb(v.0.4.0), class(v.7.3-22), data.table(v.1.14.8), hms(v.1.1.3), tidygraph(v.1.2.3), utf8(v.1.2.3), XVector(v.0.40.0), ggrepel(v.0.9.4), pillar(v.1.9.0), yulab.utils(v.0.1.0), vroom(v.1.6.4), splines(v.4.3.1), tweenr(v.2.0.2), treeio(v.1.24.3), lattice(v.0.21-9), bit(v.4.0.5), tidyselect(v.1.2.0), GO.db(v.3.17.0), Biostrings(v.2.68.1), gridExtra(v.2.3), bookdown(v.0.36), xfun(v.0.40), graphlayouts(v.1.0.1), stringi(v.1.7.12), lazyeval(v.0.2.2), ggrepel(v.0.1.3), yaml(v.2.3.7), evaluate(v.0.22), codetools(v.0.2-19), ggraph(v.2.1.0), qvalue(v.2.32.0), RVenn(v.1.1.0), ggplotify(v.0.1.2), cli(v.3.6.1), systemfonts(v.1.0.5), munsell(v.0.5.0), Rcpp(v.1.0.11), GenomeInfoDb(v.1.36.4), png(v.0.1-8), parallel(v.4.3.1), blob(v.1.2.4), DOSE(v.3.26.1), bitops(v.1.0-7), viridisLite(v.0.4.2), e1071(v.1.7-13), scales(v.1.2.1), crayon(v.1.5.2), rlang(v.1.1.1), cowplot(v.1.1.1), fastmatch(v.1.1-4) and KEGGREST(v.1.40.1)*

# Chapter 3

## Working with Sequences: Raw Data & Quality Control

last updated: 2023-10-27

### Package Install

As usual, make sure we have the right packages for this exercise

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# let's load all of the files we were using and want to have again today
p_load("tidyverse", "knitr", "readr",
       "pander", "BiocManager",
       "dplyr", "stringr")

# We also need the bioconductor packages "ShortRead" and "rfastp" for today's activity.
p_load("Rfastp", "ShortRead")
```

### 3.1 Description

This activity is intended to familiarize you with raw bioinformatic sequence files. Specifically, we'll be working with short read sequencing data generated from an Illumina platform.

## 3.2 Learning outcomes

At the end of this exercise, you should be able to:

- Load and read into R a raw gzipped fastq file.
- Inspect sequence quality and evaluate results.
- Perform quality control on raw data and save the processed output.

Note that instead of `{r}`, the below chunk uses `{bash}`, meaning this isn't r code but bash code (the language used in the terminal). The `-nc` flag ensures the files are only downloaded if they don't already exist where you are downloading them.

This may take awhile the first time you run it. The below script is a bash command that downloads these files to your computer

## 3.3 Download fastq

```
# Be sure to change this file path to the path you want your data to go
RAW_DATA_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/Data/Raw"

# if you're using Windows 10,
# in RStudio, go to Tools>Global Options... > Terminal > New Terminals open with...
# and choose WSL bash or git bash
# next, use: (be sure to put in the correct username)
#RAW_DATA_DIR="/mnt/c/Users/$USER/Desktop/Genomic_Data_Analysis/Data/Raw"

# create the destination directory if it doesn't already exist
mkdir -p $RAW_DATA_DIR

echo $RAW_DATA_DIR

# change to that directory (for this code chunk only)
cd $RAW_DATA_DIR
pwd
# Download the files.
# WARNING: curl doesn't work with relative paths
# WT unstressed (mock)
curl -L -C - -0 https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/ethan
# WT EtOH
```

```

curl -L -C - -O https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/ethanol_stress/
# msn2/4dd unstressed (mock)
curl -L -C - -O https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/ethanol_stress/
# msn2/4dd EtOH
curl -L -C - -O https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/ethanol_stress/
# These are subsamples of raw fastq files from a current project in our lab.

# Make sure names are as desired
cd $RAW_DATA_DIR

# This loops through and removes the suffix file for any OS that doesn't auto do so.
for file in *; do
    newname=$(echo "$file" | sed 's/\?raw=TRUE//')
    mv "$file" "$newname"
done

# Let's see what one of these files contains:
# if you're on windows or linux, delete the g from gzcat below
gzcat $RAW_DATA_DIR/YP5606_WT_MOCK_REP1.fastq.gz | head -n8

## /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw
## /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw
## % Total      % Received % Xferd  Average Speed   Time     Time      Time  Current
##                                         Dload  Upload Total Spent   Left Speed
## 0  0  0  0  0  0  0  0 --::-- --::-- --::-- 0  0  0  0
## 100 7257k 100 7257k 0  0 7687k 0 --::-- --::-- --::-- 7687k
## % Total      % Received % Xferd  Average Speed   Time     Time      Time  Current
##                                         Dload  Upload Total Spent   Left Speed
## 0  0  0  0  0  0  0  0 --::-- --::-- --::-- 0  0  0  0
## 100 6113k 100 6113k 0  0 6921k 0 --::-- --::-- --::-- 6921k
## % Total      % Received % Xferd  Average Speed   Time     Time      Time  Current
##                                         Dload  Upload Total Spent   Left Speed
## 0  0  0  0  0  0  0  0 --::-- --::-- --::-- 0  0  0  0
## 100 7352k 100 7352k 0  0 9602k 0 --::-- --::-- --::-- 9602k
## % Total      % Received % Xferd  Average Speed   Time     Time      Time  Current

```

34 CHAPTER 3. WORKING WITH SEQUENCES: RAW DATA & QUALITY CONTROL

```
##                                     Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 9 6
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 100 5774k 100 5774k 0 0 8306k 0 0:00:01 0:00:01 8306k
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 100 6438k 100 6438k 0 0 7890k 0 0:00:01 0:00:01 7890k
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 100 6908k 100 6908k 0 0 6737k 0 0:00:01 0:00:01 6737k
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 100 6020k 100 6020k 0 0 8490k 0 0:00:01 0:00:01 8490k
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0100 5
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0100 5
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0100 6
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 0 6797k 0 5503 0 0 8754 0 0:13:15 0:13:15 8754100 6
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 24 7521k 24 1879k 0 0 2992k 0 0:00:02 0:00:02 2992k100 75
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
## 39 6936k 39 2728k 0 0 4322k 0 0:00:01 0:00:01 4322k100 6
## % Total % Received % Xferd Average Speed Time Time Time Current
##                                         Dload Upload Total Spent Left Speed
## 0 0 0 0 0 0 0 0 0 --::-- --::-- --::-- 0 0
```

We have the data downloaded onto our system now, so let's first take a look at some of these files ourselves

The R package ShortRead allows us to look at and process raw fastq files. It has many more features than we will use today.

### 3.4 Examining fastq

Let's take a look at a fastq file

```
# If you're using windows, put your username below and uncomment this code before continuing
if(.Platform$OS.type == "windows") {
  Sys.setenv(R_USER = "C:/Users/$USERNAME")
}

# change this directory here to where you have the file saved
path_fastq_WT_MOCK_REPO1 <- path.expand("~/Desktop/Genomic_Data_Analysis/Data/Raw/YPG606_WT_MOCK_F"

fastq_WT_MOCK_REPO1 <- readFastq(path_fastq_WT_MOCK_REPO1)

# file too big? swap readFastq() for:
subsampled_fastq_WT_MOCK_REPO1 <- yield(FastqSampler(path_fastq_WT_MOCK_REPO1, n=10000)) # where n
# the fastq files we downloaded are smaller than a normal fastq file,
# because they have been subsampled down from their full size for demonstration.
```

A few quick ways to examine the fastq data object

36 CHAPTER 3. WORKING WITH SEQUENCES: RAW DATA & QUALITY CONTROL



### 38CHAPTER 3. WORKING WITH SEQUENCES: RAW DATA & QUALITY CONTROL

We see most of the nucleotides are assigned to A, C, G, or T, with one base in each read an N.

A fundamental difference between fasta and fastq files is the Quality scores contained in fastQ.

Quality scores are stored as ASCII characters representing -log<sub>10</sub> probability of base being wrong (Larger scores would be associated to more confident base calls).

A comprehensive description of phred quality can be found on the wiki page for FastQ.

To see the fastq encodings, we can run:

```
encoding(quality(fastq_WT_MOCK_REP1))
```

```
## ! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 :  
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
## ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T  
## 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51  
## U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m n  
## 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77  
## o p q r s t u v w x y z { | } ~  
## 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93
```

The ShortRead package has many functions available to allow us to collect useful metrics from our ShortRead object.

One very useful function is the `alphabetByCycle()` function which provides a quick method to summarise base occurrence of cycles.

Here we apply `alphabetByCycle()` function to the sequence information and show the occurrence of main 4 bases over first 15 cycles.

```
alph_by_cycle <- alphabetByCycle(sequence_of_reads)  
alph_by_cycle[1:4,1:15]
```

```
##          cycle  
## alphabet [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]  
##           A 21976 31733 40716 54511 67035 76250 52744 50858 52179 82996 65197  
##           C 80979 59940 67520 56801 42469 38578 34934 49605 44865 33367 47832  
##           G 91512 46565 44726 59238 53934 41211 34113 40354 43781 41308 50801  
##           T 28656 85309 70603 52948 60117 67525 101774 82748 82740 65894 59735  
##          cycle  
## alphabet [,12] [,13] [,14] [,15]  
##           A 56240 62322 63066 62605
```

```

##      C 50963 43951 43562 46231
##      G 45250 44735 43818 43027
##      T 71112 72557 73119 71702

```

We can use the `table` function to identify the number of times a sequence appears in our FastQ file's sequence reads.

```

readOccurrence <- table(sequence_of_reads)

# see the top 3 sequences that appear the highest number of times
sort(readOccurrence,decreasing = TRUE)[1:3]

## sequence_of_reads
## CTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
##                                         600
## CCCCCCCCCCTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
##                                         496
## CCCCCCCCCTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
##                                         392

```

We can identify duplicated reads (potentially arising from PCR over amplification) by using the `srduplicated()` function and the `ShortReadQ` object.

This returns a logical vector identifying which reads' sequences are duplicates (occur more than once in file). Note that the first time a sequence appears in file is not a duplicate but the second, third, fourth times etc are.

```

duplicates <- srduplicated(fastq_WT_MOCK_REP1)
duplicates[1:3]

## [1] FALSE FALSE FALSE

# we can use table() to get a quick summary of the seq duplication rate
table(duplicates)

## duplicates
## FALSE    TRUE
## 140931  82634

```

The `ShortRead` package also contains a function to generate a simple quality control report.

The `qa()` function accepts a FastQ file and returns a `FastqQA` object.

```
qa_WT_MOCK_REP1 <- qa(path_fastq_WT_MOCK_REP1)
qa_WT_MOCK_REP1
```

```
## class: FastqQA(10)
## QA elements (access with qa[["elt"]]):
##   readCounts: data.frame(1 3)
##   baseCalls: data.frame(1 5)
##   readQualityScore: data.frame(512 4)
##   baseQuality: data.frame(95 3)
##   alignQuality: data.frame(1 3)
##   frequentSequences: data.frame(50 4)
##   sequenceDistribution: data.frame(76 4)
##   perCycle: list(2)
##     baseCall: data.frame(231 4)
##     quality: data.frame(322 5)
##   perTile: list(2)
##     readCounts: data.frame(0 4)
##     medianReadQualityScore: data.frame(0 4)
##   adapterContamination: data.frame(1 1)
```

We can then use the report() function to generate a simple report.

```
myReport_WT_MOCK_REP1 <- report(qa_WT_MOCK_REP1)
myReport_WT_MOCK_REP1
```

```
## [1] "/var/folders/y1/f7wyg4vj50dgrg8drv0bn4nc0000gn/T//RtmpECN8o0/file245314480129/
```

Finally we can review the report in a browser or use the browseURL function to open it in a browser from R.

```
browseURL(myReport_WT_MOCK_REP1)
```

### 3.5 Trimming

When we observe low quality at the end of reads we may wish to remove the low quality bases for later alignment to the genome. The `trimTails()` function trims reads from the 3', removing bases which fall below a desired quality. The `trimTails()` function accepts arguments specifying the `ShortReadQ` object, the minimum number of successive bases required to be below quality cut-off for trimming and the actual cut-off score.

```

trimmed_fastq_WT_MOCK_REP1 <- trimTails(fastq_WT_MOCK_REP1, # ShortReadQ object to trim
                                         k=10, # integer number of failing letters to trigger trim
                                         a="5") # character giving letter at or below to "fail"
trimmed_fastq_WT_MOCK_REP1

## class: ShortReadQ
## length: 223565 reads; width: 16..50 cycles

```

Now we have trimmed our FastQ reads, we can export these reads for further analysis using the writeFastq() function

```

writeFastq(trimmed_fastq_WT_MOCK_REP1,
           "~/Desktop/Genomic_Data_Analysis/WT_MOCK_REP1_shortread_trimmed.fastq.gz") #path to save file

```

### 3.5.1 Automate for list of files

There are several utility programs that will provide you with QC and trim your data for you, with less input from you. We like fastp as it does some basic QC and trims your fastq files, and it does it very quickly. To make this available in R, it has been made available in the Bioconductor package Rfastp.

By default, fastp will make a html report to summarize your result. But the Rfastp wrapper allows you to look at some of them in R.

```

# create a directory for the output to go into if not already present
output_dir <- paste0(dirname(dirname(path_fastq_WT_MOCK_REP1)), "/Trimmed_rfastp")
if (!dir.exists(output_dir)) {dir.create(output_dir, recursive = TRUE)}

# if we wanted to just run a single file, we would do so like this:
rfastp_report <- rfastp(read1 = path_fastq_WT_MOCK_REP1,
                         outputFastq = paste0(output_dir, "/YPS606_WT_MOCK_REP1"))

# print out the qc summary for this sample
df_summary <- qcSummary(rfastp_report)
df_summary |> print.data.frame()

##               Before_QC      After_QC
## total_reads    2.235650e+05  2.235390e+05
## total_bases   1.117825e+07  1.117695e+07
## q20_bases     1.110373e+07  1.110280e+07
## q30_bases     1.096271e+07  1.096189e+07
## q20_rate       9.933340e-01  9.933660e-01
## q30_rate       9.807180e-01  9.807590e-01
## read1_mean_length 5.000000e+01  5.000000e+01
## gc_content     4.165340e-01  4.165420e-01

```

## 3.6 Batch file processing

That's nice, but we rarely just have a single fastq file, and we'd like to look at them all at once. Luckily, we can do that with rfastp.

First, we need to get the locations of all of the files we downloaded earlier

```
# adjust to the path where you assigned in RAW_DATA_DIR if using different than default
fq_file_dir <- dirname(path_fastq_WT_MOCK_REP1) # this just gets the path file is in.
# crate a list of all of the files
fastq.files <- list.files(path = fq_file_dir, # where to look
                           pattern = "REP[0-9].fastq.gz$", # the pattern of file name to look for
                           # Note, if you have other fastq files in the folder,
                           full.names = TRUE) # save the full path to the file

print(fastq.files)

## [1] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_MSN24_ETOH_REP1.fastq.gz"
## [2] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_MSN24_ETOH_REP2.fastq.gz"
## [3] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_MSN24_ETOH_REP3.fastq.gz"
## [4] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_MSN24_ETOH_REP4.fastq.gz"
## [5] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_MSN24_MOCK_REP1.fastq.gz"
## [6] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_MSN24_MOCK_REP2.fastq.gz"
## [7] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_MSN24_MOCK_REP3.fastq.gz"
## [8] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_MSN24_MOCK_REP4.fastq.gz"
## [9] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_WT_ETOH_REP1.fastq.gz"
## [10] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_WT_ETOH_REP2.fastq.gz"
## [11] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_WT_ETOH_REP3.fastq.gz"
## [12] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_WT_ETOH_REP4.fastq.gz"
## [13] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_WT_MOCK_REP1.fastq.gz"
## [14] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_WT_MOCK_REP2.fastq.gz"
## [15] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_WT_MOCK_REP3.fastq.gz"
## [16] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Raw/YPS606_WT_MOCK_REP4.fastq.gz"
```

Now we have all of the file paths

We can loop through all of the files to perform filtering and trimming. Note there are many arguments that can be modified. Use ?rfastp to learn more.

```
# run rfastp on all fastq files
for (i in 1:length(fastq.files)) {
  # file path to single end read
  read1 <- fastq.files[i]
  # assign output file (putting it inside of Data/Trimmed folder)
  output_name <- paste0(output_dir,
```

```

        "/",
        basename(fastq.files[i]))
json_report <- rfastp(
  read1 = read1,
  outputFastq = str_split(output_name, fixed("."))[1][1],
  disableTrimPolyG = FALSE,
  # cutLowQualFront = TRUE,
  # cutFrontWindowSize = 3,
  # cutFrontMeanQual = 10,
  # cutLowQualTail = TRUE,
  cutTailWindowSize = 1,
  # cutTailMeanQual = 5,
  minReadLength = 15,
  # trimFrontRead1 = 10,
  # adapterSequenceRead1 = 'GTGTCAGTCACTTCCAGCGG'
)
# Print the output file link in the R Markdown document
cat(paste0(
  "[Processing Complete - ",
  basename(output_name),
  "] (",
  output_name,
  ")\\n\\n"
))
}

## [Processing Complete - YPS606_MSN24_ETOH REP1.fastq.gz] (/Users/clstacy/Desktop/Genomic_Data_Analyst/MSN24/ETOH/REP1.fastq.gz)
## [Processing Complete - YPS606_MSN24_ETOH REP2.fastq.gz] (/Users/clstacy/Desktop/Genomic_Data_Analyst/MSN24/ETOH/REP2.fastq.gz)
## [Processing Complete - YPS606_MSN24_ETOH REP3.fastq.gz] (/Users/clstacy/Desktop/Genomic_Data_Analyst/MSN24/ETOH/REP3.fastq.gz)
## [Processing Complete - YPS606_MSN24_ETOH REP4.fastq.gz] (/Users/clstacy/Desktop/Genomic_Data_Analyst/MSN24/ETOH/REP4.fastq.gz)
## [Processing Complete - YPS606_MSN24 MOCK REP1.fastq.gz] (/Users/clstacy/Desktop/Genomic_Data_Analyst/MSN24/MOCK/REP1.fastq.gz)
## [Processing Complete - YPS606_MSN24 MOCK REP2.fastq.gz] (/Users/clstacy/Desktop/Genomic_Data_Analyst/MSN24/MOCK/REP2.fastq.gz)
## [Processing Complete - YPS606_MSN24 MOCK REP3.fastq.gz] (/Users/clstacy/Desktop/Genomic_Data_Analyst/MSN24/MOCK/REP3.fastq.gz)
## [Processing Complete - YPS606_MSN24 MOCK REP4.fastq.gz] (/Users/clstacy/Desktop/Genomic_Data_Analyst/MSN24/MOCK/REP4.fastq.gz)
## [Processing Complete - YPS606_WT ETOH REP1.fastq.gz] (/Users/clstacy/Desktop/Genomic_Data_Analyst/WT/ETOH/REP1.fastq.gz)
## 
```

```
## [Processing Complete - YPS606_WT_ETOH REP2.fastq.gz] (/Users/clstacy/Desktop/Genomic...
## 
## [Processing Complete - YPS606_WT_ETOH REP3.fastq.gz] (/Users/clstacy/Desktop/Genomic...
## 
## [Processing Complete - YPS606_WT_ETOH REP4.fastq.gz] (/Users/clstacy/Desktop/Genomic...
## 
## [Processing Complete - YPS606_WT_MOCK REP1.fastq.gz] (/Users/clstacy/Desktop/Genomic...
## 
## [Processing Complete - YPS606_WT_MOCK REP2.fastq.gz] (/Users/clstacy/Desktop/Genomic...
## 
## [Processing Complete - YPS606_WT_MOCK REP3.fastq.gz] (/Users/clstacy/Desktop/Genomic...
## 
## [Processing Complete - YPS606_WT_MOCK REP4.fastq.gz] (/Users/clstacy/Desktop/Genomic...
```

### 3.6.1 Running RfastP creates several files:

1. XXX\_R1.fastq.gz - FASTQ with poor quality reads filtered out
2. XXX.html - HTML file contains a QC report
3. XXX.json - JSON file with all the summary statistics

## 3.7 QC and adapters

Another common tool for quality control is called FastQC, useable via command line or GUI, available at <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/> or via pip or conda install in the command line.

To use this tool, let's get conda running on your computer. NOTE: Anaconda is already installed on computers in the computer lab. If you are using your own computer, you'll need to have conda installed (link to learn more)

First, we need to open a terminal window. We will copy code from the below code chunk into the terminal window.

```
. /opt/anaconda3/bin/activate && conda init
#. /opt/anaconda3/bin/activate && conda activate /opt/anaconda3;

# run this command in terminal to make sure conda is activated
which conda

# copy these 4 lines into terminal and run them
conda config --add channels defaults
conda config --append channels bioconda
conda config --append channels conda-forge
```

```
conda config --set channel_priority strict

# check channel order
conda config --show channels
```

Now, we need to create a conda environment with our packages. You can do so with the code below. This may take a couple of minutes the first time we run it.

```
# create an enviornment for our QC packages
if conda info --envs | grep -q QC; then echo "environment 'QC' already exists"; else conda create

# see available conda environments
conda env list

# activate our QC environment
conda activate QC

# make sure desired packages are working
which fastqc
which multiqc

# get the versions of each software
fastqc -v
multiqc --version

# it's always good coding practice to deactivate a conda environment at the end of a chunk
conda deactivate

## environment 'QC' already exists
## # conda environments:
## #
##          /Users/clstacy/Library/r-miniconda
##          /Users/clstacy/Library/r-miniconda-arm64
##          /Users/clstacy/Library/r-miniconda-arm64/envs/r-reticulate
##          /Users/clstacy/Library/r-miniconda/envs/r-reticulate
## base      * /Users/clstacy/anaconda3
## QC        /Users/clstacy/anaconda3/envs/QC
## mageck    /Users/clstacy/anaconda3/envs/mageck
## salmon    /Users/clstacy/anaconda3/envs/salmon
##          /Users/clstacy/opt/anaconda3
##          /Users/clstacy/opt/anaconda3/envs/colony_count_nm
##          /Users/clstacy/opt/anaconda3/envs/tlcc
##
## /Users/clstacy/anaconda3/envs/QC/bin/fastqc
```

```
## /Users/clstacy/anaconda3/envs/QC/bin/multiqc
## FastQC v0.12.1
## multiqc, version 1.15
```

## 3.8 Running fastqc

```
#WARNING: variables in bash you've saved in previous chunks won't be retained in later
# We need to set a variable for the folder above raw and trimmed files.
DATA_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/Data"
QC_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/QC"
# Activate conda QC environment
conda activate QC

# show which version of fastqc is active
fastqc -v

# Function to check if a command is installed, we use this next.
command_exists() {
    command -v "$1" >/dev/null 2>&1
}

if command_exists fastqc; then
    # Continue if fastqc is installed
    # first, make sure we have the folders to store the fastqc outputs
    mkdir -p $QC_DIR/fastqc/Raw
    mkdir -p $QC_DIR/fastqc/Trimmed
    # run fastqc on the raw data files
    fastqc $DATA_DIR/Raw/*.fastq.gz -o $QC_DIR/fastqc/Raw
    # run fastqc on the trimmed data files
    fastqc $DATA_DIR/Trimmed_rfastp/*.fastq.gz -o $QC_DIR/fastqc/Trimmed
    if [ $? -ne 0 ]; then
        echo "FastQC execution failed. It didn't work."
    fi
else
    echo "FastQC is not installed."
fi

# deactivate QC conda environment
conda deactivate

## FastQC v0.12.1
## application/octet-stream
## application/octet-stream
```

```
## application/octet-stream
## Started analysis of YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 5% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 10% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 15% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 20% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 25% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 30% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 35% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 40% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 45% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 50% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 55% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 60% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 65% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 70% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 75% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 80% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 85% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 90% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Approx 95% complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Analysis complete for YPS606_MSN24_ETOH_REP1.fastq.gz
## Warning: the fonts "Times" and "Times" are not available for the Java logical font "Serif", wh
## Started analysis of YPS606_MSN24_ETOH_REP2.fastq.gz
## Approx 5% complete for YPS606_MSN24_ETOH_REP2.fastq.gz
## Approx 10% complete for YPS606_MSN24_ETOH_REP2.fastq.gz
## Approx 15% complete for YPS606_MSN24_ETOH_REP2.fastq.gz
## Approx 20% complete for YPS606_MSN24_ETOH_REP2.fastq.gz
## Approx 25% complete for YPS606_MSN24_ETOH_REP2.fastq.gz
## Approx 30% complete for YPS606_MSN24_ETOH_REP2.fastq.gz
## Approx 35% complete for YPS606_MSN24_ETOH_REP2.fastq.gz
## Approx 40% complete for YPS606_MSN24_ETOH_REP2.fastq.gz
## Approx 45% complete for YPS606_MSN24_ETOH_REP2.fastq.gz
```

```
## Approx 50% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Approx 55% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Approx 60% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Approx 65% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Approx 70% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Approx 75% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Approx 80% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Approx 85% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Approx 90% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Approx 95% complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Analysis complete for YPS606_MSN24_ETOH REP2.fastq.gz
## Started analysis of YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 5% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 10% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 15% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 20% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 25% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 30% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 35% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 40% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 45% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 50% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 55% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 60% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 65% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 70% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 75% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 80% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 85% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 90% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Approx 95% complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Analysis complete for YPS606_MSN24_ETOH REP3.fastq.gz
## Started analysis of YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 5% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 10% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 15% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 20% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 25% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 30% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 35% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 40% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 45% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 50% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 55% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 60% complete for YPS606_MSN24_ETOH REP4.fastq.gz
## Approx 65% complete for YPS606_MSN24_ETOH REP4.fastq.gz
```

```
## Approx 70% complete for YPS606_MSN24_ETOH_REP4.fastq.gz
## Approx 75% complete for YPS606_MSN24_ETOH_REP4.fastq.gz
## Approx 80% complete for YPS606_MSN24_ETOH_REP4.fastq.gz
## Approx 85% complete for YPS606_MSN24_ETOH_REP4.fastq.gz
## Approx 90% complete for YPS606_MSN24_ETOH_REP4.fastq.gz
## Approx 95% complete for YPS606_MSN24_ETOH_REP4.fastq.gz
## Analysis complete for YPS606_MSN24_ETOH_REP4.fastq.gz
## Started analysis of YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 5% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 10% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 15% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 20% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 25% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 30% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 35% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 40% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 45% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 50% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 55% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 60% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 65% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 70% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 75% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 80% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 85% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 90% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Approx 95% complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Analysis complete for YPS606_MSN24_MOCK_REP1.fastq.gz
## Started analysis of YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 5% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 10% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 15% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 20% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 25% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 30% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 35% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 40% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 45% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 50% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 55% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 60% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 65% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 70% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 75% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 80% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
## Approx 85% complete for YPS606_MSN24_MOCK_REP2.fastq.gz
```

50CHAPTER 3. WORKING WITH SEQUENCES: RAW DATA & QUALITY CONTROL

```
## Approx 90% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 95% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Analysis complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Started analysis of YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 5% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 10% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 15% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 20% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 25% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 30% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 35% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 40% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 45% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 50% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 55% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 60% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 65% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 70% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 75% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 80% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 85% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 90% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 95% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Approx 100% complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Analysis complete for YPS606_MSN24_MOCK_REPO.fastq.gz
## Started analysis of YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 5% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 10% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 15% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 20% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 25% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 30% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 35% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 40% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 45% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 50% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 55% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 60% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 65% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 70% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 75% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 80% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 85% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 90% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Approx 95% complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
## Analysis complete for YPS606_MSN24_MOCK_REPO4.fastq.gz
```

```
## Started analysis of YPS606_WT_ETOH REP1.fastq.gz
## Approx 5% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 10% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 15% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 20% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 25% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 30% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 35% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 40% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 45% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 50% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 55% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 60% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 65% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 70% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 75% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 80% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 85% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 90% complete for YPS606_WT_ETOH REP1.fastq.gz
## Approx 95% complete for YPS606_WT_ETOH REP1.fastq.gz
## Analysis complete for YPS606_WT_ETOH REP1.fastq.gz
## Started analysis of YPS606_WT_ETOH REP2.fastq.gz
## Approx 5% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 10% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 15% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 20% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 25% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 30% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 35% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 40% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 45% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 50% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 55% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 60% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 65% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 70% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 75% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 80% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 85% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 90% complete for YPS606_WT_ETOH REP2.fastq.gz
## Approx 95% complete for YPS606_WT_ETOH REP2.fastq.gz
## Analysis complete for YPS606_WT_ETOH REP2.fastq.gz
## Started analysis of YPS606_WT_ETOH REP3.fastq.gz
## Approx 5% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 10% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 15% complete for YPS606_WT_ETOH REP3.fastq.gz
```

## 52CHAPTER 3. WORKING WITH SEQUENCES: RAW DATA &amp; QUALITY CONTROL

```
## Approx 20% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 25% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 30% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 35% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 40% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 45% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 50% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 55% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 60% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 65% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 70% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 75% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 80% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 85% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 90% complete for YPS606_WT_ETOH REP3.fastq.gz
## Approx 95% complete for YPS606_WT_ETOH REP3.fastq.gz
## Analysis complete for YPS606_WT_ETOH REP3.fastq.gz
## Started analysis of YPS606_WT_ETOH REP4.fastq.gz
## Approx 5% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 10% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 15% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 20% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 25% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 30% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 35% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 40% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 45% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 50% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 55% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 60% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 65% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 70% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 75% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 80% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 85% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 90% complete for YPS606_WT_ETOH REP4.fastq.gz
## Approx 95% complete for YPS606_WT_ETOH REP4.fastq.gz
## Analysis complete for YPS606_WT_ETOH REP4.fastq.gz
## Started analysis of YPS606_WT_MOCK REP1.fastq.gz
## Approx 5% complete for YPS606_WT_MOCK REP1.fastq.gz
## Approx 10% complete for YPS606_WT_MOCK REP1.fastq.gz
## Approx 15% complete for YPS606_WT_MOCK REP1.fastq.gz
## Approx 20% complete for YPS606_WT_MOCK REP1.fastq.gz
## Approx 25% complete for YPS606_WT_MOCK REP1.fastq.gz
## Approx 30% complete for YPS606_WT_MOCK REP1.fastq.gz
## Approx 35% complete for YPS606_WT_MOCK REP1.fastq.gz
```

```
## Approx 40% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 45% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 50% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 55% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 60% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 65% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 70% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 75% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 80% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 85% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 90% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Approx 95% complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Analysis complete for YPS606_WT_MOCK_REPO1.fastq.gz
## Started analysis of YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 5% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 10% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 15% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 20% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 25% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 30% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 35% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 40% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 45% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 50% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 55% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 60% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 65% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 70% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 75% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 80% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 85% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 90% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Approx 95% complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Analysis complete for YPS606_WT_MOCK_REPO2.fastq.gz
## Started analysis of YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 5% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 10% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 15% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 20% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 25% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 30% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 35% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 40% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 45% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 50% complete for YPS606_WT_MOCK_REPO3.fastq.gz
## Approx 55% complete for YPS606_WT_MOCK_REPO3.fastq.gz
```



```

## application/octet-stream
## Approx 5% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 10% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 15% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 20% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 25% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 30% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 35% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 40% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 45% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 50% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 55% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 60% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 65% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 70% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 75% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 80% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 85% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 90% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Approx 95% complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Analysis complete for YPS606_MSN24_ETOH REP1_R1.fastq.gz
## Warning: the fonts "Times" and "Times" are not available for the Java logical font "Serif", wh
## Started analysis of YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 5% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 10% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 15% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 20% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 25% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 30% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 35% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 40% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 45% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 50% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 55% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 60% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 65% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 70% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 75% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 80% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 85% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 90% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Approx 95% complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Analysis complete for YPS606_MSN24_ETOH REP2_R1.fastq.gz
## Started analysis of YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 5% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 10% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz

```

## 56 CHAPTER 3. WORKING WITH SEQUENCES: RAW DATA &amp; QUALITY CONTROL

```
## Approx 15% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 20% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 25% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 30% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 35% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 40% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 45% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 50% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 55% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 60% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 65% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 70% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 75% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 80% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 85% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 90% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Approx 95% complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Analysis complete for YPS606_MSN24_ETOH REP3_R1.fastq.gz
## Started analysis of YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 5% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 10% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 15% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 20% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 25% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 30% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 35% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 40% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 45% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 50% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 55% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 60% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 65% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 70% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 75% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 80% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 85% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 90% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Approx 95% complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Analysis complete for YPS606_MSN24_ETOH REP4_R1.fastq.gz
## Started analysis of YPS606_MSN24_MOCK REP1_R1.fastq.gz
## Approx 5% complete for YPS606_MSN24_MOCK REP1_R1.fastq.gz
## Approx 10% complete for YPS606_MSN24_MOCK REP1_R1.fastq.gz
## Approx 15% complete for YPS606_MSN24_MOCK REP1_R1.fastq.gz
## Approx 20% complete for YPS606_MSN24_MOCK REP1_R1.fastq.gz
## Approx 25% complete for YPS606_MSN24_MOCK REP1_R1.fastq.gz
## Approx 30% complete for YPS606_MSN24_MOCK REP1_R1.fastq.gz
```



## 58 CHAPTER 3. WORKING WITH SEQUENCES: RAW DATA &amp; QUALITY CONTROL

```
## Approx 55% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Approx 60% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Approx 65% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Approx 70% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Approx 75% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Approx 80% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Approx 85% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Approx 90% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Approx 95% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Approx 100% complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Analysis complete for YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Started analysis of YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 5% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 10% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 15% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 20% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 25% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 30% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 35% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 40% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 45% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 50% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 55% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 60% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 65% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 70% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 75% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 80% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 85% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 90% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Approx 95% complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Analysis complete for YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Started analysis of YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 5% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 10% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 15% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 20% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 25% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 30% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 35% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 40% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 45% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 50% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 55% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 60% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
## Approx 65% complete for YPS606_WT_ETOH_REP1_R1.fastq.gz
```



## 60CHAPTER 3. WORKING WITH SEQUENCES: RAW DATA &amp; QUALITY CONTROL

```
## Approx 90% complete for YPS606_WT_ETOH REP3_R1.fastq.gz
## Approx 95% complete for YPS606_WT_ETOH REP3_R1.fastq.gz
## Analysis complete for YPS606_WT_ETOH REP3_R1.fastq.gz
## Started analysis of YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 5% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 10% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 15% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 20% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 25% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 30% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 35% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 40% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 45% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 50% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 55% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 60% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 65% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 70% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 75% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 80% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 85% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 90% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Approx 95% complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Analysis complete for YPS606_WT_ETOH REP4_R1.fastq.gz
## Started analysis of YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 5% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 10% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 15% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 20% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 25% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 30% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 35% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 40% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 45% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 50% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 55% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 60% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 65% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 70% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 75% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 80% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 85% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 90% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Approx 95% complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Analysis complete for YPS606_WT_MOCK REP1_R1.fastq.gz
## Started analysis of YPS606_WT_MOCK REP2_R1.fastq.gz
```

```
## Approx 5% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 10% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 15% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 20% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 25% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 30% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 35% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 40% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 45% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 50% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 55% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 60% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 65% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 70% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 75% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 80% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 85% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 90% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Approx 95% complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Analysis complete for YPS606_WT_MOCK_REPO_R1.fastq.gz
## Started analysis of YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 5% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 10% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 15% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 20% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 25% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 30% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 35% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 40% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 45% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 50% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 55% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 60% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 65% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 70% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 75% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 80% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 85% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 90% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Approx 95% complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Analysis complete for YPS606_WT_MOCK_REPO3_R1.fastq.gz
## Started analysis of YPS606_WT_MOCK_REPO4_R1.fastq.gz
## Approx 5% complete for YPS606_WT_MOCK_REPO4_R1.fastq.gz
## Approx 10% complete for YPS606_WT_MOCK_REPO4_R1.fastq.gz
## Approx 15% complete for YPS606_WT_MOCK_REPO4_R1.fastq.gz
## Approx 20% complete for YPS606_WT_MOCK_REPO4_R1.fastq.gz
```

```
## Approx 25% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 30% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 35% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 40% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 45% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 50% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 55% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 60% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 65% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 70% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 75% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 80% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 85% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 90% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Approx 95% complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
## Analysis complete for YPS606_WT_MOCK_REP4_R1.fastq.gz
```

This link shows the fastqc output for the trimmed WT\_MOCK\_REP1.fastq.gz

```
browseURL("~/Desktop/Genomic_Data_Analysis/QC/fastqc/Trimmed/YPS606_WT_MOCK_REP1_R1_fa
```

We could do this for each of the html fastq files to see how they all look but with a large sample size that takes a long time and can lead to missing important information.

### 3.9 Multiqc for QC on mutliple samples

One of our favorite ways to analyze multiple samples simultaneously is MultiQC a software that combines fastQC (and other) reports

Here is the code to run it:

```
# Be sure to change this file path to the path you want to run multiqc
QC_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/QC"

# activate QC environment
conda activate QC

# run multiqc on all of the fastqc outputs
multiqc $QC_DIR/fastqc -o $QC_DIR -m fastqc -f

## 
##   /// MultiQC  | v1.15
```

```

##          multiqc | MultiQC Version v1.16 now available!
## |          multiqc | Only using modules: fastqc
## |          multiqc | Search path : /Users/clstacy/Desktop/Genomic_Data_Analysis/QC/fastqc
## | searching |                               100% 66/66
## |          fastqc | Found 32 reports
## |          multiqc | Report      : ../../Desktop/Genomic_Data_Analysis/QC/multiqc_report.html
## |          multiqc | Data       : ../../Desktop/Genomic_Data_Analysis/QC/multiqc_data
## |          multiqc | MultiQC complete

path_multiqc <- "~/Desktop/Genomic_Data_Analysis/QC/multiqc_report.html"
browseURL(path_multiqc)

```

Be sure to knit this file into a pdf or html file once you're finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

**R version 4.3.1 (2023-06-16)**

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8|en\_US.UTF-8||C||en\_US.UTF-8|en\_US.UTF-8

**attached base packages:** stats4, stats, graphics, grDevices, utils, datasets, methods and base

**other attached packages:** ShortRead(v.1.58.0), GenomicAlignments(v.1.36.0), SummarizedExperiment(v.1.30.2), MatrixGenerics(v.1.12.3), matrixStats(v.1.0.0), Rsamtools(v.2.16.0), GenomicRanges(v.1.52.1), Biostrings(v.2.68.1), GenomeInfoDb(v.1.36.4), XVector(v.0.40.0), BiocParallel(v.1.34.2), Rfasttp(v.1.10.0), org.Sc.sgd.db(v.3.17.0), AnnotationDbi(v.1.62.2), IRanges(v.2.34.1), S4Vectors(v.0.38.2), Biobase(v.2.60.0), BiocGenerics(v.0.46.0), clusterProfiler(v.4.8.2), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pander(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3), forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyverse(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)

**loaded via a namespace (and not attached):** RColorBrewer(v.1.1-3), rstudioapi(v.0.15.0), jsonlite(v.1.8.7), magrittr(v.2.0.3), farver(v.2.1.1), rmarkdown(v.2.25), ragg(v.1.2.6), fs(v.1.6.3), zlibbioc(v.1.46.0), vctrs(v.0.6.4), memoise(v.2.0.1), RCurl(v.1.98-1.12), ggtree(v.3.8.2), S4Arrays(v.1.0.6), htmltools(v.0.5.6.1), curl(v.5.1.0), gridGraphics(v.0.5-1), KernSmooth(v.2.23-22), plyr(v.1.8.9), cachem(v.1.0.8), lifecycle(v.1.0.3), pkgconfig(v.2.0.3),

*Matrix(v.1.6-1.1), R6(v.2.5.1), fastmap(v.1.1.1), gson(v.0.1.0), Genome-InfoDbData(v.1.2.10), snakecase(v.0.11.1), digest(v.0.6.33), aplot(v.0.2.2), enrichplot(v.1.20.0), colorspace(v.2.1-0), patchwork(v.1.1.3), rprojroot(v.2.0.3), textshaping(v.0.3.7), RSQLite(v.2.3.1), hwriter(v.1.3.2.1), labeling(v.0.4.3), fansi(v.1.0.5), timechange(v.0.2.0), abind(v.1.4-5), httr(v.1.4.7), polyclip(v.1.10-6), compiler(v.4.3.1), proxy(v.0.4-27), bit64(v.4.0.5), withr(v.2.5.1), downloader(v.0.4), viridis(v.0.6.4), DBI(v.1.1.3), ggforce(v.0.4.1), MASS(v.7.3-60), DelayedArray(v.0.26.7), rjson(v.0.2.21), classInt(v.0.4-10), HDO.db(v.0.99.1), units(v.0.8-4), tools(v.4.3.1), ape(v.5.7-1), scatterpie(v.0.2.1), glue(v.1.6.2), nlme(v.3.1-163), GOSemSim(v.2.26.1), sf(v.1.0-14), grid(v.4.3.1), shadowtext(v.0.1.2), reshape2(v.1.4.4), fgsea(v.1.26.0), generics(v.0.1.3), gtable(v.0.3.4), tzdb(v.0.4.0), class(v.7.3-22), data.table(v.1.14.8), hms(v.1.1.3), tidygraph(v.1.2.3), utf8(v.1.2.3), ggrepel(v.0.9.4), pillar(v.1.9.0), yulab.utils(v.0.1.0), vroom(v.1.6.4), splines(v.4.3.1), tweenr(v.2.0.2), treeio(v.1.24.3), lattice(v.0.21-9), deldir(v.1.0-9), bit(v.4.0.5), tidyselect(v.1.2.0), GO.db(v.3.17.0), gridExtra(v.2.3), bookdown(v.0.36), xfun(v.0.40), graphlayouts(v.1.0.1), stringi(v.1.7.12), lazyeval(v.0.2.2), ggrepel(v.0.1.3), yaml(v.2.3.7), evaluate(v.0.22), codetools(v.0.2-19), interp(v.1.1-4), ggraph(v.2.1.0), qvalue(v.2.32.0), RVenn(v.1.1.0), ggplotify(v.0.1.2), cli(v.3.6.1), systemfonts(v.1.0.5), munsell(v.0.5.0), Rcpp(v.1.0.11), png(v.0.1-8), parallel(v.4.3.1), blob(v.1.2.4), jpeg(v.0.1-10), latticeExtra(v.0.6-30), DOSE(v.3.26.1), bitops(v.1.0-7), viridisLite(v.0.4.2), e1071(v.1.7-13), scales(v.1.2.1), crayon(v.1.5.2), rlang(v.1.1.1), cowplot(v.1.1.1), fastmatch(v.1.1-4) and KEGGREST(v.1.40.1)*

# Chapter 4

## Read Mapping

last updated: 2023-10-27

As usual, make sure we have the right packages for this exercise

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# let's load all of the files we were using and want to have again today
p_load("tidyverse", "knitr", "readr",
       "pander", "BiocManager",
       "dplyr", "stringr")

# We also need the Bioconductor packages "Rsubread" for today's activity.
p_load("Rsubread")
```

Previously, we filtered and trimmed our raw fastq files. They should be in the folder below, unless you chose a different place to store them.

```
dir_trimmed.fq_files <- "~/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfasp"

trimmed_fastq_files <- list.files(path = dir_trimmed.fq_files,
                                    pattern = ".fastq.gz$",
                                    full.names = TRUE)
trimmed_fastq_files

## [1] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfasp/YP$606_MSN24_ETOH_REP1"
## [2] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfasp/YP$606_MSN24_ETOH_REP2"
## [3] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfasp/YP$606_MSN24_ETOH_REP3"
## [4] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfasp/YP$606_MSN24_ETOH_REP4"
## [5] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfasp/YP$606_MSN24_MOCK_REP1"
```

```
## [6] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_MSN24"
## [7] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_MSN24"
## [8] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_MSN24"
## [9] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_WT_ETC"
## [10] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_WT_ETC"
## [11] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_WT_ETC"
## [12] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_WT_ETC"
## [13] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_WT_MOC"
## [14] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_WT_MOC"
## [15] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_WT_MOC"
## [16] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP5606_WT_MOC"
```

You should see the full paths to all 16 trimmed fastq files that we will be mapping to the reference genome today.

## 4.1 Alignment

Read sequences are stored in compressed (gzipped) FASTQ files. Before the differential expression analysis can proceed, these reads must be aligned to the yeast genome and counted into annotated genes. This can be achieved with functions in the Rsubread package.

## 4.2 Retrieve the genome

We will use a bash code chunk to download the latest genome

```
# Define the destination file path
# You can change this file path to the path you want your data to go, or leave it.
REF_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/Reference"

# make that directory if it doesn't already
mkdir -p $REF_DIR

# Define the URL of reference genome
# (latest from ensembl)
url="ftp://ftp.ensembl.org/pub/release-110/fasta/saccharomyces_cerevisiae/dna/Saccharo

# Check if the file already exists at the destination location
if [ ! -f "$REF_DIR/Saccharomyces_cerevisiae.R64-1-1.dna.toplevel.fa.gz" ]; then
    echo "Reference genome not found, downloading..."
    # If the file does not exist, download it using curl
```

```

curl -o "$REF_DIR/Saccharomyces_cerevisiae.R64-1-1.dna.toplevel.fa.gz" "$url"
echo "Downloading finished"
else
    echo "File already exists at $REF_DIR Skipping download."
fi

## File already exists at /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference Skipping download

```

### 4.3 Build Rsubread Index

The first step in performing the alignment is to build an index. In order to build an index you need to have the fasta file (.fa), which can be downloaded from the UCSC genome browser. This may take several minutes to run. Building the full index using the whole genome usually takes about 30 minutes to an hr on a server for larger Eukaryotic genomes. Because yeast has a relatively small genome size, we are able to build the full index in class.

```

library(Rsubread)

# Set path of the reference fasta file
reference_genome = path.expand("~/Desktop/Genomic_Data_Analysis/Reference/Saccharomyces_cerevisiae.R64-1-1.dna.toplevel.fa")

index_reference_genome = path.expand("~/Desktop/Genomic_Data_Analysis/Reference/index_rsubread_Saccharomyces_cerevisiae.R64-1-1.dna.toplevel.fa"

# build the index
buildindex(basename=index_reference_genome, reference=reference_genome)

## =====
##      / _ _ _ | | | | _ \ | _ \ | _ \ | _ \ | _ \ | _ \ | _ \ | _ \
##      | ( _ _ | | | | | | | | | | | | | | | | | | | | | | | | |
##      \_ _ \ | | | | | | | | | | | | | | | | | | | | | | | | |
##      _ _ ) | | | | | | | | | | | | | | | | | | | | | | | | |
##      ===== | _ _ / \_ _ / | _ _ / | _ _ \ \_ _ / | _ _ \ \_ _ / | _ _ \
##      Rsubread 2.14.2
##
## //===== setting =====\\
## ||
## ||          Index name : index_rsubread_Saccharomyces_cerevisiae.R6 ...
## ||          Index space : base space
## ||          Index split : no-split
## ||          Repeat threshold : 100 repeats
## ||          Gapped index : no

```

```

## ||
## |       Free / total memory : 1.4GB / 8.0GB
## |
## |       Input files : 1 file in total
## |           o Saccharomyces_cerevisiae.R64-1-1.dna.toplevel.fa.gz
## |
## |
## |       WARNING: the free memory is lower than 3.0GB.
## |           the program may run very slow or crash.
## |
## \=====//\=====
## |
## //===== Running =====\\
## |
## |   Check the integrity of provided reference sequences ...
## |   No format issues were found
## |   Scan uninformative subreads in reference sequences ...
## |   11 uninformative subreads were found.
## |   These subreads were excluded from index building.
## |   Estimate the index size...
## |       8%,    0 mins elapsed, rate=8894.2k bps/s
## |       16%,   0 mins elapsed, rate=10182.5k bps/s
## |       24%,   0 mins elapsed, rate=10698.6k bps/s
## |       33%,   0 mins elapsed, rate=10958.8k bps/s
## |       41%,   0 mins elapsed, rate=11119.5k bps/s
## |       49%,   0 mins elapsed, rate=11242.2k bps/s
## |       58%,   0 mins elapsed, rate=11327.4k bps/s
## |       66%,   0 mins elapsed, rate=11410.7k bps/s
## |       74%,   0 mins elapsed, rate=11472.0k bps/s
## |       83%,   0 mins elapsed, rate=11506.6k bps/s
## |       91%,   0 mins elapsed, rate=11531.9k bps/s
## |
## |       WARNING: available memory is lower than 3.0 GB.
## |           The program may run very slow.
## |   Build a gapped index and/or split index into blocks to reduce memory use.
## |
## |   Build the index...
## |       8%,    0 mins elapsed, rate=189.7k bps/s
## |       16%,   0 mins elapsed, rate=216.9k bps/s
## |       24%,   0 mins elapsed, rate=204.5k bps/s
## |       33%,   0 mins elapsed, rate=199.3k bps/s
## |       41%,   0 mins elapsed, rate=204.1k bps/s
## |       49%,   0 mins elapsed, rate=204.2k bps/s
## |       58%,   0 mins elapsed, rate=203.3k bps/s
## |       66%,   0 mins elapsed, rate=205.8k bps/s
## |       74%,   0 mins elapsed, rate=205.3k bps/s

```

```

## || 83%, 0 mins elapsed, rate=205.8k bps/s
## || 91%, 0 mins elapsed, rate=205.9k bps/s
## || Save current index block...
## || [ 0.0% finished ]
## || [ 10.0% finished ]
## || [ 20.0% finished ]
## || [ 30.0% finished ]
## || [ 40.0% finished ]
## || [ 50.0% finished ]
## || [ 60.0% finished ]
## || [ 70.0% finished ]
## || [ 80.0% finished ]
## || [ 90.0% finished ]
## || [ 100.0% finished ]
## ||
## ||          Total running time: 1.6 minutes.
## ||Index /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_rsu ...
## ||
## \\=====

```

We can see the arguments available with the align function from the Rsubread package

```
args(align)
```

```

## function (index, readfile1, readfile2 = NULL, type = "rna", input_format = "gzFASTQ",
##   output_format = "BAM", output_file = paste(readfile1, "subread",
##     output_format, sep = "."),
##   phredOffset = 33, nsubreads = 10,
##   TH1 = 3, TH2 = 1, maxMismatches = 3, unique = FALSE, nBestLocations = 1,
##   indels = 5, complexIndels = FALSE, nTrim5 = 0, nTrim3 = 0,
##   minFragLength = 50, maxFragLength = 600, PE_orientation = "fr",
##   nthreads = 1, readGroupID = NULL, readGroup = NULL, keepReadOrder = FALSE,
##   sortReadsByCoordinates = FALSE, color2base = FALSE, DP_GapOpenPenalty = -1,
##   DP_GapExtPenalty = 0, DP_MismatchPenalty = 0, DP_MatchScore = 2,
##   detectSV = FALSE, useAnnotation = FALSE, annot.inbuilt = "mm39",
##   annot.ext = NULL, isGTF = FALSE, GTF.featureType = "exon",
##   GTF.attrType = "gene_id", chrAliases = NULL)
## NULL

```

This process takes some time to finish.

```
# run the alignment on all of the trimmed_fastq_files
align(index=index_reference_genome,
      readfile1=trimmed_fastq_files,
```

```

type = "rna",
input_format = "gzFASTQ",
output_format = "BAM",
unique = TRUE,
nBestLocations = 1,
sortReadsByCoordinates = TRUE,
nthreads=6
)

```

The output of the alignment are bam corresponding to each fastq file.

We can get a summary of the proportion of reads that mapped to the reference genome using the propmapped function.

```

# create an object in R listing
bam_files <- list.files(path = dir_trimmed.fq_files, pattern = ".BAM$", full.names = TRUE)
bam_files

## [1] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_MSN24"
## [2] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_MSN24"
## [3] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_MSN24"
## [4] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_MSN24"
## [5] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_MSN24"
## [6] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_MSN24"
## [7] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_MSN24"
## [8] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_MSN24"
## [9] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_WT_ETOH"
## [10] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_WT_ETOH"
## [11] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_WT_ETOH"
## [12] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_WT_ETOH"
## [13] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_WT_M001"
## [14] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_WT_M001"
## [15] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_WT_M001"
## [16] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_WT_M001"

# find the proportion of reads that mapped for each sample
props <- propmapped(files=bam_files)

props |> print()

##                                     NumTotal NumMapped PropMapped
## YPS606_MSN24_ETOH_REP1_R1.fastq.gz.subread.BAM    233278     14427   0.061845
## YPS606_MSN24_ETOH_REP2_R1.fastq.gz.subread.BAM        0         0      NaN
## YPS606_MSN24_ETOH_REP3_R1.fastq.gz.subread.BAM        0         0      NaN

```

```

## YPS606_MSN24_ETOH_REP4_R1.fastq.gz.subread.BAM 205792 178785 0.868766
## YPS606_MSN24_MOCK_REP1_R1.fastq.gz.subread.BAM 167075 143114 0.856585
## YPS606_MSN24_MOCK_REP2_R1.fastq.gz.subread.BAM 169754 146302 0.861847
## YPS606_MSN24_MOCK_REP3_R1.fastq.gz.subread.BAM 210001 178664 0.850777
## YPS606_MSN24_MOCK_REP4_R1.fastq.gz.subread.BAM 208329 177749 0.853213
## YPS606_WT_ETOH_REP1_R1.fastq.gz.subread.BAM 181587 159200 0.876715
## YPS606_WT_ETOH_REP2_R1.fastq.gz.subread.BAM 201551 176904 0.877713
## YPS606_WT_ETOH_REP3_R1.fastq.gz.subread.BAM 214745 188499 0.877781
## YPS606_WT_ETOH_REP4_R1.fastq.gz.subread.BAM 187319 164152 0.876323
## YPS606_WT_MOCK_REP1_R1.fastq.gz.subread.BAM 223539 193407 0.865205
## YPS606_WT_MOCK_REP2_R1.fastq.gz.subread.BAM 187469 161251 0.860148
## YPS606_WT_MOCK_REP3_R1.fastq.gz.subread.BAM 224767 192104 0.854681
## YPS606_WT_MOCK_REP4_R1.fastq.gz.subread.BAM 0 0 NaN

```

## 4.4 Pseudomapping with Salmon

Salmon is a widely used pseudomapper. It is not available to use in R, but we can use bash code chunks to run it in the same markdown document.

### 4.4.1 Create Conda Env

First, we need to create a new conda environment for salmon.

Depending on your computer, we might need to run this code in terminal.

```

## Warning, if you did not complete Working_with_Sequences.Rmd activity,
#      your conda might not be set up correctly for this code.

# create an environment for our pseudomapping with Salmon
# this code is "extra" because it only creates env if not already existing.
if conda info --envs | grep -q salmon; then echo "environment 'salmon' already exists"; else CONDA
# the channel priority order above is needed to get a recent version via conda.

# see available conda environments
conda env list

# activate our QC environment
conda activate salmon

# make sure desired packages are working
which salmon

# help page for using salmon

```

```
salmon -h

# it's always good coding practice to deactivate
# a conda environment at the end of a chunk
conda deactivate
```

#### 4.4.2 Download transcriptome

To make an index for Salmon, we need transcript sequences in the FASTA format.

```
# Define the destination file path
# Be sure to change this file path to the path you want your data to go
REF_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/Reference"

# make that directory if it doesn't already
mkdir -p $REF_DIR

# Define the URL of reference transcriptome
# (latest from ensembl)
url="ftp://ftp.ensembl.org/pub/release-110/fasta/saccharomyces_cerevisiae/cdna/Saccharomyces_cerevisiae.R64-1-1.cdna.all.fa.gz"

# Check if the file already exists at the destination location
if [ ! -f "$REF_DIR/Saccharomyces_cerevisiae.R64-1-1.cdna.all.fa.gz" ]; then
    echo "Reference transcriptome not found, downloading..."
    # If the file does not exist, download it using curl
    curl -o "$REF_DIR/Saccharomyces_cerevisiae.R64-1-1.cdna.all.fa.gz" "$url"
    echo "Downloading finished"
else
    echo "File already exists at $REF_DIR Skipping download."
fi

## File already exists at /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference Skipp
```

#### 4.4.3 Building the Salmon index

Salmon can index by using the command `salmon index`. A recent feature update to Salmon includes an option to map to decoys, we will use the entire genome as the decoy for our index, because the *S. cerevesiae* genome is small. You can read more at: <https://salmon.readthedocs.io/en/latest/salmon.html#preparing-transcriptome-indices-mapping-based-mode>.

```

# We need to set a variable for where the transcriptome file is
REF_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/Reference"
TRANSCRIPTOME="/Users/$USER/Desktop/Genomic_Data_Analysis/Reference/Saccharomyces_cerevisiae.R64-1-1.dna
GENOME="/Users/$USER/Desktop/Genomic_Data_Analysis/Reference/Saccharomyces_cerevisiae.R64-1-1.dna

# Activate conda salmon environment
conda activate salmon

# Run a script that generates a decoy.txt file from the genome we downloaded
grep "^.>" <(gunzip -c $GENOME) | cut -d " " -f 1 > $REF_DIR/decoys.txt
sed -i.bak -e 's/>/\//g' $REF_DIR/decoys.txt

# Combine the transcriptome and genome into a single file for indexing
cat $TRANSCRIPTOME $GENOME > $REF_DIR/gentrome.fasta.gz

# We will use the yeast, but it needs to be indexed by salmon
salmon index -t $REF_DIR/gentrome.fasta.gz -d $REF_DIR/decoys.txt -p 4 -i $REF_DIR/index_salmon_Saccharomyces_cerevisiae.R64-1-1.dna

conda deactivate

## Version Info: This is the most recent version of salmon.
## [2023-10-26 12:18:15.379] [jLog] [info] building index
## out : /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Saccharomyces_cerevisiae.R64-1-1.dna
## [2023-10-26 12:18:15.380] [puff::index::jointLog] [info] Running fixFasta
##
## [Step 1 of 4] : counting k-mers
##
## [2023-10-26 12:18:15.920] [puff::index::jointLog] [warning] Removed 41 transcripts that were shorter than the minimum length
## [2023-10-26 12:18:15.921] [puff::index::jointLog] [warning] If you wish to retain duplicate transcripts, use --keep-duplicates
## [2023-10-26 12:18:15.921] [puff::index::jointLog] [info] Replaced 0 non-ATCG nucleotides
## [2023-10-26 12:18:15.921] [puff::index::jointLog] [info] Clipped poly-A tails from 0 transcripts
## wrote 6588 cleaned references
## [2023-10-26 12:18:15.978] [puff::index::jointLog] [info] Filter size not provided; estimating filter size
## [2023-10-26 12:18:16.484] [puff::index::jointLog] [info] ntHll estimated 11513300 distinct k-mers
## Threads = 4
## Vertex length = 31
## Hash functions = 5
## Filter size = 268435456
## Capacity = 2
## Files:
## /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Saccharomyces_cerevisiae.R64-1-1.dna
## -----
## Round 0, 0:268435456
## Pass Filling Filtering

```

```

## 1   2   3
## 2   1   0
## True junctions count = 20631
## False junctions count = 50154
## Hash table size = 70785
## Candidate marks count = 195280
## -----
## Reallocating bifurcations time: 0
## True marks count: 93214
## Edges construction time: 1
## -----
## Distinct junctions = 20631
##
## TwoPaCo::buildGraphMain:: allocated with scalable_malloc; freeing.
## TwoPaCo::buildGraphMain:: Calling scalable_allocation_command(TBBMALLOC_CLEAN_ALL_BLOCKS)
## allowedIn: 14
## Max Junction ID: 20809
## seen.size():166481 kmerInfo.size():20810
## approximateContigTotalLength: 11070364
## counters for complex kmers:
## (prec>1 & succ>1)=327 | (succ>1 & isStart)=3 | (prec>1 & isEnd)=11 | (isStart & isEnd)=1
## contig count: 25029 element count: 12321058 complex nodes: 343
## # of ones in rank vector: 25028
## [2023-10-26 12:18:24.799] [puff::index::jointLog] [info] Starting the Pufferfish index
## [2023-10-26 12:18:24.800] [puff::index::jointLog] [info] Setting the index/BinaryGfa
## size = 12321058
## -----
## | Loading contigs | Time = 3.4303 ms
## -----
## size = 12321058
## -----
## | Loading contig boundaries | Time = 1.0584 ms
## -----
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## 25028
## [2023-10-26 12:18:24.826] [puff::index::jointLog] [info] Done wrapping the rank vector
## [2023-10-26 12:18:24.826] [puff::index::jointLog] [info] contig count for validation
## [2023-10-26 12:18:24.832] [puff::index::jointLog] [info] Total # of Contigs : 25,029
## [2023-10-26 12:18:24.832] [puff::index::jointLog] [info] Total # of numerical Contigs : 25,029
## [2023-10-26 12:18:24.832] [puff::index::jointLog] [info] Total # of contig vec entries : 25,029
## [2023-10-26 12:18:24.832] [puff::index::jointLog] [info] bits per offset entry 17
## [2023-10-26 12:18:24.833] [puff::index::jointLog] [info] Done constructing the contig vec
## [2023-10-26 12:18:24.837] [puff::index::jointLog] [info] # segments = 25,028
## [2023-10-26 12:18:24.837] [puff::index::jointLog] [info] total length = 12,321,058

```

```

## [2023-10-26 12:18:24.838] [puff::index::jointLog] [info] Reading the reference files ...
## [2023-10-26 12:18:24.927] [puff::index::jointLog] [info] positional integer width = 24
## [2023-10-26 12:18:24.927] [puff::index::jointLog] [info] seqSize = 12,321,058
## [2023-10-26 12:18:24.927] [puff::index::jointLog] [info] rankSize = 12,321,058
## [2023-10-26 12:18:24.927] [puff::index::jointLog] [info] edgeVecSize = 0
## [2023-10-26 12:18:24.927] [puff::index::jointLog] [info] num keys = 11,570,218
## [Building BooPHF] 0.213% elapsed: 0 min 0 sec remaining: 0 min 2 sec[Building BooPHF]
## [2023-10-26 12:18:25.456] [puff::index::jointLog] [info] mphf size = 7.22767 MB
## [2023-10-26 12:18:25.465] [puff::index::jointLog] [info] chunk size = 3,080,265
## [2023-10-26 12:18:25.465] [puff::index::jointLog] [info] chunk 0 = [0, 3,080,282)
## [2023-10-26 12:18:25.465] [puff::index::jointLog] [info] chunk 1 = [3,080,282, 6,160,547)
## [2023-10-26 12:18:25.465] [puff::index::jointLog] [info] chunk 2 = [6,160,547, 9,240,812)
## [2023-10-26 12:18:25.465] [puff::index::jointLog] [info] chunk 3 = [9,240,812, 12,321,028)
## [2023-10-26 12:18:26.046] [puff::index::jointLog] [info] finished populating pos vector
## [2023-10-26 12:18:26.046] [puff::index::jointLog] [info] writing index components
## [2023-10-26 12:18:26.093] [puff::index::jointLog] [info] finished writing dense pufferfish index
## [2023-10-26 12:18:26.096] [jLog] [info] done building index
## for info, total work write each : 2.331    total work inram from level 3 : 4.322  total work
## Bitarray      60630080 bits (100.00 %) (array + ranks )
## final hash      0 bits (0.00 %) (nb in final hash 0)

```

Notice that we combined the fasta file of the transcriptome with the fasta file of the entire genome (in that order) into the gentrome.fasta.gz file which was then indexed.

Salmon is a pseudomapper, so it doesn't create sam/bam files and is instead able to count directly from the fastq files. We will do the pseudomapping and counting all in one step in the next activity.

## 4.5 Questions

### 4.5.1 With Rsubread:

Question 1: Try aligning the fastq files allowing multi-mapping reads (set unique = FALSE), allowing for up to 6 “best” locations to be reported (nBestLocations = 6), and allow reads to be fractionally counted (fraction = TRUE). Specify the output file names (bam\_files\_multi) by substituting “.fastq.gz” with “.multi.bam” so we don’t overwrite our unique alignment bam files.

```

# Define the pattern and replacement
pattern <- "\\\\.fastq\\\\.gz$"
replacement <- "subread.multi.bam"

# Create the new file names

```

```

bam_files_multi <- gsub(pattern, replacement, trimmed_fastq_files)

# update this code to run with Rsubread multimapping, as described above.
align(index=index_reference_genome,
      readfile1=trimmed_fastq_files,
      output_file = -----,
      type = "rna",
      input_format = "gzFASTQ",
      output_format = "BAM",
      unique = ----,
      nBestLocations = ----,
      nthreads=6
)

```

Question 2: Look at the proportion of reads mapped and see if we get any more reads mapping by specifying a less stringent criteria.

#### 4.5.2 With Salmon:

Question 3: What are the pros and cons of using Salmon vs subread for mapping reads?

Be sure to knit this file into a pdf or html file once you're finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

**R version 4.3.1 (2023-06-16)**

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8|en\_US.UTF-8||C||en\_US.UTF-8|en\_US.UTF-8

**attached base packages:** stats4, stats, graphics, grDevices, utils, datasets, methods and base

**other attached packages:** Rsubread(v.2.14.2), ShortRead(v.1.58.0), GenomicAlignments(v.1.36.0), SummarizedExperiment(v.1.30.2), MatrixGenerics(v.1.12.3), matrixStats(v.1.0.0), Rsamtools(v.2.16.0), GenomicRanges(v.1.52.1), Biostrings(v.2.68.1), GenomeInfoDb(v.1.36.4), XVector(v.0.40.0), BiocParallel(v.1.34.2), Rfastp(v.1.10.0), org.Sc.sgd.db(v.3.17.0), AnnotationDbi(v.1.62.2), IRanges(v.2.34.1), S4Vectors(v.0.38.2), Biobase(v.2.60.0), BiocGenerics(v.0.46.0), clusterProfiler(v.4.8.2), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pander(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3), forcats(v.1.0.0),

*stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyverse(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)*

**loaded via a namespace (and not attached):** *RColorBrewer(v.1.1-3), rstudioapi(v.0.15.0), jsonlite(v.1.8.7), magrittr(v.2.0.3), farver(v.2.1.1), rmarkdown(v.2.25), ragg(v.1.2.6), fs(v.1.6.3), zlibbioc(v.1.46.0), vctrs(v.0.6.4), memoise(v.2.0.1), RCurl(v.1.98-1.12), ggtree(v.3.8.2), S4Arrays(v.1.0.6), htmltools(v.0.5.6.1), curl(v.5.1.0), gridGraphics(v.0.5-1), KernSmooth(v.2.23-22), plyr(v.1.8.9), cachem(v.1.0.8), lifecycle(v.1.0.3), pkgconfig(v.2.0.3), Matrix(v.1.6-1.1), R6(v.2.5.1), fastmap(v.1.1.1), gson(v.0.1.0), Genome-InfoDbData(v.1.2.10), snakecase(v.0.11.1), digest(v.0.6.33), aplot(v.0.2.2), enrichplot(v.1.20.0), colorspace(v.2.1-0), patchwork(v.1.1.3), rprojroot(v.2.0.3), textshaping(v.0.3.7), RSQLite(v.2.3.1), hwriter(v.1.3.2.1), labeling(v.0.4.3), fansi(v.1.0.5), timechange(v.0.2.0), abind(v.1.4-5), httr(v.1.4.7), polyclip(v.1.10-6), compiler(v.4.3.1), proxy(v.0.4-27), bit64(v.4.0.5), withr(v.2.5.1), downloader(v.0.4), viridis(v.0.6.4), DBI(v.1.1.3), ggforce(v.0.4.1), MASS(v.7.3-60), DelayedArray(v.0.26.7), rjson(v.0.2.21), classInt(v.0.4-10), HDO.db(v.0.99.1), units(v.0.8-4), tools(v.4.3.1), ape(v.5.7-1), scatterpie(v.0.2.1), glue(v.1.6.2), nlme(v.3.1-163), GOSemSim(v.2.26.1), sf(v.1.0-14), grid(v.4.3.1), shadowtext(v.0.1.2), reshape2(v.1.4.4), fgsea(v.1.26.0), generics(v.0.1.3), gtable(v.0.3.4), tzdb(v.0.4.0), class(v.7.3-22), data.table(v.1.14.8), hms(v.1.1.3), tidygraph(v.1.2.3), utf8(v.1.2.3), ggrepel(v.0.9.4), pillar(v.1.9.0), yulab.utils(v.0.1.0), vroom(v.1.6.4), splines(v.4.3.1), tweenr(v.2.0.2), treeio(v.1.24.3), lattice(v.0.21-9), deldir(v.1.0-9), bit(v.4.0.5), tidyselect(v.1.2.0), GO.db(v.3.17.0), gridExtra(v.2.3), bookdown(v.0.36), xfun(v.0.40), graphlayouts(v.1.0.1), stringi(v.1.7.12), lazyeval(v.0.2.2), ggfun(v.0.1.3), yaml(v.2.3.7), evaluate(v.0.22), codetools(v.0.2-19), interp(v.1.1-4), ggraph(v.2.1.0), qvalue(v.2.32.0), RVenn(v.1.1.0), ggplotify(v.0.1.2), cli(v.3.6.1), systemfonts(v.1.0.5), munsell(v.0.5.0), Rcpp(v.1.0.11), png(v.0.1-8), parallel(v.4.3.1), blob(v.1.2.4), jpeg(v.0.1-10), latticeExtra(v.0.6-30), DOSE(v.3.26.1), bitops(v.1.0-7), viridisLite(v.0.4.2), e1071(v.1.7-13), scales(v.1.2.1), crayon(v.1.5.2), rlang(v.1.1.1), cowplot(v.1.1.1), fastmatch(v.1.1-4) and KEGGREST(v.1.40.1)*



# Chapter 5

## Read Counting

last updated: 2023-10-27

As usual, make sure we have the right packages for this exercise

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# let's load all of the files we were using and want to have again today
p_load("tidyverse", "knitr", "readr",
       "pander", "BiocManager",
       "dplyr", "stringr")

# We also need the Bioconductor packages "Rsubread" for today's activity.
p_load("Rsubread")
```

### 5.1 featureCounts

We will first show how to use the `featureCounts()` function in the `Rsubread` package to generate counts from the mapped .bam files.

#### 5.1.1 Locate BAM files

Previously, we aligned our fastq files to the reference genome, generating BAM files. They should be in your “~/Desktop/Genomic\_Data\_Analysis/Data/Trimmed\_rfastp” folder, unless you chose a different place to store them.

```
# Where the bam files are located (default same as trimmed fastq file location)
bam_file_dir <- "~/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/"
```

```

# save list of all of those files with their full path
bam.files <- list.files(path = bam_file_dir,
                         pattern = ".subread.BAM$",
                         full.names = TRUE)
# make sure we see what we expect.
bam.files

## [1] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_MSN2"
## [2] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_MSN2"
## [3] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_MSN2"
## [4] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_MSN2"
## [5] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_MSN2"
## [6] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_MSN2"
## [7] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_MSN2"
## [8] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_MSN2"
## [9] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_WT_E"
## [10] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_WT_E"
## [11] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_WT_E"
## [12] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_WT_E"
## [13] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_WT_M"
## [14] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_WT_M"
## [15] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_WT_M"
## [16] "/Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp//YPS606_WT_M"

```

You should see the full paths to all 16 trimmed fastq bam files that we will be mapping to the reference genome today.

### 5.1.2 Retrieve the genome annotation

We currently have our raw reads mapped to the genome in the form of bam files. Before the differential expression analysis can proceed, these reads must be assigned and counted towards annotated genes. This can be achieved with functions in the Rsubread package, we will also see how to do this with Salmon.

We will use a bash code chunk to download the latest genome annotation

```

# Define the destination file path
REF_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/Reference"
# If this directory doesn't exist, you need to first complete the Read_Mapping.Rmd executable

# Define the URL of reference genome annotation (gtf)
# (latest from ensembl)
url="ftp://ftp.ensembl.org/pub/release-110/gtf/saccharomyces_cerevisiae/Saccharomyces_cerevisiae.GRCh38.110.gtf"

```

```

# Check if the file already exists at the destination location
if [ ! -f "$REF_DIR/Saccharomyces_cerevisiae.R64-1-1.110.gtf.gz" ]; then
    echo "Reference genome annotation not found, downloading..."
    # If the file does not exist, download it using curl
    curl -o "$REF_DIR/Saccharomyces_cerevisiae.R64-1-1.110.gtf.gz" "$url"
    echo "Downloading finished"
else
    echo "File already exists at $REF_DIR Skipping download."
fi

## File already exists at /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference Skipping download

```

Let's take a look at the first few lines of the gtf file

```

# see the header columns with metadata starting with #! and delimited with \t
read.delim(
  path.expand(
    "~/Desktop/Genomic_Data_Analysis/Reference/Saccharomyces_cerevisiae.R64-1-1.110.gtf.gz"
  ),
  header = F,
  sep = "\t",
  nrow = 10
)

## V1
## 1      #!genome-build R64-1-1
## 2      #!genome-version R64-1-1
## 3      #!genome-date 2011-09
## 4      #!genome-build-accession GCA_000146045.2
## 5      #!genebuild-last-updated 2018-10
## 6          IV
## 7          sgd
## 8          gene
## 9          8683
## 10         9756
## 11         .
## 12         -
## 13         .
## 14 gene_id YDL246C; gene_name SOR2; gene_source sgd; gene_biotype protein_coding;

# We can also take a look at the first few entries to see the columns
read.delim(
  path.expand(
    "~/Desktop/Genomic_Data_Analysis/Reference/Saccharomyces_cerevisiae.R64-1-1.110.gtf.gz"

```

```

),
header = F,
comment.char = "#",
strip.white = T,
nrows = 20 #just the first 20 lines
)

##      V1    V2        V3    V4      V5 V6 V7 V8
## 1  IV sgd     gene 8683  9756   .  -  .
## 2  IV sgd transcript 8683  9756   .  -  .
## 3  IV sgd     exon 8683  9756   .  -  .
## 4  IV sgd      CDS 8686  9756   .  -  0
## 5  IV sgd start_codon 9754  9756   .  -  0
## 6  IV sgd stop_codon 8683  8685   .  -  0
## 7  IV sgd     gene 17577 18566   .  -  .
## 8  IV sgd transcript 17577 18566   .  -  .
## 9  IV sgd     exon 17577 18566   .  -  .
## 10 IV sgd      CDS 17580 18566   .  -  0
## 11 IV sgd start_codon 18564 18566   .  -  0
## 12 IV sgd stop_codon 17577 17579   .  -  0
## 13 IV sgd     gene 1248154 1249821   .  -  .
## 14 IV sgd transcript 1248154 1249821   .  -  .
## 15 IV sgd     exon 1248154 1249821   .  -  .
## 16 IV sgd      CDS 1248157 1249821   .  -  0
## 17 IV sgd start_codon 1249819 1249821   .  -  0
## 18 IV sgd stop_codon 1248154 1248156   .  -  0
## 19 IV sgd     gene 289572 290081   .  -  .
## 20 IV sgd transcript 289572 290081   .  -  .

##
## 1
## 2
## 3      gene_id YDL246C; transcript_id YDL246C;
## 4      gene_id YDL246C; transcript_id YDL246C_mRNA; exon_number 1; gene_name SOR2; gen
## 5      gene_id YDL246C; transcript_id YDL246C_mRNA; exon_number 1; gene_name SOR2; gen
## 6      gene_id YDL246C; transcript_id YDL246C_mRNA; exon_number 1; gene_name SOR2; gen
## 7
## 8      gene_id YDL243C; transcript_id YDL243C;
## 9      gene_id YDL243C; transcript_id YDL243C_mRNA; exon_number 1; gene_name AAD4; gen
## 10     gene_id YDL243C; transcript_id YDL243C_mRNA; exon_number 1; gene_name AAD4; gen
## 11     gene_id YDL243C; transcript_id YDL243C_mRNA; exon_number 1; gene_name AAD4; gen
## 12     gene_id YDL243C; transcript_id YDL243C_mRNA; exon_number 1; gene_name AAD4; gen
## 13
## 14
## 15     gene_id YDR387C; transcript_id YDR387C_mRNA; exon_number 1; gene_name CIN10; gen
## 16     gene_id YDR387C; transcript_id YDR387C_mRNA; exon_number 1; gene_name CIN10; gen

```

```

## 17          gene_id YDR387C; transcript_id YDR387C_mRNA; exon_number 1; gene_r
## 18          gene_id YDR387C; transcript_id YDR387C_mRNA; exon_number 1; gene_r
## 19          gene_id YDR387C; transcript_id YDR387C_mRNA; exon_number 1; gene_r
## 20          gene_id YDD

```

There are 9 columns in a standard gtf file, information about each is available here: <https://useast.ensembl.org/info/website/upload/gff.html>

Note that version 2 of gff is identical to the gtf format.

### 5.1.3 Counting with FeatureCounts

```

library(Rsubread)

# Set path of the reference annotation gzipped gtf file
reference_annotation = "~/Desktop/Genomic_Data_Analysis/Reference/Saccharomyces_cerevisiae.R64-1-
```

We can see the arguments available with the align function from the Rsubread package

```
args(featureCounts)
```

```

## function (files, annot.inbuilt = "mm39", annot.ext = NULL, isGTFAnnotationFile = FALSE,
##          GTF.featureType = "exon", GTF.attrType = "gene_id", GTF.attrType.extra = NULL,
##          chrAliases = NULL, useMetaFeatures = TRUE, allowMultiOverlap = FALSE,
##          minOverlap = 1, fracOverlap = 0, fracOverlapFeature = 0,
##          largestOverlap = FALSE, nonOverlap = NULL, nonOverlapFeature = NULL,
##          readShiftType = "upstream", readShiftSize = 0, readExtension5 = 0,
##          readExtension3 = 0, read2pos = NULL, countMultiMappingReads = TRUE,
##          fraction = FALSE, isLongRead = FALSE, minMQS = 0, splitOnly = FALSE,
##          nonSplitOnly = FALSE, primaryOnly = FALSE, ignoreDup = FALSE,
##          strandSpecific = 0, juncCounts = FALSE, genome = NULL, isPairedEnd = FALSE,
##          countReadPairs = TRUE, requireBothEndsMapped = FALSE, checkFragLength = FALSE,
##          minFragLength = 50, maxFragLength = 600, countChimericFragments = TRUE,
##          autosort = TRUE, nthreads = 1, byReadGroup = FALSE, reportReads = NULL,
##          reportReadsPath = NULL, maxMOp = 10, tmpDir = ".", verbose = FALSE)
## NULL

```

The Phred offset determines the encoding for the base-calling quality string in the FASTQ file. For the Illumina 1.8 format onwards, this encoding is set at +33. However, older formats may use a +64 encoding. Users should ensure that the correct encoding is specified during alignment. If unsure, one can examine the first several quality strings in the FASTQ file. A good rule of thumb is to check whether lower-case letters are present (+64 encoding) or absent (+33).

```

# This command counts the number of each feature per fastq file,
#. generating an output we can use later.
fc <- featureCounts(bam.files,
                      annot.ext = reference_annotation,
                      isGTFAnnotationFile = TRUE,
                      GTF.featureType = "exon"
                     )

## =====
##      / _ _ _ | | | | _ \ | _ \ | _ \ | _ \ | _ \ | _ \ | _ \ |
##      | ( _ _ | | | | | _ ) | | _ ) | | _ ) | | _ ) | | _ ) | | _ )
##      \_ _ \ | | | | | _ <| | _ / | | _ | | _ / | | _ | | _ | | _ )
##      _ _ _ ) | | _ | | _ ) | | _ \ | | _ | | _ / | | _ \ | | _ | | _ )
##      | _ _ / \_ _ / | _ _ / | _ | | _ \ | | _ / | | _ \ | | _ \ | | _ )
##      Rsubread 2.14.2
##
## //===== featureCounts setting =====\\
## ||
## ||           Input files : 16 BAM files
## ||
## ||                         YPS606_MSN24_ETOH REP1_R1.fastq.gz.subread.BAM
## ||                         YPS606_MSN24_ETOH REP2_R1.fastq.gz.subread.BAM
## ||                         YPS606_MSN24_ETOH REP3_R1.fastq.gz.subread.BAM
## ||                         YPS606_MSN24_ETOH REP4_R1.fastq.gz.subread.BAM
## ||                         YPS606_MSN24_MOCK REP1_R1.fastq.gz.subread.BAM
## ||                         YPS606_MSN24_MOCK REP2_R1.fastq.gz.subread.BAM
## ||                         YPS606_MSN24_MOCK REP3_R1.fastq.gz.subread.BAM
## ||                         YPS606_MSN24_MOCK REP4_R1.fastq.gz.subread.BAM
## ||                         YPS606_WT_ETOH REP1_R1.fastq.gz.subread.BAM
## ||                         YPS606_WT_ETOH REP2_R1.fastq.gz.subread.BAM
## ||                         YPS606_WT_ETOH REP3_R1.fastq.gz.subread.BAM
## ||                         YPS606_WT_ETOH REP4_R1.fastq.gz.subread.BAM
## ||                         YPS606_WT_MOCK REP1_R1.fastq.gz.subread.BAM
## ||                         YPS606_WT_MOCK REP2_R1.fastq.gz.subread.BAM
## ||                         YPS606_WT_MOCK REP3_R1.fastq.gz.subread.BAM
## ||                         YPS606_WT_MOCK REP4_R1.fastq.gz.subread.BAM
## ||
## ||           Paired-end : no
## ||           Count read pairs : no
## ||           Annotation : Saccharomyces_cerevisiae.R64-1-1.110.gtf.gz ...
## ||           Dir for temp files : .
## ||           Threads : 1
## ||           Level : meta-feature level
## ||           Multimapping reads : counted

```

```
## || Multi-overlapping reads : not counted
## || Min overlapping bases : 1
## ||
## \\=====
## //=====
## || Load annotation file Saccharomyces_cerevisiae.R64-1-1.110.gtf.gz ...
## || Features : 7507
## || Meta-features : 7127
## || Chromosomes/contigs : 17
## ||
## || Process BAM file YPS606_MSN24_ETOH_REP1_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 233278
## || Successfully assigned alignments : 175843 (75.4%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_MSN24_ETOH_REP2_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 215810
## || Successfully assigned alignments : 161818 (75.0%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_MSN24_ETOH_REP3_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 199076
## || Successfully assigned alignments : 148581 (74.6%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_MSN24_ETOH_REP4_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 205792
## || Successfully assigned alignments : 153525 (74.6%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_MSN24_MOCK_REP1_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 167075
## || Successfully assigned alignments : 122364 (73.2%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_MSN24_MOCK_REP2_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 169754
## || Successfully assigned alignments : 126310 (74.4%)
```

```
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_MSN24_MOCK_REP3_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 210001
## || Successfully assigned alignments : 151958 (72.4%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_MSN24_MOCK_REP4_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 208329
## || Successfully assigned alignments : 153346 (73.6%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_WT_ETOH_REP1_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 181587
## || Successfully assigned alignments : 137526 (75.7%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_WT_ETOH_REP2_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 201551
## || Successfully assigned alignments : 151322 (75.1%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_WT_ETOH_REP3_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 214745
## || Successfully assigned alignments : 161909 (75.4%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_WT_ETOH_REP4_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 187319
## || Successfully assigned alignments : 141422 (75.5%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_WT_MOCK_REP1_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 223539
## || Successfully assigned alignments : 165863 (74.2%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_WT_MOCK_REP2_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
```

```

## || Total alignments : 187469
## || Successfully assigned alignments : 138324 (73.8%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_WT_MOCK_REP3_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 224767
## || Successfully assigned alignments : 163337 (72.7%)
## || Running time : 0.00 minutes
## ||
## || Process BAM file YPS606_WT_MOCK_REP4_R1.fastq.gz.subread.BAM...
## || Single-end reads are included.
## || Total alignments : 206865
## || Successfully assigned alignments : 152394 (73.7%)
## || Running time : 0.00 minutes
## ||
## || Write the final count table.
## || Write the read assignment summary.
## ||
## \\=====
//
```

We can see what all is stored in the featureCounts output object

```

names(fc)

## [1] "counts"      "annotation"   "targets"     "stat"
```

The statistics of the read mapping can be seen with `fc$stats`. This reports the numbers of unassigned reads and the reasons why they are not assigned (eg. ambiguity, multi-mapping, secondary alignment, mapping quality, fragment length, chimera, read duplicate, non-junction and so on), in addition to the number of successfully assigned reads for each library.

```

fc$stat

##                               Status YPS606_MSN24_ETOH_REP1_R1.fastq.gz.subread.BAM
## 1                         Assigned                      175843
## 2             Unassigned_Unmapped                  28887
## 3             Unassigned_Read_Type                     0
## 4             Unassigned_Singleton                     0
## 5             Unassigned_MappingQuality                   0
## 6             Unassigned_Chimera                       0
## 7             Unassigned_FragmentLength                   0
## 8             Unassigned_Duplicate                     0
```

## 9	Unassigned_MultiMapping	0
## 10	Unassigned_Secondary	0
## 11	Unassigned_NonSplit	0
## 12	Unassigned_NoFeatures	16741
## 13	Unassigned_Overlapping_Length	0
## 14	Unassigned_Ambiguity	11807
## YPS606_MSN24_ETOH REP2_R1.fastq.gz.subread.BAM		
## 1		161818
## 2		26525
## 3		0
## 4		0
## 5		0
## 6		0
## 7		0
## 8		0
## 9		0
## 10		0
## 11		0
## 12		15747
## 13		0
## 14		11720
## YPS606_MSN24_ETOH REP3_R1.fastq.gz.subread.BAM		
## 1		148581
## 2		25383
## 3		0
## 4		0
## 5		0
## 6		0
## 7		0
## 8		0
## 9		0
## 10		0
## 11		0
## 12		14166
## 13		0
## 14		10946
## YPS606_MSN24_ETOH REP4_R1.fastq.gz.subread.BAM		
## 1		153525
## 2		27007
## 3		0
## 4		0
## 5		0
## 6		0
## 7		0
## 8		0
## 9		0

```
## 10          0
## 11          0
## 12      13608
## 13          0
## 14      11652
##   YPS606_MSN24_MOCK_REP1_R1.fastq.gz.subread.BAM
## 1      122364
## 2      23961
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12      12806
## 13         0
## 14      7944
##   YPS606_MSN24_MOCK_REP2_R1.fastq.gz.subread.BAM
## 1      126310
## 2      23452
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12      11908
## 13         0
## 14      8084
##   YPS606_MSN24_MOCK_REP3_R1.fastq.gz.subread.BAM
## 1      151958
## 2      31337
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
```

```
## 11          0
## 12        16741
## 13          0
## 14        9965
##    YPS606_MSN24_MOCK_REP4_R1.fastq.gz.subread.BAM
## 1        153346
## 2        30580
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12        14454
## 13         0
## 14        9949
##    YPS606_WT_ETOH_REP1_R1.fastq.gz.subread.BAM
## 1        137526
## 2        22387
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12        11511
## 13         0
## 14        10163
##    YPS606_WT_ETOH_REP2_R1.fastq.gz.subread.BAM
## 1        151322
## 2        24647
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
```

```
## 12          14578
## 13          0
## 14         11004
##     YPS606_WT_ETOH_REP3_R1.fastq.gz.subread.BAM
## 1          161909
## 2          26246
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12         15064
## 13         0
## 14        11526
##     YPS606_WT_ETOH_REP4_R1.fastq.gz.subread.BAM
## 1          141422
## 2          23167
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12         12670
## 13         0
## 14        10060
##     YPS606_WT_MOCK_REP1_R1.fastq.gz.subread.BAM
## 1          165863
## 2          30132
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12         16586
```

```
## 13          0
## 14        10958
##    YPS606_WT_MOCK_REP2_R1.fastq.gz.subread.BAM
## 1        138324
## 2        26218
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12      13984
## 13         0
## 14      8943
##    YPS606_WT_MOCK_REP3_R1.fastq.gz.subread.BAM
## 1      163337
## 2      32663
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12      17777
## 13         0
## 14      10990
##    YPS606_WT_MOCK_REP4_R1.fastq.gz.subread.BAM
## 1      152394
## 2      29335
## 3          0
## 4          0
## 5          0
## 6          0
## 7          0
## 8          0
## 9          0
## 10         0
## 11         0
## 12      14960
## 13         0
```

```
## 14          10176
```

### 5.1.4 Counts object

The counts for the samples are stored in fc\$counts.

We can look at the dimensions of the counts to see how many genes and samples are present. The first number is the number of genes and the second number is the number of samples.

```
dim(fc$counts)
```

```
## [1] 7127    16
```

let's take a look at the first few lines of fc\$counts

```
head(fc$counts)
```

##	YPS606_MSN24_ETOH_REP1_R1.fastq.gz.subread.BAM	
##	YDL246C	0
##	YDL243C	2
##	YDR387C	6
##	YDL094C	4
##	YDR438W	5
##	YDR523C	1
##	YPS606_MSN24_ETOH_REP2_R1.fastq.gz.subread.BAM	
##	YDL246C	0
##	YDL243C	1
##	YDR387C	10
##	YDL094C	5
##	YDR438W	6
##	YDR523C	1
##	YPS606_MSN24_ETOH_REP3_R1.fastq.gz.subread.BAM	
##	YDL246C	0
##	YDL243C	1
##	YDR387C	7
##	YDL094C	7
##	YDR438W	5
##	YDR523C	0
##	YPS606_MSN24_ETOH_REP4_R1.fastq.gz.subread.BAM	
##	YDL246C	0
##	YDL243C	4
##	YDR387C	6
##	YDL094C	4

```

## YDR438W          3
## YDR523C          0
##      YPS606_MSN24_MOCK_REP1_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          1
## YDR387C          1
## YDL094C          3
## YDR438W          4
## YDR523C          0
##      YPS606_MSN24_MOCK_REP2_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          1
## YDR387C          3
## YDL094C          1
## YDR438W          1
## YDR523C          0
##      YPS606_MSN24_MOCK_REP3_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          0
## YDR387C          3
## YDL094C          3
## YDR438W          3
## YDR523C          0
##      YPS606_MSN24_MOCK_REP4_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          0
## YDR387C          6
## YDL094C          1
## YDR438W          1
## YDR523C          1
##      YPS606_WT_ETOH_REP1_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          4
## YDR387C          9
## YDL094C          2
## YDR438W          1
## YDR523C          0
##      YPS606_WT_ETOH_REP2_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          0
## YDR387C          7
## YDL094C          4
## YDR438W          3
## YDR523C          1
##      YPS606_WT_ETOH_REP3_R1.fastq.gz.subread.BAM
## YDL246C          0

```

```

## YDL243C          3
## YDR387C         12
## YDL094C          2
## YDR438W          4
## YDR523C          1
##      YPS606_WT_ETOH_REP4_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          2
## YDR387C          7
## YDL094C          5
## YDR438W          7
## YDR523C          1
##      YPS606_WT_MOCK_REP1_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          3
## YDR387C          4
## YDL094C          2
## YDR438W          1
## YDR523C          0
##      YPS606_WT_MOCK_REP2_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          1
## YDR387C          2
## YDL094C          3
## YDR438W          3
## YDR523C          0
##      YPS606_WT_MOCK_REP3_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          0
## YDR387C          9
## YDL094C          6
## YDR438W          2
## YDR523C          0
##      YPS606_WT_MOCK_REP4_R1.fastq.gz.subread.BAM
## YDL246C          0
## YDL243C          0
## YDR387C          6
## YDL094C          5
## YDR438W          4
## YDR523C          0

```

The row names of the fc\$counts matrix represent the Systematic Name for each gene (can be Entrez gene identifiers for other organisms) and the column names are the output filenames from calling the align function.

The annotation slot shows the annotation information that featureCounts used

to summarise reads over genes.

```
head(fc$annotation)
```

	GeneID	Chr	Start	End	Strand	Length
## 1	YDL246C	IV	8683	9756	-	1074
## 2	YDL243C	IV	17577	18566	-	990
## 3	YDR387C	IV	1248154	1249821	-	1668
## 4	YDL094C	IV	289572	290081	-	510
## 5	YDR438W	IV	1338274	1339386	+	1113
## 6	YDR523C	IV	1485566	1487038	-	1473

### 5.1.5 Saving fc object for future use

We will need to use this object in our next class. We can use the R function `saveRDS()` to save the R object to your computer, so it can be accessed at a later date.

```
# create a directory for the count output to go into if not already present
dir_output_counts <- path.expand("~/Desktop/Genomic_Data_Analysis/Data/Counts/Rsubread")
if (!dir.exists(dir_output_counts)) {dir.create(dir_output_counts, recursive = TRUE)}

# save the R data object
saveRDS(object = fc, file = paste0(dir_output_counts,"rsubread.yeast_fc_output.Rds"))

# often, we want to share this file as a tsv file. Here is how we can do that:
write_tsv(data.frame(
    fc$annotation[, "GeneID"],
    fc$counts,
    stringsAsFactors=FALSE),
    file=paste0(dir_output_counts,"rsubread.gene_counts.merged.yeast.tsv"))
```

### 5.1.6 RSubread QC

We can have a look at the quality scores associated with each base that has been called by the sequencing machine using the `qualityScores` function in `Rsubread`.

Let's extract quality scores for 50 reads for the fastq file .

```
# Extract quality scores
qs <- qualityScores(
    filename "~/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YP$606_MSN24_ETOH_REP1",
    nreads=50)
```

```

## qualityScores Rsubread 2.14.2
##
## Scan the input file...
## Totally 233278 reads were scanned; the sampling interval is 4665.
## Now extract read quality information...
##
## Completed successfully. Quality scores for 50 reads (equally spaced in the file) are returned.

```

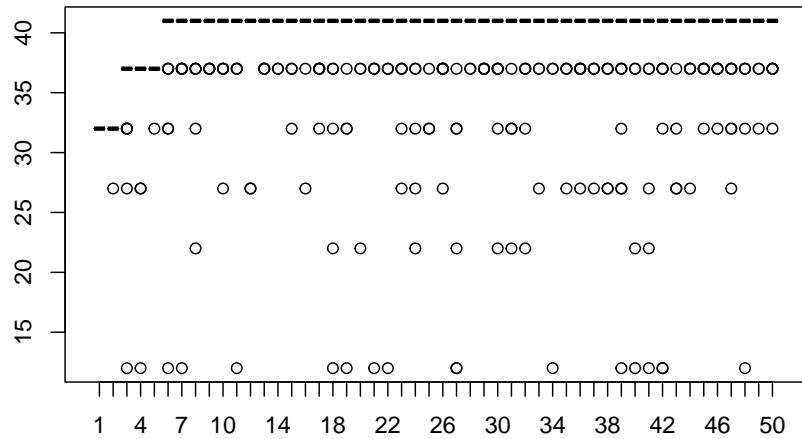
```
head(qs)
```

```

##      1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [1,] 32 32 37 37 37 41 41 41 37 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [2,] 32 32 37 37 37 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [3,] 32 32 37 37 37 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [4,] 32 32 37 37 37 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [5,] 32 32 37 37 37 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [6,] 32 32 37 37 37 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
##      26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
## [1,] 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [2,] 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [3,] 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [4,] 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [5,] 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
## [6,] 41 41 41 41 37 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
```

We are randomly sampling 50 reads from the file and seeing the quality scores. A quality score of 30 corresponds to a 1 in 1000 chance of an incorrect base call. (A quality score of 10 is a 1 in 10 chance of an incorrect base call.) To look at the overall distribution of quality scores across the sampled reads, we can look at a boxplot

```
boxplot(qs)
```



## 5.2 Salmon

Let's go through using salmon to count reads directly from the trimmed fastq.gz files

### 5.2.1 Pseudomapping & Counting

```

DATA_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfasp"
SALMON_OUT_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon"
SALMON_INDEX_DIR="/Users/$USER/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Sa

# make the analysis directory if it doesn't already exist
mkdir -p $SALMON_OUT_DIR

# activate the salmon environment
conda activate salmon

# loop through all of the fastq files
for fn in $DATA_DIR/*.fastq.gz;
do
    samp=`basename ${fn}`

```

```

echo "Processing sample ${samp}"

# run salmon
salmon quant -i $SALMON_INDEX_DIR -l A \
    -r ${fn} \
    --useVBOpt \
    -p 4 --validateMappings -o $SALMON_OUT_DIR/${samp}_quant
done

# combine all of the output files into a merged count matrix
salmon quantmerge --quants $SALMON_OUT_DIR/*_quant --column numreads -o $SALMON_OUT_DIR/salmon.gene_counts.merged.yeast.tsv

# remove the _mRNA from gene name
sed -i '' -E 's/^([^\t]+)_mRNA(\t|$/\1\t/' $SALMON_OUT_DIR/salmon.gene_counts.merged.yeast.tsv

# we can also create a table of tpm values per gene by changing the --column flag
salmon quantmerge --quants $SALMON_OUT_DIR/*_quant --column tpm \
    -o $SALMON_OUT_DIR/salmon.gene_tpm.merged.yeast.tsv

# remove the _mRNA from gene name
sed -i '' -E 's/^([^\t]+)_mRNA(\t|$/\1\t/' $SALMON_OUT_DIR/salmon.gene_tpm.merged.yeast.tsv

conda deactivate

## Processing sample YPS606_MSN24_ETOH_REP1_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## [ program ] => salmon
## [ command ] => quant
## [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Saccharomyces_cerevisiae_r64 }
## [ libType ] => { A }
## [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfaste/YS606_MSN24_ETOH_REP1_R1.fastq.gz }
## [ useVBOpt ] => { }
## [ threads ] => { 4 }
## [ validateMappings ] => { }
## [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606_MSN24_ETOH_REP1_R1.counts }
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606_MSN24_ETOH_REP1_R1.log
## [2023-10-26 16:17:04.631] [jointLog] [info] setting maxHashResizeThreads to 4
## -----
## | Loading contig table | Time = 5.8861 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 1.2828 ms
## -----

```

```

## -----
## | Loading reference lengths | Time = 134.71 us
## -----
## -----
## | Loading mphf table | Time = 9.7713 ms
## -----
## size = 12321058
## [2023-10-26 16:17:04.632] [jointLog] [info] Fragment incompatibility prior below th
## [2023-10-26 16:17:04.632] [jointLog] [info] Usage of --validateMappings implies use
## [2023-10-26 16:17:04.632] [jointLog] [info] Setting consensusSlack to selective-align
## [2023-10-26 16:17:04.632] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:04.633] [jointLog] [info] There is 1 library.
## [2023-10-26 16:17:04.635] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:04.635] [jointLog] [info] Loading dense pufferfish index.
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## -----
## | Loading contig boundaries | Time = 27.891 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 3.9378 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 53.427 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 6.1746 ms
## -----
## -----
## | Loading reference accumulative lengths | Time = 592.67 us
## -----
## [2023-10-26 16:17:04.746] [jointLog] [info] done
## [2023-10-26 16:17:04.832] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:04.833] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:04.833] [jointLog] [info] First decoy index : 6,571
## 
## 
## 
## [2023-10-26 16:17:05.026] [jointLog] [info] Automatically detected most likely libra
## 
## 

```

```

##  

##  

##  

##  

##  

##  

## [2023-10-26 16:17:05.467] [jointLog] [info] Thread saw mini-batch with a maximum of 1.32% zero  

## [2023-10-26 16:17:05.493] [jointLog] [info] Thread saw mini-batch with a maximum of 1.40% zero  

## [2023-10-26 16:17:05.495] [jointLog] [info] Thread saw mini-batch with a maximum of 1.22% zero  

## [2023-10-26 16:17:05.503] [jointLog] [info] Thread saw mini-batch with a maximum of 1.32% zero  

## [2023-10-26 16:17:05.524] [jointLog] [info] Computed 5,478 rich equivalence classes for further  

## [2023-10-26 16:17:05.524] [jointLog] [info] Counted 186,106 total reads in the equivalence cla  

## [2023-10-26 16:17:05.530] [jointLog] [info] Number of mappings discarded because of alignment  

## [2023-10-26 16:17:05.530] [jointLog] [info] Number of fragments entirely discarded because of  

## [2023-10-26 16:17:05.530] [jointLog] [info] Number of fragments discarded because they are bes  

## [2023-10-26 16:17:05.530] [jointLog] [info] Number of fragments discarded because they have on  

## [2023-10-26 16:17:05.530] [jointLog] [warning] Only 186106 fragments were mapped, but the number  

## The effective lengths have been computed using the observed mappings.  

##  

## [2023-10-26 16:17:05.530] [jointLog] [info] Mapping rate = 79.7786%  

##  

## [2023-10-26 16:17:05.530] [jointLog] [info] finished quantifyLibrary()  

## [2023-10-26 16:17:05.532] [jointLog] [info] Starting optimizer  

## [2023-10-26 16:17:05.537] [jointLog] [info] Marked 0 weighted equivalence classes as degenerat  

## [2023-10-26 16:17:05.548] [jointLog] [info] iteration = 0 | max rel diff. = 2057.36  

## [2023-10-26 16:17:06.838] [jointLog] [info] iteration = 100 | max rel diff. = 0.000215288  

## [2023-10-26 16:17:06.838] [jointLog] [info] Finished optimizer  

## [2023-10-26 16:17:06.838] [jointLog] [info] writing output  

##  

## Processing sample YPS606_MSN24_ETOH REP2_R1.fastq.gz  

## Version Info: This is the most recent version of salmon.  

## ### salmon (selective-alignment-based) v1.10.0  

## ### [ program ] => salmon  

## ### [ command ] => quant  

## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Sacchar  

## ### [ libType ] => { A }  

## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS  

## ### [ useVBOpt ] => { }  

## ### [ threads ] => { 4 }  

## ### [ validateMappings ] => { }  

## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606_MSN  

## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606  

## [2023-10-26 16:17:07.815] [jointLog] [info] setting maxHashResizeThreads to 4  

## [2023-10-26 16:17:07.816] [jointLog] [info] Fragment incompatibility prior below threshold. 1  

## [2023-10-26 16:17:07.816] [jointLog] [info] Usage of --validateMappings implies use of minScore

```

```

## [2023-10-26 16:17:07.816] [jointLog] [info] Setting consensusSlack to selective-alignment
## [2023-10-26 16:17:07.816] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:07.816] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 3.4287 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 107.12 us
## -----
## -----
## | Loading reference lengths | Time = 30.959 us
## -----
## |
## | Loading mphf table | Time = 5.9727 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## [2023-10-26 16:17:07.816] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:07.816] [jointLog] [info] Loading dense pufferfish index.
## -----
## | Loading contig boundaries | Time = 26.454 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.4621 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 27.425 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.0427 ms
## -----
## -----
## | Loading reference accumulative lengths | Time = 95.542 us
## -----
## [2023-10-26 16:17:07.886] [jointLog] [info] done
## [2023-10-26 16:17:07.955] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:07.956] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:07.956] [jointLog] [info] First decoy index : 6,571
## |
## |

```

```
##  
##  
## [2023-10-26 16:17:08.123] [jointLog] [info] Automatically detected most likely library type as  
##  
##  
##  
##  
##  
##  
##  
##  
##  
##  
## [2023-10-26 16:17:08.390] [jointLog] [info] Thread saw mini-batch with a maximum of 1.32% zero  
## [2023-10-26 16:17:08.391] [jointLog] [info] Thread saw mini-batch with a maximum of 1.40% zero  
## [2023-10-26 16:17:08.402] [jointLog] [info] Thread saw mini-batch with a maximum of 1.18% zero  
## [2023-10-26 16:17:08.408] [jointLog] [info] Thread saw mini-batch with a maximum of 1.24% zero  
## [2023-10-26 16:17:08.425] [jointLog] [info] Computed 5,294 rich equivalence classes for further  
## [2023-10-26 16:17:08.425] [jointLog] [info] Counted 173,318 total reads in the equivalence class  
## [2023-10-26 16:17:08.430] [jointLog] [info] Number of mappings discarded because of alignment  
## [2023-10-26 16:17:08.430] [jointLog] [info] Number of fragments entirely discarded because of  
## [2023-10-26 16:17:08.430] [jointLog] [info] Number of fragments discarded because they are best  
## [2023-10-26 16:17:08.430] [jointLog] [info] Number of fragments discarded because they have one  
## [2023-10-26 16:17:08.430] [jointLog] [warning] Only 173318 fragments were mapped, but the number  
## The effective lengths have been computed using the observed mappings.  
##  
## [2023-10-26 16:17:08.430] [jointLog] [info] Mapping rate = 80.3105%  
##  
## [2023-10-26 16:17:08.430] [jointLog] [info] finished quantifyLibrary()  
## [2023-10-26 16:17:08.431] [jointLog] [info] Starting optimizer  
## [2023-10-26 16:17:08.433] [jointLog] [info] Marked 0 weighted equivalence classes as degenerate  
## [2023-10-26 16:17:08.442] [jointLog] [info] iteration = 0 | max rel diff. = 1476.42  
## [2023-10-26 16:17:09.772] [jointLog] [info] iteration = 100 | max rel diff. = 0.000932656  
## [2023-10-26 16:17:09.772] [jointLog] [info] Finished optimizer  
## [2023-10-26 16:17:09.772] [jointLog] [info] writing output  
##  
## Processing sample YPS606_MSN24_ETOH REP3_R1.fastq.gz  
## Version Info: This is the most recent version of salmon.  
## ### salmon (selective-alignment-based) v1.10.0  
## ### [ program ] => salmon  
## ### [ command ] => quant  
## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Saccharomyces_cerevisiae_sc5312_index  
## ### [ libType ] => { A }  
## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_MSN24_ETOH REP3_R1.fastq.gz  
## ### [ useVBOpt ] => { }  
## ### [ threads ] => { 4 }  
## ### [ validateMappings ] => { }
```

```
## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Sai
## [2023-10-26 16:17:10.437] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:10.438] [jointLog] [info] Fragment incompatibility prior below th
## [2023-10-26 16:17:10.438] [jointLog] [info] Usage of --validateMappings implies use
## [2023-10-26 16:17:10.438] [jointLog] [info] Setting consensusSlack to selective-align
## [2023-10-26 16:17:10.438] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:10.438] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 4.2352 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 105.79 us
## -----
## |
## | Loading reference lengths | Time = 30.458 us
## -----
## |
## | Loading mphf table | Time = 6.0489 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## [2023-10-26 16:17:10.438] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:10.438] [jointLog] [info] Loading dense pufferfish index.
## -----
## | Loading contig boundaries | Time = 26.492 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.4975 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 26.471 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.0636 ms
## -----
## |
## | Loading reference accumulative lengths | Time = 66.334 us
## -----
## [2023-10-26 16:17:10.508] [jointLog] [info] done
```

```
## [2023-10-26 16:17:10.583] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:10.584] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:10.584] [jointLog] [info] First decoy index : 6,571
##
##
##
##
## [2023-10-26 16:17:10.770] [jointLog] [info] Automatically detected most likely library type as
##
##
##
##
##
##
##
##
##
##
##
## [2023-10-26 16:17:11.107] [jointLog] [info] Thread saw mini-batch with a maximum of 1.58% zero
## [2023-10-26 16:17:11.128] [jointLog] [info] Thread saw mini-batch with a maximum of 1.56% zero
## [2023-10-26 16:17:11.130] [jointLog] [info] Thread saw mini-batch with a maximum of 1.52% zero
## [2023-10-26 16:17:11.131] [jointLog] [info] Thread saw mini-batch with a maximum of 1.55% zero
## [2023-10-26 16:17:11.172] [jointLog] [info] Computed 5,306 rich equivalence classes for further
## [2023-10-26 16:17:11.172] [jointLog] [info] Counted 158,068 total reads in the equivalence cla
## [2023-10-26 16:17:11.178] [jointLog] [info] Number of mappings discarded because of alignment
## [2023-10-26 16:17:11.178] [jointLog] [info] Number of fragments entirely discarded because of
## [2023-10-26 16:17:11.178] [jointLog] [info] Number of fragments discarded because they are bes
## [2023-10-26 16:17:11.178] [jointLog] [info] Number of fragments discarded because they have on
## [2023-10-26 16:17:11.179] [jointLog] [warning] Only 158068 fragments were mapped, but the numbe
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:11.179] [jointLog] [info] Mapping rate = 79.4008%
##
## [2023-10-26 16:17:11.179] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:11.179] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:11.183] [jointLog] [info] Marked 0 weighted equivalence classes as degenerat
## [2023-10-26 16:17:11.198] [jointLog] [info] iteration = 0 | max rel diff. = 483.863
## [2023-10-26 16:17:12.422] [jointLog] [info] iteration = 100 | max rel diff. = 0.000307319
## [2023-10-26 16:17:12.423] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:12.423] [jointLog] [info] writing output
##
## Processing sample YPS606_MSN24_ETOH_REP4_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## ### [ program ] => salmon
## ### [ command ] => quant
## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Sacchar
```

```

## ### [ libType ] => { A }
## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed }
## ### [ useVBOpt ] => { }
## ### [ threads ] => { 4 }
## ### [ validateMappings ] => { }
## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon }
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Sai
## [2023-10-26 16:17:13.364] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:13.364] [jointLog] [info] Fragment incompatibility prior below th
## [2023-10-26 16:17:13.364] [jointLog] [info] Usage of --validateMappings implies use
## [2023-10-26 16:17:13.364] [jointLog] [info] Setting consensusSlack to selective-align
## [2023-10-26 16:17:13.364] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:13.364] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 5.372 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 341.25 us
## -----
## |
## | Loading reference lengths | Time = 94.625 us
## -----
## |
## | Loading mphf table | Time = 13.219 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## [2023-10-26 16:17:13.365] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:13.365] [jointLog] [info] Loading dense pufferfish index.
## Inventory entries filled: 49
## -----
## | Loading contig boundaries | Time = 34.804 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.5577 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 30.303 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.9481 ms

```

```
## -----
## |
## | Loading reference accumulative lengths | Time = 78.25 us
## |
## [2023-10-26 16:17:13.457] [jointLog] [info] done
## [2023-10-26 16:17:13.559] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:13.560] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:13.560] [jointLog] [info] First decoy index : 6,571
##
##
##
##
## [2023-10-26 16:17:13.825] [jointLog] [info] Automatically detected most likely library type as
##
## [2023-10-26 16:17:14.348] [jointLog] [info] Thread saw mini-batch with a maximum of 1.18% zero
## [2023-10-26 16:17:14.349] [jointLog] [info] Thread saw mini-batch with a maximum of 1.32% zero
## [2023-10-26 16:17:14.352] [jointLog] [info] Thread saw mini-batch with a maximum of 1.26% zero
## [2023-10-26 16:17:14.369] [jointLog] [info] Thread saw mini-batch with a maximum of 1.30% zero
##
##
##
##
## [2023-10-26 16:17:14.399] [jointLog] [info] Computed 5,260 rich equivalence classes for further
## [2023-10-26 16:17:14.399] [jointLog] [info] Counted 165,612 total reads in the equivalence clas
##
##
##
##
## [2023-10-26 16:17:14.404] [jointLog] [info] Number of mappings discarded because of alignment
## [2023-10-26 16:17:14.404] [jointLog] [info] Number of fragments entirely discarded because of
## [2023-10-26 16:17:14.404] [jointLog] [info] Number of fragments discarded because they are bes
## [2023-10-26 16:17:14.404] [jointLog] [info] Number of fragments discarded because they have or
## [2023-10-26 16:17:14.404] [jointLog] [warning] Only 165612 fragments were mapped, but the number
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:14.404] [jointLog] [info] Mapping rate = 80.4754%
##
## [2023-10-26 16:17:14.404] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:14.405] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:14.408] [jointLog] [info] Marked 0 weighted equivalence classes as degenerat
## [2023-10-26 16:17:14.416] [jointLog] [info] iteration = 0 | max rel diff. = 1859.16
## [2023-10-26 16:17:15.677] [jointLog] [info] iteration = 100 | max rel diff. = 0.000249187
## [2023-10-26 16:17:15.677] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:15.677] [jointLog] [info] writing output
##
## Processing sample YPS606_MSN24_MOCK_REP1_R1.fastq.gz
```

```
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## ### [ program ] => salmon
## ### [ command ] => quant
## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon }
## ### [ libType ] => { A }
## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed }
## ### [ useVBOpt ] => { }
## ### [ threads ] => { 4 }
## ### [ validateMappings ] => { }
## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon }
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon
## [2023-10-26 16:17:16.442] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:16.442] [jointLog] [info] Fragment incompatibility prior below threshold
## [2023-10-26 16:17:16.442] [jointLog] [info] Usage of --validateMappings implies use of --allowIncompatibility
## [2023-10-26 16:17:16.442] [jointLog] [info] Setting consensusSlack to selective-alignment
## [2023-10-26 16:17:16.442] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:16.442] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 3.5938 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 113.75 us
## -----
## |
## | Loading reference lengths | Time = 29.417 us
## -----
## |
## | Loading mphf table | Time = 5.8898 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## [2023-10-26 16:17:16.442] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:16.442] [jointLog] [info] Loading dense pufferfish index.
## -----
## | Loading contig boundaries | Time = 26.204 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.4212 ms
## -----
## size = 11570218
## -----
```

```
## | Loading positions | Time = 26.389 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.0679 ms
## -----
## |
## | Loading reference accumulative lengths | Time = 61.167 us
## -----
## [2023-10-26 16:17:16.512] [jointLog] [info] done
## [2023-10-26 16:17:16.588] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:16.589] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:16.589] [jointLog] [info] First decoy index : 6,571
##
##
##
##
## [2023-10-26 16:17:16.757] [jointLog] [info] Automatically detected most likely library type as
## 
## [2023-10-26 16:17:16.937] [jointLog] [info] Thread saw mini-batch with a maximum of 1.16% zero
## [2023-10-26 16:17:16.939] [jointLog] [info] Thread saw mini-batch with a maximum of 1.20% zero
## [2023-10-26 16:17:16.941] [jointLog] [info] Thread saw mini-batch with a maximum of 1.20% zero
##
##
##
##
## [2023-10-26 16:17:16.957] [jointLog] [info] Thread saw mini-batch with a maximum of 1.20% zero
## [2023-10-26 16:17:16.974] [jointLog] [info] Computed 5,122 rich equivalence classes for further
## [2023-10-26 16:17:16.974] [jointLog] [info] Counted 130,772 total reads in the equivalence class
##
##
##
##
## [2023-10-26 16:17:16.979] [jointLog] [info] Number of mappings discarded because of alignment
## [2023-10-26 16:17:16.979] [jointLog] [info] Number of fragments entirely discarded because of
## [2023-10-26 16:17:16.979] [jointLog] [info] Number of fragments discarded because they are best
## [2023-10-26 16:17:16.979] [jointLog] [info] Number of fragments discarded because they have one
## [2023-10-26 16:17:16.980] [jointLog] [warning] Only 130772 fragments were mapped, but the number
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:16.980] [jointLog] [info] Mapping rate = 78.2714%
##
## [2023-10-26 16:17:16.980] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:16.980] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:16.983] [jointLog] [info] Marked 0 weighted equivalence classes as degenerate
## [2023-10-26 16:17:16.994] [jointLog] [info] iteration = 0 | max rel diff. = 1687.17
```

```

## [2023-10-26 16:17:18.281] [jointLog] [info] iteration = 100 | max rel diff. = 0.000
## [2023-10-26 16:17:18.281] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:18.281] [jointLog] [info] writing output
##
## Processing sample YPS606_MSN24_MOCK_REP2_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## ### [ program ] => salmon
## ### [ command ] => quant
## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salm
## ### [ libType ] => { A }
## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_
## ### [ useVBOpt ] => { }
## ### [ threads ] => { 4 }
## ### [ validateMappings ] => { }
## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon_
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Sal
## [2023-10-26 16:17:19.020] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:19.020] [jointLog] [info] Fragment incompatibility prior below th
## [2023-10-26 16:17:19.020] [jointLog] [info] Usage of --validateMappings implies use
## [2023-10-26 16:17:19.020] [jointLog] [info] Setting consensusSlack to selective-align
## [2023-10-26 16:17:19.020] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:19.020] [jointLog] [info] There is 1 library.
##
## -----
## | Loading contig table | Time = 5.0504 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 155.67 us
## -----
## |
## | Loading reference lengths | Time = 38.25 us
## -----
## |
## | Loading mphf table | Time = 6.379 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## [2023-10-26 16:17:19.020] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:19.020] [jointLog] [info] Loading dense pufferfish index.
## Inventory entries filled: 49
##
## -----
## | Loading contig boundaries | Time = 27.051 ms
## -----
## size = 12321058

```

```
## -----  
## | Loading sequence | Time = 2.3987 ms  
## -----  
## size = 11570218  
## -----  
## | Loading positions | Time = 26.53 ms  
## -----  
## size = 20892357  
## -----  
## | Loading reference sequence | Time = 4.0417 ms  
## -----  
## -----  
## | Loading reference accumulative lengths | Time = 67.459 us  
## -----  
## [2023-10-26 16:17:19.093] [jointLog] [info] done  
## [2023-10-26 16:17:19.159] [jointLog] [info] Index contained 6,588 targets  
## [2023-10-26 16:17:19.160] [jointLog] [info] Number of decoys : 17  
## [2023-10-26 16:17:19.160] [jointLog] [info] First decoy index : 6,571  
##  
##  
##  
##  
## [2023-10-26 16:17:19.333] [jointLog] [info] Automatically detected most likely library type as  
##  
## [2023-10-26 16:17:19.522] [jointLog] [info] Thread saw mini-batch with a maximum of 1.10% zero  
## [2023-10-26 16:17:19.523] [jointLog] [info] Thread saw mini-batch with a maximum of 1.12% zero  
## [2023-10-26 16:17:19.543] [jointLog] [info] Thread saw mini-batch with a maximum of 1.14% zero  
## [2023-10-26 16:17:19.543] [jointLog] [info] Thread saw mini-batch with a maximum of 1.08% zero  
##  
##  
##  
##  
##  
##  
##  
##  
## [2023-10-26 16:17:19.562] [jointLog] [info] Computed 5,108 rich equivalence classes for further  
## [2023-10-26 16:17:19.562] [jointLog] [info] Counted 135,236 total reads in the equivalence class  
## [2023-10-26 16:17:19.569] [jointLog] [info] Number of mappings discarded because of alignment  
## [2023-10-26 16:17:19.569] [jointLog] [info] Number of fragments entirely discarded because of  
## [2023-10-26 16:17:19.569] [jointLog] [info] Number of fragments discarded because they are best  
## [2023-10-26 16:17:19.569] [jointLog] [info] Number of fragments discarded because they have or  
## [2023-10-26 16:17:19.569] [jointLog] [warning] Only 135236 fragments were mapped, but the number  
## The effective lengths have been computed using the observed mappings.  
##  
## [2023-10-26 16:17:19.569] [jointLog] [info] Mapping rate = 79.6659%
```

```

## [2023-10-26 16:17:19.569] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:19.569] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:19.572] [jointLog] [info] Marked 0 weighted equivalence classes as
## [2023-10-26 16:17:19.586] [jointLog] [info] iteration = 0 | max rel diff. = 1365.21
## [2023-10-26 16:17:20.917] [jointLog] [info] iteration = 100 | max rel diff. = 0.0013
## [2023-10-26 16:17:20.917] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:20.917] [jointLog] [info] writing output
##
## Processing sample YPS606_MSN24_MOCK_REP3_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## ### [ program ] => salmon
## ### [ command ] => quant
## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salm
## ### [ libType ] => { A }
## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_
## ### [ useVBOpt ] => { }
## ### [ threads ] => { 4 }
## ### [ validateMappings ] => { }
## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon_
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Sal
## [2023-10-26 16:17:21.594] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:21.594] [jointLog] [info] Fragment incompatibility prior below th
## [2023-10-26 16:17:21.594] [jointLog] [info] Usage of --validateMappings implies use
## [2023-10-26 16:17:21.594] [jointLog] [info] Setting consensusSlack to selective-align
## [2023-10-26 16:17:21.594] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:21.594] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 4.415 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 112.46 us
## -----
## |
## | Loading reference lengths | Time = 29.542 us
## -----
## |
## | Loading mphf table | Time = 5.8686 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## [2023-10-26 16:17:21.594] [jointLog] [info] Loading pufferfish index

```

```
## [2023-10-26 16:17:21.594] [jointLog] [info] Loading dense pufferfish index.  
## -----  
## | Loading contig boundaries | Time = 26.337 ms  
## -----  
## size = 12321058  
## -----  
## | Loading sequence | Time = 2.5303 ms  
## -----  
## size = 11570218  
## -----  
## | Loading positions | Time = 26.461 ms  
## -----  
## size = 20892357  
## -----  
## | Loading reference sequence | Time = 4.1354 ms  
## -----  
## -----  
## | Loading reference accumulative lengths | Time = 65.292 us  
## -----  
## [2023-10-26 16:17:21.664] [jointLog] [info] done  
## [2023-10-26 16:17:21.732] [jointLog] [info] Index contained 6,588 targets  
## [2023-10-26 16:17:21.733] [jointLog] [info] Number of decoys : 17  
## [2023-10-26 16:17:21.733] [jointLog] [info] First decoy index : 6,571  
##  
##  
##  
##  
##  
## [2023-10-26 16:17:21.910] [jointLog] [info] Automatically detected most likely library type as  
##  
##  
##  
##  
##  
##  
##  
##  
##  
##  
##  
##  
## [2023-10-26 16:17:22.203] [jointLog] [info] Thread saw mini-batch with a maximum of 1.62% zero  
## [2023-10-26 16:17:22.204] [jointLog] [info] Thread saw mini-batch with a maximum of 1.66% zero  
## [2023-10-26 16:17:22.228] [jointLog] [info] Thread saw mini-batch with a maximum of 1.58% zero  
## [2023-10-26 16:17:22.237] [jointLog] [info] Thread saw mini-batch with a maximum of 1.52% zero  
## [2023-10-26 16:17:22.260] [jointLog] [info] Computed 5,213 rich equivalence classes for further  
## [2023-10-26 16:17:22.260] [jointLog] [info] Counted 161,108 total reads in the equivalence cla  
## [2023-10-26 16:17:22.266] [jointLog] [info] Number of mappings discarded because of alignment  
## [2023-10-26 16:17:22.266] [jointLog] [info] Number of fragments entirely discarded because of  
## [2023-10-26 16:17:22.266] [jointLog] [info] Number of fragments discarded because they are best
```

```

## [2023-10-26 16:17:22.266] [jointLog] [info] Number of fragments discarded because t
## [2023-10-26 16:17:22.267] [jointLog] [warning] Only 161108 fragments were mapped, b
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:22.267] [jointLog] [info] Mapping rate = 76.7177%
##
## [2023-10-26 16:17:22.267] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:22.267] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:22.270] [jointLog] [info] Marked 0 weighted equivalence classes as
## [2023-10-26 16:17:22.283] [jointLog] [info] iteration = 0 | max rel diff. = 1651.76
## [2023-10-26 16:17:23.603] [jointLog] [info] iteration = 100 | max rel diff. = 0.000
## [2023-10-26 16:17:23.603] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:23.603] [jointLog] [info] writing output
##
## Processing sample YPS606_MSN24_MOCK_REP4_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## #### [ program ] => salmon
## #### [ command ] => quant
## #### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salm
## #### [ libType ] => { A }
## #### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_
## #### [ useVBOpt ] => { }
## #### [ threads ] => { 4 }
## #### [ validateMappings ] => { }
## #### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon_
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Sal
## [2023-10-26 16:17:24.172] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:24.172] [jointLog] [info] Fragment incompatibility prior below th
## [2023-10-26 16:17:24.172] [jointLog] [info] Usage of --validateMappings implies use
## [2023-10-26 16:17:24.172] [jointLog] [info] Setting consensusSlack to selective-align
## [2023-10-26 16:17:24.172] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:24.172] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 3.5423 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 131.25 us
## -----
## |
## | Loading reference lengths | Time = 38.708 us
## -----
## |
## | Loading mphf table | Time = 7.7895 ms
## -----

```



```

## [2023-10-26 16:17:25.068] [jointLog] [info] Computed 5,245 rich equivalence classes
## [2023-10-26 16:17:25.068] [jointLog] [info] Counted 163,486 total reads in the equivalence classes
## [2023-10-26 16:17:25.074] [jointLog] [info] Number of mappings discarded because of maxHashResSizeThreads = 4
## [2023-10-26 16:17:25.074] [jointLog] [info] Number of fragments entirely discarded = 0
## [2023-10-26 16:17:25.074] [jointLog] [info] Number of fragments discarded because they have zero length = 0
## [2023-10-26 16:17:25.074] [jointLog] [info] Number of fragments discarded because they have zero length = 0
## [2023-10-26 16:17:25.074] [jointLog] [info] Number of fragments discarded because they have zero length = 0
## [2023-10-26 16:17:25.075] [jointLog] [warning] Only 163486 fragments were mapped, but 163486 reads were observed.
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:25.075] [jointLog] [info] Mapping rate = 78.4749%
##
## [2023-10-26 16:17:25.075] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:25.075] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:25.078] [jointLog] [info] Marked 0 weighted equivalence classes as unmated
## [2023-10-26 16:17:25.087] [jointLog] [info] iteration = 0 | max rel diff. = 1711.22
## [2023-10-26 16:17:26.643] [jointLog] [info] iteration = 100 | max rel diff. = 0.000000
## [2023-10-26 16:17:26.644] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:26.644] [jointLog] [info] writing output
##
## Processing sample YPS606_WT_ETOH_REP1_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## #### salmon (selective-alignment-based) v1.10.0
## #### [ program ] => salmon
## #### [ command ] => quant
## #### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon }
## #### [ libType ] => { A }
## #### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/TrimmedReads }
## #### [ useVBOpt ] => { }
## #### [ threads ] => { 4 }
## #### [ validateMappings ] => { }
## #### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/SalmonCounts }
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/SalmonCounts
## [2023-10-26 16:17:27.980] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:27.980] [jointLog] [info] Fragment incompatibility prior below threshold = 0.000000
## [2023-10-26 16:17:27.980] [jointLog] [info] Usage of --validateMappings implies useVBOpt = true
## [2023-10-26 16:17:27.980] [jointLog] [info] Setting consensusSlack to selective-alignment
## [2023-10-26 16:17:27.980] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:27.980] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 4.617 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 143.33 us
## -----
## -----

```

```
## | Loading reference lengths | Time = 31.834 us
##
## [2023-10-26 16:17:27.980] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:27.980] [jointLog] [info] Loading dense pufferfish index.
##
## | Loading mphf table | Time = 6.3335 ms
##
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
##
## | Loading contig boundaries | Time = 28.873 ms
##
## size = 12321058
## 
## | Loading sequence | Time = 6.0092 ms
##
## size = 11570218
## 
## | Loading positions | Time = 49.261 ms
##
## size = 20892357
## 
## | Loading reference sequence | Time = 4.1043 ms
##
## 
## | Loading reference accumulative lengths | Time = 76.375 us
##
## [2023-10-26 16:17:28.080] [jointLog] [info] done
## [2023-10-26 16:17:28.159] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:28.160] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:28.160] [jointLog] [info] First decoy index : 6,571
##
## 
## 
## 
## [2023-10-26 16:17:28.338] [jointLog] [info] Automatically detected most likely library type as
##
## 
## 
## 
## 
## 
## 
## 
## 
## 
```

```

## [2023-10-26 16:17:28.572] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:28.576] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:28.579] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:28.581] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:28.599] [jointLog] [info] Computed 5,055 rich equivalence classes
## [2023-10-26 16:17:28.599] [jointLog] [info] Counted 149,223 total reads in the equi
## [2023-10-26 16:17:28.605] [jointLog] [info] Number of mappings discarded because of
## [2023-10-26 16:17:28.605] [jointLog] [info] Number of fragments entirely discarded b
## [2023-10-26 16:17:28.605] [jointLog] [info] Number of fragments discarded because th
## [2023-10-26 16:17:28.605] [jointLog] [info] Number of fragments discarded because th
## [2023-10-26 16:17:28.605] [jointLog] [warning] Only 149223 fragments were mapped, bu
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:28.605] [jointLog] [info] Mapping rate = 82.1771%
##
## [2023-10-26 16:17:28.605] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:28.606] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:28.608] [jointLog] [info] Marked 0 weighted equivalence classes a
## [2023-10-26 16:17:28.615] [jointLog] [info] iteration = 0 | max rel diff. = 1406.12
## [2023-10-26 16:17:29.821] [jointLog] [info] iteration = 100 | max rel diff. = 2.7103
## [2023-10-26 16:17:29.821] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:29.821] [jointLog] [info] writing output
##
## Processing sample YPS606_WT_ETOH_REP2_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## #### [ program ] => salmon
## #### [ command ] => quant
## #### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salm
## #### [ libType ] => { A }
## #### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_
## #### [ useVBOpt ] => { }
## #### [ threads ] => { 4 }
## #### [ validateMappings ] => { }
## #### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon_
## # Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Sai
## [2023-10-26 16:17:30.552] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:30.552] [jointLog] [info] Fragment incompatibility prior below th
## [2023-10-26 16:17:30.552] [jointLog] [info] Usage of --validateMappings implies use
## [2023-10-26 16:17:30.552] [jointLog] [info] Setting consensusSlack to selective-align
## [2023-10-26 16:17:30.552] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:30.552] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 3.4153 ms
## -----

```

```
## size = 25029
## -----
## | Loading contig offsets | Time = 101.58 us
## -----
## -----
## | Loading reference lengths | Time = 33.667 us
## -----
## -----
## | Loading mphf table | Time = 5.8387 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## [2023-10-26 16:17:30.553] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:30.553] [jointLog] [info] Loading dense pufferfish index.
## -----
## | Loading contig boundaries | Time = 26.221 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.3805 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 26.184 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.0112 ms
## -----
## -----
## | Loading reference accumulative lengths | Time = 61.667 us
## -----
## [2023-10-26 16:17:30.621] [jointLog] [info] done
## [2023-10-26 16:17:30.692] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:30.692] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:30.692] [jointLog] [info] First decoy index : 6,571
## 
## 
## 
## [2023-10-26 16:17:30.866] [jointLog] [info] Automatically detected most likely library type as
```

```

## 
## 
## 
## 
## 
## [2023-10-26 16:17:31.175] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:31.178] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:31.181] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:31.202] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:31.235] [jointLog] [info] Computed 5,185 rich equivalence classes
## [2023-10-26 16:17:31.235] [jointLog] [info] Counted 163,062 total reads in the equi
## [2023-10-26 16:17:31.242] [jointLog] [info] Number of mappings discarded because of
## [2023-10-26 16:17:31.242] [jointLog] [info] Number of fragments entirely discarded
## [2023-10-26 16:17:31.242] [jointLog] [info] Number of fragments discarded because th
## [2023-10-26 16:17:31.242] [jointLog] [info] Number of fragments discarded because th
## [2023-10-26 16:17:31.243] [jointLog] [warning] Only 163062 fragments were mapped, bu
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:31.243] [jointLog] [info] Mapping rate = 80.9036%
##
## [2023-10-26 16:17:31.243] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:31.243] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:31.248] [jointLog] [info] Marked 0 weighted equivalence classes as
## [2023-10-26 16:17:31.261] [jointLog] [info] iteration = 0 | max rel diff. = 2046.36
## [2023-10-26 16:17:32.605] [jointLog] [info] iteration = 100 | max rel diff. = 0.0000
## [2023-10-26 16:17:32.605] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:32.605] [jointLog] [info] writing output
##
## Processing sample YPS606_WT_ETOH REP3_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## ### [ program ] => salmon
## ### [ command ] => quant
## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salm
## ### [ libType ] => { A }
## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_
## ### [ useVBOpt ] => { }
## ### [ threads ] => { 4 }
## ### [ validateMappings ] => { }
## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon_
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Sal
## [2023-10-26 16:17:33.122] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:33.122] [jointLog] [info] Fragment incompatibility prior below thi
## [2023-10-26 16:17:33.122] [jointLog] [info] Usage of --validateMappings implies use
## [2023-10-26 16:17:33.122] [jointLog] [info] Setting consensusSlack to selective-align

```

```
## [2023-10-26 16:17:33.122] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:33.122] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 3.6836 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 114.54 us
## -----
## |
## | Loading reference lengths | Time = 31.417 us
## -----
## |
## | Loading mphf table | Time = 5.8595 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## [2023-10-26 16:17:33.123] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:33.123] [jointLog] [info] Loading dense pufferfish index.
## -----
## | Loading contig boundaries | Time = 26.271 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.3661 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 26.843 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.0524 ms
## -----
## |
## | Loading reference accumulative lengths | Time = 63.25 us
## -----
## [2023-10-26 16:17:33.192] [jointLog] [info] done
## [2023-10-26 16:17:33.257] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:33.258] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:33.258] [jointLog] [info] First decoy index : 6,571
## 
## 
##
```

```

## [2023-10-26 16:17:33.434] [jointLog] [info] Automatically detected most likely library
##
##
##
##
##
##
##
## [2023-10-26 16:17:33.727] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:33.749] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:33.763] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:33.764] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:33.789] [jointLog] [info] Computed 5,300 rich equivalence classes
## [2023-10-26 16:17:33.789] [jointLog] [info] Counted 171,053 total reads in the equivalence
## [2023-10-26 16:17:33.795] [jointLog] [info] Number of mappings discarded because of
## [2023-10-26 16:17:33.795] [jointLog] [info] Number of fragments entirely discarded because of
## [2023-10-26 16:17:33.795] [jointLog] [info] Number of fragments discarded because they
## [2023-10-26 16:17:33.795] [jointLog] [info] Number of fragments discarded because they
## [2023-10-26 16:17:33.796] [jointLog] [warning] Only 171053 fragments were mapped, but
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:33.796] [jointLog] [info] Mapping rate = 79.654%
##
## [2023-10-26 16:17:33.796] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:33.796] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:33.800] [jointLog] [info] Marked 0 weighted equivalence classes as
## [2023-10-26 16:17:33.813] [jointLog] [info] iteration = 0 | max rel diff. = 2209.18
## [2023-10-26 16:17:35.060] [jointLog] [info] iteration = 100 | max rel diff. = 6.514
## [2023-10-26 16:17:35.060] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:35.060] [jointLog] [info] writing output
##
## Processing sample YPS606_WT_ETOH_REP4_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## ### [ program ] => salmon
## ### [ command ] => quant
## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon }
## ### [ libType ] => { A }
## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed }
## ### [ useVBOpt ] => { }
## ### [ threads ] => { 4 }
## ### [ validateMappings ] => { }
## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon }
```

```
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS600
## [2023-10-26 16:17:35.696] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:35.696] [jointLog] [info] Fragment incompatibility prior below threshold. 1
## [2023-10-26 16:17:35.696] [jointLog] [info] Usage of --validateMappings implies use of minScore
## [2023-10-26 16:17:35.696] [jointLog] [info] Setting consensusSlack to selective-alignment defa
## [2023-10-26 16:17:35.696] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:35.696] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 4.5769 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 98.542 us
## -----
## -----
## | Loading reference lengths | Time = 28.625 us
## -----
## -----
## | Loading mphf table | Time = 5.858 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## [2023-10-26 16:17:35.696] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:35.696] [jointLog] [info] Loading dense pufferfish index.
## -----
## | Loading contig boundaries | Time = 26.133 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.4112 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 26.536 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.02 ms
## -----
## -----
## | Loading reference accumulative lengths | Time = 67.542 us
## -----
## [2023-10-26 16:17:35.766] [jointLog] [info] done
## [2023-10-26 16:17:35.832] [jointLog] [info] Index contained 6,588 targets
```

```

## [2023-10-26 16:17:35.833] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:35.833] [jointLog] [info] First decoy index : 6,571
##
##
##
##
## [2023-10-26 16:17:36.010] [jointLog] [info] Automatically detected most likely library
##
##
##
##
##
##
##
##
## [2023-10-26 16:17:36.272] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:36.276] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:36.285] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:36.309] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:36.355] [jointLog] [info] Computed 5,218 rich equivalence classes
## [2023-10-26 16:17:36.355] [jointLog] [info] Counted 151,388 total reads in the equivalence
## [2023-10-26 16:17:36.362] [jointLog] [info] Number of mappings discarded because of
## [2023-10-26 16:17:36.362] [jointLog] [info] Number of fragments entirely discarded because of
## [2023-10-26 16:17:36.362] [jointLog] [info] Number of fragments discarded because they
## [2023-10-26 16:17:36.362] [jointLog] [info] Number of fragments discarded because they
## [2023-10-26 16:17:36.362] [jointLog] [warning] Only 151388 fragments were mapped, but
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:36.362] [jointLog] [info] Mapping rate = 80.8183%
##
## [2023-10-26 16:17:36.362] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:36.363] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:36.366] [jointLog] [info] Marked 0 weighted equivalence classes as
## [2023-10-26 16:17:36.379] [jointLog] [info] iteration = 0 | max rel diff. = 2134.75
## [2023-10-26 16:17:37.618] [jointLog] [info] iteration = 100 | max rel diff. = 0.0007
## [2023-10-26 16:17:37.618] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:37.618] [jointLog] [info] writing output
##
## Processing sample YPS606_WT_MOCK_REP1_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## ### [ program ] => salmon
## ### [ command ] => quant
## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon }
## ### [ libType ] => { A }

```

```
## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS
## ### [ useVBOpt ] => { }
## ### [ threads ] => { 4 }
## ### [ validateMappings ] => { }
## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606_WT_
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606_WT_
## [2023-10-26 16:17:38.289] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:38.289] [jointLog] [info] Fragment incompatibility prior below threshold. 1
## [2023-10-26 16:17:38.289] [jointLog] [info] Usage of --validateMappings implies use of minScore
## [2023-10-26 16:17:38.289] [jointLog] [info] Setting consensusSlack to selective-alignment defa
## [2023-10-26 16:17:38.289] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:38.289] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 3.5544 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 95.5 us
## -----
## -----
## | Loading reference lengths | Time = 33.75 us
## -----
## -----
## | Loading mphf table | Time = 6.1033 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## [2023-10-26 16:17:38.290] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:38.290] [jointLog] [info] Loading dense pufferfish index.
## Inventory entries filled: 49
## -----
## | Loading contig boundaries | Time = 26.378 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.3573 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 25.936 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.0018 ms
## -----
```

```
## -----  
## | Loading reference accumulative lengths | Time = 65.083 us  
## -----  
## [2023-10-26 16:17:38.359] [jointLog] [info] done  
## [2023-10-26 16:17:38.424] [jointLog] [info] Index contained 6,588 targets  
## [2023-10-26 16:17:38.425] [jointLog] [info] Number of decoys : 17  
## [2023-10-26 16:17:38.425] [jointLog] [info] First decoy index : 6,571  
##  
##  
##  
##  
##  
## [2023-10-26 16:17:38.591] [jointLog] [info] Automatically detected most likely library  
##  
##  
##  
##  
##  
##  
##  
##  
##  
##  
##  
## [2023-10-26 16:17:38.858] [jointLog] [info] Thread saw mini-batch with a maximum of  
## [2023-10-26 16:17:38.873] [jointLog] [info] Thread saw mini-batch with a maximum of  
## [2023-10-26 16:17:38.875] [jointLog] [info] Thread saw mini-batch with a maximum of  
## [2023-10-26 16:17:38.879] [jointLog] [info] Thread saw mini-batch with a maximum of  
## [2023-10-26 16:17:38.896] [jointLog] [info] Computed 5,303 rich equivalence classes  
## [2023-10-26 16:17:38.896] [jointLog] [info] Counted 177,062 total reads in the equivalence  
## [2023-10-26 16:17:38.901] [jointLog] [info] Number of mappings discarded because of  
## [2023-10-26 16:17:38.901] [jointLog] [info] Number of fragments entirely discarded by the  
## [2023-10-26 16:17:38.901] [jointLog] [info] Number of fragments discarded because they  
## [2023-10-26 16:17:38.901] [jointLog] [info] Number of fragments discarded because they  
## [2023-10-26 16:17:38.902] [jointLog] [warning] Only 177062 fragments were mapped, but  
## The effective lengths have been computed using the observed mappings.  
##  
## [2023-10-26 16:17:38.902] [jointLog] [info] Mapping rate = 79.2085%  
##  
## [2023-10-26 16:17:38.902] [jointLog] [info] finished quantifyLibrary()  
## [2023-10-26 16:17:38.902] [jointLog] [info] Starting optimizer  
## [2023-10-26 16:17:38.905] [jointLog] [info] Marked 0 weighted equivalence classes as  
## [2023-10-26 16:17:38.914] [jointLog] [info] iteration = 0 | max rel diff. = 2091.9  
## [2023-10-26 16:17:40.167] [jointLog] [info] iteration = 100 | max rel diff. = 0.00000  
## [2023-10-26 16:17:40.167] [jointLog] [info] Finished optimizer  
## [2023-10-26 16:17:40.167] [jointLog] [info] writing output  
##  
## Processing sample YPS606_WT_MOCK_REP2_R1.fastq.gz  
## Version Info: This is the most recent version of salmon.
```

```
## ### salmon (selective-alignment-based) v1.10.0
## [ program ] => salmon
## [ command ] => quant
## [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Saccharomyces_cerevisiae_sc2562 }
## [ libType ] => { A }
## [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPSP606_WT_1.fq.gz }
## [ useVBOpt ] => { }
## [ threads ] => { 4 }
## [ validateMappings ] => { }
## [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPSP606_WT_1.counts }
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPSP606_WT_1.log
## [2023-10-26 16:17:41.512] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:41.512] [jointLog] [info] Fragment incompatibility prior below threshold. 1000000000
## [2023-10-26 16:17:41.512] [jointLog] [info] Usage of --validateMappings implies use of minScore=0
## [2023-10-26 16:17:41.512] [jointLog] [info] Setting consensusSlack to selective-alignment default
## [2023-10-26 16:17:41.512] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:41.512] [jointLog] [info] There is 1 library.
## -----
## | Loading contig table | Time = 3.6063 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 99.667 us
## -----
## |
## | Loading reference lengths | Time = 28.917 us
## -----
## |
## | Loading mphf table | Time = 6.1221 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## [2023-10-26 16:17:41.512] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:41.512] [jointLog] [info] Loading dense pufferfish index.
## -----
## | Loading contig boundaries | Time = 26.439 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.4007 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 26.514 ms
```

```

## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.0067 ms
## -----
## |
## | Loading reference accumulative lengths | Time = 71.792 us
## -----
## [2023-10-26 16:17:41.582] [jointLog] [info] done
## [2023-10-26 16:17:41.649] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:41.650] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:41.650] [jointLog] [info] First decoy index : 6,571
##
##
##
##
## [2023-10-26 16:17:41.820] [jointLog] [info] Automatically detected most likely library
##
## [2023-10-26 16:17:42.024] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:42.025] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:42.035] [jointLog] [info] Thread saw mini-batch with a maximum of
##
##
##
##
## [2023-10-26 16:17:42.040] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:42.057] [jointLog] [info] Computed 5,174 rich equivalence classes
## [2023-10-26 16:17:42.057] [jointLog] [info] Counted 147,314 total reads in the equi
##
##
##
##
##
## [2023-10-26 16:17:42.061] [jointLog] [warning] 0.00160026% of fragments were shorter
## If this fraction is too large, consider re-building the index with a smaller k.
## The minimum read size found was 27.
##
##
## [2023-10-26 16:17:42.061] [jointLog] [info] Number of mappings discarded because of
## [2023-10-26 16:17:42.061] [jointLog] [info] Number of fragments entirely discarded
## [2023-10-26 16:17:42.061] [jointLog] [info] Number of fragments discarded because th
## [2023-10-26 16:17:42.061] [jointLog] [info] Number of fragments discarded because th
## [2023-10-26 16:17:42.062] [jointLog] [warning] Only 147314 fragments were mapped, bu
## The effective lengths have been computed using the observed mappings.
##

```

```
## [2023-10-26 16:17:42.062] [jointLog] [info] Mapping rate = 78.5805%
##
## [2023-10-26 16:17:42.062] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:42.062] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:42.064] [jointLog] [info] Marked 0 weighted equivalence classes as degenerate
## [2023-10-26 16:17:42.072] [jointLog] [info] iteration = 0 | max rel diff. = 1936.84
## [2023-10-26 16:17:43.425] [jointLog] [info] iteration = 100 | max rel diff. = 0.000465223
## [2023-10-26 16:17:43.426] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:43.426] [jointLog] [info] writing output
##
## Processing sample YPS606_WT_MOCK_REP3_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## ### [ program ] => salmon
## ### [ command ] => quant
## ### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Sacchar
## ### [ libType ] => { A }
## ### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS
## ### [ useVBOpt ] => { }
## ### [ threads ] => { 4 }
## ### [ validateMappings ] => { }
## ### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606_WT_
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606_WT_
## -----
## | Loading contig table | Time = 5.9265 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 141.08 us
## -----
## |
## | Loading reference lengths | Time = 31.125 us
## -----
## |
## | Loading mphf table | Time = 6.0722 ms
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## [2023-10-26 16:17:44.099] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:44.099] [jointLog] [info] Fragment incompatibility prior below threshold. I
## [2023-10-26 16:17:44.099] [jointLog] [info] Usage of --validateMappings implies use of minScore
## [2023-10-26 16:17:44.099] [jointLog] [info] Setting consensusSlack to selective-alignment defa
## [2023-10-26 16:17:44.099] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:44.099] [jointLog] [info] There is 1 library.
## [2023-10-26 16:17:44.100] [jointLog] [info] Loading pufferfish index
```

```
## [2023-10-26 16:17:44.100] [jointLog] [info] Loading dense pufferfish index.
## Inventory entries filled: 49
## -----
## | Loading contig boundaries | Time = 26.983 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.6532 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 27.375 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.1746 ms
## -----
## -----
## | Loading reference accumulative lengths | Time = 70.5 us
## -----
## [2023-10-26 16:17:44.174] [jointLog] [info] done
## [2023-10-26 16:17:44.244] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:44.245] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:44.245] [jointLog] [info] First decoy index : 6,571
##
##
##
##
## [2023-10-26 16:17:44.414] [jointLog] [info] Automatically detected most likely library
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
##
## [2023-10-26 16:17:44.757] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:44.763] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:44.767] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:44.788] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:44.822] [jointLog] [info] Computed 5,175 rich equivalence classes
## [2023-10-26 16:17:44.822] [jointLog] [info] Counted 173,912 total reads in the equivalence classes
## [2023-10-26 16:17:44.829] [jointLog] [info] Number of mappings discarded because of
## [2023-10-26 16:17:44.829] [jointLog] [info] Number of fragments entirely discarded
```

```

## [2023-10-26 16:17:44.829] [jointLog] [info] Number of fragments discarded because they are best
## [2023-10-26 16:17:44.829] [jointLog] [info] Number of fragments discarded because they have or
## [2023-10-26 16:17:44.829] [jointLog] [warning] Only 173912 fragments were mapped, but the number
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:44.829] [jointLog] [info] Mapping rate = 77.3743%
##
## [2023-10-26 16:17:44.829] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:44.830] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:44.833] [jointLog] [info] Marked 0 weighted equivalence classes as degenerates
## [2023-10-26 16:17:44.846] [jointLog] [info] iteration = 0 | max rel diff. = 1677.12
## [2023-10-26 16:17:46.093] [jointLog] [info] iteration = 100 | max rel diff. = 7.79079e-05
## [2023-10-26 16:17:46.093] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:46.093] [jointLog] [info] writing output
##
## Processing sample YPS606_WT_MOCK_REP4_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## ### salmon (selective-alignment-based) v1.10.0
## #### [ program ] => salmon
## #### [ command ] => quant
## #### [ index ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Reference/index_salmon_Saccharomyces_cerevisiae_r6_100M }
## #### [ libType ] => { A }
## #### [ unmatedReads ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Trimmed_rfastp/YPS606_WT_MOCK_REP4_R1.fastq.gz }
## #### [ useVBOpt ] => { }
## #### [ threads ] => { 4 }
## #### [ validateMappings ] => { }
## #### [ output ] => { /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606_WT_MOCK_REP4_R1.counts }
## Logs will be written to /Users/clstacy/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/YPS606_WT_MOCK_REP4_R1.log
## [2023-10-26 16:17:46.664] [jointLog] [info] setting maxHashResizeThreads to 4
## [2023-10-26 16:17:46.664] [jointLog] [info] Fragment incompatibility prior below threshold. 1000000000
## [2023-10-26 16:17:46.664] [jointLog] [info] Usage of --validateMappings implies use of minScore
## [2023-10-26 16:17:46.664] [jointLog] [info] Setting consensusSlack to selective-alignment default
## [2023-10-26 16:17:46.664] [jointLog] [info] parsing read library format
## [2023-10-26 16:17:46.664] [jointLog] [info] There is 1 library.
##
## -----
## | Loading contig table | Time = 3.7872 ms
## -----
## size = 25029
## -----
## | Loading contig offsets | Time = 103.96 us
## -----
## |
## | Loading reference lengths | Time = 30.875 us
## -----
## |
## | Loading mphf table | Time = 6.0066 ms

```

```
## -----
## size = 12321058
## Number of ones: 25028
## Number of ones per inventory item: 512
## Inventory entries filled: 49
## [2023-10-26 16:17:46.665] [jointLog] [info] Loading pufferfish index
## [2023-10-26 16:17:46.665] [jointLog] [info] Loading dense pufferfish index.
## -----
## | Loading contig boundaries | Time = 26.387 ms
## -----
## size = 12321058
## -----
## | Loading sequence | Time = 2.409 ms
## -----
## size = 11570218
## -----
## | Loading positions | Time = 26.577 ms
## -----
## size = 20892357
## -----
## | Loading reference sequence | Time = 4.0797 ms
## -----
## -----
## | Loading reference accumulative lengths | Time = 66.167 us
## -----
## [2023-10-26 16:17:46.735] [jointLog] [info] done
## [2023-10-26 16:17:46.812] [jointLog] [info] Index contained 6,588 targets
## [2023-10-26 16:17:46.813] [jointLog] [info] Number of decoys : 17
## [2023-10-26 16:17:46.813] [jointLog] [info] First decoy index : 6,571
##
##
##
##
## [2023-10-26 16:17:46.987] [jointLog] [info] Automatically detected most likely libra
##
##
##
##
##
##
##
##
##
##
## [2023-10-26 16:17:47.303] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:47.319] [jointLog] [info] Thread saw mini-batch with a maximum of
## [2023-10-26 16:17:47.324] [jointLog] [info] Thread saw mini-batch with a maximum of
```

```
## [2023-10-26 16:17:47.348] [jointLog] [info] Thread saw mini-batch with a maximum of 1.14% zeroed
## [2023-10-26 16:17:47.371] [jointLog] [info] Computed 5,176 rich equivalence classes for further processing
## [2023-10-26 16:17:47.371] [jointLog] [info] Counted 163,570 total reads in the equivalence classes
## [2023-10-26 16:17:47.378] [jointLog] [info] Number of mappings discarded because of alignment quality
## [2023-10-26 16:17:47.378] [jointLog] [info] Number of fragments entirely discarded because of alignment quality
## [2023-10-26 16:17:47.378] [jointLog] [info] Number of fragments discarded because they are best mapped
## [2023-10-26 16:17:47.378] [jointLog] [info] Number of fragments discarded because they have one mapping
## [2023-10-26 16:17:47.378] [jointLog] [warning] Only 163570 fragments were mapped, but the number of reads is 163570
## The effective lengths have been computed using the observed mappings.
##
## [2023-10-26 16:17:47.378] [jointLog] [info] Mapping rate = 79.0709%
##
## [2023-10-26 16:17:47.378] [jointLog] [info] finished quantifyLibrary()
## [2023-10-26 16:17:47.380] [jointLog] [info] Starting optimizer
## [2023-10-26 16:17:47.383] [jointLog] [info] Marked 0 weighted equivalence classes as degenerate
## [2023-10-26 16:17:47.396] [jointLog] [info] iteration = 0 | max rel diff. = 1702.19
## [2023-10-26 16:17:48.642] [jointLog] [info] iteration = 100 | max rel diff. = 0.000341007
## [2023-10-26 16:17:48.642] [jointLog] [info] Finished optimizer
## [2023-10-26 16:17:48.642] [jointLog] [info] writing output
##
## Version Info: This is the most recent version of salmon.
## [2023-10-26 16:17:49.211] [mergeLog] [info] samples: [ /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.211] [mergeLog] [info] sample names : [ YPS606_MSN24_ETOH_REP1_R1.fastq.gz
## [2023-10-26 16:17:49.211] [mergeLog] [info] output column : NUMREADS
## [2023-10-26 16:17:49.211] [mergeLog] [info] output file : /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.out
## [2023-10-26 16:17:49.211] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.224] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.236] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.248] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.259] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.272] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.283] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.294] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.305] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.318] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.329] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.340] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.351] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.362] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.373] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.383] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## Version Info: This is the most recent version of salmon.
## [2023-10-26 16:17:49.570] [mergeLog] [info] samples: [ /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.fastq.gz
## [2023-10-26 16:17:49.570] [mergeLog] [info] sample names : [ YPS606_MSN24_ETOH_REP1_R1.fastq.gz
## [2023-10-26 16:17:49.570] [mergeLog] [info] output column : TPM
## [2023-10-26 16:17:49.570] [mergeLog] [info] output file : /Users/clstacy/Desktop/Genomic_Data_Analysis/MSN24/MSN24_ETOH/MSN24_ETOH_R1.out
```

```

## [2023-10-26 16:17:49.570] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.583] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.594] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.606] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.616] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.628] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.639] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.650] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.661] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.674] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.685] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.696] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.707] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.717] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.728] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...
## [2023-10-26 16:17:49.739] [mergeLog] [info] Parsing /Users/clstacy/Desktop/Genomic_L...

```

This script loops through each sample and invokes salmon using default mostly options. The `-i` argument tells salmon where to find the index `-l A` tells salmon that it should automatically determine the library type of the sequencing reads (e.g. stranded vs. unstranded etc.). The `-r` arguments tell salmon where to find the SE reads for this sample (notice, salmon will accept gzipped FASTQ files directly). Finally, the `-p 4` argument tells salmon to make use of 4 threads and the `-o` argument specifies the directory where salmon's quantification results should be written. The `-useVBOpt` flag sets to use variational Bayesian EM algorithm rather than the 'standard EM' to optimize abundance estimates (more accurate). Salmon exposes many different options to the user that enable extra features or modify default behavior. However, the purpose and behavior of all of those options is beyond the scope of this introductory tutorial. You can read about salmon's many options in the documentation.

### 5.3 Questions

1. Identify which gene has the highest counts across all samples for both salmon and Rsubread outputs.
2. Redo the counting over the exons, rather than the genes (specify `useMetaFeatures = FALSE`) with RSubread. Use the bam files generated doing alignment reporting only unique reads, and call the `featureCounts` object `fc.exon`. Check the dimension of the `counts` slot to see how much larger it is.
3. What differences do you notice in the count values from Salmon vs Rsubread?

4. CHALLENGE: Download the full size fastq files from OneDrive & use Salmon to get the read counts on the non-subsampled files.

Be sure to knit this file into a pdf or html file once you're finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

### R version 4.3.1 (2023-06-16)

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8|en\_US.UTF-8||C||en\_US.UTF-8|en\_US.UTF-8

**attached base packages:** stats4, stats, graphics, grDevices, utils, datasets, methods and base

**other attached packages:** Rsubread(v.2.14.2), ShortRead(v.1.58.0), GenomicAlignments(v.1.36.0), SummarizedExperiment(v.1.30.2), MatrixGenerics(v.1.12.3), matrixStats(v.1.0.0), Rsamtools(v.2.16.0), GenomicRanges(v.1.52.1), Biostrings(v.2.68.1), GenomeInfoDb(v.1.36.4), XVector(v.0.40.0), BiocParallel(v.1.34.2), Rfastp(v.1.10.0), org.Sc.sgd.db(v.3.17.0), AnnotationDbi(v.1.62.2), IRanges(v.2.34.1), S4Vectors(v.0.38.2), Biobase(v.2.60.0), BiocGenerics(v.0.46.0), clusterProfiler(v.4.8.2), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pander(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3),forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyverse(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)

**loaded via a namespace (and not attached):** RColorBrewer(v.1.1-3), rstudioapi(v.0.15.0), jsonlite(v.1.8.7), magrittr(v.2.0.3), farver(v.2.1.1), rmarkdown(v.2.25), fs(v.1.6.3), zlibbioc(v.1.46.0), vctrs(v.0.6.4), memoise(v.2.0.1), RCurl(v.1.98-1.12), ggtree(v.3.8.2), S4Arrays(v.1.0.6), htmltools(v.0.5.6.1), gridGraphics(v.0.5-1), plyr(v.1.8.9), cachem(v.1.0.8), lifecycle(v.1.0.3), pkgconfig(v.2.0.3), Matrix(v.1.6-1.1), R6(v.2.5.1), fastmap(v.1.1.1), gson(v.0.1.0), GenomeInfoDbData(v.1.2.10), snakecase(v.0.11.1), digest(v.0.6.33), aplot(v.0.2.2), enrichplot(v.1.20.0), colorspace(v.2.1-0), patchwork(v.1.1.3), rprojroot(v.2.0.3), RSQLite(v.2.3.1), hwriter(v.1.3.2.1), fansi(v.1.0.5), timechange(v.0.2.0), abind(v.1.4-5), httr(v.1.4.7), polyclip(v.1.10-6), compiler(v.4.3.1), bit64(v.4.0.5), withr(v.2.5.1), downloader(v.0.4), viridis(v.0.6.4), DBI(v.1.1.3), ggforce(v.0.4.1), MASS(v.7.3-60), DelayedArray(v.0.26.7), rjson(v.0.2.21), HDO.db(v.0.99.1), tools(v.4.3.1), ape(v.5.7-1), scatterpie(v.0.2.1), glue(v.1.6.2), nlme(v.3.1-163), GOSemSim(v.2.26.1), grid(v.4.3.1), shadowtext(v.0.1.2), reshape2(v.1.4.4), fgsea(v.1.26.0), generics(v.0.1.3), gtable(v.0.3.4), tzdb(v.0.4.0), data.table(v.1.14.8), hms(v.1.1.3), tidygraph(v.1.2.3), utf8(v.1.2.3), ggrepel(v.0.9.4), pillar(v.1.9.0), vroom(v.1.6.4), yulab.utils(v.0.1.0), splines(v.4.3.1), tweenr(v.2.0.2), treeio(v.1.24.3),

*lattice(v.0.21-9), deldir(v.1.0-9), bit(v.4.0.5), tidyselect(v.1.2.0), GO.db(v.3.17.0), gridExtra(v.2.3), bookdown(v.0.36), xfun(v.0.40), graphlayouts(v.1.0.1), stringi(v.1.7.12), lazyeval(v.0.2.2), ggrepel(v.0.1.3), yaml(v.2.3.7), evaluate(v.0.22), codetools(v.0.2-19), interp(v.1.1-4), ggraph(v.2.1.0), archive(v.1.1.5), qvalue(v.2.32.0), RVenn(v.1.1.0), ggplotify(v.0.1.2), cli(v.3.6.1), munsell(v.0.5.0), Rcpp(v.1.0.11), png(v.0.1-8), parallel(v.4.3.1), blob(v.1.2.4), jpeg(v.0.1-10), latticeExtra(v.0.6-30), DOSE(v.3.26.1), bitops(v.1.0-7), viridisLite(v.0.4.2), scales(v.1.2.1), crayon(v.1.5.2), rlang(v.1.1.1), cowplot(v.1.1.1), fastmatch(v.1.1-4) and KEGGREST(v.1.40.1)*

# Chapter 6

## Differential Expression: EdgeR

last updated: 2023-10-27

### Package Install

As usual, make sure we have the right packages for this exercise

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# let's load all of the files we were using and want to have again today
p_load("tidyverse", "knitr", "readr",
       "pander", "BiocManager",
       "dplyr", "stringr",
       "statmod", # required dependency, need to load manually on some macOS versions.
       "purrr", # for working with lists (beautify column names)
       "webshot2", # allow for pdf of output table.
       "reactable") # for pretty tables.

# We also need these Bioconductor packages today.
p_load("edgeR", "AnnotationDbi", "org.Sc.sgd.db")
```

### 6.1 Description

This will be our first differential expression analysis workflow, converting gene counts across samples into meaningful information about genes that appear to be significantly differentially expressed between samples

## 6.2 Learning outcomes

At the end of this exercise, you should be able to:

- Generate a table of sample metadata.
- Filter low counts and normalize count data.
- Utilize the edgeR package to identify differentially expressed genes.

```
library(edgeR)
library(org.Sc.sgd.db)
# for ease of use, set max number of digits after decimal
options(digits=3)
```

## 6.3 Loading in the featureCounts object

We saved this file in the last exercise (`05_Read_Counting.Rmd`) from the `RSubread` package. Now we can load that object back in and assign it to the variable `fc`. Be sure to change the file path if you have saved it in a different location.

```
path_fc_object <- path.expand("~/Desktop/Genomic_Data_Analysis/Data/Counts/Rsubread/rsu
counts_subset <- readRDS(file = path_fc_object)$counts
```

We generated those counts on a subset of the fastq files, but we can load the complete count file with the command below. This file has been generated with the full size fastq files with Salmon.

```
counts <-
read.delim(
  'https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/ethanol_stress/
  sep = "\t",
  header = T,
  row.names = 1
)
```

So far, we've been able to process all of the fastq files without much information about what each sample is in the experimental design. Now, we need the metadata for the samples. Note that the order matters for these files

To find the order of files we need, we can get just the part of the column name before the first “.” symbol with this command:

```

str_split_fixed(counts |> colnames(), "\\\\.", n = 2)[, 1] |> cat()

## YPS606_MSN24_ETOH_REP1_R1 YPS606_MSN24_ETOH_REP2_R1 YPS606_MSN24_ETOH_REP3_R1 YPS606_MSN24_ETOH_REP4_R1

sample_metadata <- tribble(
  ~Sample, ~Genotype, ~Condition,
  "YPS606_MSN24_ETOH_REP1_R1", "msn24dd", "EtOH",
  "YPS606_MSN24_ETOH_REP2_R1", "msn24dd", "EtOH",
  "YPS606_MSN24_ETOH_REP3_R1", "msn24dd", "EtOH",
  "YPS606_MSN24_ETOH_REP4_R1", "msn24dd", "EtOH",
  "YPS606_MSN24_MOCK_REP1_R1", "msn24dd", "unstressed",
  "YPS606_MSN24_MOCK_REP2_R1", "msn24dd", "unstressed",
  "YPS606_MSN24_MOCK_REP3_R1", "msn24dd", "unstressed",
  "YPS606_MSN24_MOCK_REP4_R1", "msn24dd", "unstressed",
  "YPS606_WT_ETOH_REP1_R1", "WT", "EtOH",
  "YPS606_WT_ETOH_REP2_R1", "WT", "EtOH",
  "YPS606_WT_ETOH_REP3_R1", "WT", "EtOH",
  "YPS606_WT_ETOH_REP4_R1", "WT", "EtOH",
  "YPS606_WT_MOCK_REP1_R1", "WT", "unstressed",
  "YPS606_WT_MOCK_REP2_R1", "WT", "unstressed",
  "YPS606_WT_MOCK_REP3_R1", "WT", "unstressed",
  "YPS606_WT_MOCK_REP4_R1", "WT", "unstressed") |>
  # Create a new column that combines the Genotype and Condition value
  mutate(Group = factor(
    paste(Genotype, Condition, sep = "."),
    levels = c(
      "WT.unstressed", "WT.EtOH",
      "msn24dd.unstressed", "msn24dd.EtOH"
    )
  )) |>
  # make Condition and Genotype a factor (with baseline as first level) for edgeR
  mutate(
    Genotype = factor(Genotype,
                      levels = c("WT", "msn24dd")),
    Condition = factor(Condition,
                      levels = c("unstressed", "EtOH"))
  )
)

```

Now, let's create a design matrix with this information

```

group <- sample_metadata$Group
design <- model.matrix(~ 0 + group)
design

```

```

##      groupWT.unstressed groupWT.EtOH groupmsn24dd.unstressed groupmsn24dd.EtOH
## 1          0          0          0          1
## 2          0          0          0          1
## 3          0          0          0          1
## 4          0          0          0          1
## 5          0          0          1          0
## 6          0          0          1          0
## 7          0          0          1          0
## 8          0          0          1          0
## 9          0          1          0          0
## 10         0          1          0          0
## 11         0          1          0          0
## 12         0          1          0          0
## 13         1          0          0          0
## 14         1          0          0          0
## 15         1          0          0          0
## 16         1          0          0          0
## # attr(", "assign")
## [1] 1 1 1 1
## # attr(", "contrasts")
## # attr(", "contrasts")$group
## [1] "contr.treatment"

colnames(design) <- levels(group)
design

##      WT.unstressed WT.EtOH msn24dd.unstressed msn24dd.EtOH
## 1          0          0          0          1
## 2          0          0          0          1
## 3          0          0          0          1
## 4          0          0          0          1
## 5          0          0          1          0
## 6          0          0          1          0
## 7          0          0          1          0
## 8          0          0          1          0
## 9          0          1          0          0
## 10         0          1          0          0
## 11         0          1          0          0
## 12         0          1          0          0
## 13         1          0          0          0
## 14         1          0          0          0
## 15         1          0          0          0
## 16         1          0          0          0
## # attr(", "assign")
## [1] 1 1 1 1

```

```
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
```

## 6.4 Count loading and Annotation

The count matrix is used to construct a DGEList class object. This is the main data class in the edgeR package. The DGEList object is used to store all the information required to fit a generalized linear model to the data, including library sizes and dispersion estimates as well as counts for each gene.

```
y <- DGEList(counts, group=group)
colnames(y) <- sample_metadata$Sample
y$samples
```

	group	lib.size	norm.factors
## YPS606_MSN24_ETOH_REP1_R1	msn24dd.EtOH	17409481	1
## YPS606_MSN24_ETOH_REP2_R1	msn24dd.EtOH	14055425	1
## YPS606_MSN24_ETOH_REP3_R1	msn24dd.EtOH	13127876	1
## YPS606_MSN24_ETOH_REP4_R1	msn24dd.EtOH	16655559	1
## YPS606_MSN24_MOCK_REP1_R1	msn24dd.unstressed	12266723	1
## YPS606_MSN24_MOCK_REP2_R1	msn24dd.unstressed	11781244	1
## YPS606_MSN24_MOCK_REP3_R1	msn24dd.unstressed	11340274	1
## YPS606_MSN24_MOCK_REP4_R1	msn24dd.unstressed	13024330	1
## YPS606_WT_ETOH_REP1_R1	WT.EtOH	15422048	1
## YPS606_WT_ETOH_REP2_R1	WT.EtOH	14924728	1
## YPS606_WT_ETOH_REP3_R1	WT.EtOH	14738753	1
## YPS606_WT_ETOH_REP4_R1	WT.EtOH	12203133	1
## YPS606_WT_MOCK_REP1_R1	WT.unstressed	13592206	1
## YPS606_WT_MOCK_REP2_R1	WT.unstressed	12921965	1
## YPS606_WT_MOCK_REP3_R1	WT.unstressed	13128396	1
## YPS606_WT_MOCK_REP4_R1	WT.unstressed	15568155	1

Human-readable gene symbols can also be added to complement the gene ID for each gene, using the annotation in the org.Sc.sgd.db package.

```
y$genes <- AnnotationDbi::select(org.Sc.sgd.db, keys=rownames(y), columns="GENENAME")

## 'select()' returned 1:1 mapping between keys and columns

head(y$genes)
```

```
##      ORF      SGD GENENAME
## 1 YIL170W S000001432    HXT12
## 2 YIL175W S000001437    <NA>
## 3 YPL276W S000006197    <NA>
## 4 YFL056C S000001838    AAD6
## 5 YCL074W S000000579    <NA>
## 6 YAR061W S000000087    <NA>
```

## 6.5 Filtering to remove low counts

Genes with very low counts across all libraries provide little evidence for differential expression. In addition, the pronounced discreteness of these counts interferes with some of the statistical approximations that are used later in the pipeline. These genes should be filtered out prior to further analysis. Here, we will retain a gene only if it is expressed at a count-per-million (CPM) above 0.7 in at least four samples.

```
keep <- rowSums(cpm(y) > 0.7) >= 4
y <- y[keep,]
summary(keep)
```

```
##      Mode   FALSE    TRUE
## logical     956    5615
```

Where did those cutoff numbers come from?

As a general rule, we don't want to exclude a gene that is expressed in only one group, so a cutoff number equal to the number of replicates can be a good starting point. For counts, a good threshold can be chosen by identifying the CPM that corresponds to a count of 10, which in this case would be about 0.7:

```
cpm(10, mean(y$samples$lib.size))
```

```
##      [,1]
## [1,] 0.72
```

Smaller CPM thresholds are usually appropriate for larger libraries.

## 6.6 Normalization for composition bias

TMM normalization is performed to eliminate composition biases between libraries. This generates a set of normalization factors, where the product of

these factors and the library sizes defines the effective library size. The calcNormFactors function returns the DGEList argument with only the norm.factors changed.

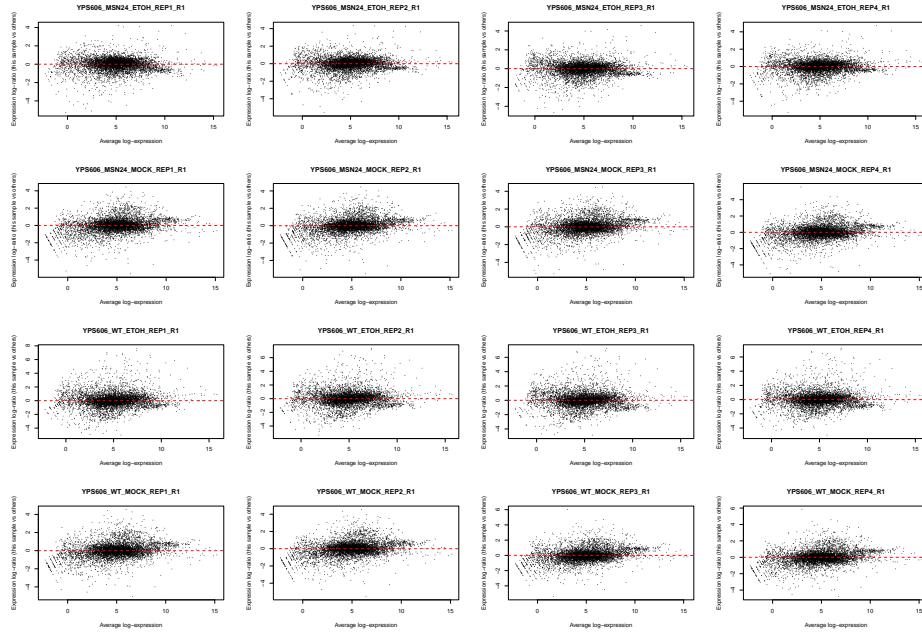
```
y <- calcNormFactors(y)
y$samples
```

	group	lib.size	norm.factors
## YPS606_MSN24_ETOH REP1_R1	msn24dd.EtOH	17409481	1.239
## YPS606_MSN24_ETOH REP2_R1	msn24dd.EtOH	14055425	1.102
## YPS606_MSN24_ETOH REP3_R1	msn24dd.EtOH	13127876	1.108
## YPS606_MSN24_ETOH REP4_R1	msn24dd.EtOH	16655559	1.007
## YPS606_MSN24_MOCK REP1_R1	msn24dd.unstressed	12266723	1.038
## YPS606_MSN24_MOCK REP2_R1	msn24dd.unstressed	11781244	1.003
## YPS606_MSN24_MOCK REP3_R1	msn24dd.unstressed	11340274	0.960
## YPS606_MSN24_MOCK REP4_R1	msn24dd.unstressed	13024330	0.984
## YPS606_WT_ETOH REP1_R1	WT.EtOH	15422048	0.839
## YPS606_WT_ETOH REP2_R1	WT.EtOH	14924728	0.941
## YPS606_WT_ETOH REP3_R1	WT.EtOH	14738753	0.988
## YPS606_WT_ETOH REP4_R1	WT.EtOH	12203133	0.971
## YPS606_WT_MOCK REP1_R1	WT.unstressed	13592206	0.990
## YPS606_WT_MOCK REP2_R1	WT.unstressed	12921965	1.038
## YPS606_WT_MOCK REP3_R1	WT.unstressed	13128396	0.900
## YPS606_WT_MOCK REP4_R1	WT.unstressed	15568155	0.951

The normalization factors multiply to unity across all libraries. A normalization factor below unity indicates that the library size will be scaled down, as there is more suppression (i.e., composition bias) in that library relative to the other libraries. This is also equivalent to scaling the counts upwards in that sample. Conversely, a factor above unity scales up the library size and is equivalent to downscaling the counts. The performance of the TMM normalization procedure can be examined using mean-difference (MD) plots. This visualizes the library size-adjusted log-fold change between two libraries (the difference) against the average log-expression across those libraries (the mean). The below command plots an MD plot, comparing sample 1 against an artificial library constructed from the average of all other samples.

## 6.7 MDS plots

```
for (sample in 1:nrow(y$samples)) {
  plotMD(cpm(y, log=TRUE), column=sample)
  abline(h=0, col="red", lty=2, lwd=2)
}
```



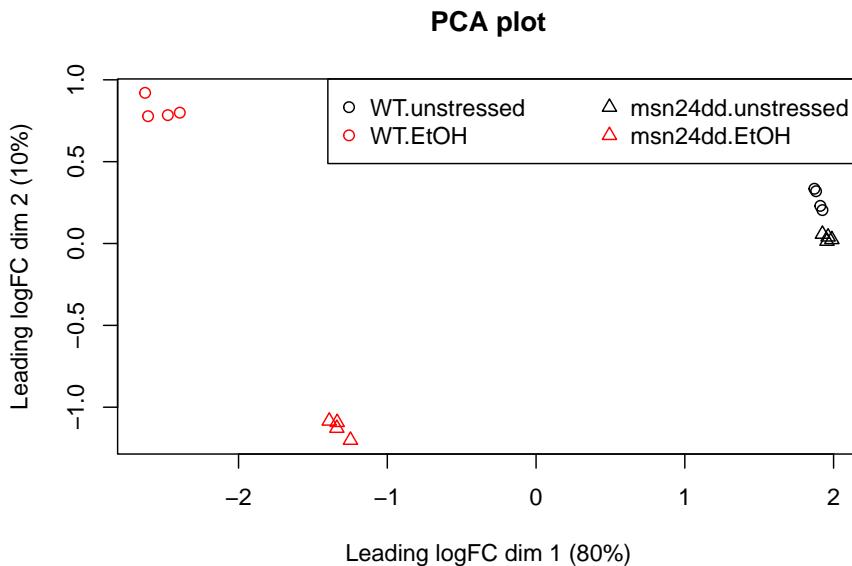
## 6.8 Exploring differences between libraries

The data can be explored by generating multi-dimensional scaling (MDS) plots. This visualizes the differences between the expression profiles of different samples in two dimensions. The next plot shows the MDS plot for the yeast heatshock data.

```

points <- c(1,1,2,2)
colors <- rep(c("black", "red"),8)
plotMDS(y, col=colors[group], pch=points[group])
legend("topright", legend=levels(group),
       pch=points, col=colors, ncol=2)
title(main="PCA plot")

```

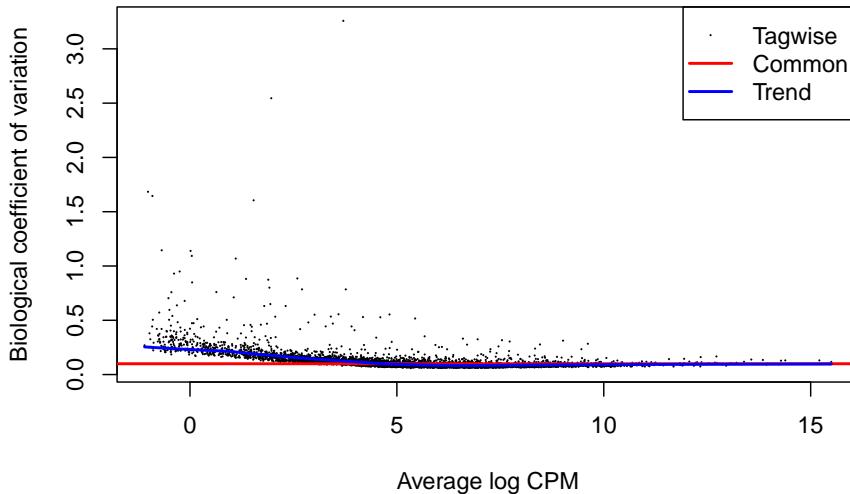


## 6.9 Estimate Dispersion

The trended NB dispersion is estimated using the `estimateDisp` function. This returns the `DGEList` object with additional entries for the estimated NB dispersions for all genes. These estimates can be visualized with `plotBCV`, which shows the root-estimate, i.e., the biological coefficient of variation for each gene

```
y <- estimateDisp(y, design, robust=TRUE)
plotBCV(y)
title(main="Biological Coefficient of Variation (BCV) vs gene abundance")
```

### Biological Coefficient of Variation (BCV) vs gene abundance



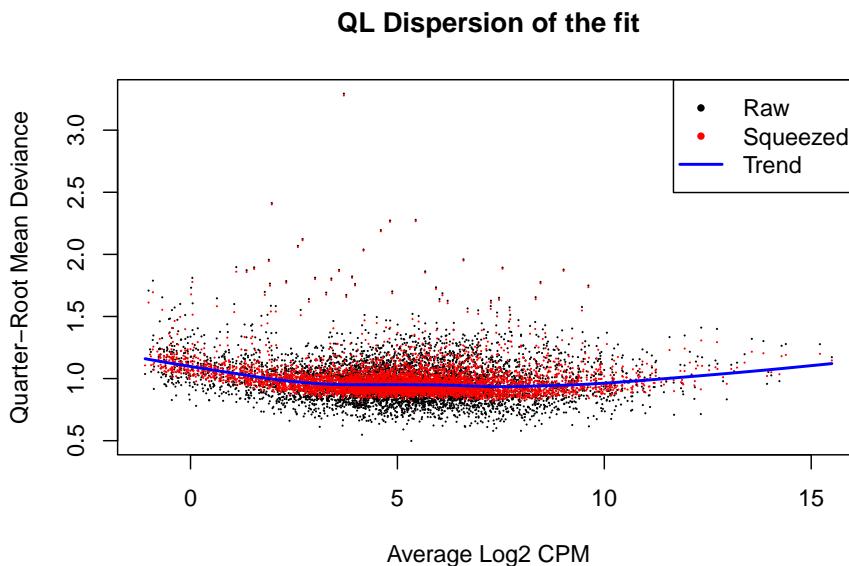
In general, the trend in the NB dispersions should decrease smoothly with increasing abundance. This is because the expression of high-abundance genes is expected to be more stable than that of low-abundance genes. Any substantial increase at high abundances may be indicative of batch effects or trended biases. The value of the trended NB dispersions should range between 0.005 to 0.05 for laboratory-controlled biological systems like mice or cell lines, though larger values will be observed for patient-derived data ( $> 0.1$ )

For the QL dispersions, estimation can be performed using the `glmQLFit` function. This returns a `DGEGLM` object containing the estimated values of the GLM coefficients for each gene

```
fit <- glmQLFit(y, design, robust=TRUE)
head(fit$coefficients)
```

	WT.unstressed	WT.EtOH	msn24dd.unstressed	msn24dd.EtOH
## YIL170W	-15.1	-13.10	-15.87	-13.21
## YFL056C	-11.1	-11.01	-11.07	-10.42
## YAR061W	-13.7	-13.30	-13.36	-13.36
## YGR014W	-8.5	-8.74	-8.41	-8.66
## YPR031W	-10.5	-11.89	-10.45	-11.86
## YIL003W	-10.6	-12.07	-10.72	-11.97

```
plotQLDisp(fit)
title(main="QL Dispersion of the fit")
```



EB squeezing of the raw dispersion estimators towards the trend reduces the uncertainty of the final estimators. The extent of this moderation is determined by the value of the prior df, as estimated from the data. Large estimates for the prior df indicate that the QL dispersions are less variable between genes, meaning that stronger EB moderation can be performed. Small values for the prior df indicate that the dispersions are highly variable, meaning that strong moderation would be inappropriate

Setting `robust=TRUE` in `glmQLFit` is strongly recommended. This causes `glmQLFit` to estimate a vector of `df.prior` values, with lower values for outlier genes and larger values for the main body of genes.

## 6.10 Testing for differential expression

The final step is to actually test for significant differential expression in each gene, using the QL F-test. The contrast of interest can be specified using the `makeContrasts` function. Here, genes are detected that are DE between the stressed and unstressed. This is done by defining the null hypothesis as heat stressed - unstressed = 0.

```

# generate contrasts we are interested in learning about
my.contrasts <- makeContrasts(EtOHvsMOCK.WT = WT.EtOH - WT.unstressed,
                               EtOHvsMOCK.MSN24dd = msn24dd.EtOH - msn24dd.unstressed,
                               EtOH.MSN24ddvsWT = msn24dd.EtOH - WT.EtOH,
                               MOCK.MSN24ddvsWT = msn24dd.unstressed - WT.unstressed,
                               EtOHvsWT.MSN24ddvsWT = (msn24dd.EtOH-msn24dd.unstressed)-(WT.EtOH
                               levels=design)

# This contrast looks at the difference in the stress responses between mutant and WT
res <- glmQLFTest(fit, contrast = my.contrasts[, "EtOHvsWT.MSN24ddvsWT"])

# let's take a quick look at the results
topTags(res, n=10)

## Coefficient: 1*WT.unstressed -1*WT.EtOH -1*msn24dd.unstressed 1*msn24dd.EtOH
##          ORF      SGD GENENAME logFC logCPM   F  PValue     FDR
## YMR105C YMR105C S000004711    PGM2 -6.84  9.70 1608 2.91e-24 1.64e-20
## YMR196W YMR196W S000004809    <NA> -5.15  8.36  877 5.58e-21 1.06e-17
## YKL035W YKL035W S000001518    UGP1 -3.84  10.78 868 5.65e-21 1.06e-17
## YDR516C YDR516C S000002924    EMI2 -4.01  9.08  795 1.65e-20 2.31e-17
## YBR126C YBR126C S000000330    TPS1 -3.46  9.81  693 8.80e-20 9.32e-17
## YLR258W YLR258W S000004248    GSY2 -4.86  8.25  680 1.10e-19 9.32e-17
## YPR149W YPR149W S000006353    NCE102 -4.24  7.95  790 1.16e-19 9.32e-17
## YDR001C YDR001C S000002408    NTH1 -2.89  7.08  650 1.89e-19 1.33e-16
## YHL021C YHL021C S000001013    AIM17 -4.21  6.88  635 3.51e-19 2.19e-16
## YML100W YML100W S000004566    TSL1 -7.12  9.81 1003 5.62e-19 3.15e-16

# generate a beautiful table for the pdf/html file.
topTags(res, n=Inf) |> data.frame() |>
  arrange(FDR) |>
  mutate(logFC=round(logFC,2)) |>
  # mutate(across(where(is.numeric), signif, 3)) |>
  mutate_if(is.numeric, signif, 3) |>
  remove_rownames() |>
  reactable(
    searchable = TRUE,
    showSortable = TRUE,
    columns = list(ORF = colDef(
      cell = function(value) {
        # Render as a link
        url <-
          sprintf("https://www.yeastgenome.org/locus/%s", value)
        htmltools::tags$a(href = url, target = "_blank", as.character(value))
      }
    )
  )
}

```

```
)  
)
```

Search

ORF ↑	SGD ↑	GENENA ↑ ME	↓ logFC	↓ logCPM	↑ F	↑
<a href="#">YMR105C</a>	S000004711	PGM2	-6.84	9.7	1610	2
<a href="#">YMR196W</a>	S000004809		-5.15	8.36	877	5
<a href="#">YKL035W</a>	S000001518	UGP1	-3.84	10.8	868	5
<a href="#">YDR516C</a>	S000002924	EMI2	-4.01	9.08	795	1
<a href="#">YBR126C</a>	S000000330	TPS1	-3.46	9.81	693	
<a href="#">YLR258W</a>	S000004248	GSY2	-4.86	8.25	680	
<a href="#">YPR149W</a>	S000006353	NCE102	-4.24	7.95	790	1
<a href="#">YDR001C</a>	S000002408	NTH1	-2.89	7.08	650	1
<a href="#">YHL021C</a>	S000001013	AIM17	-4.21	6.88	635	3
<a href="#">YML100W</a>	S000004566	TSL1	-7.12	9.81	1000	5

1–10 of 5615 rows      Previous    **1**    2    3    4    5    ...    562    Next

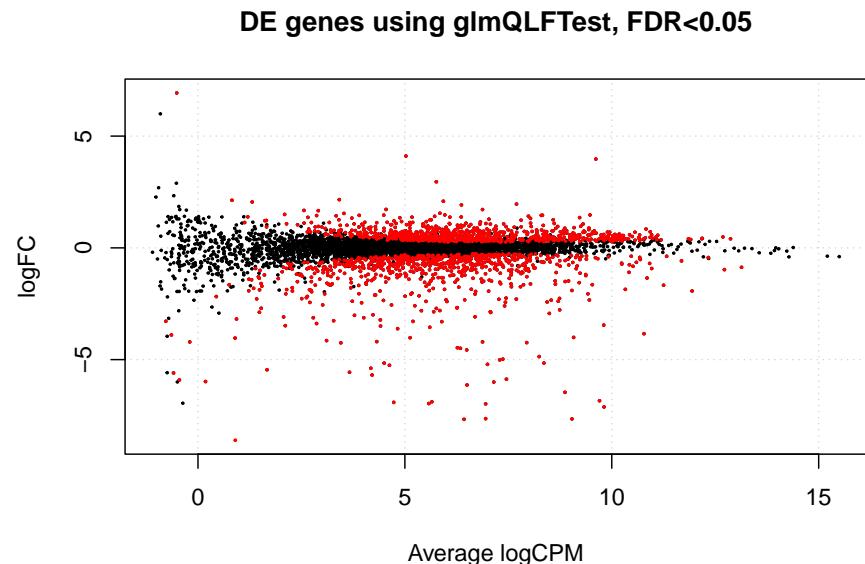
```
is.de <- decideTestsDGE(res,
                         p.value=0.05,
                         lfc = 0) # this allows you to set a cutoff, BUT...
# if you want to compare against a FC that isn't 0, should use glmTreat instead.
```

```
summary(is.de)

##          1*WT.unstressed -1*WT.EtOH -1*msn24dd.unstressed 1*msn24dd.EtOH
## Down                                761
## NotSig                               4031
## Up                                    823
```

Let's take a quick look at the differential expression

```
plotSmear(res, de.tags=rownames(res)[is.de!=0])
title(main="DE genes using glmQLFTest, FDR<0.05")
```



Here is how we can save our output file(s).

```
# Choose topTags destination
dir_output_edgeR <-
  path.expand("~/Desktop/Genomic_Data_Analysis/Analysis/edgeR/")
if (!dir.exists(dir_output_edgeR)) {
  dir.create(dir_output_edgeR, recursive = TRUE)
}

# for sharing with others, the topTags output is convenient.
topTags(res, n = Inf) |> data.frame() |>
```

```

arrange(desc(logFC)) |>
  mutate(logFC = round(logFC, 2)) |>
  # mutate(across(where(is.numeric), signif, 3)) |>
  mutate_if(is.numeric, signif, 3) |>
  write_tsv(x=_, file = paste0(dir_output_edgeR, "yeast_topTags_edgeR.tsv"))

# for subsequent analysis, let's save the res object as an R data object.
saveRDS(object = res, file = paste0(dir_output_edgeR, "yeast_res_edgeR.Rds"))

# we might also want our y object list
saveRDS(object = y, file = paste0(dir_output_edgeR, "yeast_y_edgeR.Rds"))

```

## 6.11 Looking at all contrasts at once

If we want results from all contrasts, we need to loop through them in edgeR, and then combine the results. We will look more at the results of this in a later activity.

```

# One way is to not specify just one contrast, like this:
res_all <- glmQLFTest(fit, contrast = my.contrasts)

res_all |>
  topTags(n=Inf) |>
  data.frame() |>
  head()

##          ORF      SGD GENENAME logFC.EtOHvsMOCK.WT
## YDR516C YDR516C S000002924     EMI2            7.04
## YGR008C YGR008C S000003240     STF2            7.23
## YNL141W YNL141W S000005085     AAH1           -8.23
## YLR258W YLR258W S000004248     GSY2            7.56
## YMR105C YMR105C S000004711     PGM2            7.63
## YER103W YER103W S000000905     SSA4            7.77
##          logFC.EtOHvsMOCK.MSN24dd logFC.EtOH.MSN24ddvsWT logFC.MOCK.MSN24ddvsWT
## YDR516C             3.030           -4.717           -0.710
## YGR008C             2.020           -6.118           -0.906
## YNL141W            -9.064           -0.971           -0.133
## YLR258W             2.692           -5.239           -0.376
## YMR105C             0.794           -6.981           -0.140
## YER103W             7.122           -0.796           -0.149
##          logFC.EtOHvsWT.MSN24ddvsWT logCPM      F    PValue      FDR
## YDR516C              -4.007     9.08 3136 2.13e-32 6.24e-29
## YGR008C              -5.212     7.00 3125 2.22e-32 6.24e-29

```

```

## YNL141W          -0.838   7.19 2965 4.30e-32 8.05e-29
## YLR258W          -4.863   8.25 2747 1.12e-31 1.40e-28
## YMR105C          -6.841   9.70 2723 1.25e-31 1.40e-28
## YER103W          -0.647  10.59 2536 3.03e-31 2.83e-28

# alternatively, we can loop to get DE genes in each contrast.
# here we are just saving which genes are DE per contrast
decideTests_edgeR_tmp <- list()
for (i in 1:ncol(my.contrasts)){

  current.res <- glmQLFTTest(fit, contrast = my.contrasts[,paste0(dimnames(my.contrast
  # current.res <- eBayes(current.res)
  decideTests_edgeR_tmp[[i]] <- current.res |> decideTests(p.value = 0.05, lfc = 0)
  as.data.frame()

}

decideTests_edgeR <- list_cbind(decideTests_edgeR_tmp) |>
  rownames_to_column("gene")

head(decideTests_edgeR)

##      gene -1*WT.unstressed 1*WT.EtOH -1*msn24dd.unstressed 1*msn24dd.EtOH
## 1 YIL170W           1                   1                   1
## 2 YFL056C           0                   0                   1
## 3 YAR061W           0                   0                   0
## 4 YGR014W          -1                  -1                  -1
## 5 YPR031W          -1                  -1                  -1
## 6 YIL003W          -1                  -1                  -1
## -1*WT.EtOH 1*msn24dd.EtOH -1*WT.unstressed 1*msn24dd.unstressed
## 1           0                   0                   0
## 2           1                   0                   0
## 3           0                   0                   0
## 4           0                   0                   0
## 5           0                   0                   0
## 6           0                   0                   0
## 1*WT.unstressed -1*WT.EtOH -1*msn24dd.unstressed 1*msn24dd.EtOH
## 1           0                   0                   0
## 2           1                   0                   1
## 3           0                   0                   0
## 4           0                   0                   0
## 5           0                   0                   0
## 6           0                   0                   0

```

```
# save this file for future analysis
write_tsv(decideTests_edgeR, "~/Documents/GitHub/GenomicDataAnalysis_Fa23/analysis/yeast_decideTests.Rtsv")

# for subsequent analysis, let's also save the res_all object as an R data object.
saveRDS(object = res_all, file = paste0(dir_output_edgeR, "yeast_res_all_edgeR.Rds"))
```

## 6.12 Questions

Question 1: How many genes were upregulated and downregulated in the contrast we looked at in todays activity? Be sure to clarify the cutoffs used for determining significance.

Question 2: Which gene has the lowest pvalue with a postive log2 fold change?

Question 3: Choose one of the contrasts in `my.contrasts` that we didn't test together, and identify the top 3 most differentially expressed genes.

Question 4: In the contrast you chose, give a brief description of the biological interpretation of that contrast.

Question 5: In the example above, we tested for differential expression of any magnitude. Often, we only care about changes of at least a certain magnitude. In this case, we need to use a different command. using the same data, test for genes with differential expression of at least 1 log2 fold change using the `glmTreat` function in edgeR. How do these results compare to DE genes without a logFC cutoff?

## 6.13 A template set of code chunks for doing this is below:

We already loaded in the salmon counts as the object `counts` above. This code chunk just re-downloads that same file.

```
path_salmon_counts <- 'https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/ethanol_vs_water_salmon_counts.txt'

counts <- read.delim(
  path_salmon_counts,
  sep = "\t",
  header = T,
  row.names = 1
)
```

```

# We are reusing the sample_metadata, group, etc that we assigned above

# create DGEList with salmon counts
y <- DGEList(counts, group=group)
colnames(y) <- sample_metadata$Sample

# add gene names
y$genes <- AnnotationDbi::select(org.Sc.sgd.db, keys=rownames(y),
                                   columns="GENENAME")

## 'select()' returned 1:1 mapping between keys and columns

# filter low counts
keep <- rowSums(cpm(y) > 60) >= 4
y <- y[keep,]

# calculate norm factors
y <- calcNormFactors(y)

# estimate dispersion
y <- estimateDisp(y, design, robust=TRUE)

# generate the fit
fit <- glmQLFit(y, design, robust=TRUE)

# Note that, unlike other edgeR functions such as glmLRT and glmQLFTest,
# glmTreat can only accept a single contrast.
# If contrast is a matrix with multiple columns, then only the first column will be used

# Implement a test against FC at least 1 the test our contrast of interest
tr <- glmTreat(fit,
               contrast = my.contrasts[, "EtOHvsWT.MSN24ddvsWT"],
               lfc=1)

# generate a beautiful table for the pdf/html file.
topTags(tr, n = Inf) |>
  data.frame() |>
  arrange(FDR) |>
  mutate(logFC = round(logFC, 2)) |>
  # mutate(across(where(is.numeric), signif, 3)) |>
  mutate_if(is.numeric, signif, 3) |>
  remove_rownames() |>
  reactable(
    searchable = TRUE,

```

6.13. A TEMPLATE SET OF CODE CHUNKS FOR DOING THIS IS BELOW:155

```
showSortable = TRUE,
columns = list(ORF = colDef(
  cell = function(value) {
    # Render as a link
    url <-
      sprintf("https://www.yeastgenome.org/locus/%s", value)
    htmltools::tags$a(href = url, target = "_blank", as.character(value))
  }
))
```

<input type="text" value="Search"/>							
ORF ↑	SGD ↑	GENENA ↑ ME	↓ logFC	↑ unshrunk. logFC	↓ logCPM	↑	
<a href="#">YMR105C</a>	S000004711	PGM2	-6.84	-6.84	9.76	8	
<a href="#">YML100W</a>	S000004566	TSL1	-7.12	-7.12	9.87	1	
<a href="#">YKL035W</a>	S000001518	UGP1	-3.84	-3.84	10.8	3	
<a href="#">YMR196W</a>	S000004809		-5.15	-5.15	8.42		
<a href="#">YDR516C</a>	S000002924	EMI2	-4	-4	9.14	2	
<a href="#">YPR149W</a>	S000006353	NCE102	-4.24	-4.24	7.94	4	
<a href="#">YBR126C</a>	S000000330	TPS1	-3.45	-3.45	9.86	1	
<a href="#">YLR258W</a>	S000004248	GSY2	-4.86	-4.86	8.3	3	
<a href="#">YFR053C</a>	S000001949	HXX1	-7.65	-7.65	9.1		
<a href="#">YHL021C</a>	S000001013	AIM17	-4.2	-4.2	6.93	1	

1–10 of 2372 rows      Previous **1** 2 3 4 5 ... 238 Next

```
# write the table to a tsv file
topTags(tr, n=Inf) |>
  data.frame() |>
  arrange(FDR) |>
  mutate(logFC=round(logFC,2)) |>
  # mutate(across(where(is.numeric), signif, 3)) |>
  mutate_if(is.numeric, signif, 3) |>
```

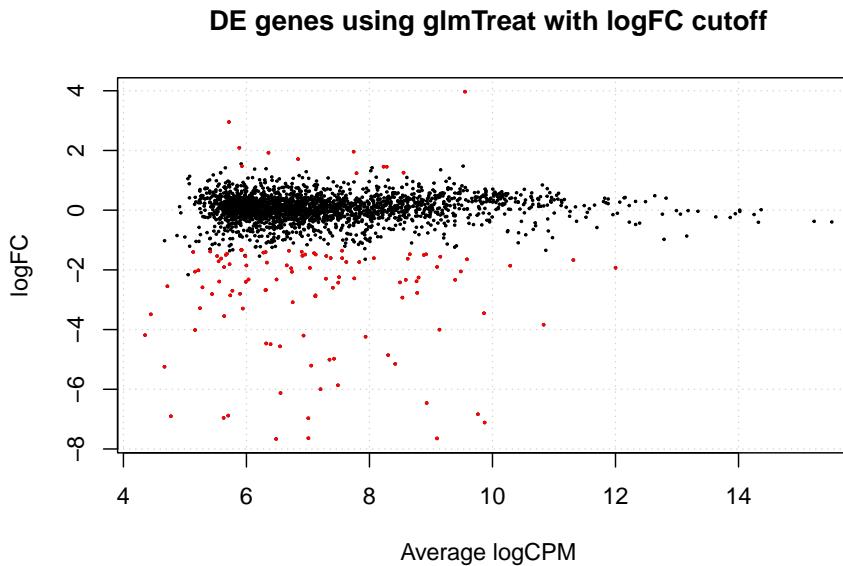
6.13. A TEMPLATE SET OF CODE CHUNKS FOR DOING THIS IS BELOW:157

```
write_tsv(x=_, file = paste0(dir_output_edgeR, "yeast_lfc1topTags_edgeR.tsv"))

# summarize the DE genes
is.de_tr <- decideTestsDGE(tr, p.value=0.05)
summary(is.de_tr)

##          1*WT.unstressed -1*WT.EtOH -1*msn24dd.unstressed 1*msn24dd.EtOH
## Down                               106
## NotSig                            2255
## Up                                11

# visualize results
plotSmear(tr, de.tags=rownames(tr)[is.de_tr!=0])
title(main="DE genes using glmTreat with logFC cutoff")
```



Be sure to knit this file into a pdf or html file once you're finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

**R version 4.3.1 (2023-06-16)**

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8||en\_US.UTF-8||C||en\_US.UTF-8||en\_US.UTF-8

**attached base packages:** *stats4, stats, graphics, grDevices, utils, datasets, methods* and *base*

**other attached packages:** *edgeR(v.3.42.4), limma(v.3.56.2), reactable(v.0.4.4), webshot2(v.0.1.1), statmod(v.1.5.0), Rsubread(v.2.14.2), ShortRead(v.1.58.0), GenomicAlignments(v.1.36.0), SummarizedExperiment(v.1.30.2), MatrixGenerics(v.1.12.3), matrixStats(v.1.0.0), Rsamtools(v.2.16.0), GenomicRanges(v.1.52.1), Biostrings(v.2.68.1), GenomeInfoDb(v.1.36.4), XVector(v.0.40.0), BiocParallel(v.1.34.2), Rfastp(v.1.10.0), org.Sc.sgd.db(v.3.17.0), AnnotationDbi(v.1.62.2), IRanges(v.2.34.1), S4Vectors(v.0.38.2), Biobase(v.2.60.0), BiocGenerics(v.0.46.0), clusterProfiler(v.4.8.2), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pander(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3),forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrrr(v.1.0.2), readr(v.2.1.4), tidyverse(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)*

**loaded via a namespace (and not attached):** *splines(v.4.3.1), later(v.1.3.1), bitops(v.1.0-7), ggplotify(v.0.1.2), polyclip(v.1.10-6), lifecycle(v.1.0.3), rprojroot(v.2.0.3), vroom(v.1.6.4), processx(v.3.8.2), lattice(v.0.21-9), MASS(v.7.3-60), crosstalk(v.1.2.0), magrittr(v.2.0.3), rmarkdown(v.2.25), yaml(v.2.3.7), cowplot(v.1.1.1), chromote(v.0.1.2), DBI(v.1.1.3), RColorBrewer(v.1.1-3), abind(v.1.4-5), zlibbioc(v.1.46.0), ggraph(v.2.1.0), RCurl(v.1.98-1.12), yulab.utils(v.0.1.0), tweenr(v.2.0.2), GenomeInfoDbData(v.1.2.10), enrichplot(v.1.20.0), ggrepel(v.0.9.4), codetools(v.0.2-19), DelayedArray(v.0.26.7), DOSE(v.3.26.1), ggforce(v.0.4.1), tidyselect(v.1.2.0), aplot(v.0.2.2), farver(v.2.1.1), viridis(v.0.6.4), webshot(v.0.5.5), jsonlite(v.1.8.7), ellipsis(v.0.3.2), tidygraph(v.1.2.3), tools(v.4.3.1), treeio(v.1.24.3), Rcpp(v.1.0.11), glue(v.1.6.2), gridExtra(v.2.3), xfun(v.0.40), qvalue(v.2.32.0), websocket(v.1.4.1), withr(v.2.5.1), fastmap(v.1.1.1), latticeExtra(v.0.6-30), fansi(v.1.0.5), digest(v.0.6.33), timechange(v.0.2.0), R6(v.2.5.1), gridGraphics(v.0.5-1), colorspace(v.2.1-0), GO.db(v.3.17.0), jpeg(v.0.1-10), RSQLite(v.2.3.1), utf8(v.1.2.3), generics(v.0.1.3), data.table(v.1.14.8), graphlayouts(v.1.0.1), httr(v.1.4.7), htmlwidgets(v.1.6.2), S4Arrays(v.1.0.6), scatterpie(v.0.2.1), pkgconfig(v.2.0.3), gtable(v.0.3.4), blob(v.1.2.4), hwriter(v.1.3.2.1), shadowtext(v.0.1.2), htmltools(v.0.5.6.1), bookdown(v.0.36), fgsea(v.1.26.0), scales(v.1.2.1), png(v.0.1-8), snakecase(v.0.11.1), ggfun(v.0.1.3), rstudioapi(v.0.15.0), tzdb(v.0.4.0), reshape2(v.1.4.4), rjson(v.0.2.21), nlme(v.3.1-163), cachem(v.1.0.8), RVenn(v.1.1.0), parallel(v.4.3.1), HDO.db(v.0.99.1), pillar(v.1.9.0), grid(v.4.3.1), vctrs(v.0.6.4), promises(v.1.2.1), archive(v.1.1.5), evaluate(v.0.22), cli(v.3.6.1), locfit(v.1.5-9.8), compiler(v.4.3.1), rlang(v.1.1.1), crayon(v.1.5.2), interp(v.1.1-4), reactR(v.0.5.0), ps(v.1.7.5), plyr(v.1.8.9), fs(v.1.6.3), stringi(v.1.7.12), viridisLite(v.0.4.2), deldir(v.1.0-9), munsell(v.0.5.0), lazyeval(v.0.2.2), GOSemSim(v.2.26.1), Matrix(v.1.6-1.1), hms(v.1.1.3), patchwork(v.1.1.3), bit64(v.4.0.5), KEGGREST(v.1.40.1), memoise(v.2.0.1), ggtree(v.3.8.2),*

6.13. A TEMPLATE SET OF CODE CHUNKS FOR DOING THIS IS BELOW:159

*fastmatch(v.1.1-4), bit(v.4.0.5), downloader(v.0.4), ape(v.5.7-1) and gson(v.0.1.0)*



# Chapter 7

## Differential Expression: DESeq2

last updated: 2023-10-27

### Package Install

As usual, make sure we have the right packages for this exercise

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# let's load all of the files we were using and want to have again today
p_load("tidyverse", "knitr", "readr",
       "pander", "BiocManager",
       "dplyr", "stringr",
       "purrr", # for working with lists (beautify column names)
       "reactable") # for pretty tables.

# We also need these Bioconductor packages today.
p_load("DESeq2", "AnnotationDbi", "org.Sc.sgd.db")
```

### 7.1 Description

This will be our second differential expression analysis workflow, converting gene counts across samples into meaningful information about genes that appear to be significantly differentially expressed between samples. This is inspired heavily by: <http://bioconductor.org/packages-devel/bioc/vignettes/DESeq2/inst/doc/DESeq2.html>.

## 7.2 Learning outcomes

At the end of this exercise, you should be able to:

- Utilize the DESeq2 package to identify differentially expressed genes.

```
library(DESeq2)
library(org.Sc.sgd.db)
library(tidyverse)
library(reactable)
# for ease of use, set max number of digits after decimal
options(digits=3)
```

## 7.3 Loading in the featureCounts object

We saved this file at the end the exercise (Read\_Counting.Rmd). Now we can load that object back in and assign it to the variable fc. Be sure to change the file path if you have saved it in a different location. This is the same way we started the edgeR analysis.

```
path_fc_object <- path.expand("~/Desktop/Genomic_Data_Analysis/Data/Counts/Rsubread/rsu
fc <- readRDS(file = path_fc_object)
```

If you don't have that file for any reason, the below code chunk will load a copy of it from Github.

```
counts <- read.delim('https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data
  sep = "\t",
  header = T,
  row.names = 1
)

# clean the column names to remove the fastq.gz_quant
colnames(counts) <- str_split_fixed(counts %>% colnames(), "\\\.", n = 2)[, 1]
```

We will create the data frame again that has all of the metadata information.

```
sample_metadata <- tribble(
  ~Sample,                      ~Genotype,      ~Condition,
  "YPS606_MSN24_ETOH_REP1_R1",  "msn24dd",    "EtOH",
  "YPS606_MSN24_ETOH_REP2_R1",  "msn24dd",    "EtOH",
```

```

"YPS606_MSN24_ETOH REP3_R1", "msn24dd", "EtOH",
"YPS606_MSN24_ETOH REP4_R1", "msn24dd", "EtOH",
"YPS606_MSN24_MOCK REP1_R1", "msn24dd", "unstressed",
"YPS606_MSN24_MOCK REP2_R1", "msn24dd", "unstressed",
"YPS606_MSN24_MOCK REP3_R1", "msn24dd", "unstressed",
"YPS606_MSN24_MOCK REP4_R1", "msn24dd", "unstressed",
"YPS606_WT_ETOH REP1_R1", "WT", "EtOH",
"YPS606_WT_ETOH REP2_R1", "WT", "EtOH",
"YPS606_WT_ETOH REP3_R1", "WT", "EtOH",
"YPS606_WT_ETOH REP4_R1", "WT", "EtOH",
"YPS606_WT_MOCK REP1_R1", "WT", "unstressed",
"YPS606_WT_MOCK REP2_R1", "WT", "unstressed",
"YPS606_WT_MOCK REP3_R1", "WT", "unstressed",
"YPS606_WT_MOCK REP4_R1", "WT", "unstressed") %>%
# make Condition and Genotype a factor (with baseline as first level) for DESeq2
mutate(
  Genotype = factor(Genotype,
                     levels = c("WT", "msn24dd")),
  Condition = factor(Condition,
                     levels = c("unstressed", "EtOH"))
)

```

## 7.4 Count loading and Annotation

The count matrix is used to construct a `DESeqDataSet` class object. This is the main data class in the `DESeq2` package. The `DESeqDataSet` object is used to store all the information required to fit a generalized linear model to the data, including library sizes and dispersion estimates as well as counts for each gene.

Because we used the `featureCounts` function (Liao, Smyth, and Shi 2013) in the `Rsubread` package, the matrix of read counts can be directly provided from the `"counts"` element in the list output. The count matrix and column data can typically be read into R from flat files using base R functions such as `read.csv` or `read.delim`.

With the count matrix, `cts`, and the sample information, `coldata`, we can construct a `DESeqDataSet`:

```

# notice the different design specification
dds <- DESeqDataSetFromMatrix(countData = round(cts),
                               colData = sample_metadata,
                               design = ~ 1 + Genotype + Condition + Genotype:Condition)

## converting counts to integer mode

```

```

# simplify the column names to make them pretty
colnames(dds) <- str_split_fixed(colnames(dds), "\\\\.", n = 2)[, 1]

# take a look at the dds object
dds

## class: DESeqDataSet
## dim: 6571 16
## metadata(1): version
## assays(1): counts
## rownames(6571): YIL170W YIL175W ... YJL134W YER096W
## rowData names(0):
## colnames(16): YPS606_MSN24_ETOH REP1_R1 YPS606_MSN24_ETOH REP2_R1 ...
##   YPS606_WT_MOCK REP3_R1 YPS606_WT_MOCK REP4_R1
## colData names(3): Sample Genotype Condition

# compare this to the edgeR process below:
# y <- DGEList(counts, group=group)
# colnames(y) <- sample_metadata$GEOAccession
# y

```

## 7.5 Filtering to remove low counts

While it is not necessary to pre-filter low count genes before running the DESeq2 functions, there are two reasons which make pre-filtering useful: by removing rows in which there are very few reads, we reduce the memory size of the dds data object, and we increase the speed of count modeling within DESeq2. It can also improve visualizations, as features with no information for differential expression are not plotted in dispersion plots or MA-plots.

Here we perform pre-filtering to keep only rows that have a count of at least 10 for a minimal number of samples. The count of 10 is a reasonable choice for bulk RNA-seq. A recommendation for the minimal number of samples is to specify the smallest group size, e.g. here there are 4 treated samples. If there are not discrete groups, one can use the minimal number of samples where non-zero counts would be considered interesting. One can also omit this step entirely and just rely on the independent filtering procedures available in results(), either IHW or genefilter. See independent filtering section.

```

smallestGroupSize <- 4
keep <- rowSums(counts(dds) >= 10) >= smallestGroupSize
dds <- dds[keep,]

```

```
# Equivalent version in edgeR:
# keep <- rowSums(cpm(y) > 60) >= 4
# y <- y[keep,]
# summary(keep)
```

## 7.6 Testing for differential expression

The standard differential expression analysis steps are wrapped into a single function, DESeq. The estimation steps performed by this function are described below, in the manual page for `?DESeq` and in the Methods section of the DESeq2 publication (Love, Huber, and Anders 2014).

Results tables are generated using the function `results`, which extracts a results table with log2 fold changes, p values and adjusted p values. With no additional arguments to `results`, the log2 fold change and Wald test p value will be for the last variable in the design formula, and if this is a factor, the comparison will be the *last level* of this variable over the *reference level*. However, the order of the variables of the design do not matter so long as the user specifies the comparison to build a results table for, using the name or contrast arguments of `results`.

Details about the comparison are printed to the console, directly above the results table. The text, condition treated vs untreated, tells you that the estimates are of the logarithmic fold change  $\log_2(\text{treated}/\text{untreated})$ .

```
# Now that we have a DESeq2 object, we can perform differential expression.
dds <- DESeq(dds)

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

resultsNames(dds)
```

```

## [1] "Intercept"                      "Genotype_msn24dd_vs_WT"
## [3] "Condition_EtOH_vs_unstressed"   "Genotypemsn24dd.ConditionEtOH"

# create a model matrix
mod_mat <- model.matrix(design(dds), colData(dds))

# define coefficient vectors for each group
WT_MOCK <- colMeans(mod_mat[dds$Genotype == "WT" & dds$Condition == "unstressed", ])
WT_EtOH <- colMeans(mod_mat[dds$Genotype == "WT" & dds$Condition == "EtOH", ])
MSN24_MOCK <- colMeans(mod_mat[dds$Genotype == "msn24dd" & dds$Condition == "unstressed"])
MSN24dd_EtOH <- colMeans(mod_mat[dds$Genotype == "msn24dd" & dds$Condition == "EtOH", ])

```

The nice thing about this approach is that we do not need to worry about any of this, the weights come from our `colMeans()` call automatically. And now, any contrasts that we make will take these weights into account:

```

res <- results(dds)
res

## log2 fold change (MLE): Genotypemsn24dd.ConditionEtOH
## Wald test p-value: Genotypemsn24dd.ConditionEtOH
## DataFrame with 5622 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue      padj
##           <numeric>      <numeric>      <numeric>      <numeric>      <numeric>
## YIL170W    14.8499     0.9768970   0.878689  1.111767 2.66238e-01 0.443362781
## YFL056C    265.8871    0.8156191   0.183405  4.447097 8.70386e-06 0.000074707
## YAR061W    20.1326    -0.6364363   0.468363 -1.358852 1.74194e-01 0.329294856
## YGR014W    2615.6548   -0.0174106   0.128224 -0.135782 8.91994e-01 0.942098024
## YPR031W    237.3713   -0.1163443   0.226110 -0.514547 6.06870e-01 0.755325201
## ...
## YDL086C-A  15.0128    -0.7051445   0.626958 -1.12471 0.26071321 0.43661890
## YJR067C    145.6068   -0.0844778   0.213231 -0.39618 0.69197231 0.81490383
## YDR030C    80.1245    -0.3096978   0.281000 -1.10213 0.27040547 0.44831011
## YJL134W    1389.3306   0.2569824   0.147556  1.74159 0.08157979 0.19181998
## YER096W    250.0039   -0.6634104   0.214676 -3.09029 0.00199959 0.00931377

```

We could have equivalently produced this results table with the following more specific command. Because `Genotypemsn24dd:ConditionEtOH` is the last variable in the design, we could optionally leave off the contrast argument to extract the comparison of the two levels of `Genotypemsn24dd:ConditionEtOH`.

```

res <- results(dds,
               contrast = (MSN24dd_EtOH - MSN24_MOCK) - (WT_EtOH - WT_MOCK)
               )

```

```

res %>%
  data.frame() %>%
  rownames_to_column("ORF") %>%
  # add the gene names
  left_join(AnnotationDbi::select(org.Sc.sgd.db, keys=.ORF, columns="GENENAME"), by="ORF") %>%
  relocate(GENENAME, .after = ORF) %>%
  arrange(padj) %>%
  mutate(log2FoldChange = round(log2FoldChange, 2)) %>%
  mutate(across(where(is.numeric), signif, 3)) %>%
  reactable(
    searchable = TRUE,
    showSortable = TRUE,
    columns = list(ORF = colDef(
      cell = function(value) {
        # Render as a link
        url <-
          sprintf("https://www.yeastgenome.org/locus/%s", value)
        htmltools::tags$a(href = url, target = "_blank", as.character(value))
      }
    )))
  )

## 'select()' returned 1:1 mapping between keys and columns

## Warning: There was 1 warning in `mutate()` .
## i In argument: `across(where(is.numeric), signif, 3)` .
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \((x) mean(x, na.rm = TRUE))
```

ORF ↑	GENENA ↓ ME	baseMean ↑	log2Fold Change ↑	lfcSE ↓	stat ↓	↑
<a href="#">YMR105C</a>	PGM2	11700	-6.85	0.172	-39.7	
<a href="#">YML100W</a>	TSL1	12600	-7.13	0.21	-34	1.
<a href="#">YKL035W</a>	UGP1	24500	-3.85	0.124	-31.1	3.
<a href="#">YFR053C</a>	HXX1	7350	-7.67	0.253	-30.3	4.
<a href="#">YMR196W</a>		4590	-5.17	0.173	-29.9	1..
<a href="#">YDR516C</a>	EMI2	7550	-4.02	0.147	-27.3	4..
<a href="#">YPR149W</a>	NCE102	3410	-4.25	0.158	-26.9	3..
<a href="#">YBR126C</a>	TPS1	12500	-3.47	0.134	-26	8..
<a href="#">YLR258W</a>	GSY2	4240	-4.88	0.188	-26	1..
<a href="#">YER053C</a>	PIC2	2310	-4.99	0.192	-25.9	2..

1–10 of 5622 rows      Previous    **1**    2    3    4    5    ...    563    Next

```
# filter based on padj and a lfc cutoff
res_sig <- subset(res, padj<.01)
res_lfc <- subset(res_sig, abs(log2FoldChange) > 1)

# let's compare the summaries before and after setting a lfc cutoff:
summary(res, alpha=0.05)
```

```

## 
## out of 5622 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 832, 15%
## LFC < 0 (down)    : 815, 14%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

summary(res_lfc, alpha=0.05)

## 
## out of 354 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)      : 76, 21%
## LFC < 0 (down)    : 278, 79%
## outliers [1]       : 0, 0%
## low counts [2]     : 0, 0%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

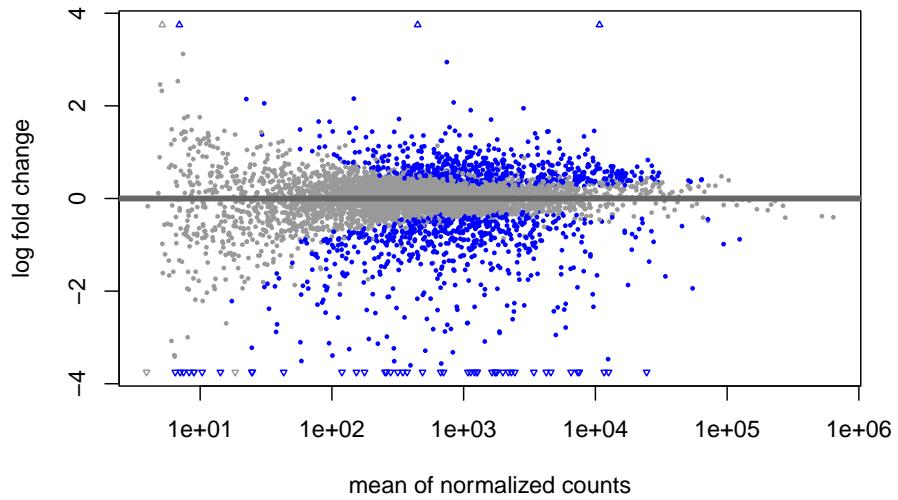
head(res_lfc)

## log2 fold change (MLE): 0,0,0,+1
## Wald test p-value: 0,0,0,+1
## DataFrame with 6 rows and 6 columns
##   baseMean log2FoldChange      lfcSE      stat      pvalue      padj
##   <numeric>      <numeric> <numeric> <numeric> <numeric> <numeric>
## YER091C 2846.4044      1.94771  0.267390  7.28414 3.23735e-13 8.42610e-12
## YJR127C  819.0282     -1.34748  0.150234 -8.96917 2.98759e-19 1.21712e-17
## YAL040C 3530.0007     -1.05979  0.141324 -7.49903 6.42942e-14 1.78941e-12
## YLR456W  57.3756      1.06661  0.333378  3.19940 1.37712e-03 6.76761e-03
## YMR173W  295.1969     -3.23614  0.268425 -12.05605 1.80230e-33 1.49008e-31
## YFR017C  252.8558     -5.39554  0.489053 -11.03265 2.65916e-28 1.84565e-26

```

Let's take a quick look at the differential expression

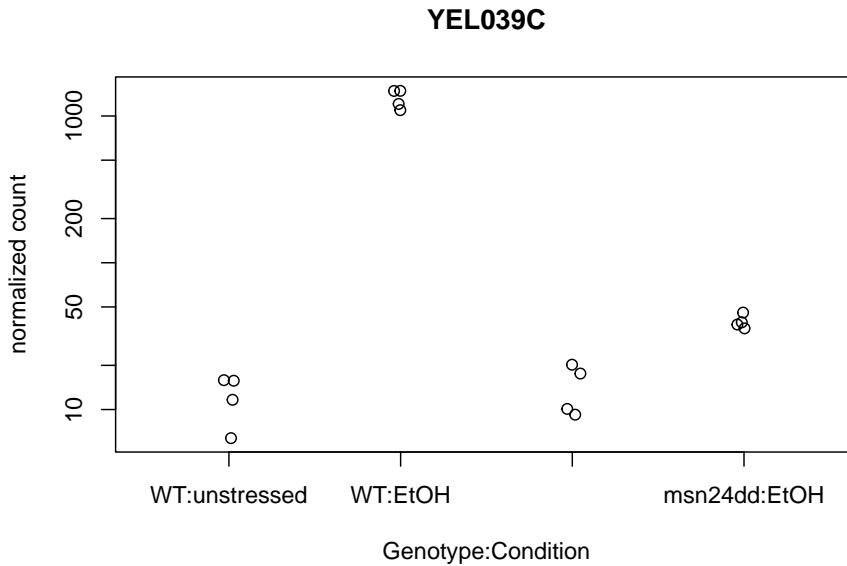
```
DESeq2::plotMA(res, alpha=0.01)
```



Plot an individual gene:

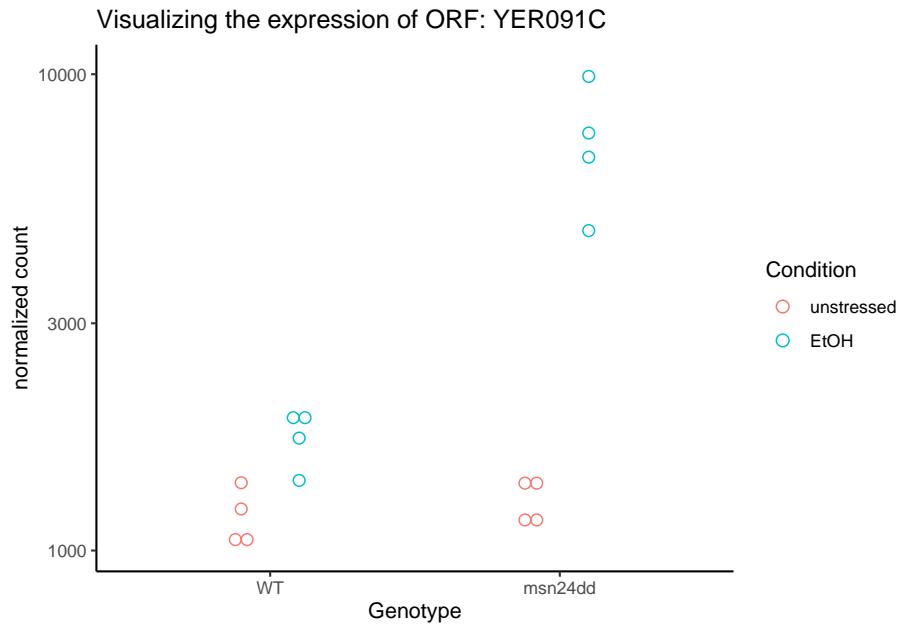
```
gene <- "YER091C"

# Here is the default visualization. Depending on screen size, the xlab
# might not show all of the groups.
plotCounts(dds, gene="YEL039C", intgroup=c("Genotype","Condition"),
           xlab="Genotype:Condition")
```



```
# Make the plot prettier with ggplot(). Note the returnData=TRUE let's us do this.
plotCounts(dds, gene=gene, intgroup=c("Genotype", "Condition"),
           xlab="Genotype:Condition", returnData = TRUE) %>%
  rownames_to_column("Sample") %>%
  ggplot(aes(x=Genotype, y=count, color=Condition, shape=Condition)) +
  geom_dotplot(binaxis = "y", stackdir = "center", dotsize=0.75,
                position=position_dodge(0.4), # this seperates by Condition a bit
                fill=NA) +
  labs(x="Genotype",
       y="normalized count",
       title=paste0("Visualizing the expression of ORF: ", gene))
  ) +
  scale_y_log10() +
  theme_classic()

## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```



We need to make sure and save our output file(s).

```
# Choose topTags destination
dir_output_DESeq2 <-
  path.expand("~/Desktop/Genomic_Data_Analysis/Analysis/DESeq2/")
if (!dir.exists(dir_output_DESeq2)) {
  dir.create(dir_output_DESeq2, recursive = TRUE)
}

# for sharing with others, a tsv for the res output is convenient.
# Depending on what people need, we can save res object as is or beautify it.
res %>%
  data.frame() %>%
  rownames_to_column("ORF") %>%
  left_join(AnnotationDbi::select(org.Sc.sgd.db, keys=.ORF, columns="GENENAME"), by="ORF")
  relocate(GENENAME, .after = ORF) %>%
  # arrange(padj) %>%
  # mutate(log2FoldChange = round(log2FoldChange, 2)) %>%
  # mutate(across(where(is.numeric), signif, 3)) %>%
  write_tsv(., file = paste0(dir_output_DESeq2, "yeast_res_DESeq2.tsv"))

## 'select()' returned 1:1 mapping between keys and columns
```

```
# for subsequent analysis, let's save the res object as an R data object.
saveRDS(object = res, file = paste0(dir_output_DESeq2, "yeast_res_DESeq2.Rds"))
```

## 7.7 Questions

Question 1: How many genes were upregulated and downregulated in the contrast we looked at in this activity? Be sure to clarify the cutoffs used for determining significance.

Question 2: Choose one of the contrasts in `my.contrasts` that we didn't test together, and identify the top 3 most differentially expressed genes.

Question 3: In the contrast you chose, give a brief description of the biological interpretation of that contrast.

Question 4: We analyzed differential expression of the counts generated by the full Salmon counts. Load in the counts generated by using the subset samples and look at the same contrast we did in class. What differences and similarities do you see?

A template for doing this is below:

```
path_subset_counts <- path.expand("~/Desktop/Genomic_Data_Analysis/Data/Counts/Salmon/salmon.genes.txt")

# If you don't have that file, uncomment the code below and run it instead.
# read.delim('https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/ethanol_stress/counts.txt')

subset_counts <- read.delim(file = path_subset_counts,
  sep = "\t",
  header = T,
  row.names = 1
)

# We are reusing the sample_metadata, group, etc that we assigned above

# create DESeqDataSet with salmon counts (round needed for nonintegers)
dds_subset <- DESeqDataSetFromMatrix(countData = round(subset_counts),
  colData = sample_metadata,
  design = ~ 1 + Genotype + Condition + Genotype:Condition)

## converting counts to integer mode
```

```

# simplify the column names to make them pretty
colnames(dds_subset) <- str_split_fixed(colnames(dds_subset), "\\.", n = 2)[, 1]

# filter low counts
keep_subset <- rowSums(counts(dds_subset) >= 10) >= smallestGroupSize
dds_subset <- dds_subset[keep_subset,]

# generate the fit
dds_subset <- DESeq(dds_subset)

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

# test our contrast of interest
res_subset <- results(dds_subset,
                      contrast = (MSN24dd_EtOH - MSN24_MOCK) - (WT_EtOH - WT_MOCK)
                     )

# generate a beautiful table for the pdf/html file.
res_subset %>%
  data.frame() %>%
  rownames_to_column("ORF") %>%
  # add the gene names
  left_join(AnnotationDbi::select(org.Sc.sgd.db, keys=.ORF, columns="GENENAME"), by="ORF")
  relocate(GENENAME, .after = ORF) %>%
  arrange(padj) %>%
  mutate(log2FoldChange = round(log2FoldChange, 2)) %>%
  mutate(across(where(is.numeric), signif, 3)) %>%
  reactable(
    searchable = TRUE,
    showSortable = TRUE,
    columns = list(ORF = colDef(
      cell = function(value) {
        # Render as a link
      }
    )
  )

```

```
url <-
  sprintf("https://www.yeastgenome.org/locus/%s", value)
  htmltools::tags$a(href = url, target = "_blank", as.character(value))
}
))
)

## 'select()' returned 1:1 mapping between keys and columns
```

ORF ↑	GENENA ME	baseMean	log2Fold Change	lfcSE	stat	↑
<a href="#">YKL035W</a>	UGP1	280	-4.04	0.236	-17.1	6
<a href="#">YPR149W</a>	NCE102	37	-4.35	0.452	-9.62	4
<a href="#">YML100W</a>	TSL1	142	-8.21	0.896	-9.17	7
<a href="#">YPL004C</a>	LSP1	71.9	-2.8	0.307	-9.12	3
<a href="#">YMR105C</a>	PGM2	133	-6.24	0.741	-8.42	3
<a href="#">YKL150W</a>	MCR1	57.2	-3.12	0.373	-8.37	5
<a href="#">YDR077W</a>	SED1	286	-1.53	0.186	-8.21	2
<a href="#">YOL155C</a>	HPF1	197	-1.62	0.222	-7.29	3
<a href="#">YJR045C</a>	SSC1	210	-1.28	0.177	-7.23	4
<a href="#">YBR126C</a>	TPS1	143	-2.75	0.382	-7.21	5

1–10 of 2542 rows      Previous      **1**    2    3    4    5    ...    255    Next

```
# summarize the DE genes
summary(res_subset, alpha=0.05)
```

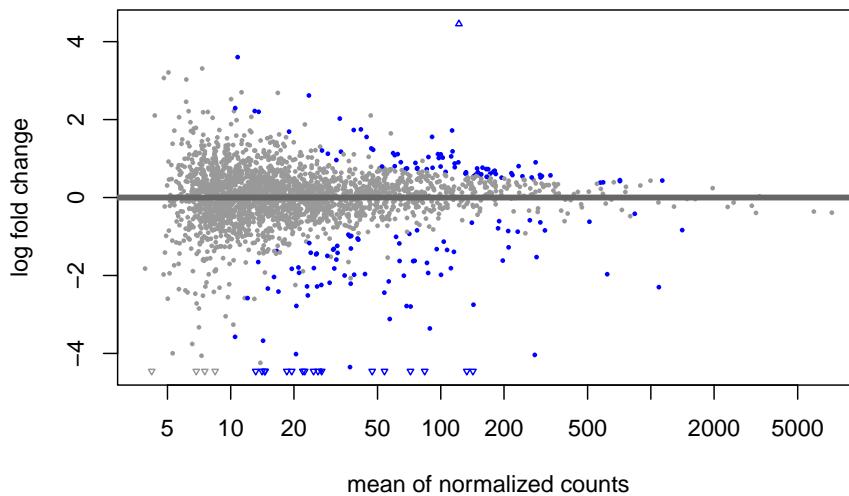
```
##
## out of 2542 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up) : 76, 3%
```

```

## LFC < 0 (down)      : 99, 3.9%
## outliers [1]        : 0, 0%
## low counts [2]      : 690, 27%
## (mean count < 10)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

# visualize results
DESeq2:::plotMA(res_subset, alpha=0.05)

```



Be sure to knit this file into a pdf or html file once you're finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

**R version 4.3.1 (2023-06-16)**

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8|en\_US.UTF-8||C||en\_US.UTF-8|en\_US.UTF-8

**attached base packages:** stats4, stats, graphics, grDevices, utils, datasets, methods and base

**other attached packages:** DESeq2(v.1.40.2), edgeR(v.3.42.4), limma(v.3.56.2), reactable(v.0.4.4), webshot2(v.0.1.1), statmod(v.1.5.0), Rsubread(v.2.14.2),

*ShortRead(v.1.58.0), GenomicAlignments(v.1.36.0), SummarizedExperiment(v.1.30.2), MatrixGenerics(v.1.12.3), matrixStats(v.1.0.0), Rsamtools(v.2.16.0), GenomicRanges(v.1.52.1), Biostrings(v.2.68.1), GenomeInfoDb(v.1.36.4), XVector(v.0.40.0), BiocParallel(v.1.34.2), Rfastp(v.1.10.0), org.Sc.sgd.db(v.3.17.0), AnnotationDbi(v.1.62.2), IRanges(v.2.34.1), S4Vectors(v.0.38.2), Biobase(v.2.60.0), BiocGenerics(v.0.46.0), clusterProfiler(v.4.8.2), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pander(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3), forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyverse(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)*

**loaded via a namespace (and not attached):** splines(v.4.3.1), later(v.1.3.1), bitops(v.1.0-7), ggplotify(v.0.1.2), polyclip(v.1.10-6), lifecycle(v.1.0.3), rprojroot(v.2.0.3), vroom(v.1.6.4), processx(v.3.8.2), lattice(v.0.21-9), MASS(v.7.3-60), crosstalk(v.1.2.0), magrittr(v.2.0.3), rmarkdown(v.2.25), yaml(v.2.3.7), cowplot(v.1.1.1), chromote(v.0.1.2), DBI(v.1.1.3), RColorBrewer(v.1.1-3), abind(v.1.4-5), zlibbioc(v.1.46.0), ggraph(v.2.1.0), RCurl(v.1.98-1.12), yulab.utils(v.0.1.0), tweenr(v.2.0.2), GenomeInfoDbData(v.1.2.10), enrichplot(v.1.20.0), ggrepel(v.0.9.4), codetools(v.0.2-19), DelayedArray(v.0.26.7), DOSE(v.3.26.1), ggforce(v.0.4.1), tidyselect(v.1.2.0), aplot(v.0.2.2), farver(v.2.1.1), viridis(v.0.6.4), webshot(v.0.5.5), jsonlite(v.1.8.7), ellipsis(v.0.3.2), tidygraph(v.1.2.3), tools(v.4.3.1), treeio(v.1.24.3), Rcpp(v.1.0.11), glue(v.1.6.2), gridExtra(v.2.3), xfun(v.0.40), qvalue(v.2.32.0), websocket(v.1.4.1), withr(v.2.5.1), fastmap(v.1.1.1), latticeExtra(v.0.6-30), fansi(v.1.0.5), digest(v.0.6.33), timechange(v.0.2.0), R6(v.2.5.1), gridGraphics(v.0.5-1), colorspace(v.2.1-0), GO.db(v.3.17.0), jpeg(v.0.1-10), RSQLite(v.2.3.1), utf8(v.1.2.3), generics(v.0.1.3), data.table(v.1.14.8), graphlayouts(v.1.0.1), httr(v.1.4.7), htmlwidgets(v.1.6.2), S4Arrays(v.1.0.6), scatterpie(v.0.2.1), pkgconfig(v.2.0.3), gtable(v.0.3.4), blob(v.1.2.4), hwriter(v.1.3.2.1), shadowtext(v.0.1.2), htmltools(v.0.5.6.1), bookdown(v.0.36), fgsea(v.1.26.0), scales(v.1.2.1), png(v.0.1-8), snakecase(v.0.11.1), ggrepel(v.0.1.3), rstudioapi(v.0.15.0), tzdb(v.0.4.0), reshape2(v.1.4.4), rjson(v.0.2.21), nlme(v.3.1-163), cachem(v.1.0.8), RVenn(v.1.1.0), parallel(v.4.3.1), HDO.db(v.0.99.1), pillar(v.1.9.0), grid(v.4.3.1), vctrs(v.0.6.4), promises(v.1.2.1), archive(v.1.1.5), evaluate(v.0.22), cli(v.3.6.1), locfit(v.1.5-9.8), compiler(v.4.3.1), rlang(v.1.1.1), crayon(v.1.5.2), interp(v.1.1-4), reactR(v.0.5.0), ps(v.1.7.5), plyr(v.1.8.9), fs(v.1.6.3), stringi(v.1.7.12), viridisLite(v.0.4.2), deldir(v.1.0-9), munsell(v.0.5.0), lazyeval(v.0.2.2), GOSemSim(v.2.26.1), Matrix(v.1.6-1.1), hms(v.1.1.3), patchwork(v.1.1.3), bit64(v.4.0.5), KEGGREST(v.1.40.1), memoise(v.2.0.1), ggtree(v.3.8.2), fastmatch(v.1.1-4), bit(v.4.0.5), downloader(v.0.4), ape(v.5.7-1) and gson(v.0.1.0)

# Chapter 8

## Differential Expression: limma

last updated: 2023-10-27

### Install Packages

As usual, make sure we have the right packages for this exercise

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# let's load all of the files we were using and want to have again today
p_load("tidyverse", "knitr", "readr",
       "pander", "BiocManager",
       "dplyr", "stringr",
       "statmod", # required dependency, need to load manually on some macOS versions.
       "Glimma", # beautifies limma results
       "purrr", # for working with lists (beautify column names)
       "reactable") # for pretty tables.

# We also need these Bioconductor packages today.
p_load("edgeR", "AnnotationDbi", "org.Sc.sgd.db", "ggVennDiagram")
#NOTE: edgeR loads limma as a dependency
```

### 8.1 Description

This will be our last differential expression analysis workflow, converting gene counts across samples into meaningful information about genes that appear to be significantly differentially expressed between samples

## 8.2 Learning Objectives

At the end of this exercise, you should be able to:

- Generate a table of sample metadata.
- Filter low counts and normalize count data.
- Utilize the limma package to identify differentially expressed genes.

```
library(limma)
library(org.Sc.sgd.db)
# for ease of use, set max number of digits after decimal
options(digits=3)
```

## 8.3 Loading in the count data file

We are downloading the counts for the non-subsampled fastq files from a Github repository using the code below. Just as in previous exercises, assign the data to the variable `counts`. You can change the file path if you have saved it to your computer in a different location.

```
counts <- read.delim('https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/
  sep = "\t",
  header = T,
  row.names = 1
)
```

If you don't have that file for any reason, the below code chunk will load a copy of it from Github.

To find the order of files we need, we can get just the part of the column name before the first “.” symbol with this command:

```
str_split_fixed(counts %>% colnames(), "\\\.", n = 2)[, 1]
```

```
## [1] "YPS606_MSN24_ETOH_REP1_R1" "YPS606_MSN24_ETOH_REP2_R1"
## [3] "YPS606_MSN24_ETOH_REP3_R1" "YPS606_MSN24_ETOH_REP4_R1"
## [5] "YPS606_MSN24_MOCK_REP1_R1" "YPS606_MSN24_MOCK_REP2_R1"
## [7] "YPS606_MSN24_MOCK_REP3_R1" "YPS606_MSN24_MOCK_REP4_R1"
## [9] "YPS606_WT_ETOH_REP1_R1"     "YPS606_WT_ETOH_REP2_R1"
## [11] "YPS606_WT_ETOH_REP3_R1"    "YPS606_WT_ETOH_REP4_R1"
## [13] "YPS606_WT_MOCK_REP1_R1"   "YPS606_WT_MOCK_REP2_R1"
## [15] "YPS606_WT_MOCK_REP3_R1"   "YPS606_WT_MOCK_REP4_R1"
```

```

sample_metadata <- tribble(
  ~Sample,           ~Genotype,      ~Condition,
  "YPS606_MSN24_ETOH_REP1_R1", "msn24dd", "EtOH",
  "YPS606_MSN24_ETOH_REP2_R1", "msn24dd", "EtOH",
  "YPS606_MSN24_ETOH_REP3_R1", "msn24dd", "EtOH",
  "YPS606_MSN24_ETOH_REP4_R1", "msn24dd", "EtOH",
  "YPS606_MSN24_MOCK_REP1_R1", "msn24dd", "unstressed",
  "YPS606_MSN24_MOCK_REP2_R1", "msn24dd", "unstressed",
  "YPS606_MSN24_MOCK_REP3_R1", "msn24dd", "unstressed",
  "YPS606_MSN24_MOCK_REP4_R1", "msn24dd", "unstressed",
  "YPS606_WT_ETOH_REP1_R1",    "WT",        "EtOH",
  "YPS606_WT_ETOH_REP2_R1",    "WT",        "EtOH",
  "YPS606_WT_ETOH_REP3_R1",    "WT",        "EtOH",
  "YPS606_WT_ETOH_REP4_R1",    "WT",        "EtOH",
  "YPS606_WT_MOCK_REP1_R1",    "WT",        "unstressed",
  "YPS606_WT_MOCK_REP2_R1",    "WT",        "unstressed",
  "YPS606_WT_MOCK_REP3_R1",    "WT",        "unstressed",
  "YPS606_WT_MOCK_REP4_R1",    "WT",        "unstressed") %>%
# Create a new column that combines the Genotype and Condition value
mutate(Group = factor(
  paste(Genotype, Condition, sep = "."),
  levels = c(
    "WT.unstressed", "WT.EtOH",
    "msn24dd.unstressed", "msn24dd.EtOH"
  )
)) %>%
# make Condition and Genotype a factor (with baseline as first level) for edgeR
mutate(
  Genotype = factor(Genotype,
                     levels = c("WT", "msn24dd")),
  Condition = factor(Condition,
                     levels = c("unstressed", "EtOH"))
)

```

Now, let's create a design matrix with this information

```

group <- sample_metadata$Group
design <- model.matrix(~ 0 + group)

# beautify column names
colnames(design) <- levels(group)
design

##      WT.unstressed WT.EtOH msn24dd.unstressed msn24dd.EtOH

```

```

## 1      0      0      0      1
## 2      0      0      0      1
## 3      0      0      0      1
## 4      0      0      0      1
## 5      0      0      1      0
## 6      0      0      1      0
## 7      0      0      1      0
## 8      0      0      1      0
## 9      0      1      0      0
## 10     0      1      0      0
## 11     0      1      0      0
## 12     0      1      0      0
## 13     1      0      0      0
## 14     1      0      0      0
## 15     1      0      0      0
## 16     1      0      0      0
## attr(),"assign")
## [1] 1 1 1 1
## attr(),"contrasts")
## attr(),"contrasts")$group
## [1] "contr.treatment"

```

## 8.4 Count loading and Annotation

The count matrix is used to construct a DGEList class object. This is the main data class in the edgeR package. The DGEList object is used to store all the information required to fit a generalized linear model to the data, including library sizes and dispersion estimates as well as counts for each gene.

```

y <- DGEList(counts, group=group)
colnames(y) <- sample_metadata$Sample
y$samples

##                                     group lib.size norm.factors
## YPS606_MSN24_ETOH_REP1_R1    msn24dd.EtOH 17409481      1
## YPS606_MSN24_ETOH_REP2_R1    msn24dd.EtOH 14055425      1
## YPS606_MSN24_ETOH_REP3_R1    msn24dd.EtOH 13127876      1
## YPS606_MSN24_ETOH_REP4_R1    msn24dd.EtOH 16655559      1
## YPS606_MSN24_MOCK_REP1_R1   msn24dd.unstressed 12266723      1
## YPS606_MSN24_MOCK_REP2_R1   msn24dd.unstressed 11781244      1
## YPS606_MSN24_MOCK_REP3_R1   msn24dd.unstressed 11340274      1
## YPS606_MSN24_MOCK_REP4_R1   msn24dd.unstressed 13024330      1
## YPS606_WT_ETOH_REP1_R1       WT.EtOH 15422048      1
## YPS606_WT_ETOH_REP2_R1       WT.EtOH 14924728      1

```

```

## YPS606_WT_ETOH_REP3_R1           WT.EtOH 14738753    1
## YPS606_WT_ETOH_REP4_R1           WT.EtOH 12203133    1
## YPS606_WT_MOCK_REP1_R1          WT.unstressed 13592206    1
## YPS606_WT_MOCK_REP2_R1          WT.unstressed 12921965    1
## YPS606_WT_MOCK_REP3_R1          WT.unstressed 13128396    1
## YPS606_WT_MOCK_REP4_R1          WT.unstressed 15568155    1

```

Human-readable gene symbols can also be added to complement the gene ID for each gene, using the annotation in the org.Sc.sgd.db package.

```

y$genes <- AnnotationDbi::select(org.Sc.sgd.db, keys=rownames(y), columns="GENENAME")

## 'select()' returned 1:1 mapping between keys and columns

head(y$genes)

##      ORF      SGD GENENAME
## 1 YIL170W S000001432   HXT12
## 2 YIL175W S000001437     <NA>
## 3 YPL276W S000006197     <NA>
## 4 YFL056C S000001838   AAD6
## 5 YCL074W S000000579     <NA>
## 6 YAR061W S000000087     <NA>

```

## 8.5 Filtering to remove low counts

Genes with very low counts across all libraries provide little evidence for differential expression. In addition, the pronounced discreteness of these counts interferes with some of the statistical approximations that are used later in the pipeline. These genes should be filtered out prior to further analysis. Here, we will retain a gene only if it is expressed at a count-per-million (CPM) above 60 in at least four samples.

```

keep <- rowSums(cpm(y) > 0.7) >= 4
y <- y[keep,]
summary(keep)

```

```

##      Mode   FALSE    TRUE
## logical     956    5615

```

Where did those cutoff numbers come from?

As a general rule, we don't want to exclude a gene that is expressed in only one group, so a cutoff number equal to the number of replicates can be a good starting point. For counts, a good threshold can be chosen by identifying the CPM that corresponds to a count of 10, which in this case would be about 60 (due to our fastq files being subsets of the full reads):

```
cpm(10, mean(y$samples$lib.size))
```

```
##      [,1]
## [1,] 0.72
```

Smaller CPM thresholds are usually appropriate for larger libraries.

## 8.6 Normalization for composition bias

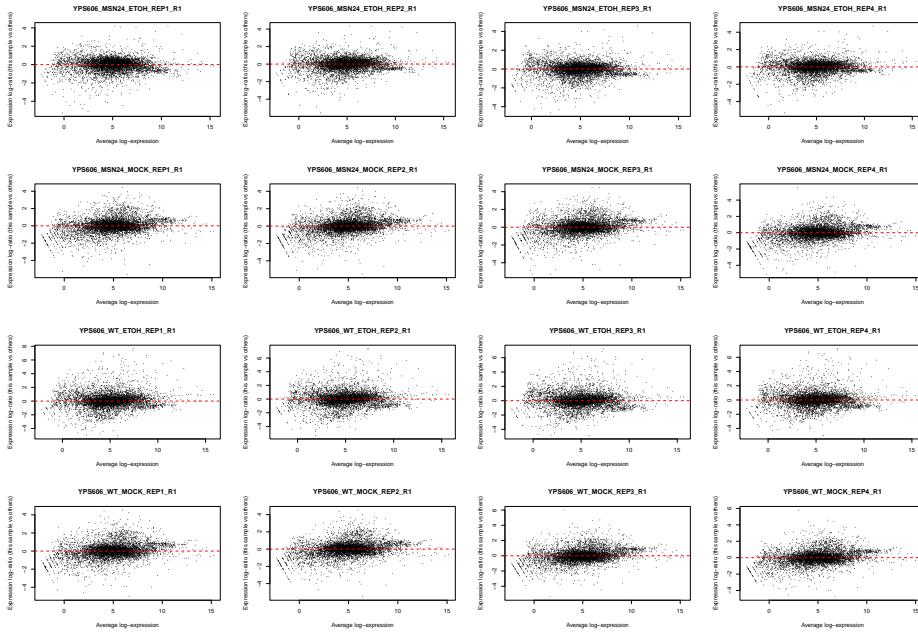
TMM normalization is performed to eliminate composition biases between libraries. This generates a set of normalization factors, where the product of these factors and the library sizes defines the effective library size. The calcNormFactors function returns the DGEList argument with only the norm.factors changed.

```
y <- calcNormFactors(y)
y$samples
```

	group	lib.size	norm.factors
## YPS606_MSN24_ETOH_REP1_R1	msn24dd.EtOH	17409481	1.239
## YPS606_MSN24_ETOH_REP2_R1	msn24dd.EtOH	14055425	1.102
## YPS606_MSN24_ETOH_REP3_R1	msn24dd.EtOH	13127876	1.108
## YPS606_MSN24_ETOH_REP4_R1	msn24dd.EtOH	16655559	1.007
## YPS606_MSN24_MOCK_REP1_R1	msn24dd.unstressed	12266723	1.038
## YPS606_MSN24_MOCK_REP2_R1	msn24dd.unstressed	11781244	1.003
## YPS606_MSN24_MOCK_REP3_R1	msn24dd.unstressed	11340274	0.960
## YPS606_MSN24_MOCK_REP4_R1	msn24dd.unstressed	13024330	0.984
## YPS606_WT_ETOH_REP1_R1	WT.EtOH	15422048	0.839
## YPS606_WT_ETOH_REP2_R1	WT.EtOH	14924728	0.941
## YPS606_WT_ETOH_REP3_R1	WT.EtOH	14738753	0.988
## YPS606_WT_ETOH_REP4_R1	WT.EtOH	12203133	0.971
## YPS606_WT_MOCK_REP1_R1	WT.unstressed	13592206	0.990
## YPS606_WT_MOCK_REP2_R1	WT.unstressed	12921965	1.038
## YPS606_WT_MOCK_REP3_R1	WT.unstressed	13128396	0.900
## YPS606_WT_MOCK_REP4_R1	WT.unstressed	15568155	0.951

The normalization factors multiply to unity across all libraries. A normalization factor below unity indicates that the library size will be scaled down, as there is more suppression (i.e., composition bias) in that library relative to the other libraries. This is also equivalent to scaling the counts upwards in that sample. Conversely, a factor above unity scales up the library size and is equivalent to downscaling the counts. The performance of the TMM normalization procedure can be examined using mean-difference (MD) plots. This visualizes the library size-adjusted log-fold change between two libraries (the difference) against the average log-expression across those libraries (the mean). The below command plots an MD plot, comparing sample 1 against an artificial library constructed from the average of all other samples.

```
for (sample in 1:nrow(y$samples)) {
  plotMD(cpm(y, log=TRUE), column=sample)
  abline(h=0, col="red", lty=2, lwd=2)
}
```



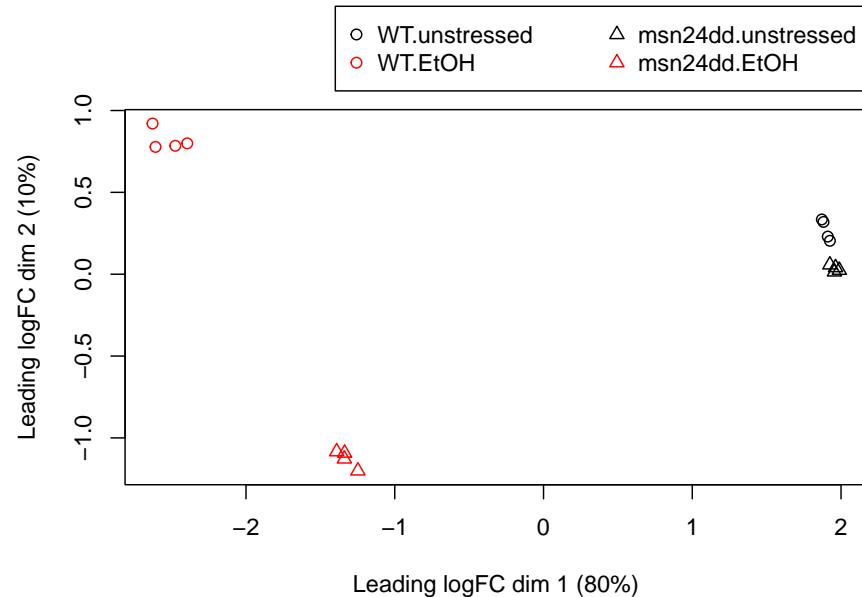
## 8.7 Exploring differences between libraries

The data can be explored by generating multi-dimensional scaling (MDS) plots. This visualizes the differences between the expression profiles of different samples in two dimensions. The next plot shows the MDS plot for the yeast heatshock data.

```

points <- c(1,1,2,2)
colors <- rep(c("black", "red"),8)
plotMDS(y, col=colors[group], pch=points[group])
# legend("bottomright", legend=levels(group),
#        # pch=points, col=colors, ncol=2)
legend("bottomright", legend=levels(group),
       pch=points, col=colors, ncol=2,
       inset=c(0,1.05), xpd=TRUE)

```

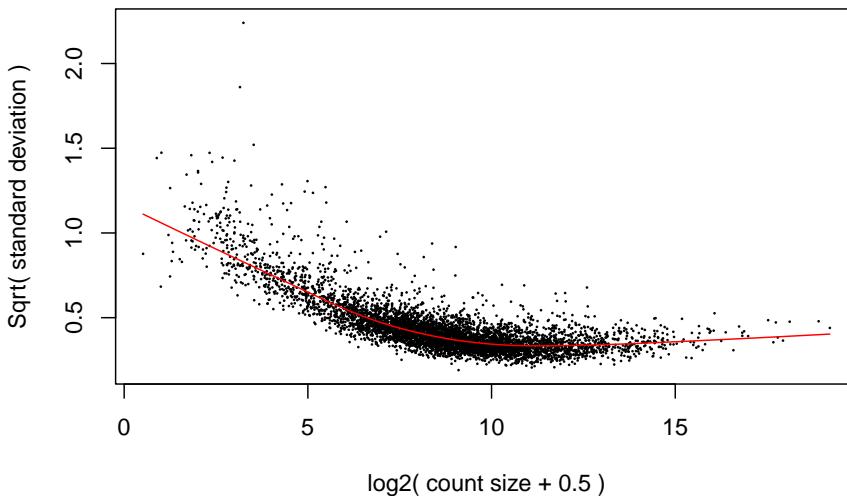


## 8.8 Estimate Dispersion

This is the first step in a limma analysis that differs from the edgeR workflow.

```
y <- voom(y, design, plot = T)
```

### voom: Mean-variance trend



```
# compare this to the edgeR function estimateDisp, which uses a NB distribution.
# y <- estimateDisp(y, design, robust=TRUE)
# plotBCV(y)
```

What is `voom` doing?

- Counts are transformed to log2 counts per million reads (CPM), where “per million reads” is defined based on the normalization factors we calculated earlier
- A linear model is fitted to the log2 CPM for each gene, and the residuals are calculated
- A smoothed curve is fitted to the  $\text{sqrt}(\text{residual standard deviation})$  by average expression (see red line in plot above)
- The smoothed curve is used to obtain weights for each gene and sample that are passed into limma along with the log2 CPMs.

Limma uses the `lmFit` function. This returns a `MArrayLM` object containing the weighted least squares estimates for each gene.

```
fit <- lmFit(y, design)
head(coef(fit))
```

```

##          WT.unstressed WT.EtOH msn24dd.unstressed msn24dd.EtOH
## YIL170W      -2.154    0.936        -3.239     0.851
## YFL056C       3.921    4.044        3.958     4.888
## YAR061W       0.135    0.746        0.666     0.641
## YGR014W       7.666    7.319        7.796     7.436
## YPR031W       4.711    2.735        4.857     2.818
## YIL003W       4.589    2.530        4.468     2.662

# edgeR equivalent
# fit <- glmQLFit(y, design, robust=TRUE)
# head(fit$coefficients)
# plotQLDisp(fit)

```

Comparisons between groups (log fold-changes) are obtained as *contrasts* of these fitted linear models:

## 8.9 Testing for differential expression

The final step is to actually test for significant differential expression in each gene, using the QL F-test. The contrast of interest can be specified using the `makeContrasts` function in limma, the same one that is used by edgeR.

```

# generate contrasts we are interested in learning about
my.contrasts <- makeContrasts(EtOHvsMOCK.WT = WT.EtOH - WT.unstressed,
                               EtOHvsMOCK.MSN24dd = msn24dd.EtOH - msn24dd.unstressed,
                               EtOH.MSN24ddvsWT = msn24dd.EtOH - WT.EtOH,
                               MOCK.MSN24ddvsWT = msn24dd.unstressed - WT.unstressed,
                               EtOHvsWT.MSN24ddvsWT = (msn24dd.EtOH-msn24dd.unstressed)-(WT.EtOH
                               levels=design)

# fit the linear model to these contrasts
res_all <- contrasts.fit(fit, my.contrasts)

# This looks at all of our contrasts in my.contrasts
res_all <- eBayes(res_all)

# eBayes is the alternative to glmQLFTest in edgeR
# This contrast looks at the difference in the stress responses between mutant and WT
# res <- glmQLFTest(fit, contrast = my.contrasts)

top.table <- topTable(res_all, sort.by = "F", n = Inf)
head(top.table, 20)

```

```

##          ORF      SGD GENENAME EtOHvsMOCK.WT EtOHvsMOCK.MSN24dd
## YER103W YER103W S000000905    SSA4      7.77      7.122
## YDR516C YDR516C S000002924    EMI2      7.03      3.031
## YCL040W YCL040W S000000545    GLK1      8.51      6.833
## YMR105C YMR105C S000004711    PGM2      7.62      0.792
## YLL039C YLL039C S000003962    UBI4      5.75      3.840
## YJL052W YJL052W S000003588    TDH1      10.02     9.028
## YOR317W YOR317W S000005844    FAA1      5.36      4.624
## YBL039C YBL039C S000000135    URA7      -6.93     -5.470
## YGL037C YGL037C S000003005    PNC1      6.10      3.849
## YHR104W YHR104W S000001146    GRE3      4.94      2.519
## YGR254W YGR254W S000003486    ENO1      7.83      7.590
## YBR126C YBR126C S000000330    TPS1      5.36      1.908
## YPL012W YPL012W S000005933    RRP12     -5.12     -4.315
## YDR399W YDR399W S000002807    HPT1      -5.12     -5.460
## YHR170W YHR170W S000001213    NMD3      -4.26     -3.542
## YLR258W YLR258W S000004248    GSY2      7.54      2.699
## YGR159C YGR159C S000003391    NSR1      -6.88     -5.983
## YMR196W YMR196W S000004809    <NA>      7.36      2.198
## YLL026W YLL026W S000003949    HSP104     5.70      3.659
## YML100W YML100W S000004566    TSL1      7.79      0.658
##          EtOH.MSN24ddvsWT MOCK.MSN24ddvsWT EtOHvsWT.MSN24ddvsWT AveExpr   F
## YER103W      -0.797      -0.15215     -0.645      7.81 3407
## YDR516C      -4.710      -0.71026     -4.000      6.13 2659
## YCL040W      -2.077      -0.39710     -1.680      8.06 2600
## YMR105C      -6.969      -0.14072     -6.829      6.08 2204
## YLL039C      -2.135      -0.22780     -1.907      7.23 2067
## YJL052W      -1.209      -0.22146     -0.988      8.83 2000
## YOR317W      -0.979      -0.24259     -0.736      7.06 1964
## YBL039C      1.423       0.03529      1.458      6.17 1942
## YGL037C      -2.856      -0.60381     -2.252      6.60 1920
## YHR104W      -2.568      -0.14350     -2.424      6.99 1910
## YGR254W      -0.725      -0.48483     -0.240      10.64 1849
## YBR126C      -3.646      -0.19755     -3.448      7.99 1789
## YPL012W       0.814       0.00468      0.809      6.77 1761
## YDR399W      -0.422      -0.08242     -0.339      6.27 1697
## YHR170W       0.687      -0.03394      0.721      6.27 1675
## YLR258W      -5.233      -0.38790     -4.845      5.01 1626
## YGR159C       0.761      -0.13994      0.901      7.21 1606
## YMR196W      -4.953       0.20940     -5.162      5.42 1604
## YLL026W      -1.492       0.54538     -2.038      7.84 1571
## YML100W      -7.508      -0.37240     -7.136      5.91 1557
##          P.Value adj.P.Val
## YER103W 3.36e-30  1.89e-26
## YDR516C 5.53e-29  1.33e-25
## YCL040W 7.11e-29  1.33e-25

```

```
## YMR105C 4.59e-28  6.45e-25
## YLL039C 9.49e-28  1.07e-24
## YJL052W 1.37e-27  1.29e-24
## YOR317W 1.68e-27  1.29e-24
## YBL039C 1.91e-27  1.29e-24
## YGL037C 2.17e-27  1.29e-24
## YHR104W 2.31e-27  1.29e-24
## YGR254W 3.32e-27  1.70e-24
## YBR126C 4.84e-27  2.27e-24
## YPL012W 5.77e-27  2.49e-24
## YDR399W 8.75e-27  3.51e-24
## YHR170W 1.01e-26  3.80e-24
## YLR258W 1.42e-26  4.97e-24
## YGR159C 1.63e-26  5.16e-24
## YMR196W 1.65e-26  5.16e-24
## YLL026W 2.08e-26  6.15e-24
## YML100W 2.31e-26  6.47e-24
```

```
top.table %>%
  tibble() %>%
  arrange(adj.P.Val) %>%
  mutate(across(where(is.numeric), signif, 3)) %>%
  reactable()
```

ORF	SGD	GENENAME	EtOHvsMO CK.WT	EtOHvsMO CK.MSN24 dd	EtOH.MSN 24ddvsWT	MO N24
YER103W	S000000905	SSA4	7.77	7.12	-0.797	
YDR516C	S000002924	EMI2	7.03	3.03	-4.71	
YCL040W	S000000545	GLK1	8.51	6.83	-2.08	
YMR105C	S000004711	PGM2	7.62	0.792	-6.97	
YLL039C	S000003962	UBI4	5.75	3.84	-2.13	
YJL052W	S000003588	TDH1	10	9.03	-1.21	
YOR317W	S000005844	FAA1	5.36	4.62	-0.979	
YBL039C	S000000135	URA7	-6.93	-5.47	1.42	
YGL037C	S000003005	PNC1	6.1	3.85	-2.86	
YHR104W	S000001146	GRE3	4.94	2.52	-2.57	

1–10 of 5615 rows

Previous **1** 2 3 4 5 ... 562 Next

```
# edgeR equivalent below:

# let's take a quick look at the results
# topTags(res, n=10)
#
## generate a beautiful table for the pdf/html file.
# topTags(res, n=Inf) %>% data.frame() %>%
```

```
#   arrange(FDR) %>%
#   mutate(logFC=round(logFC,2)) %>%
#   mutate(across(where(is.numeric), signif, 3)) %>%
#   reactable()
```

```
# Let's see how many genes in total are significantly different in any contrast
length(which(top.table$adj.P.Val < 0.05))
```

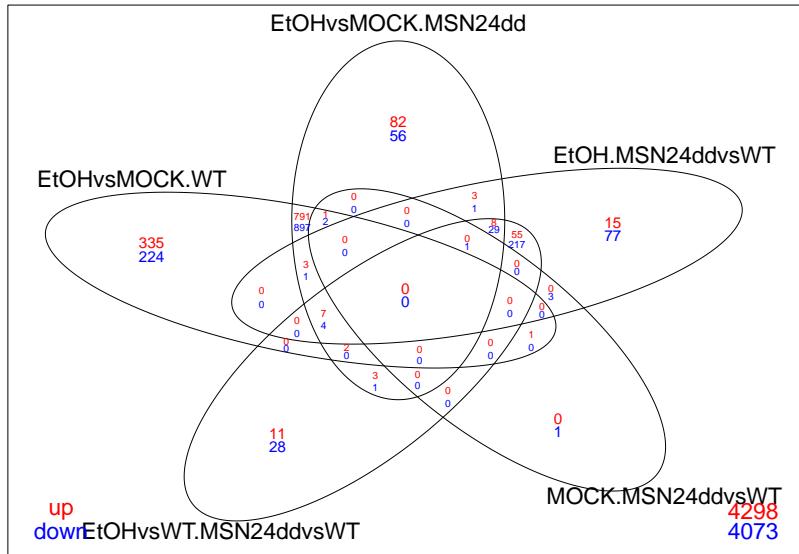
```
## [1] 4911
```

```
# let's summarize this and break it down by contrast.
res_all %>%
  decideTests(p.value = 0.05, lfc = 0) %>%
  summary()
```

	EtOHvsMOCK.WT	EtOHvsMOCK.MSN24dd	EtOH.MSN24ddvsWT	MOCK.MSN24ddvsWT
## Down	2260	2145	1247	10
## NotSig	1102	1285	2920	5595
## Up	2253	2185	1448	10
## EtOHvsWT.MSN24ddvsWT				
## Down		756		
## NotSig		4065		
## Up		794		

```
# we can save the decideTests output for graphing
decide_tests_res_all_limma <- res_all %>%
  decideTests(p.value = 0.05, lfc = 0)

# Bonus: limma allows us to create a venn diagram of these contrasts
# up & downregulated genes
res_all %>%
  decideTests(p.value = 0.05, lfc = 1) %>%
  vennDiagram(include=c("up", "down"),
              lwd=0.75,
              mar=rep(2,4), # increase margin size
              counts.col= c("red", "blue"),
              show.include=TRUE)
```



## 8.10 Examining a specific contrast

It is interesting to see all of the contrasts simultaneously, but often we may want to look at just a single contrast (and get the corresponding probabilities). Here is how we do that:

```
# fit the linear model to these contrasts
res <- contrasts.fit(fit, my.contrasts[, "EtOHvsWT.MSN24ddvsWT"])

# This contrast looks at the difference in the stress responses between mutant and WT
res <- eBayes(res)

# Note that there is no longer an "F" column, because we only look at one contrast.
top.table <- topTable(res, sort.by = "P", n = Inf)
head(top.table, 20)
```

##	ORF	SGD	GENENAME	logFC	AveExpr	t	P.Value	adj.P.Val	B
##	YMR105C	YMR105C	S000004711	PGM2	-6.83	6.08	-39.1	2.72e-22	1.53e-18
##	YKL035W	YKL035W	S000001518	UGP1	-3.83	9.33	-33.5	8.75e-21	2.46e-17
##	YML100W	YML100W	S000004566	TSL1	-7.14	5.91	-32.2	2.12e-20	3.96e-17
##	YBR126C	YBR126C	S000000330	TPS1	-3.45	7.99	-28.7	2.66e-19	3.52e-16
##	YPR149W	YPR149W	S000006353	NCE102	-4.25	7.34	-28.5	3.13e-19	3.52e-16

```

## YMR196W YMR196W S000004809 <NA> -5.16 5.42 -26.4 1.70e-18 1.59e-15 32.1
## YDR516C YDR516C S000002924 EMI2 -4.00 6.13 -26.2 2.03e-18 1.63e-15 32.1
## YKL150W YKL150W S000001633 MCR1 -2.93 7.61 -25.2 4.48e-18 3.14e-15 31.5
## YPL004C YPL004C S000005925 LSP1 -2.77 8.09 -25.1 5.06e-18 3.15e-15 31.4
## YDR001C YDR001C S000002408 NTH1 -2.89 6.09 -24.6 7.76e-18 4.36e-15 31.0
## YFR053C YFR053C S000001949 HXK1 -7.63 4.07 -22.7 4.47e-17 2.28e-14 27.7
## YHR104W YHR104W S000001146 GRE3 -2.42 6.99 -21.8 1.13e-16 5.28e-14 28.3
## YER053C YER053C S000000855 PIC2 -4.95 4.63 -21.6 1.37e-16 5.68e-14 27.8
## YHL021C YHL021C S000001013 AIM17 -4.19 4.94 -21.5 1.42e-16 5.68e-14 28.0
## YLR258W YLR258W S000004248 GSY2 -4.85 5.01 -21.4 1.64e-16 6.13e-14 27.6
## YDR074W YDR074W S000002481 TPS2 -2.33 7.40 -19.6 1.11e-15 3.90e-13 26.0
## YDR258C YDR258C S000002666 HSP78 -4.47 4.96 -19.4 1.28e-15 4.24e-13 25.9
## YDR342C YDR342C S000002750 HXT7 -5.92 5.97 -18.7 2.95e-15 9.21e-13 25.1
## YGR008C YGR008C S000003240 STF2 -5.20 3.59 -17.7 9.30e-15 2.75e-12 23.2
## YGR088W YGR088W S000003320 CTT1 -6.16 3.75 -17.6 1.05e-14 2.95e-12 23.5

```

```

top.table %>%
  tibble() %>%
  arrange(adj.P.Val) %>%
  mutate(across(where(is.numeric), signif, 3)) %>%
  reactable()

```

ORF	SGD	GENENAME	logFC	AveExpr	t	
YMR105C	S000004711	PGM2	-6.83	6.08	-39.1	2
YKL035W	S000001518	UGP1	-3.83	9.33	-33.5	8
YML100W	S000004566	TSL1	-7.14	5.91	-32.2	2
YBR126C	S000000330	TPS1	-3.45	7.99	-28.7	2
YPR149W	S000006353	NCE102	-4.25	7.34	-28.5	2
YMR196W	S000004809		-5.16	5.42	-26.4	
YDR516C	S000002924	EMI2	-4	6.13	-26.2	2
YKL150W	S000001633	MCR1	-2.93	7.61	-25.2	4
YPL004C	S000005925	LSP1	-2.77	8.09	-25.1	5
YDR001C	S000002408	NTH1	-2.89	6.09	-24.6	7

1–10 of 5615 rows

Previous 1 2 3 4 5 ... 562 Next

```

is.de <- decideTests(res, p.value=0.05)
summary(is.de)

##      [,1]
## Down    756
## NotSig 4065
## Up     794

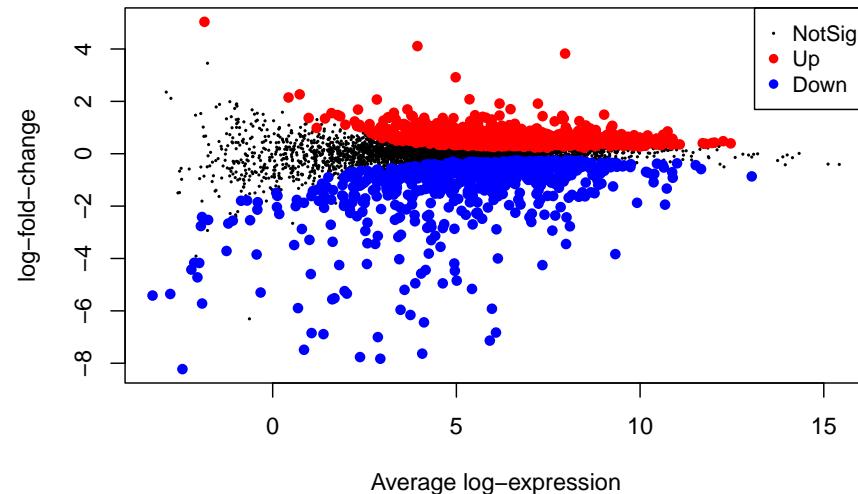
```

## 8.11 Visualization

We can visualize limma results using some built-in limma functions.

### 8.11.1 MA lot

```
# visualize results
limma::plotMA(res, status=is.de)
```



We need to make sure and save our output file(s).

```
# Choose topTags destination
dir_output_limma <-
  path.expand("~/Desktop/Genomic_Data_Analysis/Analysis/limma/")
if (!dir.exists(dir_output_limma)) {
  dir.create(dir_output_limma, recursive = TRUE)
}

# for sharing with others, the topTags output is convenient.
top.table %>% tibble() %>%
  arrange(desc(adj.P.Val)) %>%
  mutate(adj.P.Val = round(adj.P.Val, 2)) %>%
  mutate(across(where(is.numeric), signif, 3)) %>%
```

```

write_tsv(., file = paste0(dir_output_limma, "yeast_topTags_limma.tsv"))

# for subsequent analysis, let's save the res object as an R data object.
saveRDS(object = res, file = paste0(dir_output_limma, "yeast_res_limma.Rds"))

# we might also want our y object list
saveRDS(object = y, file = paste0(dir_output_limma, "yeast_y_limma.Rds"))

```

## 8.12 treat() testing

We can use the `limma` command `treat()` to test against a fold-change cutoff. `res` (or `fit`) can be either before or after `eBayes` has been run. Note that we need to use

```

lfc1_res <- treat(res,
                    lfc=1,
                    robust = TRUE)
# treat is a limma command that can be run on fit
lfc1_top.table <- topTreat(lfc1_res, n=Inf, p.value=0.05)

# print the genes with DE significantly beyond the cutoff
lfc1_top.table

```

##	ORF	SGD	GENENAME	logFC	AveExpr	t	P.Value	adj.P.Val
## YMR105C	YMR105C	S000004711	PGM2	-6.83	6.078	-33.74	2.77e-23	1.55e-19
## YKL035W	YKL035W	S000001518	UGP1	-3.83	9.326	-24.66	7.44e-20	2.09e-16
## YML100W	YML100W	S000004566	TSL1	-7.14	5.910	-27.70	1.92e-19	3.59e-16
## YMR196W	YMR196W	S000004809	<NA>	-5.16	5.424	-21.35	2.66e-18	3.73e-15
## YBR126C	YBR126C	S000000330	TPS1	-3.45	7.986	-20.39	8.23e-18	8.56e-15
## YPR149W	YPR149W	S000006353	NCE102	-4.25	7.342	-22.00	9.14e-18	8.56e-15
## YFR053C	YFR053C	S000001949	HXK1	-7.63	4.070	-19.81	1.66e-17	1.33e-14
## YDR516C	YDR516C	S000002924	EMI2	-4.00	6.129	-19.37	2.87e-17	2.02e-14
## YER053C	YER053C	S000000855	PIC2	-4.95	4.629	-17.28	4.60e-16	2.87e-13
## YLR258W	YLR258W	S000004248	GSY2	-4.85	5.009	-17.00	6.78e-16	3.81e-13
## YKL150W	YKL150W	S000001633	MCR1	-2.93	7.612	-16.69	1.05e-15	5.38e-13
## YHL021C	YHL021C	S000001013	AIM17	-4.19	4.944	-16.57	1.24e-15	5.80e-13
## YPL004C	YPL004C	S000005925	LSP1	-2.77	8.090	-16.01	2.82e-15	1.22e-12
## YDR001C	YDR001C	S000002408	NTH1	-2.89	6.090	-15.85	3.59e-15	1.44e-12
## YGR008C	YGR008C	S000003240	STF2	-5.20	3.589	-14.07	5.82e-14	2.18e-11
## YGR088W	YGR088W	S000003320	CTT1	-6.16	3.749	-14.84	9.98e-14	3.50e-11
## YDR258C	YDR258C	S000002666	HSP78	-4.47	4.959	-15.08	1.25e-13	4.13e-11
## YHR104W	YHR104W	S000001146	GRE3	-2.42	6.988	-12.54	7.96e-13	2.48e-10
## YMR250W	YMR250W	S000004862	GAD1	-4.58	4.042	-12.43	9.74e-13	2.88e-10

## YLR177W	YLR177W S000004167	<NA>	-3.55	4.568	-11.79	8.56e-12	2.40e-09
## YHR087W	YHR087W S000001129	RTC3	-7.76	2.377	-11.68	1.03e-11	2.66e-09
## YDR074W	YDR074W S000002481	TPS2	-2.33	7.398	-11.16	1.04e-11	2.66e-09
## YBR085C-A	YBR085C-A S000007522	<NA>	-2.84	4.630	-10.87	1.82e-11	4.45e-09
## YBR072W	YBR072W S000000276	HSP26	-4.95	3.882	-10.69	2.60e-11	6.08e-09
## YFR015C	YFR015C S000001911	GSY1	-5.96	3.481	-11.00	3.50e-11	7.86e-09
## YER067W	YER067W S000000869	RGI1	-6.44	4.118	-11.22	4.78e-11	1.03e-08
## YGR248W	YGR248W S000003480	SOL4	-6.89	1.382	-10.00	1.07e-10	2.22e-08
## YDR033W	YDR033W S000002440	MRH1	-2.61	7.040	-10.66	1.14e-10	2.29e-08
## YBL064C	YBL064C S000000160	PRX1	-2.32	5.593	-9.75	1.79e-10	3.47e-08
## YPR160W	YPR160W S000006364	GPH1	-7.00	2.863	-10.55	2.20e-10	4.12e-08
## YOR315W	YOR315W S000005842	SFG1	4.11	3.940	9.63	2.32e-10	4.21e-08
## YMR090W	YMR090W S000004696	<NA>	-3.30	4.314	-9.61	2.44e-10	4.28e-08
## YMR031C	YMR031C S000004633	EIS1	-2.70	5.442	-9.74	6.30e-10	1.07e-07
## YFR052C-A	YFR052C-A S000028768	<NA>	-6.85	1.058	-8.95	1.02e-09	1.68e-07
## YEL011W	YEL011W S000000737	GLC3	-4.44	4.163	-10.05	1.17e-09	1.87e-07
## YML054C	YML054C S000004518	CYB2	-2.80	4.320	-8.70	1.80e-09	2.81e-07
## YBR183W	YBR183W S000000387	YPC1	-4.21	2.566	-8.54	2.54e-09	3.85e-07
## YGL037C	YGL037C S000003005	PNC1	-2.25	6.605	-8.41	3.41e-09	5.04e-07
## YNL007C	YNL007C S000004952	SIS1	-2.29	7.054	-8.82	3.83e-09	5.51e-07
## YEL039C	YEL039C S000000765	CYC7	-5.25	1.956	-8.27	4.75e-09	6.67e-07
## YDL124W	YDL124W S000002282	<NA>	-2.87	4.947	-8.12	6.69e-09	9.16e-07
## YDR342C	YDR342C S000002750	HXT7	-5.92	5.968	-12.31	8.67e-09	1.16e-06
## YMR173W	YMR173W S000004784	DDR48	-3.18	3.420	-7.99	9.06e-09	1.16e-06
## YFL014W	YFL014W S000001880	HSP12	-7.83	2.930	-9.85	9.08e-09	1.16e-06
## YPL240C	YPL240C S000006161	HSP82	-2.36	7.691	-8.39	1.22e-08	1.52e-06
## YOL052C-A	YOL052C-A S000005413	DDR2	-8.22	-2.456	-7.76	1.57e-08	1.91e-06
## YFR017C	YFR017C S000001913	IGD1	-5.56	1.621	-7.75	2.04e-08	2.44e-06
## YML128C	YML128C S000004597	MSC1	-2.39	4.202	-7.55	2.57e-08	3.01e-06
## YLL026W	YLL026W S000003949	HSP104	-2.04	7.838	-7.54	2.63e-08	3.02e-06
## YLR178C	YLR178C S000004168	TFS1	-2.83	4.206	-7.53	3.22e-08	3.62e-06
## YLR152C	YLR152C S000004142	<NA>	-2.42	3.464	-7.40	3.69e-08	4.06e-06
## YMR261C	YMR261C S000004874	TPS3	-1.74	7.291	-7.13	7.15e-08	7.72e-06
## YGR070W	YGR070W S000003302	ROM1	-2.34	3.863	-7.10	7.65e-08	7.98e-06
## YMR169C	YMR169C S000004779	ALD3	-4.03	3.446	-7.79	7.68e-08	7.98e-06
## YLL039C	YLL039C S000003962	UBI4	-1.91	7.235	-6.99	1.01e-07	1.03e-05
## YCR091W	YCR091W S000000687	KIN82	-2.46	3.418	-6.98	1.03e-07	1.03e-05
## YOR161C	YOR161C S000005687	PNS1	-3.81	4.249	-8.20	1.07e-07	1.06e-05
## YKL151C	YKL151C S000001634	NNR2	-2.25	5.572	-6.95	1.13e-07	1.09e-05
## YBL075C	YBL075C S000000171	SSA3	-1.86	5.368	-6.75	1.84e-07	1.75e-05
## YJL042W	YJL042W S000003578	MHP1	-1.76	6.012	-6.55	3.00e-07	2.81e-05
## YNL160W	YNL160W S000005104	YGP1	-2.69	4.669	-6.71	3.36e-07	3.09e-05
## YNR034W-A	YNR034W-A S000007525	EGO4	-7.49	0.853	-6.55	5.60e-07	5.08e-05
## YDL204W	YDL204W S000002363	RTN2	-3.44	2.791	-6.31	6.20e-07	5.53e-05
## YBL015W	YBL015W S000000111	ACH1	-2.05	5.858	-6.34	7.13e-07	6.26e-05
## YBR169C	YBR169C S000000373	SSE2	-2.69	4.305	-6.15	8.37e-07	7.23e-05

## YPL230W	YPL230W S000006151	USV1	-4.25	1.810	-6.11	1.43e-06	1.22e-04
## YBR230C	YBR230C S000000434	OM14	-2.33	5.257	-6.31	1.51e-06	1.27e-04
## YFL051C	YFL051C S000001843	<NA>	2.92	4.980	6.49	1.81e-06	1.49e-04
## YNL015W	YNL015W S000004960	PBI2	-2.58	3.872	-5.80	2.17e-06	1.77e-04
## YOR298C-A	YOR298C-A S000007253	MBF1	-1.61	7.482	-5.58	3.65e-06	2.92e-04
## YNL274C	YNL274C S000005218	GOR1	-2.52	3.327	-5.56	3.84e-06	3.04e-04
## YBR214W	YBR214W S000000418	SDS24	-2.27	5.543	-5.56	3.93e-06	3.07e-04
## YDR277C	YDR277C S000002685	MTH1	-2.63	2.580	-5.43	5.36e-06	4.13e-04
## YGR281W	YGR281W S000003513	YOR1	-1.53	7.181	-5.39	6.05e-06	4.56e-04
## YOR173W	YOR173W S000005699	DCS2	-2.95	2.521	-5.39	6.09e-06	4.56e-04
## YER073W	YER073W S000000875	ALD5	1.91	6.175	5.54	6.59e-06	4.87e-04
## YBR149W	YBR149W S000000353	ARA1	-1.62	7.634	-5.32	7.18e-06	5.24e-04
## YLL023C	YLL023C S000003946	POM33	-1.96	6.435	-5.60	8.36e-06	6.02e-04
## YOR347C	YOR347C S000005874	PYK2	-3.42	2.578	-5.24	9.04e-06	6.43e-04
## YLR327C	YLR327C S000004319	TMA10	-5.90	0.689	-5.23	1.37e-05	9.55e-04
## YOR052C	YOR052C S000005578	TMC1	-1.94	3.794	-5.08	1.38e-05	9.55e-04
## YGR019W	YGR019W S000003251	UGA1	-1.81	4.934	-5.02	1.61e-05	1.10e-03
## YDL039C	YDL039C S000002197	PRM7	2.08	5.358	5.25	1.74e-05	1.18e-03
## YDL181W	YDL181W S000002340	INH1	-1.62	5.447	-4.86	2.42e-05	1.62e-03
## YKL037W	YKL037W S000001520	AIM26	-5.72	-1.921	-4.83	2.68e-05	1.77e-03
## YDR216W	YDR216W S000002624	ADR1	-2.13	3.634	-4.81	2.76e-05	1.80e-03
## YGR086C	YGR086C S000003318	PIL1	-1.51	8.762	-4.78	2.98e-05	1.92e-03
## YER066C-A	YER066C-A S000002959	<NA>	-5.36	-2.787	-4.77	3.07e-05	1.96e-03
## YDR275W	YDR275W S000002683	BSC2	-2.22	2.441	-4.74	3.32e-05	2.10e-03
## YHR092C	YHR092C S000001134	HXT4	-3.09	3.492	-5.14	3.54e-05	2.21e-03
## YDR533C	YDR533C S000002941	HSP31	-2.04	3.459	-4.71	3.64e-05	2.25e-03
## YMR145C	YMR145C S000004753	NDE1	-1.57	8.147	-4.68	3.89e-05	2.37e-03
## YNL194C	YNL194C S000005138	<NA>	-5.30	-0.328	-4.69	4.03e-05	2.43e-03
## YIL056W	YIL056W S000001318	VHR1	-1.63	5.441	-4.67	4.07e-05	2.43e-03
## YPL014W	YPL014W S000005935	CIP1	-2.05	4.793	-4.77	5.47e-05	3.23e-03
## YAL065C	YAL065C S000001817	<NA>	5.04	-1.857	4.53	5.80e-05	3.39e-03
## YKL201C	YKL201C S000001684	MNN4	-2.18	5.141	-4.91	6.06e-05	3.51e-03
## YJL141C	YJL141C S000003677	YAK1	-1.72	5.050	-4.50	6.27e-05	3.59e-03
## YPL061W	YPL061W S000005982	ALD6	3.82	7.959	5.49	6.35e-05	3.60e-03
## YAL060W	YAL060W S000000056	BDH1	-1.94	6.745	-4.87	6.51e-05	3.65e-03
## YBR117C	YBR117C S000000321	TKL2	-3.36	1.632	-4.57	6.90e-05	3.84e-03
## YDR185C	YDR185C S000002593	UPS3	-2.26	2.111	-4.41	7.95e-05	4.37e-03
## YNR014W	YNR014W S000005297	<NA>	-5.52	1.682	-4.93	8.81e-05	4.80e-03
## YER054C	YER054C S000000856	GIP2	-3.49	0.585	-4.36	9.10e-05	4.92e-03
## YDR513W	YDR513W S000002921	GRX2	-1.49	6.726	-4.25	1.20e-04	6.43e-03
## YER067C-A	YER067C-A S000028748	<NA>	-5.41	-3.270	-4.21	1.38e-04	7.24e-03
## YIL136W	YIL136W S000001398	OM45	-2.39	2.294	-4.20	1.38e-04	7.24e-03
## YPR184W	YPR184W S000006388	GDB1	-1.85	5.573	-4.24	1.57e-04	8.16e-03
## YMR251W-A	YMR251W-A S000004864	HOR7	-3.13	4.424	-4.60	1.60e-04	8.22e-03
## YOL155C	YOL155C S000005515	HPF1	-1.87	9.918	-4.41	1.72e-04	8.78e-03
## YBR139W	YBR139W S000000343	<NA>	-1.62	6.133	-4.06	1.98e-04	1.00e-02

```

## YOR185C    YOR185C S000005711    GSP2 -1.88   4.302  -4.00  2.33e-04  1.17e-02
## YBR161W    YBR161W S000000365    CSH1 -1.69   3.783  -3.96  2.59e-04  1.29e-02
## YBR054W    YBR054W S000000258    YR02 -4.60   1.038  -4.14  2.61e-04  1.29e-02
## YBL049W    YBL049W S000000145    MOH1 -4.72   -2.048 -3.92  2.90e-04  1.42e-02
## YOR345C    YOR345C S000005872    <NA> -4.42   -2.214 -3.91  2.98e-04  1.44e-02
## YDR345C    YDR345C S000002753    HXT3 -2.43   7.998  -4.45  3.27e-04  1.56e-02
## YCL040W    YCL040W S000000545    GLK1 -1.68   8.056  -3.87  3.29e-04  1.56e-02
## YAL005C    YAL005C S000000004    SSA1 -1.94   10.680 -4.16  3.44e-04  1.62e-02
## YJL107C    YJL107C S000003643    <NA> -1.89   2.503  -3.83  3.60e-04  1.68e-02
## YCR021C    YCR021C S000000615    HSP30 -5.34   2.019  -4.31  3.70e-04  1.72e-02
## YMR081C    YMR081C S000004686    ISF1 -3.85   -0.437 -3.81  3.86e-04  1.78e-02
## YKL096W    YKL096W S000001579    CWP1 -2.48   5.702  -4.24  4.91e-04  2.24e-02
## YGR143W    YGR143W S000003375    SKN1 -1.52   4.422  -3.64  6.00e-04  2.71e-02
## YDL022W    YDL022W S000002180    GPD1 -1.64   8.332  -3.65  8.10e-04  3.64e-02
## YKR049C    YKR049C S000001757    FMP46 -2.16   2.104  -3.50  8.54e-04  3.78e-02
## YNL200C    YNL200C S000005144    NNR1 -2.11   2.949  -3.50  8.56e-04  3.78e-02
## YNL195C    YNL195C S000005139    <NA> -3.29   1.001  -3.45  9.91e-04  4.35e-02
## YCL035C    YCL035C S000000540    GRX1 -1.54   5.312  -3.43  1.00e-03  4.37e-02
## YPL087W    YPL087W S000006008    YDC1 -1.54   6.261  -3.44  1.06e-03  4.60e-02
## YOR374W    YOR374W S000005901    ALD4 -1.75   6.685  -3.58  1.09e-03  4.65e-02
## YPR026W    YPR026W S000006230    ATH1 -1.51   4.864  -3.37  1.18e-03  4.97e-02
## YMR016C    YMR016C S000004618    SOK2  1.47    5.508  3.37  1.18e-03  4.97e-02

# for subsequent analysis, let's save the output file as a tsv
# and the res object as an R data object.
lfc1_top.table %>% tibble() %>%
  arrange(desc(adj.P.Val)) %>%
  mutate(adj.P.Val = round(adj.P.Val, 2)) %>%
  mutate(across(where(is.numeric), signif, 3)) %>%
  write_tsv(., file = paste0(dir_output_limma, "yeast_lfc1_topTreat_limma.tsv"))

saveRDS(object = lfc1_res, file = paste0(dir_output_limma, "yeast_lfc1_res_limma.Rds"))

```

### 8.12.1 Visualize DE genes from Treat using lfc=1

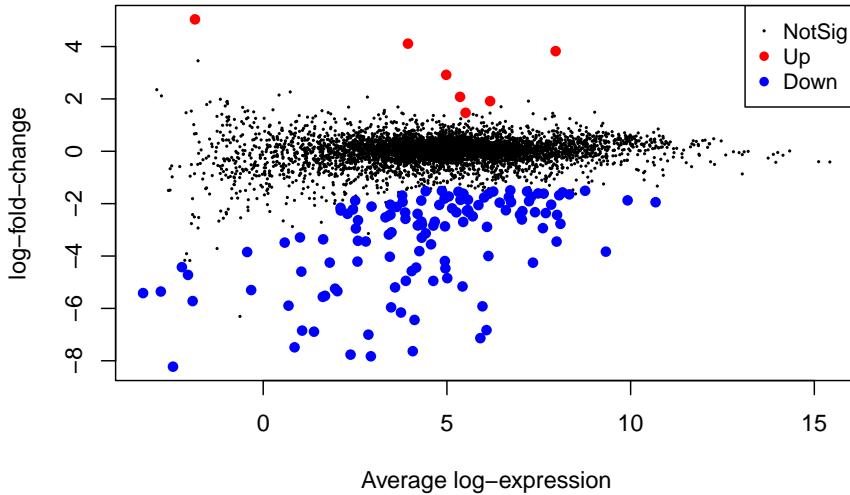
```

is.de.lfc1 <- decideTests(lfc1_res, p.value=0.05)
summary(is.de.lfc1)

##          [,1]
## Down      126
## NotSig   5482
## Up        7

```

```
# visualize results
limma:::plotMA(lfc1_res, status=is.de.lfc1)
```



## 8.13 Comparing DE analysis softwares

We have went through some example DE workflows with edgeR, DESeq2, and limma-voom. Since we have saved our outputs for each analysis, we can compare their outcomes now.

```
# load in all of the DE results for the difference of difference contrast
path_output_edgeR <- "~/Desktop/Genomic_Data_Analysis/Analysis/edgeR/yeast_topTags_edgeR.tsv"
path_output_DESeq2 <- "~/Desktop/Genomic_Data_Analysis/Analysis/DESeq2/yeast_res_DESeq2.tsv"
path_output_limma <- "~/Desktop/Genomic_Data_Analysis/Analysis/limma/yeast_topTags_limma.tsv"

topTags_edgeR <- read.delim(path_output_edgeR)
topTags_DESeq2 <- read.delim(path_output_DESeq2)
topTags_limma <- read.delim(path_output_limma)

sig_cutoff <- 0.01
FC_cutoff <- 1
# NOTE: we need to be very careful applying an FC cutoff like this
```

```

## edgeR
# get genes that are upregulated
up_edgeR_DEG <- topTags_edgeR %>%
  dplyr::filter(FDR < sig_cutoff & logFC > FC_cutoff) %>%
  pull(ORF)

down_edgeR_DEG <- topTags_edgeR %>%
  dplyr::filter(FDR < sig_cutoff & logFC < -FC_cutoff) %>%
  pull(ORF)

## DESeq2
up_DESeq2_DEG <- topTags_DESeq2 %>%
  dplyr::filter(padj < sig_cutoff & log2FoldChange > FC_cutoff) %>%
  pull(ORF)

down_DESeq2_DEG <- topTags_DESeq2 %>%
  dplyr::filter(padj < sig_cutoff & log2FoldChange < -FC_cutoff) %>%
  pull(ORF)

## limma
up_limma_DEG <- topTags_limma %>%
  dplyr::filter(adj.P.Val < sig_cutoff & logFC > FC_cutoff) %>%
  pull(ORF)

down_limma_DEG <- topTags_limma %>%
  dplyr::filter(adj.P.Val < sig_cutoff & logFC < -FC_cutoff) %>%
  pull(ORF)

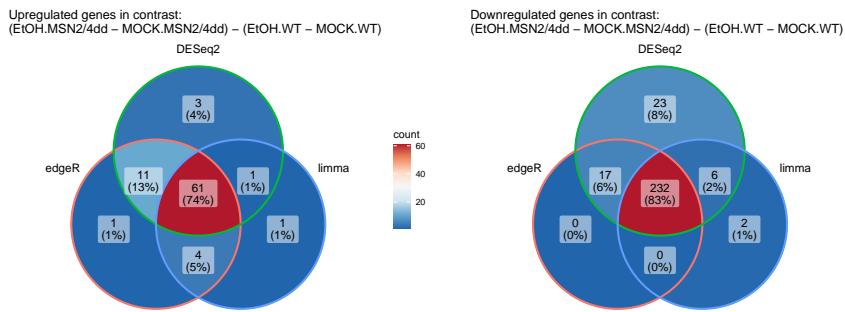
up_DEG_results_list <- list(up_edgeR_DEG,
                             up_DESeq2_DEG,
                             up_limma_DEG)

# visualize the GO results list as a venn diagram
ggVennDiagram(up_DEG_results_list,
               category.names = c("edgeR", "DESeq2", "limma")) +
  scale_x_continuous(expand = expansion(mult = .2)) +
  scale_fill_distiller(palette = "RdBu")
) +
  ggtitle("Upregulated genes in contrast: \n(EtOH.MSN2/4dd - MOCK.MSN2/4dd) - (EtOH.WT

# Now let's do the same for downregulated genes:
down_DEG_results_list <- list(down_edgeR_DEG,
                               down_DESeq2_DEG,
                               down_limma_DEG)

```

```
ggVennDiagram(down_DEG_results_list,
  category.names = c("edgeR", "DESeq2", "limma")) +
  scale_x_continuous(expand = expansion(mult = .2)) +
  scale_fill_distiller(palette = "RdBu"
) +
  ggtitle("Downregulated genes in contrast: \n(EtOH.MSN2/4dd - MOCK.MSN2/4dd) - (EtOH.WT - MOCK.WT)
```



## 8.14 Correlation between logFC estimates across softwares

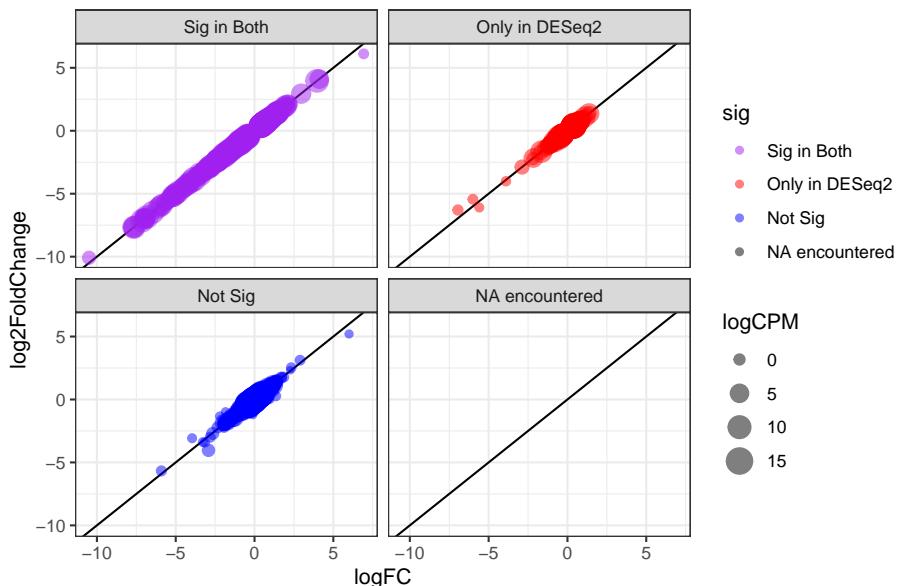
```
# Custom labels for facet headers
custom_labels <- c("purple" = "Sig in Both",
                  "red" = "Only in edgeR",
                  "blue" = "Only in DESeq2",
                  "black" = "Not Sig",
                  "grey" = "NA encountered")

# compare edgeR & DESeq2
full_join(topTags_edgeR, topTags_DESeq2,
          by = join_by(ORF, SGD, GENENAME)) %>%
  mutate(edgeR_sig = ifelse(FDR < sig_cutoff, "red", "black")) %>%
  mutate(DESeq2_sig = ifelse(padj < sig_cutoff, "blue", "black")) %>%
  mutate(sig = factor(case_when(
    edgeR_sig == "red" & DESeq2_sig == "blue" ~ "purple",
    edgeR_sig == "red" & DESeq2_sig != "blue" ~ "red",
    edgeR_sig != "red" & DESeq2_sig == "blue" ~ "blue",
    edgeR_sig != "red" & DESeq2_sig != "blue" ~ "black",
    TRUE ~ "grey" # if none of these are met
  ), levels = c("purple", "red", "blue", "black", "grey"), labels = c("Sig in Both", "Only in edgeR", "Only in DESeq2", "Not Sig", "NA encountered")))
```

```
ggplot(aes(x=logFC, y=log2FoldChange, color = sig, size=logCPM)) +
  geom_abline(slope = 1,) +
  geom_point(alpha=0.5) +
  scale_color_manual(values=c("purple", "red", "blue", "black", "grey")) + # use color
  theme_bw() +
  facet_wrap(~sig, labeller = labeller(new_column = custom_labels)) +
  ggtitle("Comparing genewise logFC estimates between edgeR and DESeq2")
```

## Warning: Removed 11 rows containing missing values (`geom\_point()`).

Comparing genewise logFC estimates between edgeR and DESeq2

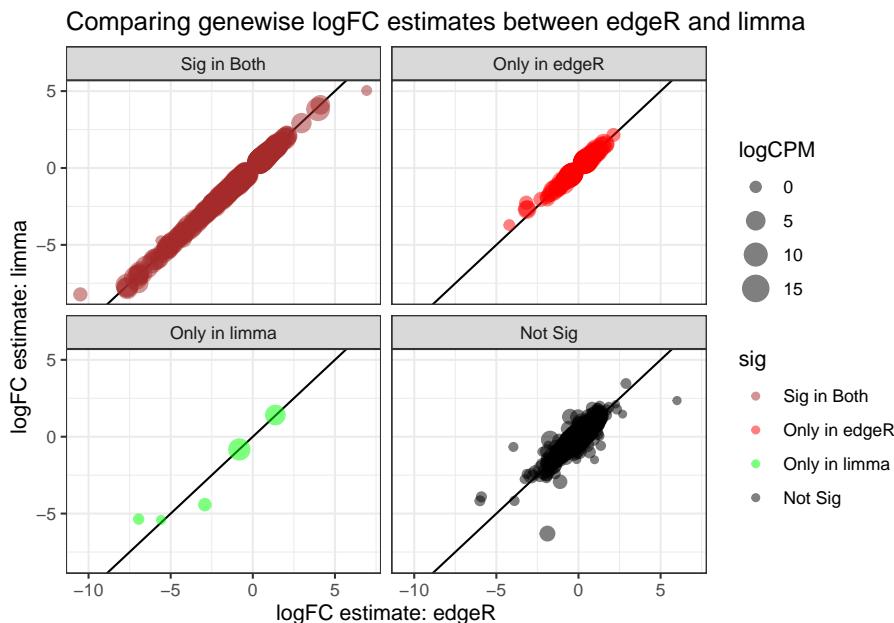


```
# compare edgeR & limma
full_join(topTags_edgeR, topTags_limma,
          by = join_by(ORF, SGD, GENENAME)) %>%
  mutate(edgeR_sig = ifelse(FDR < sig_cutoff, "red", "black")) %>%
  mutate(limma_sig = ifelse(adj.P.Val < sig_cutoff, "green", "black")) %>%
  mutate(sig = factor(case_when(
    edgeR_sig == "red" & limma_sig == "green" ~ "brown",
    edgeR_sig == "red" & limma_sig != "green" ~ "red",
    edgeR_sig != "red" & limma_sig == "green" ~ "green",
    edgeR_sig != "red" & limma_sig != "green" ~ "black",
    TRUE ~ "grey" # if none of these are met
  ), levels = c("brown", "red", "green", "black", "grey"), labels = c("Sig in Both", "(0, 5, 10, 15)"))
  ggpplot(aes(x=logFC.x, y=logFC.y, color = sig, size=logCPM)) +
```

```

geom_abline(slope = 1,) +
geom_point(alpha=0.5) +
scale_color_manual(values=c("brown", "red", "green", "black", "grey")) + # use colors given
theme_bw() +
facet_wrap(~sig, labeller = labeller(new_column = custom_labels)) +
ggttitle("Comparing genewise logFC estimates between edgeR and limma") +
labs(x="logFC estimate: edgeR", y="logFC estimate: limma")

```



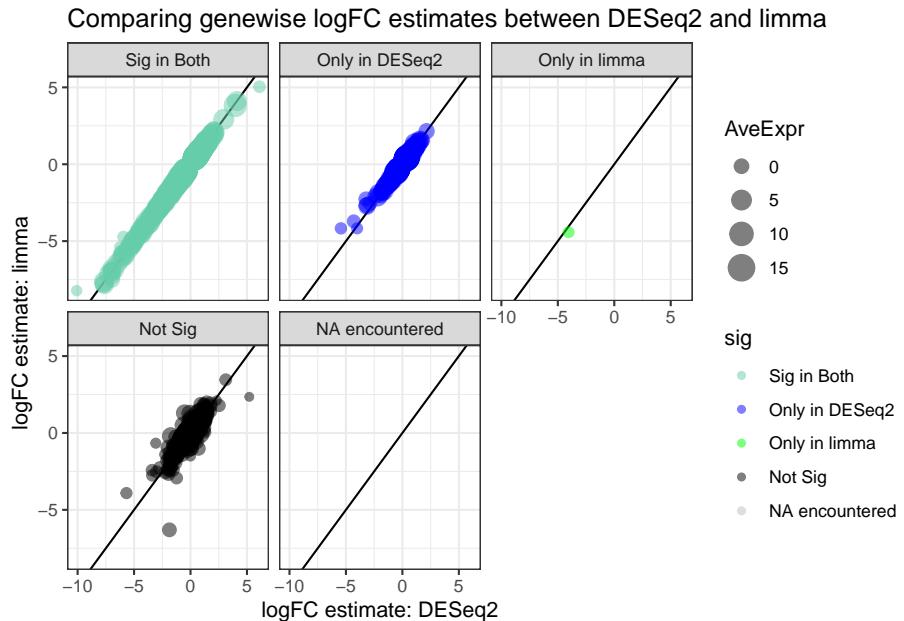
```

# compare DESeq2 & limma
full_join(topTags_DESeq2, topTags_limma,
          by = join_by(ORF, SGD, GENENAME)) %>%
mutate(DESeq2_sig = ifelse(padj < sig_cutoff, "blue", "black")) %>%
mutate(limma_sig = ifelse(adj.P.Val < sig_cutoff, "green", "black")) %>%
mutate(sig = factor(case_when(
  DESeq2_sig == "blue" & limma_sig == "green" ~ "aquamarine3",
  DESeq2_sig == "blue" & limma_sig != "green" ~ "blue",
  DESeq2_sig != "blue" & limma_sig == "green" ~ "green",
  DESeq2_sig != "blue" & limma_sig != "green" ~ "black",
  TRUE ~ "grey" # if none of these are met
), levels = c("aquamarine3", "blue", "green", "black", "grey"), labels = c("Sig in Both", "Only in edgeR", "Only in limma", "Not Sig"))
), ggplot(aes(x=log2FoldChange, y=logFC, color = sig, size=AveExpr)) +
  geom_abline(slope = 1,) +
  geom_point(alpha=0.5) +
  scale_color_manual(values=c("aquamarine3", "blue", "green", "black", "grey")) + # use colors given
  theme_bw()

```

```
theme_bw() +
facet_wrap(~sig, labeller = labeller(new_column = custom_labels, drop=FALSE)) +
ggtitle("Comparing genewise logFC estimates between DESeq2 and limma") +
labs(x="logFC estimate: DESeq2", y="logFC estimate: limma")
```

```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```



## 8.15 Questions

Question 1: How many genes were upregulated and downregulated in the contrast we looked at in today's activity? Be sure to clarify the cutoffs used for determining significance.

Question 2: What are the pros and cons of applying a logFC cutoff to a differential expression analysis?

Be sure to knit this file into a pdf or html file once you're finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

R version 4.3.1 (2023-06-16)

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8|en\_US.UTF-8||C||en\_US.UTF-8||en\_US.UTF-8

**attached base packages:** *stats4, stats, graphics, grDevices, utils, datasets, methods* and *base*

**other attached packages:** *Glimma(v.2.10.0), DESeq2(v.1.40.2), edgeR(v.3.42.4), limma(v.3.56.2), reactable(v.0.4.4), webshot2(v.0.1.1), statmod(v.1.5.0), Rsubread(v.2.14.2), ShortRead(v.1.58.0), GenomicAlignments(v.1.36.0), SummarizedExperiment(v.1.30.2), MatrixGenerics(v.1.12.3), matrixStats(v.1.0.0), Rsamtools(v.2.16.0), GenomicRanges(v.1.52.1), Biostrings(v.2.68.1), GenomeInfoDb(v.1.36.4), XVector(v.0.40.0), BiocParallel(v.1.34.2), Rfastp(v.1.10.0), org.Sc.sgd.db(v.3.17.0), AnnotationDbi(v.1.62.2), IRanges(v.2.34.1), S4Vectors(v.0.38.2), Biobase(v.2.60.0), BiocGenerics(v.0.46.0), clusterProfiler(v.4.8.2), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pandoc(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3), forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyrr(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)*

**loaded via a namespace (and not attached):** *splines(v.4.3.1), later(v.1.3.1), bitops(v.1.0-7), ggplotify(v.0.1.2), polyclip(v.1.10-6), lifecycle(v.1.0.3), sf(v.1.0-14), rprojroot(v.2.0.3), vroom(v.1.6.4), processx(v.3.8.2), lattice(v.0.21-9), MASS(v.7.3-60), crosstalk(v.1.2.0), magrittr(v.2.0.3), rmarkdown(v.2.25), yaml(v.2.3.7), cowplot(v.1.1.1), chromote(v.0.1.2), DBI(v.1.1.3), RColorBrewer(v.1.1-3), abind(v.1.4-5), zlibbioc(v.1.46.0), ggraph(v.2.1.0), RCurl(v.1.98-1.12), yulab.utils(v.0.1.0), tweenr(v.2.0.2), GenomeInfoDbData(v.1.2.10), enrichplot(v.1.20.0), ggrepel(v.0.9.4), units(v.0.8-4), codetools(v.0.2-19), DelayedArray(v.0.26.7), DOSE(v.3.26.1), ggforce(v.0.4.1), tidyselect(v.1.2.0), aplot(v.0.2.2), farver(v.2.1.1), viridis(v.0.6.4), webshot(v.0.5.5), jsonlite(v.1.8.7), e1071(v.1.7-13), ellipsis(v.0.3.2), tidygraph(v.1.2.3), tools(v.4.3.1), treeio(v.1.24.3), Rcpp(v.1.0.11), glue(v.1.6.2), gridExtra(v.2.3), xfun(v.0.40), qvalue(v.2.32.0), websocket(v.1.4.1), withr(v.2.5.1), fastmap(v.1.1.1), latticeExtra(v.0.6-30), fansi(v.1.0.5), digest(v.0.6.33), timechange(v.0.2.0), R6(v.2.5.1), gridGraphics(v.0.5-1), colorspace(v.2.1-0), GO.db(v.3.17.0), jpeg(v.0.1-10), RSQLite(v.2.3.1), utf8(v.1.2.3), generics(v.0.1.3), data.table(v.1.14.8), class(v.7.3-22), graphlayouts(v.1.0.1), httr(v.1.4.7), htmlwidgets(v.1.6.2), S4Arrays(v.1.0.6), scatterpie(v.0.2.1), pkgconfig(v.2.0.3), gtable(v.0.3.4), blob(v.1.2.4), hwriter(v.1.3.2.1), shadowtext(v.0.1.2), htmltools(v.0.5.6.1), bookdown(v.0.36), fgsea(v.1.26.0), scales(v.1.2.1), png(v.0.1-8), snakecase(v.0.11.1), ggfun(v.0.1.3), rstudioapi(v.0.15.0), tzdb(v.0.4.0), reshape2(v.1.4.4), rjson(v.0.2.21), nlme(v.3.1-163), proxy(v.0.4-27), cachem(v.1.0.8), KernSmooth(v.2.23-22), RVenn(v.1.1.0), parallel(v.4.3.1), HDO.db(v.0.99.1), pillar(v.1.9.0), grid(v.4.3.1), vctrs(v.0.6.4), promises(v.1.2.1), archive(v.1.1.5), evaluate(v.0.22), cli(v.3.6.1), locfit(v.1.5-9.8), compiler(v.4.3.1), rlang(v.1.1.1), crayon(v.1.5.2), labeling(v.0.4.3), classInt(v.0.4-10), interp(v.1.1-4), re-*

*actR(v.0.5.0), ps(v.1.7.5), plyr(v.1.8.9), fs(v.1.6.3), stringi(v.1.7.12), viridisLite(v.0.4.2), deldir(v.1.0-9), munsell(v.0.5.0), lazyeval(v.0.2.2), GOSemSim(v.2.26.1), Matrix(v.1.6-1.1), hms(v.1.1.3), patchwork(v.1.1.3), bit64(v.4.0.5), KEGGREST(v.1.40.1), memoise(v.2.0.1), ggtree(v.3.8.2), fastmatch(v.1.1-4), bit(v.4.0.5), downloader(v.0.4), ape(v.5.7-1) and gson(v.0.1.0)*

# Chapter 9

## Visualizing Differential Expression Results

last updated: 2023-10-27

### Install Packages

As usual, make sure we have the right packages for this exercise

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# let's load all of the files we were using and want to have again today
p_load("tidyverse", "knitr", "readr",
       "pander", "BiocManager",
       "dplyr", "stringr",
       "purrr", # for working with lists (beautify column names)
       "scales", "viridis", # for ggplot
       "reactable") # for pretty tables.

# We also need these packages today.
p_load("DESeq2", "edgeR", "AnnotationDbi", "org.Sc.sgd.db",
       "ggrepel",
       "Glimma",
       "ggVennDiagram", "ggplot2")
```

### 9.1 Description

This exercises shows more ways differential expression analysis data can be visualized.

## 9.2 Learning Outcomes

At the end of this exercise, you should be able to:

- Visualize Differential Expression Results
- Interpret MA and volcano plots

```
library(org.Sc.sgd.db)

# load in all of the DE results for the difference of difference contrast
path_output_edgeR <- "~/Desktop/Genomic_Data_Analysis/Analysis/edgeR/yeast_topTags_edgeR"
path_output_DESeq2 <- "~/Desktop/Genomic_Data_Analysis/Analysis/DESeq2/yeast_res_DESeq2"
path_output_limma <- "~/Desktop/Genomic_Data_Analysis/Analysis/limma/yeast_topTags_limma"

# if you don't have these files, we generated them in previous lessons.
# you can remove the "##" from the chunks below to grab them from the interwebs.
# path_output_edgeR <- "https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/analysis/edgeR/yeast_topTags_edgeR"
# path_output_DESeq2 <- "https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/analysis/DESeq2/yeast_res_DESeq2"
# path_output_limma <- "https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/analysis/limma/yeast_topTags_limma"

topTags_edgeR <- read.delim(path_output_edgeR)
topTags_DESeq2 <- read.delim(path_output_DESeq2)
topTags_limma <- read.delim(path_output_limma)

sig_cutoff <- 0.05
FC_cutoff <- 1

## edgeR
# get genes that are upregulated
up_edgeR_DEG <- topTags_edgeR %>%
  dplyr::filter(FDR < sig_cutoff & logFC > FC_cutoff) %>%
  pull(ORF)

down_edgeR_DEG <- topTags_edgeR %>%
  dplyr::filter(FDR < sig_cutoff & logFC < -FC_cutoff) %>%
  pull(ORF)

## DESeq2
up_DESeq2_DEG <- topTags_DESeq2 %>%
  dplyr::filter(padj < sig_cutoff & log2FoldChange > FC_cutoff) %>%
  pull(ORF)

down_DESeq2_DEG <- topTags_DESeq2 %>%
  dplyr::filter(padj < sig_cutoff & log2FoldChange < -FC_cutoff) %>%
```

```

pull(ORF)

## limma
up_limma_DEG <- topTags_limma %>%
  dplyr::filter(adj.P.Val < sig_cutoff & logFC > FC_cutoff) %>%
  pull(ORF)

down_limma_DEG <- topTags_limma %>%
  dplyr::filter(adj.P.Val < sig_cutoff & logFC < -FC_cutoff) %>%
  pull(ORF)

up_DEG_results_list <- list(up_edgeR_DEG,
                             up_DESeq2_DEG,
                             up_limma_DEG)

```

### 9.3 MA-plot

MA plots display a log ratio (M) vs an average (A) in order to visualize the differences between two groups. In general we would expect the expression of genes to remain consistent between conditions and so the MA plot should be similar to the shape of a trumpet with most points residing on a y intercept of 0. DESeq2 has a built in function for creating the MA plot that we have used before (`plotMA()`), but we can also make our own:

```

# assign pvalue and logFC cutoffs for coloring DE genes
sig_cutoff <- 0.01
FC_label_cutoff <- 3

#plot MA for edgeR using ggplot2
topTags_edgeR %>%
  mutate(`Significant FDR` = case_when(
    FDR < sig_cutoff ~ "Yes",
    .default = "No"),
    delabel = case_when(FDR < sig_cutoff & abs(logFC) > FC_label_cutoff ~ ORF,
    .default = NA)) %>%
  ggplot(aes(x=logCPM, y=logFC, color = `Significant FDR`, label = delabel)) +
  geom_point(size=1) +
  scale_y_continuous(limits=c(-5, 5), oob=squish) +
  geom_hline(yintercept = 0, colour="darkgrey", linewidth=1, linetype="longdash") +
  labs(x="mean of normalized counts", y="log fold change") +
  # ggrepel::geom_text_repel(size = 1.5) +
  scale_color_manual(values = c("black", "red")) +
  theme_bw() +

```

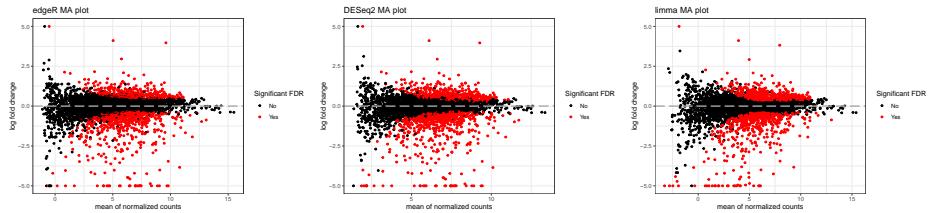
```

ggtile("edgeR MA plot")

#plot MA for DESeq2 using ggplot2
topTags_DESeq2 %>%
  mutate(
    `Significant FDR` = case_when(padj < sig_cutoff ~ "Yes",
                                    .default = "No"),
    delabel = case_when(
      padj < sig_cutoff & abs(log2FoldChange) > FC_label_cutoff ~ ORF,
      .default = NA)
  ) %>%
  ggplot(aes(log(baseMean), log2FoldChange, color = `Significant FDR`, label = delabel),
         geom_point(size=1) +
         scale_y_continuous(limits=c(-5, 5), oob=squish) +
         geom_hline(yintercept = 0, colour="darkgrey", linewidth=1, linetype="longdash") +
         labs(x="mean of normalized counts", y="log fold change") +
         # ggrepel::geom_text_repel(size = 1.5) +
         scale_color_manual(values = c("black", "red")) +
         theme_bw() +
         ggtile("DESeq2 MA plot")

#plot MA for limma using ggplot2
topTags_limma %>%
  mutate(
    `Significant FDR` = case_when(adj.P.Val < sig_cutoff ~ "Yes",
                                    .default = "No"),
    delabel = case_when(
      adj.P.Val < sig_cutoff & abs(logFC) > FC_label_cutoff ~ ORF,
      .default = NA)
  ) %>%
  ggplot(aes(AveExpr, logFC, color = `Significant FDR`, label = delabel)) +
  geom_point(size=1) +
  scale_y_continuous(limits=c(-5, 5), oob=squish) +
  geom_hline(yintercept = 0, colour="darkgrey", linewidth=1, linetype="longdash") +
  labs(x="mean of normalized counts", y="log fold change") +
  # ggrepel::geom_text_repel(size = 1.5) +
  scale_color_manual(values = c("black", "red")) +
  theme_bw() +
  ggtile("limma MA plot")

```



## 9.4 Volcano Plot

```
# change the dimensions of the output figure by clicking the gear icon in top right corner of the
# viewer pane

topTags_edgeR %>%
  mutate(`Significant FDR` = case_when(
    FDR < sig_cutoff ~ "Yes",
    .default = "No"),
    delabel = case_when(FDR < sig_cutoff & abs(logFC) > FC_label_cutoff ~ ORF,
      .default = NA)) %>%
  ggplot(aes(x = logFC, -log10(FDR), color = `Significant FDR`, label = delabel)) +
  geom_point(size = 1) +
  ggrepel::geom_text_repel(size = 1.5) +
  labs(x = "log fold change", y = "-log10(adjusted p-value)") +
  theme_bw() +
  guides(color="none") +
  scale_color_manual(values = c("black", "red")) +
  ggtitle("edgeR Volcano plot")

## Warning: Removed 5556 rows containing missing values (`geom_text_repel()`).

topTags_DESeq2 %>%
  mutate(
    `Significant FDR` = case_when(padj < sig_cutoff ~ "Yes",
      .default = "No"),
    delabel = case_when(
      padj < sig_cutoff & abs(log2FoldChange) > FC_label_cutoff ~ ORF,
      .default = NA)
  ) %>%
  ggplot(aes(log2FoldChange, -log10(padj), color = `Significant FDR`, label = delabel)) +
  geom_point(size = 1) +
  ggrepel::geom_text_repel(size = 1.5) +
  labs(x = "log fold change", y = "-log10(adjusted p-value)") +
```

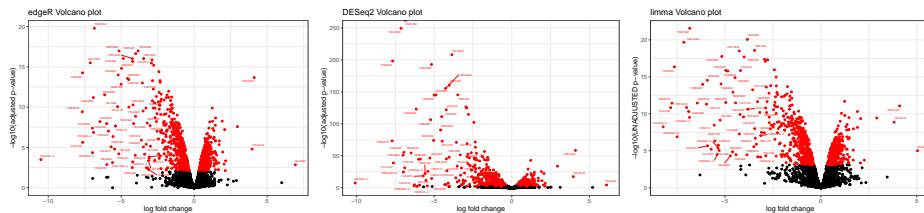
```
theme_bw() +
guides(color="none") +
scale_color_manual(values = c("black", "red")) +
ggtitle("DESeq2 Volcano plot")
```

```
## Warning: Removed 5559 rows containing missing values (`geom_text_repel()`).
```

```
## Warning: ggrepel: 14 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

```
topTags_limma %>%
mutate(
`Significant FDR` = case_when(adj.P.Val < sig_cutoff ~ "Yes",
.default = "No"),
delabel = case_when(
adj.P.Val < sig_cutoff & abs(logFC) > FC_label_cutoff ~ ORF,
.default = NA)
) %>%
ggplot(aes(x=logFC, y=-log10(P.Value), color = `Significant FDR`, label = delabel)) +
geom_point(size = 1) +
ggrepel::geom_text_repel(size = 1.5) +
labs(x = "log fold change", y = "-log10(UNADJUSTED p-value)") +
theme_bw() +
guides(color="none") +
scale_color_manual(values = c("black", "red")) +
ggtitle("limma Volcano plot")
```

```
## Warning: Removed 5557 rows containing missing values (`geom_text_repel()`).
```



## 9.5 Using Glimma for an interactive visualization

### 9.5.1 MA plots

```
# load in res objects for both limma and edgeR
res_limma <- readRDS("~/Desktop/Genomic_Data_Analysis/Analysis/limma/yeast_res_limma.Rds")
res_edgeR <- readRDS("~/Desktop/Genomic_Data_Analysis/Analysis/edgeR/yeast_res_edgeR.Rds")

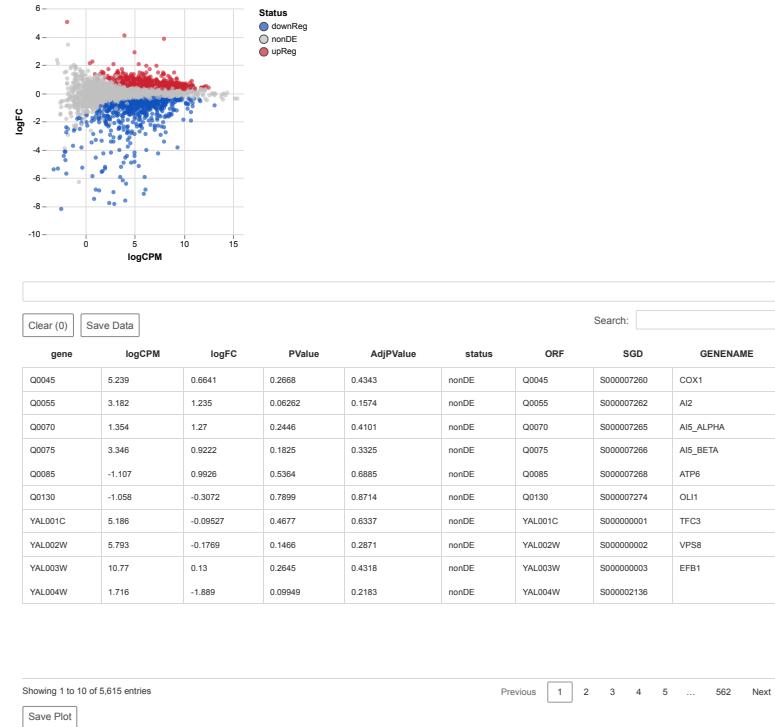
# code to pull it from github:
# res_limma <- read_rds("https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/analysis/yeast_res_limma.Rds")
# res_edgeR <- read_rds("https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/analysis/yeast_res_edgeR.Rds")

# load in the DGE lists for each
y_limma <- readRDS("~/Desktop/Genomic_Data_Analysis/Analysis/limma/yeast_y_limma.Rds")
y_edgeR <- readRDS("~/Desktop/Genomic_Data_Analysis/Analysis/edgeR/yeast_y_edgeR.Rds")

# again, alternative code to pull from github
# y_limma <- read_rds("https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/analysis/yeast_y_limma.Rds")
# y_edgeR <- read_rds("https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/analysis/yeast_y_edgeR.Rds")

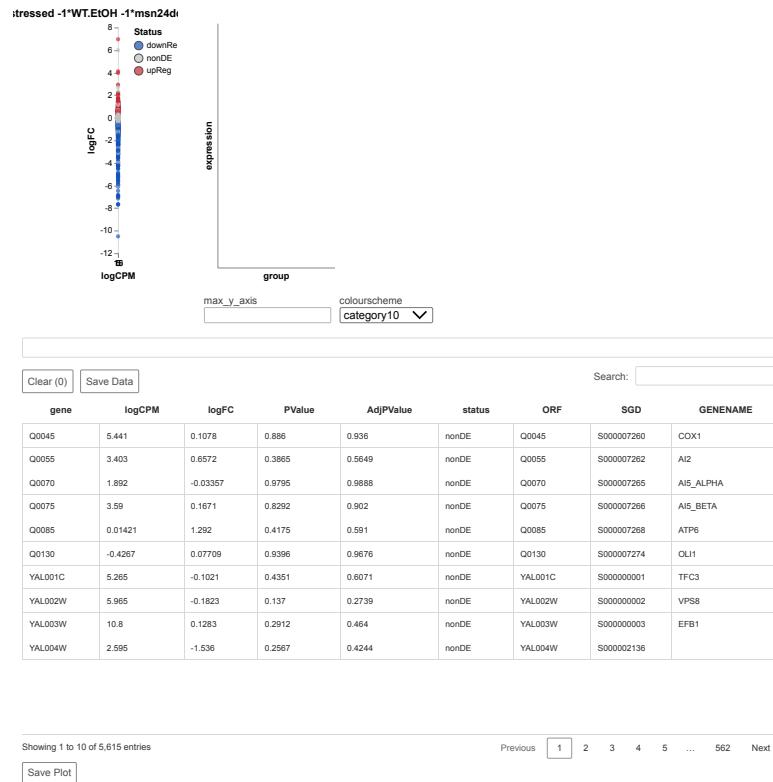
glimmaMA(res_limma, dge = y_limma)
```

216 CHAPTER 9. VISUALIZING DIFFERENTIAL EXPRESSION RESULTS



```
glimmaMA(res_edgeR, dge = y_edgeR)
```

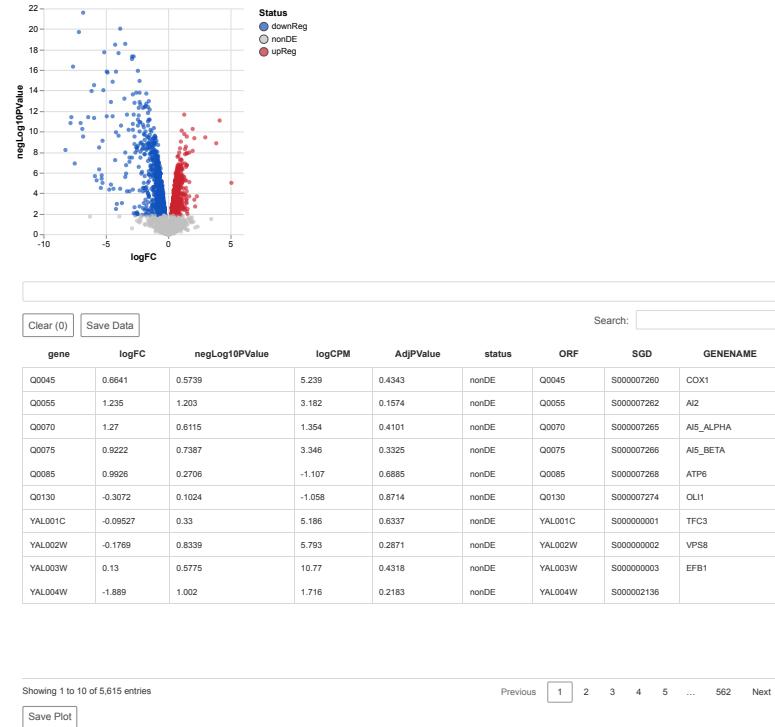
```
## Warning in buildXYData(table, status, main, display.columns, anno, counts, :
## count transform requested but not all count values are integers.
```



### 9.5.2 Volcano Plots

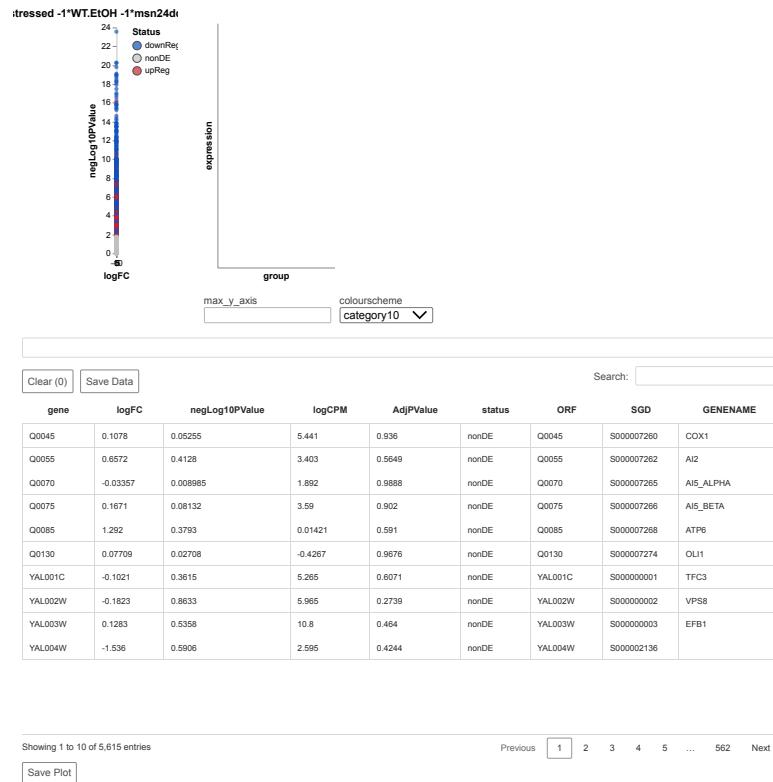
```
glimmaVolcano(res_limma, dge = y_limma)
```

218 CHAPTER 9. VISUALIZING DIFFERENTIAL EXPRESSION RESULTS



```
glimmaVolcano(res_edgeR, dge = y_edgeR)
```

```
## Warning in buildXYData(table, status, main, display.columns, anno, counts, :
## count transform requested but not all count values are integers.
```



## 9.6 Generating bar graph summaries

This visualization approach compresses relevant information, so it's generally a discouraged approach for visualizing DE data. However, it is done, so if it is useful for your study, here is how you could do it.

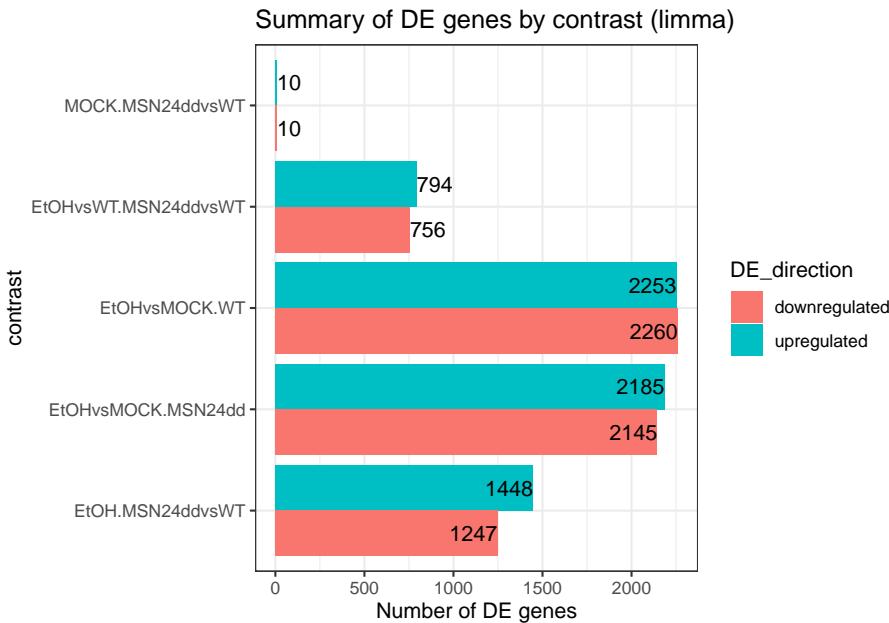
```

# let's use the res_all object from the 08_DE_limma exercise:
res_all_limma <- read_rds('https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/analysis/yeast_decideTests_all_edgeR.R')

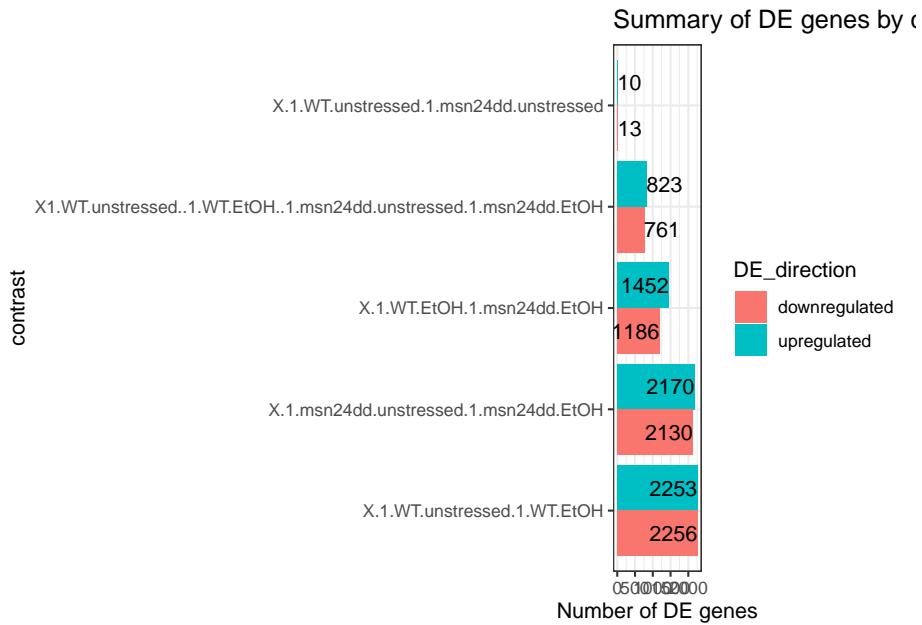
# read.delim(
#   'https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/analysis/yeast_decideTests_all_edgeR.R',
#   sep = "\t",
#   header = T,
#   row.names = 1
# ) %>% rownames_to_column("gene")

res_all_limma %>%
  decideTests(p.value = 0.05, lfc = 0) %>%
  as.data.frame() %>%
  rownames_to_column("gene") %>%
  pivot_longer(c(-gene), names_to = "contrast", values_to = "DE_direction") %>%
  group_by(contrast) %>%
  summarise(
    upregulated = sum(DE_direction == 1),
    downregulated = sum(DE_direction == -1)
  ) %>%
  pivot_longer(c(-contrast), names_to = "DE_direction", values_to = "n_genes") %>%
  ggplot(aes(x = contrast, y = n_genes, fill = DE_direction)) +
  geom_col(position = "dodge") +
  theme_bw() +
  coord_flip() +
  geom_text(aes(label = n_genes),
            position = position_dodge(width = .9),
            hjust = "inward") +
  labs(y = "Number of DE genes") +
  ggtitle("Summary of DE genes by contrast (limma)")

```

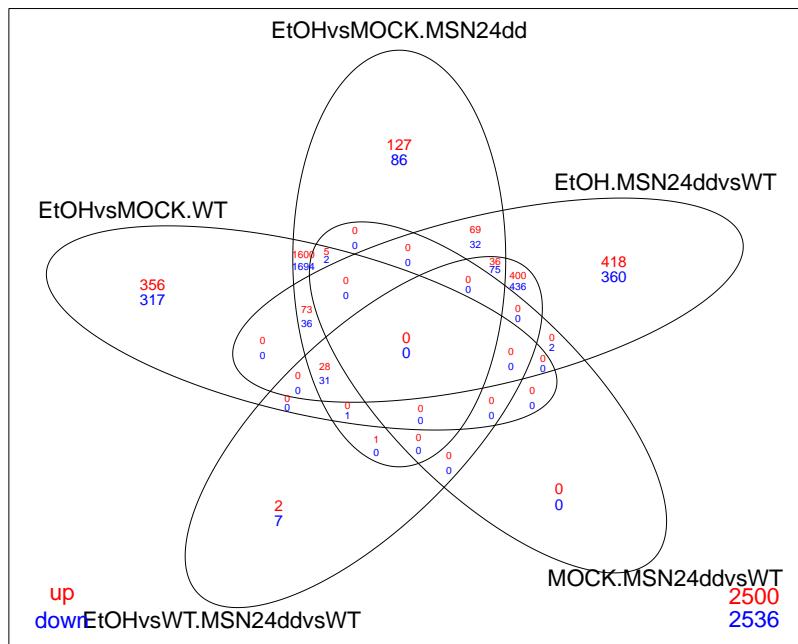


```
# how to do the same for edgeR
decideTests_all_edgeR %>%
  pivot_longer(-gene, names_to = "contrast", values_to = "DE_direction") %>%
  group_by(contrast) %>%
  summarise(
    upregulated = sum(DE_direction == 1),
    downregulated = sum(DE_direction == -1)
  ) %>%
  pivot_longer(-contrast, names_to = "DE_direction", values_to = "n_genes") %>%
  mutate(contrast = fct_reorder(contrast, 1/(1+n_genes))) %>%
  ggplot(aes(x = contrast, y = n_genes, fill = DE_direction)) +
  geom_col(position = "dodge") +
  theme_bw() +
  coord_flip() +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  geom_text(aes(label = n_genes),
            position = position_dodge(width = .9),
            hjust = "inward") +
  labs(y="Number of DE genes") +
  ggtitle("Summary of DE genes by contrast (edgeR)")
```

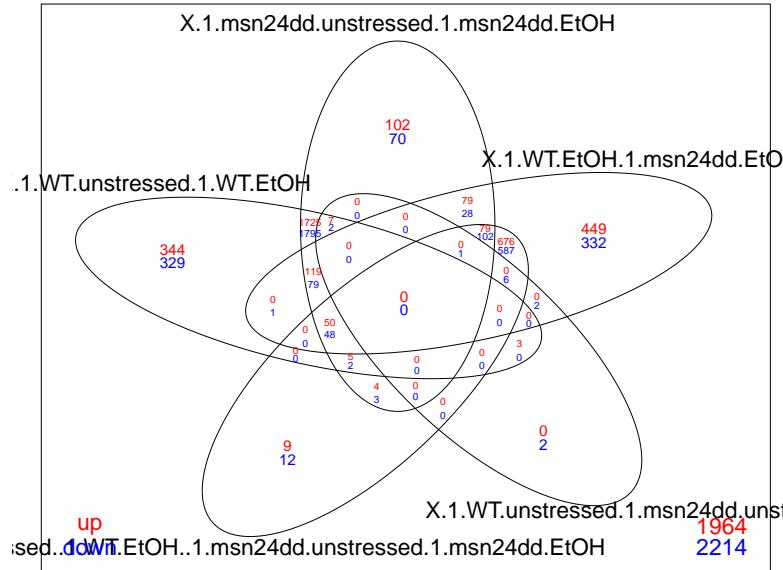


If we want to show the same amount of information, in a more informative way, a venn diagram is often a better alternative. Here's an easy way to get that visualization if you use either edgeR or limma for your analysis.

```
# same as before, we can make the plot from the decideTests output
res_all_limma %>%
  decideTests(p.value = 0.01, lfc = 0) %>%
  vennDiagram(include=c("up", "down"),
              lwd=0.75,
              mar=rep(2,4), # increase margin size
              counts.col= c("red", "blue"),
              show.include=TRUE)
```



```
decideTests_all_edgeR %>%
  column_to_rownames("gene") %>%
  vennDiagram(include=c("up", "down"),
              lwd=0.75,
              mar=rep(4,4), # increase margin size
              counts.col= c("red", "blue"),
              show.include=TRUE)
```



Venn diagrams are useful for showing gene counts as well as there overlaps between contrasts. A useful gui based web-page for creating venn diagrams includes: <https://eulerr.co/>. If you enjoy coding, it also exists as an R package (<https://cran.r-project.org/web/packages/eulerr/index.html>).

## 9.7 Exercise

1. Modify the code below to find out how many genes are upregulated ( $p.value < 0.01$  and  $|lfc| > 1$ ) in the ethanol stress response of both WT cells and msn2/4 mutants.

```
# here are all of the contrasts  
colnames(res.all.limma)
```

```

## [1] "EtOHvsMOCK.WT"           "EtOHvsMOCK.MSN24dd"    "EtOH.MSN24ddvsWT"
## [4] "MOCK.MSN24ddvsWT"       "EtOHvsWT.MSN24ddvsWT"

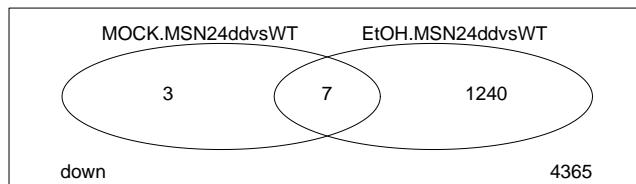
# select the correct two and replace them below
res_all_limma %>%
  decideTests(p.value = 0.05, lfc = 0) %>%

```

```

data.frame() %>%
  # change the columns selected in this select command
  dplyr::select(c("MOCK.MSN24ddvsWT", "EtOH.MSN24ddvsWT")) %>%
  vennDiagram(include="down",
              lwd=0.75,
              mar=rep(0,4), # increase margin size
              # counts.col= c("red", "blue"),
              show.include=TRUE
)

```



```
pander::pander(sessionInfo())
```

**R version 4.3.1 (2023-06-16)**

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8|en\_US.UTF-8||C||en\_US.UTF-8|en\_US.UTF-8

**attached base packages:** stats4, stats, graphics, grDevices, utils, datasets, methods and base

**other attached packages:** ggrepel(v.0.9.4), viridis(v.0.6.4), viridisLite(v.0.4.2), scales(v.1.2.1), Glimma(v.2.10.0), DESeq2(v.1.40.2), edgeR(v.3.42.4), limma(v.3.56.2), reactable(v.0.4.4), webshot2(v.0.1.1), statmod(v.1.5.0), Rsubread(v.2.14.2), ShortRead(v.1.58.0), GenomicAlignments(v.1.36.0), SummarizedExperiment(v.1.30.2), MatrixGenerics(v.1.12.3), matrixStats(v.1.0.0), Rsamtools(v.2.16.0), GenomicRanges(v.1.52.1), Biostrings(v.2.68.1), GenomeInfoDb(v.1.36.4), XVector(v.0.40.0), BiocParallel(v.1.34.2), Rfastp(v.1.10.0), org.Sc.sgd.db(v.3.17.0), AnnotationDbi(v.1.62.2), IRanges(v.2.34.1), S4Vectors(v.0.38.2), Biobase(v.2.60.0), BiocGenerics(v.0.46.0), clusterProfiler(v.4.8.2), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pander(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3), forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyrr(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)

**loaded via a namespace (and not attached):** *splines(v.4.3.1), later(v.1.3.1), bitops(v.1.0-7), ggplotify(v.0.1.2), polyclip(v.1.10-6), lifecycle(v.1.0.3), sf(v.1.0-14), rprojroot(v.2.0.3), vroom(v.1.6.4), processx(v.3.8.2), lattice(v.0.21-9), MASS(v.7.3-60), crosstalk(v.1.2.0), magrittr(v.2.0.3), rmarkdown(v.2.25), yaml(v.2.3.7), cowplot(v.1.1.1), chromote(v.0.1.2), DBI(v.1.1.3), RColorBrewer(v.1.1-3), abind(v.1.4-5), zlibbioc(v.1.46.0), ggraph(v.2.1.0), RCurl(v.1.98-1.12), yulab.utils(v.0.1.0), tweenr(v.2.0.2), GenomeInfoDbData(v.1.2.10), enrichplot(v.1.20.0), units(v.0.8-4), codetools(v.0.2-19), DelayedArray(v.0.26.7), DOSE(v.3.26.1), ggforce(v.0.4.1), tidyselect(v.1.2.0), aplot(v.0.2.2), farver(v.2.1.1), webshot(v.0.5.5), jsonlite(v.1.8.7), e1071(v.1.7-13), ellipsis(v.0.3.2), tidygraph(v.1.2.3), tools(v.4.3.1), treeio(v.1.24.3), Rcpp(v.1.0.11), glue(v.1.6.2), gridExtra(v.2.3), xfun(v.0.40), qvalue(v.2.32.0), websocket(v.1.4.1), withr(v.2.5.1), fastmap(v.1.1.1), latticeExtra(v.0.6-30), fansi(v.1.0.5), digest(v.0.6.33), timechange(v.0.2.0), R6(v.2.5.1), gridGraphics(v.0.5-1), colorspace(v.2.1-0), GO.db(v.3.17.0), jpeg(v.0.1-10), RSQLite(v.2.3.1), utf8(v.1.2.3), generics(v.0.1.3), data.table(v.1.14.8), class(v.7.3-22), graphlayouts(v.1.0.1), httr(v.1.4.7), htmlwidgets(v.1.6.2), S4Arrays(v.1.0.6), scatterpie(v.0.2.1), pkgconfig(v.2.0.3), gtable(v.0.3.4), blob(v.1.2.4), hwriter(v.1.3.2.1), shadowtext(v.0.1.2), htmltools(v.0.5.6.1), bookdown(v.0.36), fgsea(v.1.26.0), png(v.0.1-8), snakecase(v.0.11.1), ggfun(v.0.1.3), rstudioapi(v.0.15.0), tzdb(v.0.4.0), reshape2(v.1.4.4), rjson(v.0.2.21), nlme(v.3.1-163), proxy(v.0.4-27), cachem(v.1.0.8), KernSmooth(v.2.23-22), RVenn(v.1.1.0), parallel(v.4.3.1), HDO.db(v.0.99.1), pillar(v.1.9.0), grid(v.4.3.1), vctrs(v.0.6.4), promises(v.1.2.1), archive(v.1.1.5), evaluate(v.0.22), cli(v.3.6.1), locfit(v.1.5-9.8), compiler(v.4.3.1), rlang(v.1.1.1), crayon(v.1.5.2), labeling(v.0.4.3), classInt(v.0.4-10), interp(v.1.1-4), reactR(v.0.5.0), ps(v.1.7.5), plyr(v.1.8.9), fs(v.1.6.3), stringi(v.1.7.12), deldir(v.1.0-9), munsell(v.0.5.0), lazyeval(v.0.2.2), GOSemSim(v.2.26.1), Matrix(v.1.6-1.1), hms(v.1.1.3), patchwork(v.1.1.3), bit64(v.4.0.5), KEGGREST(v.1.40.1), memoise(v.2.0.1), ggtext(v.3.8.2), fastmatch(v.1.1-4), bit(v.4.0.5), downloader(v.0.4), ape(v.5.7-1) and gson(v.0.1.0)*

# **Chapter 10**

## **Clustering**

last updated: 2023-10-27

### **10.1 Description**

This activity is intended to familiarize you with hierarchical clustering using Cluster 3.0 and visualization using Java TreeView.

### **10.2 Learning outcomes**

At the end of this exercise, you should be able to:

- Create a preclustering (PCL) file to load into Cluster 3.0.
- Perform hierarchical clustering with different settings.
- Visualize clustered data with TreeView
- Generate gene lists for clusters of interest for downstream functional analysis (e.g., GO enrichment)

### **10.3 Cluster 3.0**

There are lots of software packages that will perform clustering analysis. One of the original programs for hierarchical clustering was designed by Michael Eisen, which has been converted to an open source package with the current version of 3.0. Files generated following clustering analysis can be visualized using Java TreeView.

### 10.3.1 Generating a PCL file

Cluster reads in tab-delimited text files with a minimum of 1 column with the gene IDs, columns of your expression values (generally logFC, but can be TPMs), and then a row with column names. I also include an extra column with gene annotations and gene weight (GWEIGHT, all set to 1 to start) and experiment weight (EWEIGHT, also set to 1 for all). More on what weights are to follow. To open a PCL file, select from the Cluster drop-down menu “File” -> “Open Data.”

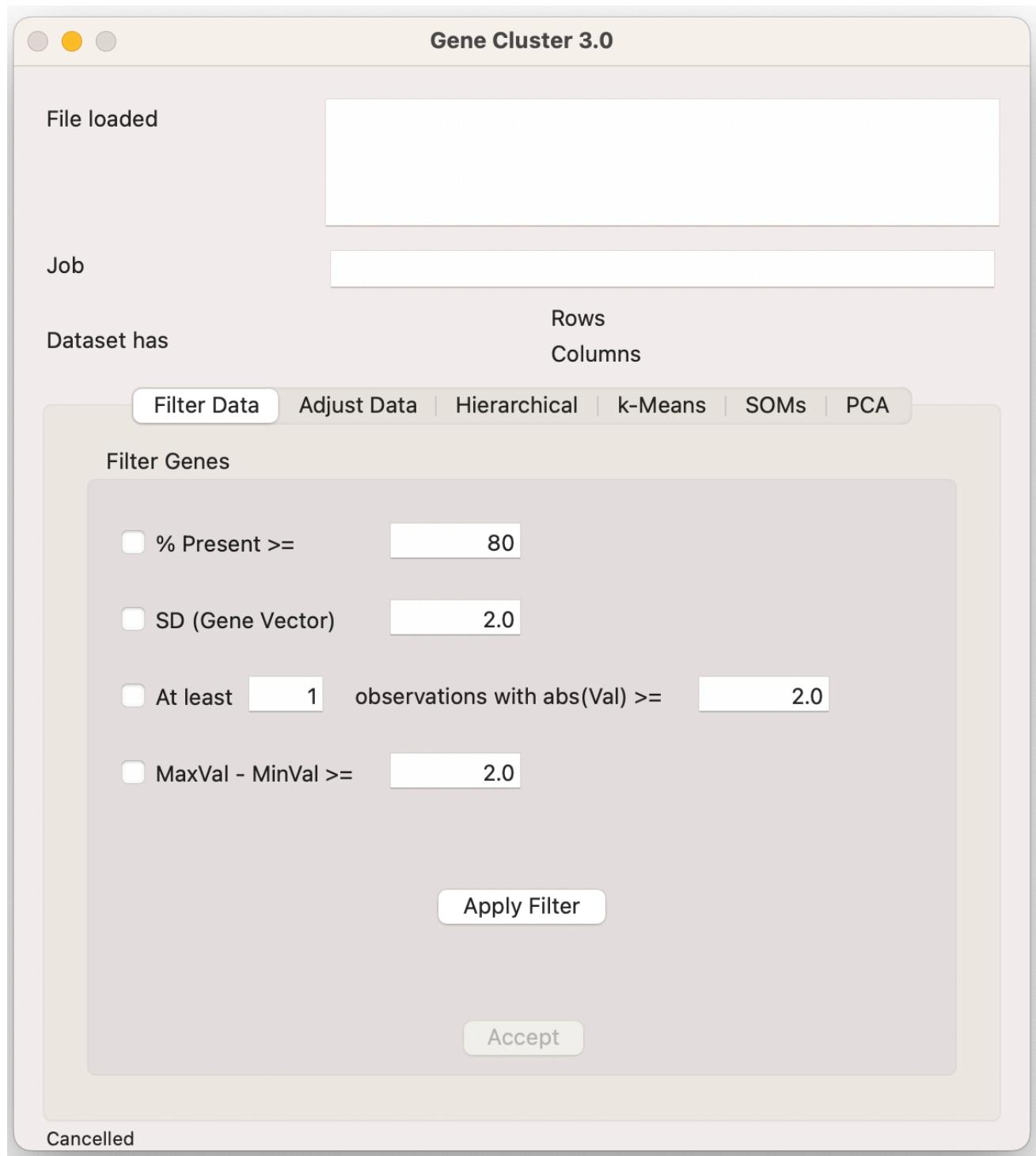
ID	NAME	GWEIGHT	logFC: YJM11	logFC: YPS60	logFC: YPS161
EWEIGHT			1	1	1
YAL001C	YAL001C T	1	0.382109	0.99966	1.218873
YAL002W	YAL002W	1	1.702246	1.461561	1.442073
YAL003W	YAL003W	1	-1.206682	-0.893712	-0.790667
YAL005C	YAL005C S	1	5.448996	5.948278	5.991898
YAL007C	YAL007C E	1	0.330388	1.21717	1.234555
YAL008W	YAL008W	1	1.788012	2.049009	1.816677
YAL009W	YAL009W	1	0.2843	0.412543	0.238627
YAL010C	YAL010C N	1	0.476174	0.289204	0.411537
YAL013W	YAL013W	1	0.110431	-0.623527	-0.832608
YAL014C	YAL014C S	1	0.807763	-0.018814	-0.21106
YAL015C	YAL015C N	1	-0.915333	-1.254105	-1.72367
YAL016C-A	#N/A	1	2.489218	0.178638	0.617952
YAL016W	YAL016W	1	0.022187	1.330222	1.007308
YAL017W	YAL017W	1	1.936083	2.659995	2.772925
YAL018C	YAL018C Y	1	4.717415	0.988252	0.672135
YAL019W	YAL019W	1	-2.131094	-1.364105	-0.844628
YAL021C	YAL021C C	1	-0.642581	0.070374	-0.206866
YAL022C	YAL022C F	1	0.244989	0.363592	0.158775
YAL023C	YAL023C P	1	-0.074167	0.761547	0.591847
YAL024C	YAL024C L	1	-0.611652	-0.549697	-0.114888
YAL025C	YAL025C N	1	-6.051175	-5.805034	-5.800037
YAL026C	YAL026C D	1	-0.75401	-0.066808	0.122285
YAL026C-A	#N/A	1	-1.016489	-0.394508	-0.617914
YAL028W	YAL028W	1	3.501061	3.590619	3.717687
YAL029C	YAL029C N	1	-1.75696	-0.738218	-0.41407

### 10.3.2 Filtering data

Cluster allows filtering on:

- **% Present  $\geq X$ .** Genes with missing values above that cutoff are removed from the analysis.
- **SD (Gene Vector)  $\geq X$ .** Genes with standard deviations above that cutoff are removed from the analysis.
- **At least X Observations with  $\text{abs}(\text{Val}) \geq Y$ .** Genes with fewer than the selected number of observations above a cutoff are removed from the analysis. E.g., At least 1 observation with a logFC of  $+/ - 1$ .
- **MaxVal-MinVal  $\geq X$ .** Genes whose maximum minus minimum values are less than the cutoff are removed.

I generally filter on 80% or 100% present, and will often only include significantly differentially expressed genes in my PCL file (instead of applying a specific filter).



### 10.3.3 Hierarchical Clustering

Cluster allows you to perform hierarchical clustering on genes, arrays (i.e., samples/experiments), or both. Check the “Cluster” box for one or both, and then choose your similarity metric. The most common are Pearson correlation (either centered or uncentered) and Euclidian distance. Finally, you click on the linkage type to start the clustering (centroid, single, complete, or average).

By default, all experiments (arrays) are treated equally (set to 1). Sometimes you have more than one type of sample than another (e.g., 6 treatments and 3 controls). This unbalanced design means that the treatment groups will disproportionately influence the clustering. The “Calculate weights” tab reapportions how much each experiment affects the clustering, ideally up-weighting the controls and down-weighting the treatments.

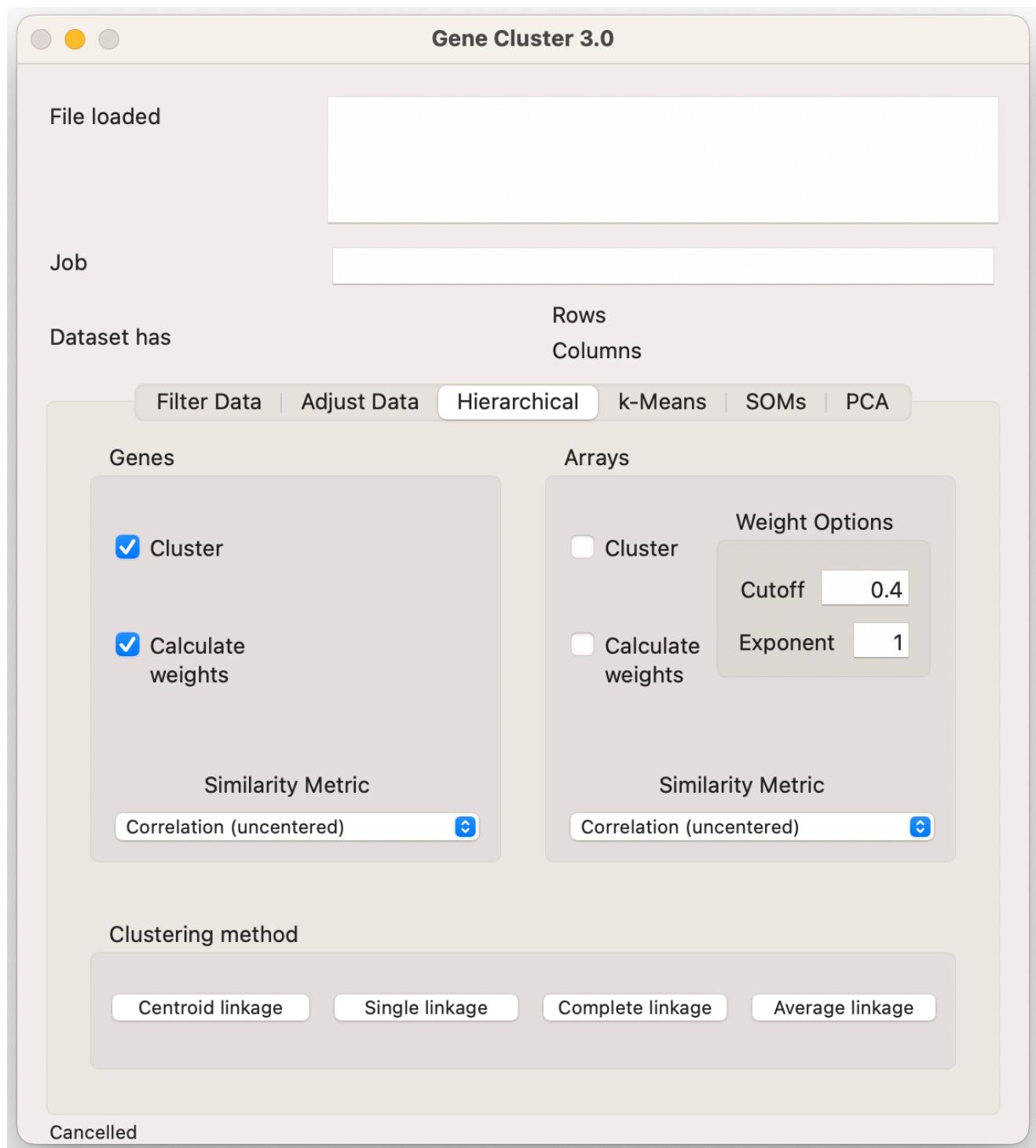
This is implemented through the following equation, where L is the local density score for each row (i):

$$L(i) = \sum_{j \text{ with } d(i,j) < k} \left( \frac{k - d(i,j)}{k} \right)^n$$

The user supplies the exponent value (n) and the cutoff (k). Common values for the cutoff are 0.7 to 1, and 0.4 to 0.8 for the exponent. The clustered data file will show the re-calculated weights, which you can use to refine your weighting choices.

The outputs of Cluster will be a clustered data table (JobName.cdt), and the gene (g) and/or array (a)tree files (JobName.gtr, JobName.atr).

**Make sure your job name is informative (e.g., EtOH\_Response\_CenteredPearson\_Centroid)**



### 10.3.4 K-Means Clustering

Cluster also allows for  $k$ -means clustering, where you can organize genes into  $k$  clusters using the same similarity metric options as for hierarchical clustering. You can use the following code to estimate the optimal number of clusters via three methods (wss, silhouette, and gap statistic):

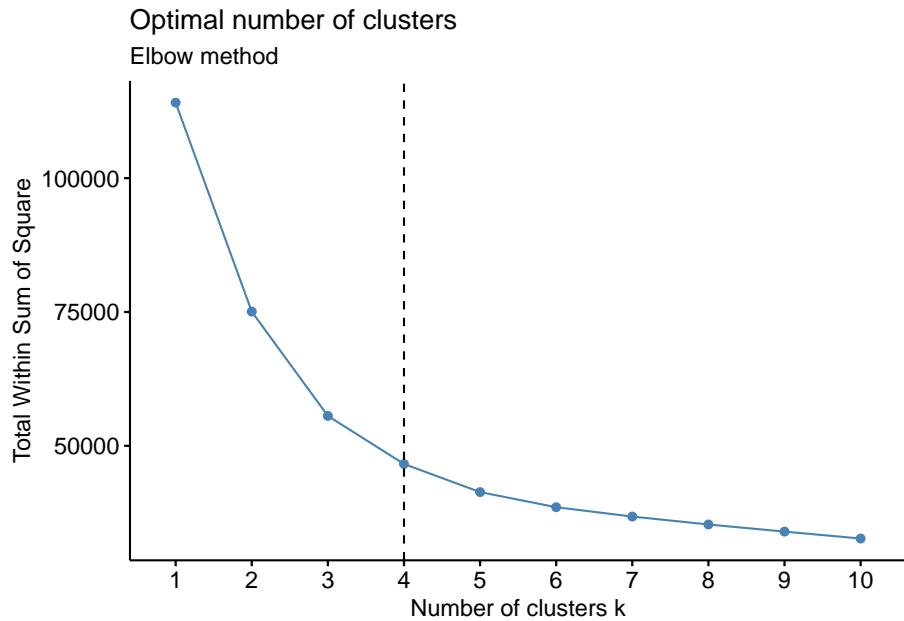
```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# let's load all of the files we were using and want to have again today
p_load("readr", "factoextra", "NbClust")

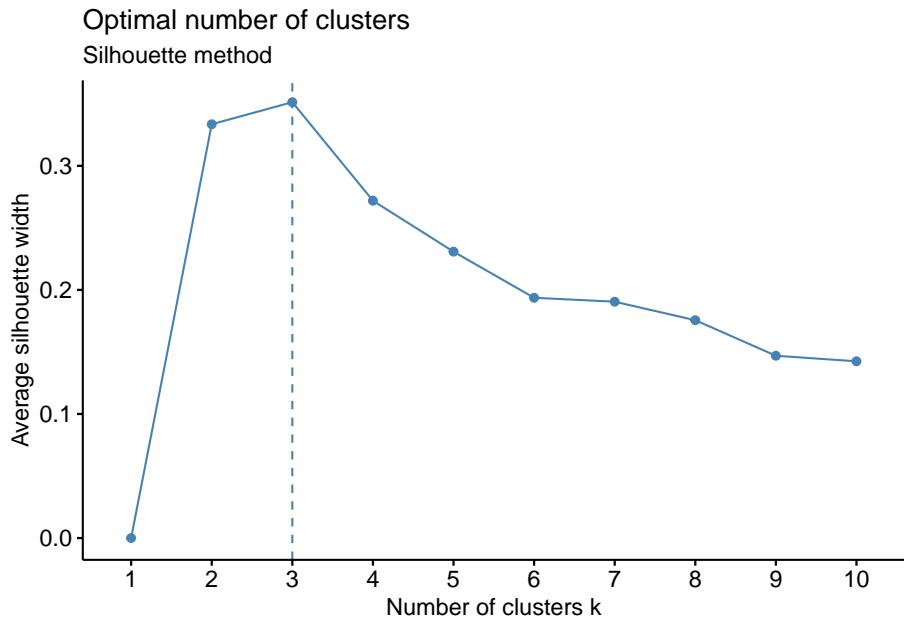
# Import From Text (readr) to load the pcl file. Change code below to PCL_file <- readr::read_csv("PCL.csv")
PCL_file <- data.table::fread("https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/PCL.csv")

# removing the 3 columns and 1 row that do not contain logFC data
PCL_nbclust = PCL_file[,-c(1,2,3)]
PCL_nbclust = PCL_nbclust[-1,]

# Elbow method
fviz_nbclust(PCL_nbclust, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2) +
  labs(subtitle = "Elbow method")
```



```
# Silhouette method  
fviz_nbclust(PCL_nbclust, kmeans, method = "silhouette") +  
  labs(subtitle = "Silhouette method")
```



```
# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
set.seed(123)
fviz_nbclust(PCL_nbclust, kmeans, nstart = 25, method = "gap_stat", nboot = 50) +
  labs(subtitle = "Gap statistic method")

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```









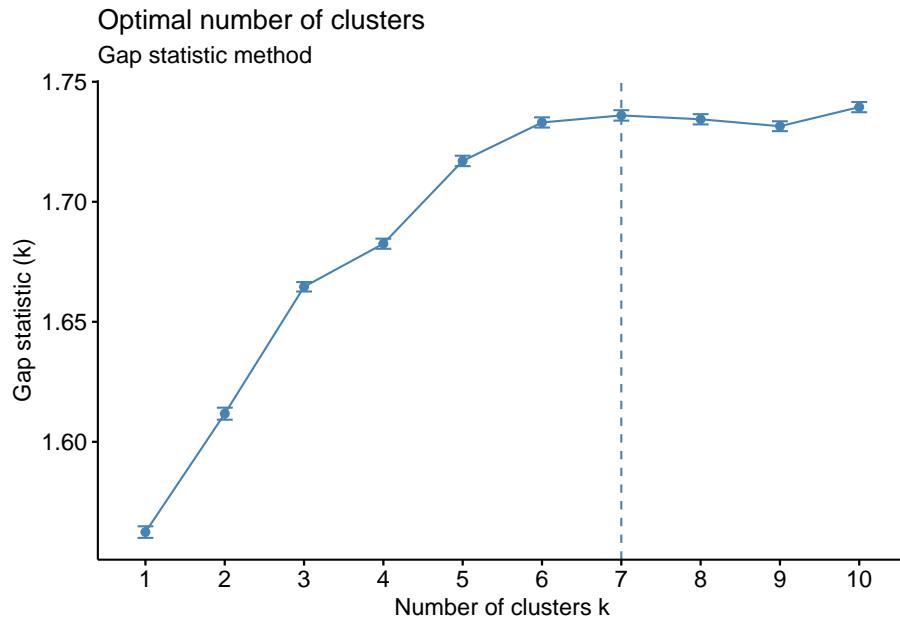






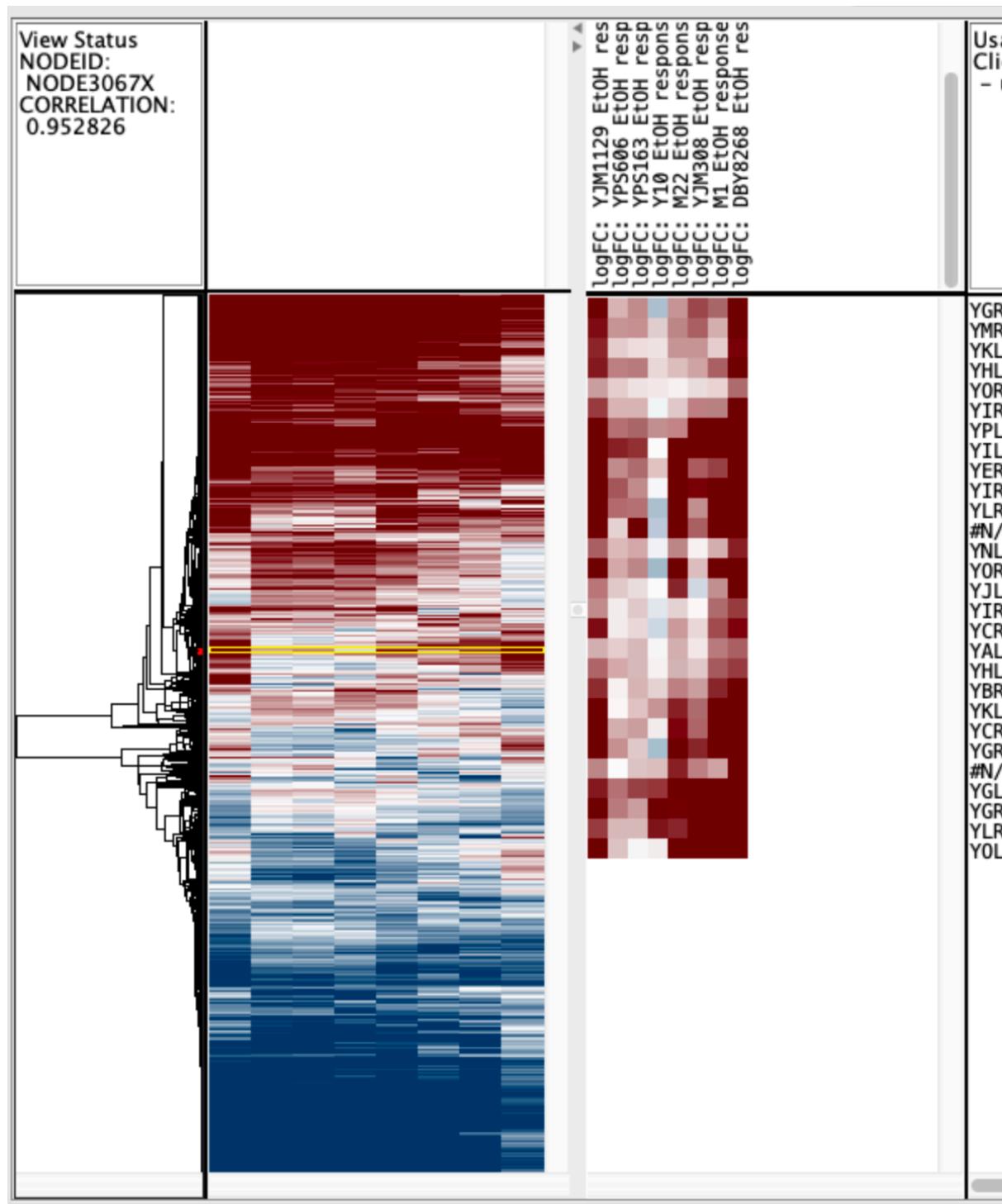






## 10.4 Visualizing Clusters with Java TreeView

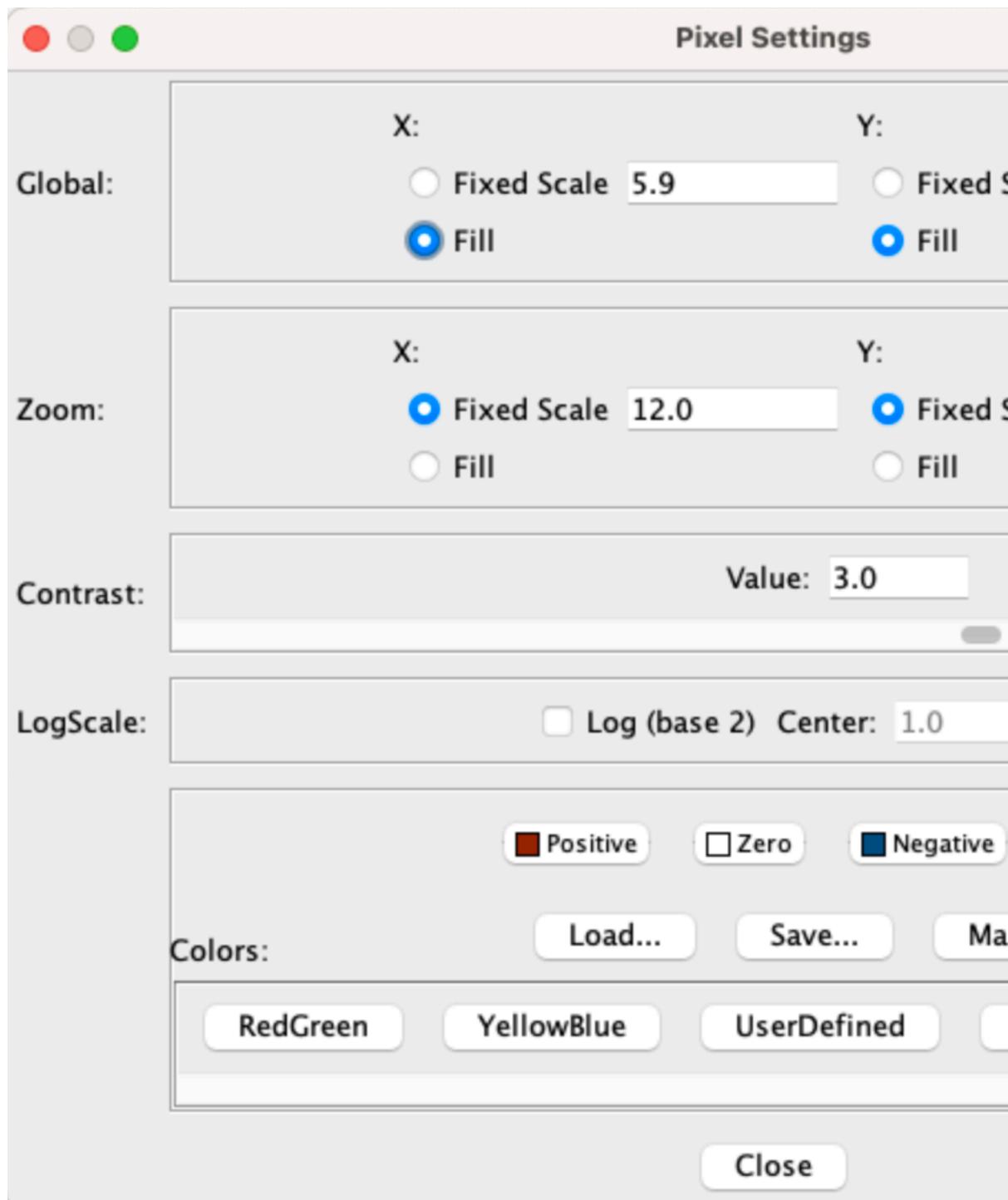
Treeview uses the cdt file to generate a heat map of the clustered data and the gtr/atr files to draw the tree (similar to a phylogenetic tree). Here's an example heat map.



The large panel in the middle is the heat map of all the data (global), with each row being the expression values for a single gene, and each column being a single experiment/sample. The inset shows a zoomed image for a selected portion of the tree, and those are obtained by clicking on the heat map to select a single gene, and then moving the cursor into the tree region and pressing the “up” arrow on the keyboard to move node-by-node up the tree. On top of the inset heat map are the sample names, and to the right are the gene names and annotations. The far top left shows the correlation value for that particular node the tree.

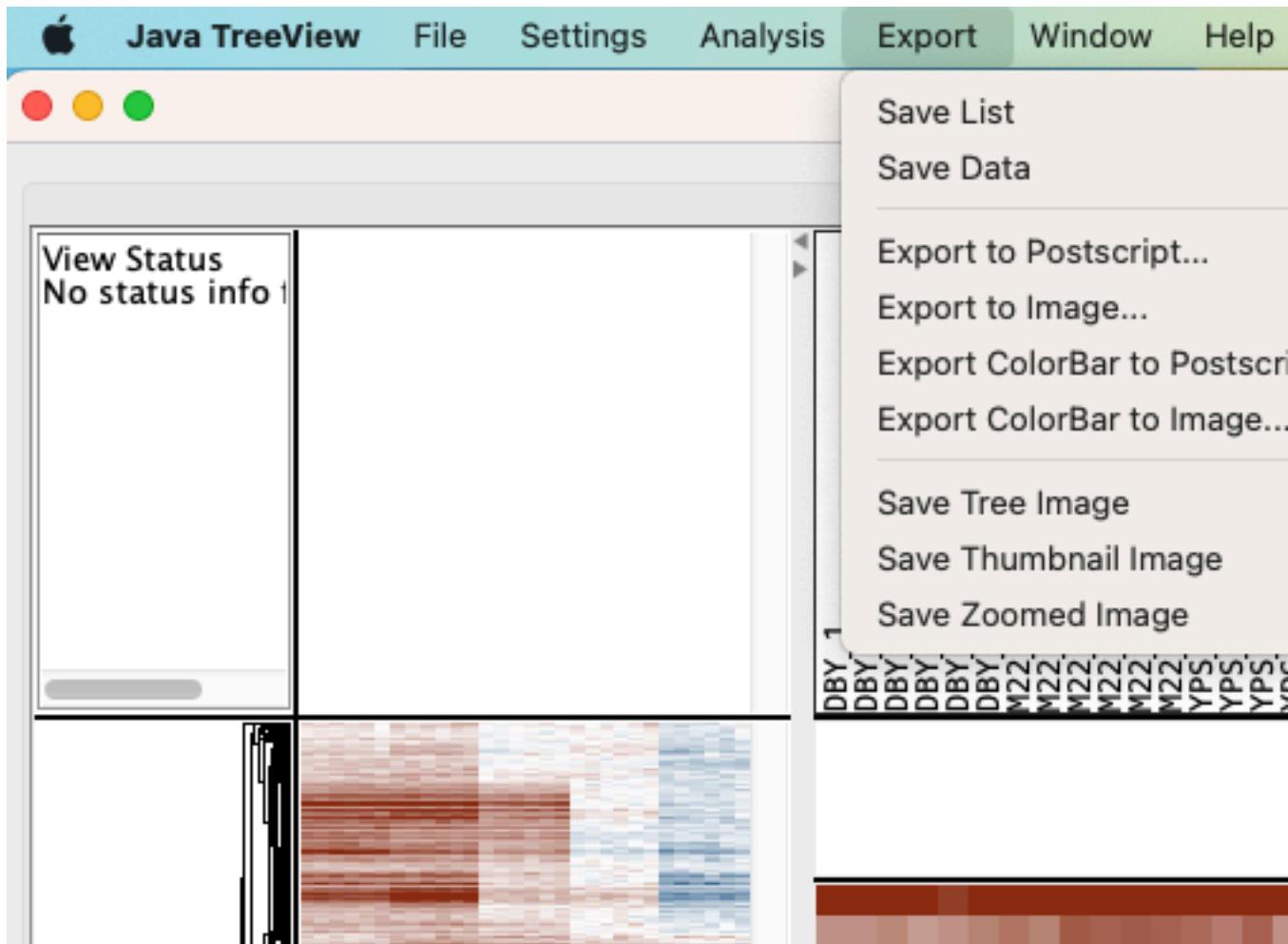
#### 10.4.1 Changing pixel settings.

The colors on the heat map are user defined by the pixel settings tab, where you can set the max logFC for the color scheme and change the colors for the positive, negative, and zero values. The default for the global tree is to not show the full scale, but you can set it to “fill” to fit the entire screen.

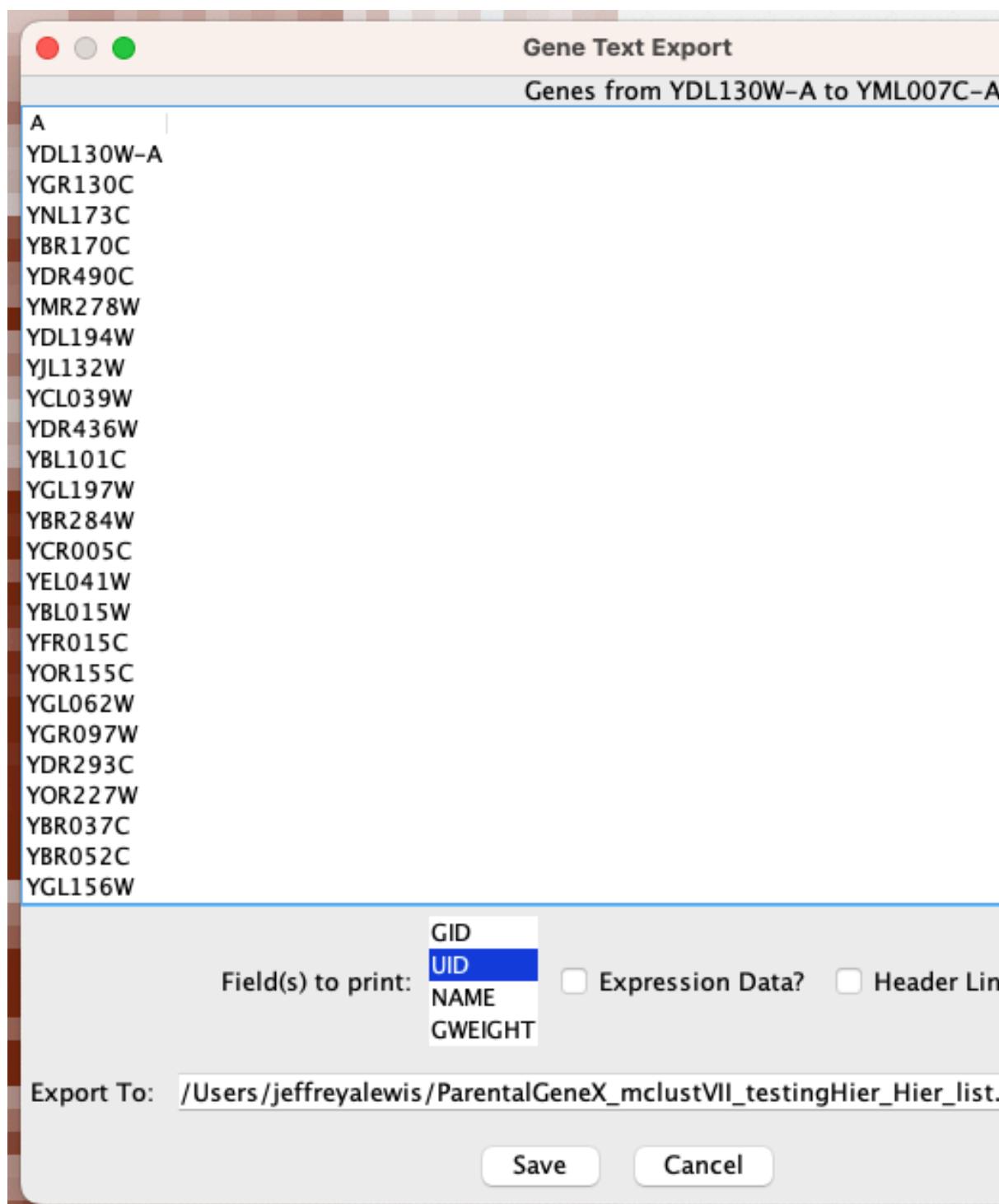


### 10.4.2 Selecting groups of genes.

When picking out clusters of genes, we often want to know their functional enrichments. We can select groups of genes by clicking on the “Export” tab and then “Save List.”



The resulting list can then be either copy and pasted into Excel or saved directly as a text file. The lists can then be used as inputs for clusterProfiler or online enrichment analysis tools (such as the Princeton Go Finder).



## 10.5 Performing clustering on yeast stress data

Now it's your turn to play around with data. Download the Gasch\_2000\_stress.pcl file from OneDrive (Data Files -> Msn24\_EtOH -> Clustering) and visualize clustering outputs using different methods:

- 1) Compare with and without filtering on 80% present, and compare with and without filtering on having a certain number of values above an logFC of  $|1|$ .
- 2) Compare different similarity metrics. The ones most commonly used are (Pearson) correlation (centered or uncentered) and Euclidean difference, but see what clustering with other metrics looks like.
- 3) What happens to the heat map and tree look when you use different linkage methods (centroid, single, complete, or average)?

## 10.6 Questions

1. Using the Gasch\_2000 dataset, filter the data on 80% present, and then cluster with uncentered correlation, calculated weights on arrays (so click on the box in "Genes") with a cutoff of 0.7 and an exponent of 1, and click centroid linkage.

Search for the gene *DCG1* (a gene of unknown function). Based on the genes immediately surround *DCG1* in the heat map, what would you predict the function of *DCG1* to be and why?

2. Download the DE\_yeast\_TF\_stress.txt dataset from the OneDrive (in the same Clustering folder as for the Gasch\_2000 dataset). These data include logFC and FDR corrected-pvalues for both the NaCl (salt) response and the EtOH response for WT yeast, an *msn2/4* deletion mutant (which we've looked at before), and deletion mutants for two other transcription factors, *yap1*, and *skn7*.

Create a PCL file combining just the logFC data for the EtOH and NaCl responses for the WT and mutant strains (so, leave out the WT vs mutant comparisons). Cluster the genes using the Correlation (uncentered), which is the uncentered Pearson correlation, and click "Centroid linkage." Save the job with a name that denotes those choices. Then, change the similarity metric to Euclidean distance and repeat the clustering (still Centroid linkage).

How does using Euclidean distance affect the clustering and why? When might you want to use Euclidean distance as your similarity metric?

3. Repeat the clustering using Absolute correlation (uncentered) and Centroid linkage. How does this affect the clustering and why? Can you think of a circumstance where Absolute correlation would be useful?
4. Make two new PCL files separating the ethanol responses and salt responses, and cluster the data separately. This time cluster on arrays as well as genes. Try different filters and clustering methods until you find one that you feel captures the data, and note your clustering parameters.

Based on your clustering, which transcription factor looks to be most responsible for the regulating the ethanol response? Which transcription factors seems most responsibel for the salt response? Looking back at the FDR corrected p-values for each TF's response (WT v mutant comparisons), does this match your expectations from the clustering. Why might clustering and differential expression analysis yield different answers to the question of which TF is most important for a response?

5. Using the clustering that you settled on for question #4, identify the single main cluster for each that contains genes affected by the muations in *msn2/4*. Make a figure highlighting those clusters by exporting the thumbnail images and importing into PowerPoint (or another graphics program if you prefer) and drawing a line next to the clusters (e.g., Figure 4 from this paper). Use the Princeton GO term Finder to identify BP enrichments for those clusters, and annotate the top 5 terms to the figure. Save as a PDF to embed into your homework document.

# Chapter 11

## KEGG Analysis

last updated: 2023-10-27

### Package Install

As usual, make sure we have the right packages for this exercise

```
if (!require("pacman")) install.packages("pacman"); library(pacman)

# temporary install to fix bug in package
pacman::p_install_version("dbplyr", version = "2.3.4")

## 
## Version of dbplyr (v. 2.3.4) is suitable

# let's load all of the files we were using and want to have again today
p_load("tidyverse", "knitr", "readr",
       "pander", "BiocManager",
       "dplyr", "stringr", "DOSE",
       "purrr", # for working with lists (beautify column names)
       "reactable", # for pretty tables.
       "BiocFileCache" # for saving downloaded data files
       )

# a package from github to install (using pacman library to install)
p_install_gh("noriakis/ggkegg")

## Skipping install of 'ggkegg' from a github remote, the SHA1 (0cece6db) has not changed since 1
##   Use `force = TRUE` to force installation
```

```
##  
## The following packages were installed:  
## ggkegg  
  
# We also need these Bioconductor packages today.  
p_load("edgeR",  
       "AnnotationDbi", "org.Sc.sgd.db",  
       "pathview", "clusterProfiler", "ggupset",  
       "KEGGgraph", "ggkegg", "patchwork",  
       "igraph", "tidygraph", "ggfx")
```

## 11.1 Description

In this class exercise, we will explore the use of KEGG pathways in genomic data analysis. KEGG is a valuable resource for understanding biological pathways and functions associated with genomic data.

## 11.2 Learning Outcomes

At the end of this exercise, you should be able to:

- Understand the basics of KEGG pathways.
- Learn how to retrieve and analyze pathway information using R.
- Apply KEGG analysis to genomic data.
- Identify paralog differences within KEGG pathways

```
library(edgeR)  
library(org.Sc.sgd.db)  
# for ease of use, set max number of digits after decimal  
options(digits=3)
```

## 11.3 Loading in the edgeR DE gene file output.

```
# Choose topTags destination  
dir_output_edgeR <-  
  path.expand("~/Desktop/Genomic_Data_Analysis/Analysis/edgeR/")  
  
# for this, let's load the res objects as an R data object.  
res <- read_rds(file = paste0(dir_output_edgeR, "yeast_res_edgeR.Rds"))
```

```

# the below code downloads the file from the internet if you don't have it on your computer
# comment out the above line of code if it gives an error.
if (!exists("res", envir = .GlobalEnv)) {
  # If variable doesn't exist, load from
  url <- "https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/yeast_res_all_edgeR.RData"
  res <- read_rds(url)
  # assign("res", data, envir = .GlobalEnv)
  cat(paste("Loaded res from", url, "\n"))
  rm("url", envir = .GlobalEnv)
} else {
  url <- "https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/yeast_res_all_edgeR.RData"
  cat(paste("res object is already loaded. Skipping loading from", url, "\n"))
  rm("url", envir = .GlobalEnv)
}

## res object is already loaded. Skipping loading from https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/yeast_res_all_edgeR.RData

# let's also get the res object for all of the contrasts at once, (from GitHub)
res_all <- read_rds("https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/yeast_res_all_edgeR.RData")

```

## 11.4 KEGG Analysis

Recall that the `res` object from the `edgeR` Differential Expression analysis was defined as `glmQLFTest(fit, contrast = my.contrasts[, "EtOHvsWT.MSN24ddvsWT"])`. In other words, we are looking at the difference in the ethanol stress response of the `Msn2/4` mutant and the `WT` cells. Recall that positive log<sub>2</sub> fold changes correspond to relatively higher expression of that gene in the `Msn2/4` mutant cells in response stress compared to how `WT` cells change during stress.

### 11.4.1 Using `kegg()` from `limma`

One approach is to do a simple KEGG enrichment with the `limma` package, using the function `kegg()`. This approach is nice, because we can directly load in the `res` object we created in the `edgeR` workflow, set our FDR cutoff, and the process happens automatically. See [https://www.kegg.jp/kegg/catalog/org\\_list.html](https://www.kegg.jp/kegg/catalog/org_list.html) or <https://rest.kegg.jp/list/organism> for list of organisms available.

```

# test for overrepresentation of KEGG pathways in given gene list
k <- kegg(res,
           species.KEGG = "sce", # three-letter KEGG species identifier.
           FDR = 0.01) #FDR cutoff we want in deciding which genes to include.

```

```
# we can see the p-value results for both up & down enrichment
k
```

		Pathway	N	Up	Down
##		Glycolysis / Gluconeogenesis	51	5	11
## sce00010		Citrate cycle (TCA cycle)	30	1	10
## sce00020		Pentose phosphate pathway	27	3	6
## sce00030		Pentose and glucuronate interconversions	10	0	3
## sce00040		Fructose and mannose metabolism	20	0	7
## sce00051		Galactose metabolism	22	0	7
## sce00052		Ascorbate and aldarate metabolism	11	2	4
## sce00053		Fatty acid biosynthesis	13	2	1
## sce00061		Fatty acid elongation	8	1	0
## sce00062		Fatty acid degradation	20	7	5
## sce00071		Steroid biosynthesis	18	6	3
## sce00100		Ubiquinone and other terpenoid-quinone biosynthesis	10	0	5
## sce00130		Oxidative phosphorylation	71	0	27
## sce00190		Arginine biosynthesis	16	3	1
## sce00220		Purine metabolism	57	9	8
## sce00230		Pyrimidine metabolism	31	5	2
## sce00240		Alanine, aspartate and glutamate metabolism	27	5	5
## sce00250		Glycine, serine and threonine metabolism	30	4	5
## sce00260		Monobactam biosynthesis	3	1	0
## sce00261		Cysteine and methionine metabolism	43	12	3
## sce00270		Valine, leucine and isoleucine degradation	14	6	4
## sce00280		Valine, leucine and isoleucine biosynthesis	12	8	0
## sce00290		Lysine biosynthesis	12	6	0
## sce00300		Lysine degradation	16	4	5
## sce00310		Arginine and proline metabolism	23	3	4
## sce00330		Carbapenem biosynthesis	3	1	0
## sce00332		Histidine metabolism	14	2	3
## sce00340		Tyrosine metabolism	13	3	2
## sce00350		Phenylalanine metabolism	7	2	0
## sce00360		Tryptophan metabolism	20	3	6
## sce00380		Phenylalanine, tyrosine and tryptophan biosynthesis	17	3	0
## sce00400		beta-Alanine metabolism	13	3	6
## sce00410		Taurine and hypotaurine metabolism	3	0	3
## sce00430		Phosphonate and phosphinate metabolism	4	1	0
## sce00440		Selenocompound metabolism	12	2	0
## sce00450		Cyanoamino acid metabolism	5	0	1
## sce00460		D-Amino acid metabolism	3	0	1
## sce00470		Glutathione metabolism	23	1	5
## sce00480		Starch and sucrose metabolism	39	1	20
## sce00500		N-Glycan biosynthesis	30	2	1
## sce00510		Other glycan degradation	1	0	1
## sce00511					

## sce00513	Various types of N-glycan biosynthesis	31	4	1
## sce00514	Other types of O-glycan biosynthesis	14	2	0
## sce00515	Mannose type O-glycan biosynthesis	7	0	0
## sce00520	Amino sugar and nucleotide sugar metabolism	31	0	7
## sce00561	Glycerolipid metabolism	31	2	13
## sce00562	Inositol phosphate metabolism	21	0	1
## sce00563	Glycosylphosphatidylinositol (GPI)-anchor biosynthesis	25	0	2
## sce00564	Glycerophospholipid metabolism	40	2	10
## sce00565	Ether lipid metabolism	8	0	3
## sce00590	Arachidonic acid metabolism	3	0	2
## sce00592	alpha-Linolenic acid metabolism	4	0	1
## sce00600	Sphingolipid metabolism	14	0	3
## sce00620	Pyruvate metabolism	48	12	12
## sce00630	Glyoxylate and dicarboxylate metabolism	27	4	6
## sce00640	Propanoate metabolism	13	2	1
## sce00650	Butanoate metabolism	9	3	4
## sce00660	C5-Branched dibasic acid metabolism	3	1	0
## sce00670	One carbon pool by folate	15	2	0
## sce00680	Methane metabolism	20	3	3
## sce00730	Thiamine metabolism	16	0	1
## sce00740	Riboflavin metabolism	16	0	1
## sce00750	Vitamin B6 metabolism	11	2	1
## sce00760	Nicotinate and nicotinamide metabolism	20	2	3
## sce00770	Pantothenate and CoA biosynthesis	23	10	4
## sce00780	Biotin metabolism	6	0	0
## sce00785	Lipoic acid metabolism	12	1	2
## sce00790	Folate biosynthesis	11	0	1
## sce00791	Atrazine degradation	2	0	0
## sce00860	Porphyrin metabolism	17	0	2
## sce00900	Terpenoid backbone biosynthesis	19	5	1
## sce00909	Sesquiterpenoid and triterpenoid biosynthesis	2	0	0
## sce00910	Nitrogen metabolism	7	2	1
## sce00920	Sulfur metabolism	15	3	1
## sce00970	Aminoacyl-tRNA biosynthesis	40	4	1
## sce00999	Biosynthesis of various plant secondary metabolites	3	2	0
## sce01040	Biosynthesis of unsaturated fatty acids	10	1	0
## sce01100	Metabolic pathways	766	86	132
## sce01110	Biosynthesis of secondary metabolites	339	56	65
## sce01200	Carbon metabolism	104	10	24
## sce01210	2-Oxocarboxylic acid metabolism	42	13	6
## sce01212	Fatty acid metabolism	23	4	1
## sce01230	Biosynthesis of amino acids	122	32	7
## sce01232	Nucleotide metabolism	42	6	3
## sce01240	Biosynthesis of cofactors	126	18	18
## sce01250	Biosynthesis of nucleotide sugars	22	0	6
## sce02010	ABC transporters	18	1	3

## sce03008	Ribosome biogenesis in eukaryotes	74	32	1
## sce03010	Ribosome	174	61	1
## sce03013	Nucleocytoplasmic transport	65	8	2
## sce03015	mRNA surveillance pathway	48	6	1
## sce03018	RNA degradation	61	6	3
## sce03020	RNA polymerase	31	14	0
## sce03022	Basal transcription factors	32	2	0
## sce03030	DNA replication	31	4	0
## sce03040	Spliceosome	75	4	4
## sce03050	Proteasome	36	0	0
## sce03060	Protein export	22	1	1
## sce03082	ATP-dependent chromatin remodeling	35	3	1
## sce03083	Polycomb repressive complex	13	2	2
## sce03250	Viral life cycle - HIV-1	15	0	1
## sce03410	Base excision repair	25	2	0
## sce03420	Nucleotide excision repair	49	5	0
## sce03430	Mismatch repair	20	1	0
## sce03440	Homologous recombination	21	0	0
## sce03450	Non-homologous end-joining	10	0	0
## sce04011	MAPK signaling pathway - yeast	108	7	12
## sce04070	Phosphatidylinositol signaling system	23	0	1
## sce04111	Cell cycle - yeast	129	12	2
## sce04113	Meiosis - yeast	117	9	15
## sce04120	Ubiquitin mediated proteolysis	50	1	1
## sce04122	Sulfur relay system	7	0	1
## sce04130	SNARE interactions in vesicular transport	20	0	0
## sce04136	Autophagy - other	23	0	1
## sce04138	Autophagy - yeast	90	2	8
## sce04139	Mitophagy - yeast	40	1	3
## sce04141	Protein processing in endoplasmic reticulum	92	1	17
## sce04144	Endocytosis	79	1	11
## sce04145	Phagosome	35	0	0
## sce04146	Peroxisome	39	2	8
## sce04148	Efferocytosis	24	0	4
## sce04213	Longevity regulating pathway - multiple species	38	1	17
## sce04392	Hippo signaling pathway - multiple species	8	2	0
## sce04814	Motor proteins	24	2	1
##	P.Up P.Down			
## sce00010	4.59e-01 1.13e-02			
## sce00020	9.35e-01 4.71e-04			
## sce00030	4.21e-01 4.86e-02			
## sce00040	1.00e+00 7.17e-02			
## sce00051	1.00e+00 2.47e-03			
## sce00052	1.00e+00 4.55e-03			
## sce00053	2.47e-01 1.90e-02			
## sce00061	3.14e-01 7.50e-01			

```
## sce00062 5.17e-01 1.00e+00
## sce00071 1.02e-03 4.45e-02
## sce00100 3.13e-03 2.71e-01
## sce00130 1.00e+00 1.69e-03
## sce00190 1.00e+00 2.93e-10
## sce00220 1.57e-01 8.18e-01
## sce00230 5.55e-02 2.13e-01
## sce00240 1.27e-01 8.35e-01
## sce00250 7.96e-02 1.30e-01
## sce00260 2.61e-01 1.80e-01
## sce00261 2.39e-01 1.00e+00
## sce00270 2.03e-04 8.24e-01
## sce00280 6.85e-04 4.53e-02
## sce00290 1.11e-06 1.00e+00
## sce00300 2.45e-04 1.00e+00
## sce00310 4.42e-02 1.75e-02
## sce00330 3.23e-01 1.97e-01
## sce00332 2.39e-01 1.00e+00
## sce00340 3.47e-01 1.62e-01
## sce00350 9.69e-02 3.84e-01
## sce00360 1.18e-01 1.00e+00
## sce00380 2.49e-01 1.16e-02
## sce00400 1.79e-01 1.00e+00
## sce00410 9.69e-02 9.52e-04
## sce00430 1.00e+00 1.02e-03
## sce00440 3.05e-01 1.00e+00
## sce00450 2.80e-01 1.00e+00
## sce00460 1.00e+00 4.13e-01
## sce00470 1.00e+00 2.73e-01
## sce00480 8.77e-01 7.52e-02
## sce00500 9.72e-01 9.65e-11
## sce00510 7.49e-01 9.59e-01
## sce00511 1.00e+00 1.01e-01
## sce00513 2.81e-01 9.63e-01
## sce00514 3.47e-01 1.00e+00
## sce00515 1.00e+00 1.00e+00
## sce00520 1.00e+00 3.16e-02
## sce00561 7.65e-01 3.67e-06
## sce00562 1.00e+00 8.94e-01
## sce00563 1.00e+00 7.35e-01
## sce00564 8.74e-01 5.25e-03
## sce00565 1.00e+00 3.90e-02
## sce00590 1.00e+00 2.85e-02
## sce00592 1.00e+00 3.47e-01
## sce00600 1.00e+00 1.62e-01
## sce00620 6.17e-04 2.31e-03
```

```
## sce00630 2.03e-01 4.86e-02
## sce00640 3.14e-01 7.50e-01
## sce00650 3.69e-02 8.57e-03
## sce00660 2.39e-01 1.00e+00
## sce00670 3.79e-01 1.00e+00
## sce00680 2.49e-01 3.29e-01
## sce00730 1.00e+00 8.18e-01
## sce00740 1.00e+00 8.18e-01
## sce00750 2.47e-01 6.90e-01
## sce00760 5.29e-01 3.29e-01
## sce00770 9.09e-06 1.97e-01
## sce00780 1.00e+00 1.00e+00
## sce00785 6.65e-01 3.46e-01
## sce00790 1.00e+00 6.90e-01
## sce00791 1.00e+00 1.00e+00
## sce00860 1.00e+00 5.24e-01
## sce00900 2.03e-02 8.68e-01
## sce00909 1.00e+00 1.00e+00
## sce00910 1.18e-01 5.26e-01
## sce00920 1.36e-01 7.98e-01
## sce00970 4.63e-01 9.86e-01
## sce00999 2.13e-02 1.00e+00
## sce01040 5.97e-01 1.00e+00
## sce01100 5.52e-03 3.24e-11
## sce01110 1.16e-06 1.37e-07
## sce01200 4.18e-01 7.56e-05
## sce01210 3.33e-05 2.46e-01
## sce01212 1.34e-01 9.14e-01
## sce01230 6.35e-09 9.70e-01
## sce01232 1.53e-01 8.11e-01
## sce01240 2.36e-02 8.12e-02
## sce01250 1.00e+00 1.88e-02
## sce02010 8.06e-01 2.71e-01
## sce03008 1.21e-15 1.00e+00
## sce03010 3.84e-23 1.00e+00
## sce03013 2.00e-01 9.92e-01
## sce03015 2.35e-01 9.94e-01
## sce03018 4.39e-01 9.54e-01
## sce03020 7.75e-08 1.00e+00
## sce03022 7.80e-01 1.00e+00
## sce03030 2.81e-01 1.00e+00
## sce03040 9.02e-01 9.53e-01
## sce03050 1.00e+00 1.00e+00
## sce03060 8.65e-01 9.04e-01
## sce03082 5.97e-01 9.76e-01
## sce03083 3.14e-01 3.84e-01
```

```

## sce03250 1.00e+00 7.98e-01
## sce03410 6.53e-01 1.00e+00
## sce03420 4.24e-01 1.00e+00
## sce03430 8.38e-01 1.00e+00
## sce03440 1.00e+00 1.00e+00
## sce03450 1.00e+00 1.00e+00
## sce04011 8.41e-01 4.09e-01
## sce04070 1.00e+00 9.14e-01
## sce04111 4.47e-01 1.00e+00
## sce04113 6.99e-01 1.99e-01
## sce04120 9.90e-01 9.95e-01
## sce04122 1.00e+00 5.26e-01
## sce04130 1.00e+00 1.00e+00
## sce04136 1.00e+00 9.14e-01
## sce04138 9.97e-01 7.01e-01
## sce04139 9.74e-01 7.84e-01
## sce04141 1.00e+00 9.67e-03
## sce04144 9.99e-01 1.69e-01
## sce04145 1.00e+00 1.00e+00
## sce04146 8.65e-01 3.79e-02
## sce04148 1.00e+00 2.19e-01
## sce04213 9.69e-01 3.55e-08
## sce04392 1.49e-01 1.00e+00
## sce04814 6.30e-01 9.23e-01

```

Interesting, but note those are in order of the pathway ID, which are the left-most rowname values above. If we want to sort the rows based on their significance, we can use the `topKEGG` function to do so, like this:

```

# extract the top KEGG pathways from kegg output
topKEGG(k, sort="down")

```

	Pathway	N	Up	Down
## sce01100	Metabolic pathways	766	86	132
## sce00500	Starch and sucrose metabolism	39	1	20
## sce00190	Oxidative phosphorylation	71	0	27
## sce04213	Longevity regulating pathway - multiple species	38	1	17
## sce01110	Biosynthesis of secondary metabolites	339	56	65
## sce00561	Glycerolipid metabolism	31	2	13
## sce01200	Carbon metabolism	104	10	24
## sce00020	Citrate cycle (TCA cycle)	30	1	10
## sce00410	beta-Alanine metabolism	13	3	6
## sce00430	Taurine and hypotaurine metabolism	3	0	3
## sce00130	Ubiquinone and other terpenoid-quinone biosynthesis	10	0	5
## sce00620	Pyruvate metabolism	48	12	12

```

## sce00051          Fructose and mannose metabolism 20 0 7
## sce00052          Galactose metabolism 22 0 7
## sce00564          Glycerophospholipid metabolism 40 2 10
## sce00650          Butanoate metabolism 9 3 4
## sce04141          Protein processing in endoplasmic reticulum 92 1 17
## sce00010          Glycolysis / Gluconeogenesis 51 5 11
## sce00380          Tryptophan metabolism 20 3 6
## sce00310          Lysine degradation 16 4 5
##                  P.Up   P.Down
## sce01100 5.52e-03 3.24e-11
## sce00500 9.72e-01 9.65e-11
## sce00190 1.00e+00 2.93e-10
## sce04213 9.69e-01 3.55e-08
## sce01110 1.16e-06 1.37e-07
## sce00561 7.65e-01 3.67e-06
## sce01200 4.18e-01 7.56e-05
## sce00020 9.35e-01 4.71e-04
## sce00410 9.69e-02 9.52e-04
## sce00430 1.00e+00 1.02e-03
## sce00130 1.00e+00 1.69e-03
## sce00620 6.17e-04 2.31e-03
## sce00051 1.00e+00 2.47e-03
## sce00052 1.00e+00 4.55e-03
## sce00564 8.74e-01 5.25e-03
## sce00650 3.69e-02 8.57e-03
## sce04141 1.00e+00 9.67e-03
## sce00010 4.59e-01 1.13e-02
## sce00380 2.49e-01 1.16e-02
## sce00310 4.42e-02 1.75e-02

```

Notice I chose to sort on p-value for the “downregulated” genes (aka lower in the mutant). We could change “down” to “up” in the code above to sort on the other column.

### 11.4.2 Pathway Visualization

So, we’ve found out which KEGG pathways are enriched in our gene list for this comparison. A common thing we will want to do is visualize those enriched pathways. The Bioconductor package `pathview` allows us to visualize these pathways. If you prefer web based tools, it has a free web version at: <https://pathview.uncc.edu>

In addition to just the genes that are DE, we can also include information about their changes by coloring the corresponding parts of the graph accordingly.

```
# this makes sure you have pathview and tries to install it if you don't
if (!requireNamespace("pathview", quietly = TRUE))
  BiocManager::install("pathview")
library("pathview")

# save the logFC values from the res object to a new variable
gene_data_logFC <- res$table %>% dplyr::select(logFC)

# put the ORF names as names for each entry
fold_change_geneList <- setNames(object = data.frame(res)$logFC,
                                    nm = data.frame(res)$ORF)

# we can see what this look like with head()
head(fold_change_geneList)

##   YIL170W  YFL056C  YAR061W  YGR014W  YPR031W  YIL003W
##   1.00593  0.82290 -0.63346 -0.00538 -0.10371  0.25097

# the pathview command saves the file to your current directory.
# let's create a place for that information to go.
path_to_kegg_images <- "~/Desktop/Genomic_Data_Analysis/Analysis/KEGG/"
if (!dir.exists(path_to_kegg_images)) {
  dir.create(path_to_kegg_images, recursive = TRUE)
}

# move to this place so images save there.
setwd(path_to_kegg_images)

# now, we can run the pathview command.
# you can try changing the pathway.id below to one of the shown examples
# this will let you see the different pathways.
test <- pathview(gene.data = fold_change_geneList,
                  # pathway.id = "sce01100", #metabolic pathways
                  pathway.id = "sce00010", #glycolysis/gluconeogenesis
                  # pathway.id = "sce03050", # proteasome cycle
                  # pathway.id = "sce00020", # TCA cycle
                  # pathway.id = "sce00500", # starch & sucrose metabolism (trehalose)
                  # pathway.id = "sce00030", #PPP
                  species      = "sce",
                  gene.idtype="orf",
                  # expand.node = TRUE,
                  split.group=T,
                  map.symbol = T,
                  is.signal=F,
                  kegg.native = F,
```

```

match.data = T,
node.sum='max.abs', # this determines how to choose which gene to
multi.state = T,
bins=20,
same.layer = F,
pdf.size=c(7,7),
limit = list(gene=5, cpd=1))

## 'select()' returned 1:1 mapping between keys and columns

## [1] "Note: 148 of 5615 unique input IDs unmapped."

## Info: Getting gene ID data from KEGG...

## Info: Done with data retrieval!

## Warning in .local(from, to, graph): edges replaced: '139|61', '139|62',
## '118|121'

## Info: Working in directory /Users/clstacy/Desktop/Genomic_Data_Analysis/Analysis/KEGG

## Info: Writing image file sce00010.pathview.pdf

```

Of course, you might want more control over the process. In that case, we can use another package. For example, we could use `clusterProfiler` that we've used before. If you want to learn more, this is a useful guide: <https://yulab-smu.top/biomedical-knowledge-mining-book/clusterprofiler-kegg.html>. In this analysis, we need to manually select the genes that we want to run KEGG enrichment on, and save that character vector of gene names.

```

# turn res object into a list of genes
# Recall that these genes are those that are DE between mutant stress response and
# the WT stress response.
DE_genes <- res %>%
  data.frame() %>%
  filter(PValue < 0.01 & abs(logFC)>0.5
        ) %>%
  pull(ORF)

```

To run the enrichment, we can use the `enrichKEGG()` function from the `clusterProfiler` package. The argument “gene” for this function requires a vector of Entrez gene id's. Right now we have a vector of gene IDs, but they are the ORF names instead of entrez IDs, so for most organisms we need to convert them.

```
# how can we find our organism & its code? try this clusterProfiler command:
search_kegg_organism('yeast', by='common_name')
```

```
##      kegg_code      scientific_name    common_name
## 708      sce  Saccharomyces cerevisiae budding yeast
## 820      spo Schizosaccharomyces pombe fission yeast
```

Thankfully, the `clusterProfiler` package comes with the function `bitr` (Biological Id TRanslator) to translate geneIDs.

Note: the `sce` genome database is coded differently than many genome databases, so it requests the `ORF` instead of the `entrezID`, so we can directly use the `DE_genes` vector instead of the `entrez ID`. If you don't work with yeast, you'll probably need to use the `entrez ID` list for analyses that you do on your own.

```
# convert gene IDs to entrez IDs
entrez_ids <- bitr(DE_genes, fromType = "ORF", toType = "ENTREZID", OrgDb = org.Sc.sgd.db)

## 'select()' returned 1:1 mapping between keys and columns

# Run our KEGG enrichment
kegg_results <- enrichKEGG(gene = DE_genes, #entrez_ids$ENTREZID,
                           organism = 'sce' #options: https://www.genome.jp/kegg/catalog/org_list
                           )

## Reading KEGG annotation online: "https://rest.kegg.jp/link/sce/pathway"...
## Reading KEGG annotation online: "https://rest.kegg.jp/list/pathway/sce"...
## take a peak at the results
head(kegg_results)

##          category                      subcategory           ID
## sce01110     Metabolism        Global and overview maps sce01110
## sce00500     Metabolism        Carbohydrate metabolism sce00500
## sce04213 Organismal Systems                               Aging sce04213
## sce00620     Metabolism        Carbohydrate metabolism sce00620
## sce01200     Metabolism        Global and overview maps sce01200
## sce00770     Metabolism  Metabolism of cofactors and vitamins sce00770
##                                         Descri
## sce01110 Biosynthesis of secondary metabolites - Saccharomyces cerevisiae (budding y
## sce00500   Starch and sucrose metabolism - Saccharomyces cerevisiae (budding y
```

```
## sce04213 Longevity regulating pathway - multiple species - Saccharomyces cerevisiae
## sce00620                                     Pyruvate metabolism - Saccharomyces cerevisiae
## sce01200                                     Carbon metabolism - Saccharomyces cerevisiae
## sce00770          Pantothenate and CoA biosynthesis - Saccharomyces cerevisiae
##           GeneRatio   BgRatio   pvalue p.adjust    qvalue
## sce01110    108/385 351/2467 4.48e-15 4.26e-13 3.11e-13
## sce00500    20/385  41/2467 5.18e-07 1.96e-05 1.43e-05
## sce04213    19/385  38/2467 6.19e-07 1.96e-05 1.43e-05
## sce00620    22/385  52/2467 2.91e-06 6.90e-05 5.05e-05
## sce01200    36/385 112/2467 5.89e-06 1.09e-04 7.99e-05
## sce00770    13/385  23/2467 6.90e-06 1.09e-04 7.99e-05
##
## sce01110 YPL028W/YER091C/YKR067W/YMR169C/YDR178W/YER026C/YKL085W/YGR088W/YNL274C/YII
## sce00500
## sce04213
## sce00620
## sce01200
## sce00770
##           Count
## sce01110    108
## sce00500     20
## sce04213     19
## sce00620     22
## sce01200     36
## sce00770     13
```

```
# create a table for the html file
data.frame(kegg_results) %>% reactable()
```

	category	subcategory	ID	Description	GeneRatio	BgR:
sce0110	Metabolism	Global and overview maps	sce0110	Biosynthesis of secondary metabolites - <i>Saccharomyces cerevisiae</i> (budding yeast)	108/385	351%

```
# Remove " - Saccharomyces cerevisiae" from each description entry
kegg_results$result$Description <- kegg_results$result$Description %>% print() %>% str_replace_all

## [1] "Biosynthesis of secondary metabolites - Saccharomyces cerevisiae (budding yeast)"
## [2] "Starch and sucrose metabolism - Saccharomyces cerevisiae (budding yeast)"
## [3] "Longevity regulating pathway - multiple species - Saccharomyces cerevisiae (budding yeast)"
## [4] "Pyruvate metabolism - Saccharomyces cerevisiae (budding yeast)"
```

```

## [5] "Carbon metabolism - Saccharomyces cerevisiae (budding yeast)"
## [6] "Pantothenate and CoA biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [7] "2-Oxocarboxylic acid metabolism - Saccharomyces cerevisiae (budding yeast)"
## [8] "Ribosome biogenesis in eukaryotes - Saccharomyces cerevisiae (budding yeast)"
## [9] "Valine, leucine and isoleucine degradation - Saccharomyces cerevisiae (budding yeast)"
## [10] "Glyoxylate and dicarboxylate metabolism - Saccharomyces cerevisiae (budding yeast)"
## [11] "beta-Alanine metabolism - Saccharomyces cerevisiae (budding yeast)"
## [12] "Valine, leucine and isoleucine biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [13] "Biosynthesis of amino acids - Saccharomyces cerevisiae (budding yeast)"
## [14] "RNA polymerase - Saccharomyces cerevisiae (budding yeast)"
## [15] "Ribosome - Saccharomyces cerevisiae (budding yeast)"
## [16] "Tryptophan metabolism - Saccharomyces cerevisiae (budding yeast)"
## [17] "Glycerolipid metabolism - Saccharomyces cerevisiae (budding yeast)"
## [18] "Biosynthesis of cofactors - Saccharomyces cerevisiae (budding yeast)"
## [19] "Glycine, serine and threonine metabolism - Saccharomyces cerevisiae (budding yeast)"
## [20] "Citrate cycle (TCA cycle) - Saccharomyces cerevisiae (budding yeast)"
## [21] "Lysine degradation - Saccharomyces cerevisiae (budding yeast)"
## [22] "Fatty acid degradation - Saccharomyces cerevisiae (budding yeast)"
## [23] "Oxidative phosphorylation - Saccharomyces cerevisiae (budding yeast)"
## [24] "Glycolysis / Gluconeogenesis - Saccharomyces cerevisiae (budding yeast)"
## [25] "Cysteine and methionine metabolism - Saccharomyces cerevisiae (budding yeast)"
## [26] "Ubiquinone and other terpenoid-quinone biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [27] "Ascorbate and aldarate metabolism - Saccharomyces cerevisiae (budding yeast)"
## [28] "Purine metabolism - Saccharomyces cerevisiae (budding yeast)"
## [29] "Lysine biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [30] "Alanine, aspartate and glutamate metabolism - Saccharomyces cerevisiae (budding yeast)"
## [31] "Fructose and mannose metabolism - Saccharomyces cerevisiae (budding yeast)"
## [32] "Pentose phosphate pathway - Saccharomyces cerevisiae (budding yeast)"
## [33] "Galactose metabolism - Saccharomyces cerevisiae (budding yeast)"
## [34] "Methane metabolism - Saccharomyces cerevisiae (budding yeast)"
## [35] "Lipoic acid metabolism - Saccharomyces cerevisiae (budding yeast)"
## [36] "Tyrosine metabolism - Saccharomyces cerevisiae (budding yeast)"
## [37] "Propanoate metabolism - Saccharomyces cerevisiae (budding yeast)"
## [38] "Arginine and proline metabolism - Saccharomyces cerevisiae (budding yeast)"
## [39] "Biosynthesis of nucleotide sugars - Saccharomyces cerevisiae (budding yeast)"
## [40] "Glycerophospholipid metabolism - Saccharomyces cerevisiae (budding yeast)"
## [41] "Peroxisome - Saccharomyces cerevisiae (budding yeast)"
## [42] "Histidine metabolism - Saccharomyces cerevisiae (budding yeast)"
## [43] "Nicotinate and nicotinamide metabolism - Saccharomyces cerevisiae (budding yeast)"
## [44] "Sulfur metabolism - Saccharomyces cerevisiae (budding yeast)"
## [45] "Nucleotide metabolism - Saccharomyces cerevisiae (budding yeast)"
## [46] "Arginine biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [47] "Amino sugar and nucleotide sugar metabolism - Saccharomyces cerevisiae (budding yeast)"
## [48] "Glutathione metabolism - Saccharomyces cerevisiae (budding yeast)"
## [49] "Fatty acid metabolism - Saccharomyces cerevisiae (budding yeast)"
## [50] "Pentose and glucuronate interconversions - Saccharomyces cerevisiae (budding yeast)"

```

```
## [51] "Fatty acid biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [52] "Vitamin B6 metabolism - Saccharomyces cerevisiae (budding yeast)"
## [53] "Polycomb repressive complex - Saccharomyces cerevisiae (budding yeast)"
## [54] "Pyrimidine metabolism - Saccharomyces cerevisiae (budding yeast)"
## [55] "Meiosis - yeast - Saccharomyces cerevisiae (budding yeast)"
## [56] "MAPK signaling pathway - yeast - Saccharomyces cerevisiae (budding yeast)"
## [57] "Steroid biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [58] "ABC transporters - Saccharomyces cerevisiae (budding yeast)"
## [59] "Efferocytosis - Saccharomyces cerevisiae (budding yeast)"
## [60] "Selenocompound metabolism - Saccharomyces cerevisiae (budding yeast)"
## [61] "Terpenoid backbone biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [62] "Other types of O-glycan biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [63] "Sphingolipid metabolism - Saccharomyces cerevisiae (budding yeast)"
## [64] "Protein processing in endoplasmic reticulum - Saccharomyces cerevisiae (budding yeast)"
## [65] "One carbon pool by folate - Saccharomyces cerevisiae (budding yeast)"
## [66] "Porphyrin metabolism - Saccharomyces cerevisiae (budding yeast)"
## [67] "RNA degradation - Saccharomyces cerevisiae (budding yeast)"
## [68] "Endocytosis - Saccharomyces cerevisiae (budding yeast)"
## [69] "Biosynthesis of unsaturated fatty acids - Saccharomyces cerevisiae (budding yeast)"
## [70] "Folate biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [71] "Various types of N-glycan biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [72] "DNA replication - Saccharomyces cerevisiae (budding yeast)"
## [73] "Mitophagy - yeast - Saccharomyces cerevisiae (budding yeast)"
## [74] "Autophagy - other - Saccharomyces cerevisiae (budding yeast)"
## [75] "Motor proteins - Saccharomyces cerevisiae (budding yeast)"
## [76] "Glycosylphosphatidylinositol (GPI)-anchor biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [77] "Viral life cycle - HIV-1 - Saccharomyces cerevisiae (budding yeast)"
## [78] "Cell cycle - yeast - Saccharomyces cerevisiae (budding yeast)"
## [79] "Riboflavin metabolism - Saccharomyces cerevisiae (budding yeast)"
## [80] "Phenylalanine, tyrosine and tryptophan biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [81] "Thiamine metabolism - Saccharomyces cerevisiae (budding yeast)"
## [82] "Nucleocytoplasmic transport - Saccharomyces cerevisiae (budding yeast)"
## [83] "Autophagy - yeast - Saccharomyces cerevisiae (budding yeast)"
## [84] "N-Glycan biosynthesis - Saccharomyces cerevisiae (budding yeast)"
## [85] "Spliceosome - Saccharomyces cerevisiae (budding yeast)"
## [86] "Inositol phosphate metabolism - Saccharomyces cerevisiae (budding yeast)"
## [87] "Homologous recombination - Saccharomyces cerevisiae (budding yeast)"
## [88] "Phosphatidylinositol signaling system - Saccharomyces cerevisiae (budding yeast)"
## [89] "Base excision repair - Saccharomyces cerevisiae (budding yeast)"
## [90] "Basal transcription factors - Saccharomyces cerevisiae (budding yeast)"
## [91] "mRNA surveillance pathway - Saccharomyces cerevisiae (budding yeast)"
## [92] "ATP-dependent chromatin remodeling - Saccharomyces cerevisiae (budding yeast)"
## [93] "Nucleotide excision repair - Saccharomyces cerevisiae (budding yeast)"
## [94] "Ubiquitin mediated proteolysis - Saccharomyces cerevisiae (budding yeast)"
## [95] "Aminoacyl-tRNA biosynthesis - Saccharomyces cerevisiae (budding yeast)"
```

## 11.5 Visualize on the KEGG website

The first way we might want to visualize this plot is part of a KEGG pathway. We can open an html window with genes that are enriched highlighted, like this. Run one of these lines below and it will open a new window in your browser.

```
browseKEGG(kegg_results, 'sce00500') # starch & sucrose metabolism
browseKEGG(kegg_results, 'sce04213') # longevity
```

## 11.6 Comparing Paralogs in common pathways

Using the `res_all` object instead, we can compare logFC across contrasts. We can create a side by side of the WT EtOH response and the Msn2/4 EtOH response.

```
# choose the pathway we want to visualize:

pathway_to_graph <- "sce00010" # glycolysis/gluconeogenesis
# pathway_to_graph <- "sce00020" # TCA cycle
# pathway_to_graph <- "sce04213" # longevity
# pathway_to_graph <- "sce00500" # starch & sucrose metabolism
# pathway_to_graph <- "sce00620" # pyruvate metabolism

# get the ID for the pathway we want to see
pathway_number <- kegg_results %>%
  data.frame() %>%
  mutate(row_number = row_number()) %>%
  filter(ID == pathway_to_graph) %>%
  pull(row_number)

# this saves the kegg list reaction mappings to KEGG_reactions for plotting
# where these came from: "https://rest.kegg.jp/list/reaction/"
if (!exists("KEGG_reactions")) {
  # save the kegglist reaction information
  KEGG_reactions <- KEGGREST::keggList("reaction") %>%
    as.data.frame()
  colnames(KEGG_reactions) <- "long_reaction_description"
  KEGG_reactions <- KEGG_reactions %>%
    tibble::rownames_to_column("reaction") %>%
    dplyr::mutate(reaction_description = str_split_i(
      long_reaction_description, ";", 1)
    ) %>%
```

```

dplyr::select(reaction, reaction_description) %>%
  dplyr::mutate(reaction = paste0('rn:', reaction))
}

# save the kegg_data as a ggkegg object
KEGG_data <- kegg_results %>%
  ggkegg(
    convert_first = FALSE,
    convert_collapse = "\n",
    pathway_number = pathway_number, # changing this to change the pathway.
    convert_org = c("sce"),
    delete_zero_degree = TRUE,
    return_igraph = FALSE
  )

# process this data to visualize
graph.data <- KEGG_data$data %>%
  filter(type == "gene") %>%
  mutate(showname = strsplit(name, " ") %>% str_remove_all("sce:")) %>%
  mutate(showname = gsub('c\\(\\|\\)|"|"|,|\'|', ' ', showname)) %>%
  separate_rows(showname, sep = " ") %>%
  left_join(rownames_to_column(res_all$table),
            by = c("showname" = "rowname")) %>%
  left_join(KEGG_reactions, by = "reaction") %>%
  mutate(gene_name = AnnotationDbi::select(org.Sc.sgd.db,
                                            keys = showname,
                                            columns = "GENENAME")$GENENAME) %>%
  mutate(gene_name = coalesce(gene_name, showname))

## 'select()' returned many:1 mapping between keys and columns

# find our fc values needed for color scale
max_fc <-
  ceiling(max(c(
    abs(graph.data$logFC.EtOHvsMOCK.MSN24dd),
    abs(graph.data$logFC.EtOHvsMOCK.WT)
  ), na.rm = TRUE))

# create graph for WT stress response
WT.graph <- graph.data %>%
  ggplot(aes(x = x, y = y)) +
  overlay_raw_map(pathway_to_graph) +
  ggrepel::geom_label_repel(
    aes(label = gene_name, fill = logFC.EtOHvsMOCK.WT),
    box.padding = 0.05,

```

```

label.padding = 0.05,
direction = "y",
size = 2,
max.overlaps = 100,
label.r = 0.002,
seed = 123
) +
theme_void() +
scale_fill_gradientn(
colours = rev(RColorBrewer::brewer.pal(n = 10, name = "RdYlBu")),
limits = c(-max_fc, max_fc)
) +
theme(plot.margin = unit(c(0, 0, 0, 0), "cm")) +
labs(fill = "logFC", title = "      WT Response to Ethanol")

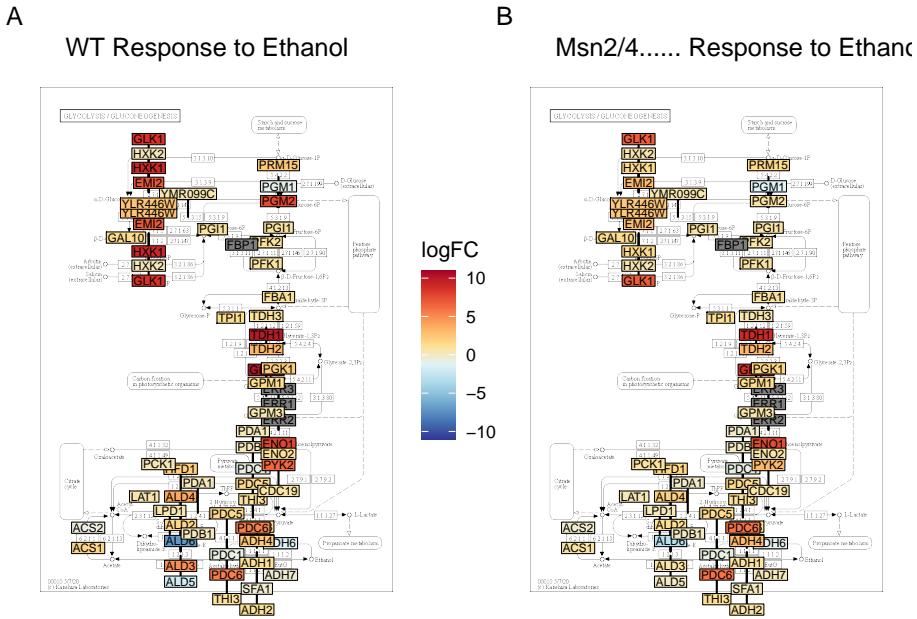
# create graph for mutant stress response
msn24.graph <- graph.data %>%
ggplot(aes(x = x, y = y)) +
overlay_raw_map(pathway_to_graph) +
ggrepel::geom_label_repel(
aes(label = gene_name, fill = logFC.EtOHvsMOCK.MSN24dd),
box.padding = 0.05,
label.padding = 0.05,
direction = "y",
size = 2,
max.overlaps = 100,
label.r = 0.002,
seed = 123
) +
theme_void() +
guides(fill = FALSE) +
scale_fill_gradientn(
colours = rev(RColorBrewer::brewer.pal(n = 10, name = "RdYlBu")),
limits = c(-max_fc, max_fc)
) +
theme(plot.margin = unit(c(0, 0, 0, 0), "cm")) +
labs(fill = "logFC", title = "      Msn2/4  Response to Ethanol")

# generate the graph
side_by_side_graph <- WT.graph + msn24.graph +
patchwork::plot_annotation(tag_levels = 'A')

# display the graph

```

## side\_by\_side\_graph



Once we are happy with our graph. We might want to save it. We can save it like this

```
# command to save
ggsave(paste0(path_to_kegg_images, pathway_to_graph, ".sidebyside.pdf"),
       plot = side_by_side_graph,
       device = "pdf",
       width=10,
       height=6,
       dpi=300
)
```

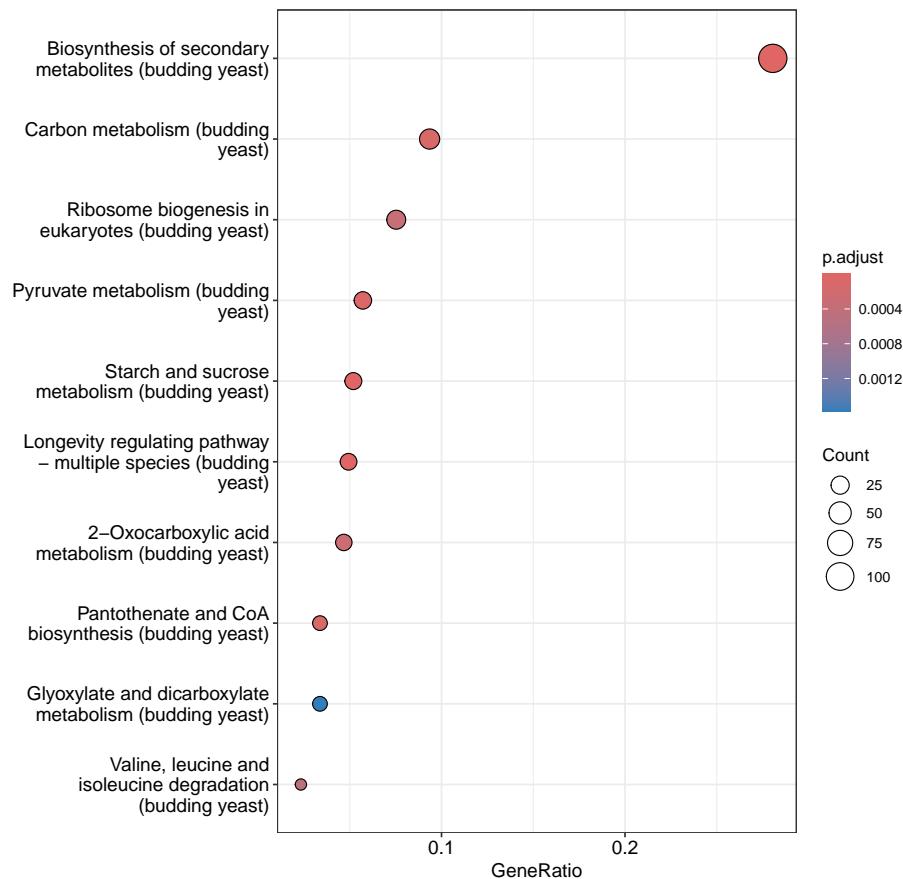
## 11.7 Additional KEGG-related analyses

clusterProfiler let's us visualize the output of `enrichKEGG()`, that we named as `kegg_results` in this exercise, using the `res` object.

## 11.8 Dotplot

We can create a dotplot from this object as shown below. The dotplot shows KEGG categories that are enriched, the adjusted p.values, the number of genes in the KEGG category, and the proportion of genes in the KEGG pathway that are included in the DE gene list.

```
# Plot the KEGG pathway enrichment results
dotplot(kegg_results, showCategory = 10)
```



## 11.9 cnetplot

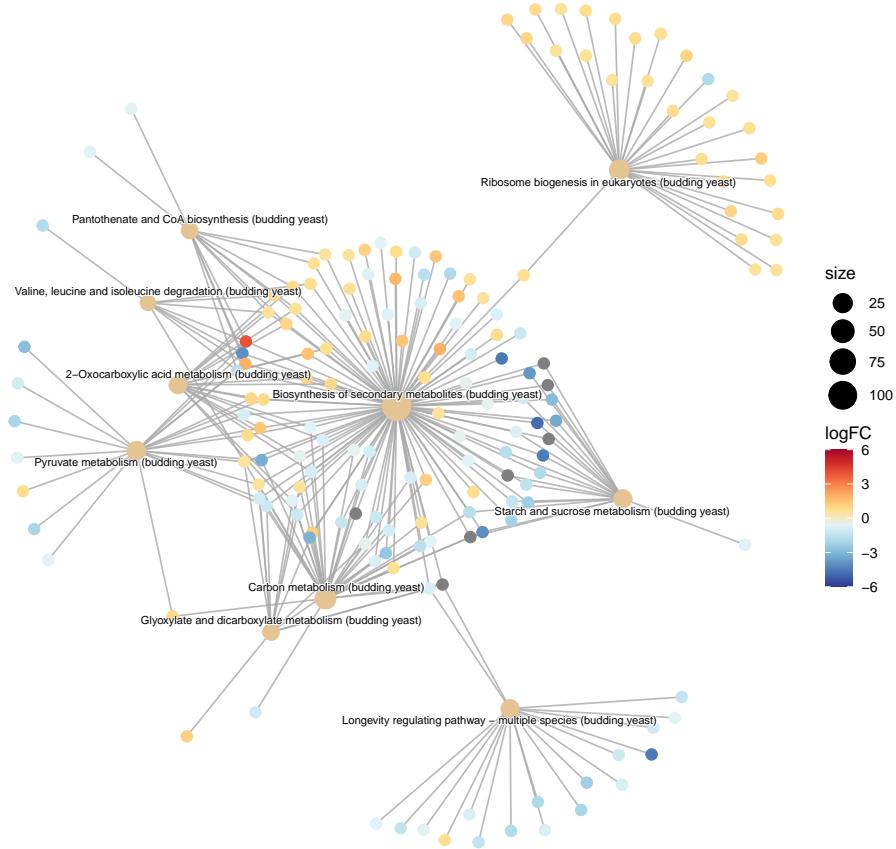
The `dotplot()` only shows the most significant or selected enriched terms, while we may want to know which genes are involved in these significant terms. In

order to consider the potentially biological complexities in which a gene may belong to multiple annotation categories and provide log2FC information, we can use the `cnetplot()` function to extract the complex association. The `cnetplot()` depicts the linkages of genes and KEGG pathways as a network. We can project the network into 2D space, or we can create a circular version of the graph. See each below.

```
# cnetplot
cnetplot(kegg_results,
  showCategory=10,
  node_label="category",
  color.params=list(foldChange = fold_change_geneList),
  cex.params=list(gene_label = 0.1,
                  category_label=0.5),
  max.overlaps=100) +
# change the color scale
scale_colour_gradientn(colours = rev(RColorBrewer::brewer.pal(n = 10, name = "RdYlBu")), limits
  labs(colour="logFC")

## Scale for size is already present.
## Adding another scale for size, which will replace the existing scale.
## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.

## Warning: Removed 167 rows containing missing values (`geom_text_repel()`).
```



```
# circular cnetplot
cnetplot(kegg_results,
  showCategory=6,
  circular=TRUE, # this changes the output graphing
  node_label="category",
  color.params=list(foldChange = fold_change_geneList),
  cex.params=list(gene_label = 0.5,
    category_label=0.5),
  max.overlaps=100) +
# change the color scale
scale_colour_gradientn(colours = rev(RColorBrewer::brewer.pal(n = 10, name = "RdYlBu"))
  labs(colour="logFC")

## Scale for size is already present.
## Adding another scale for size, which will replace the existing scale.
```

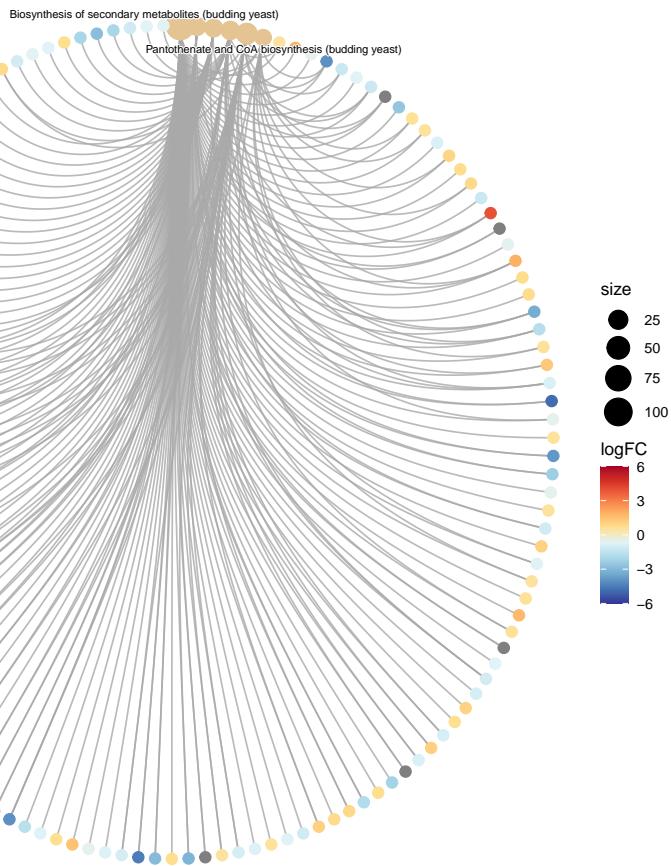
```

## Scale for colour is already present.
## Adding another scale for colour, which will replace the existing scale.

## Warning: Removed 137 rows containing missing values (`geom_text_repel()`).

## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

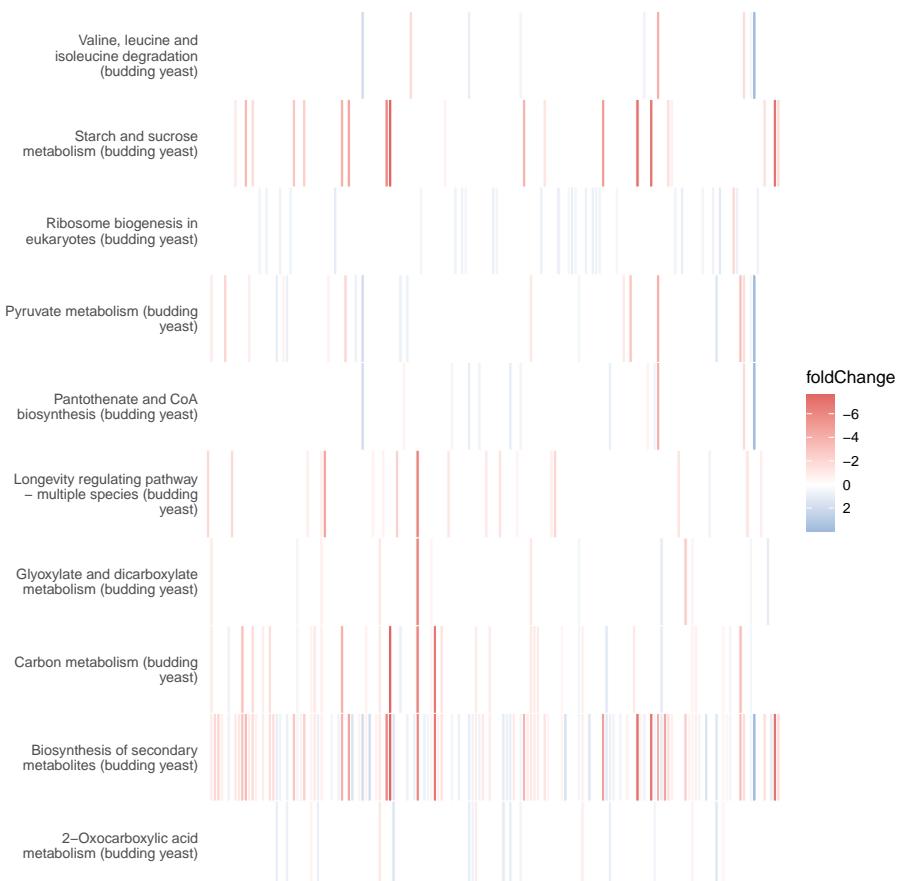
```



## 11.10 heatplot

The `heatplot` is similar to `cnetplot`, while displaying the relationships as a heatmap. The gene-concept network may become too complicated if you want to show a large number significant terms. The heatplot can simplify the result and more easy to identify expression patterns.

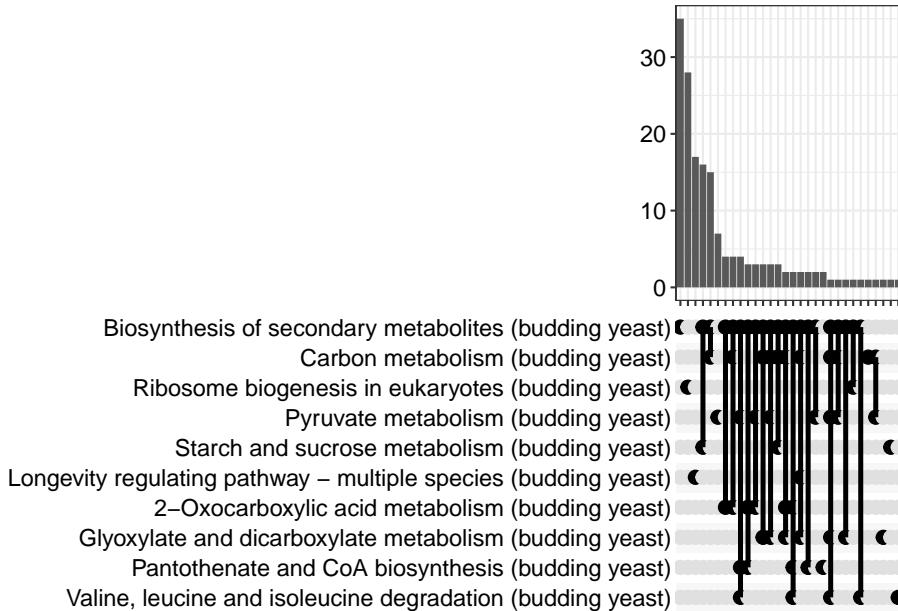
```
# heatplot
heatplot(kegg_results, showCategory=10, foldChange = fold_change_geneList) +
  # the below code hides the messy overlapping gene names.
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```



## 11.11 upsetplot

The `upsetplot` shows the overlaps and unique contributions of KEGG pathways across multiple gene sets. It helps to identify which KEGG pathways are shared or distinct between different conditions or experimental groups, providing insights into the common and specific KEGG pathways enriched in each set of genes. Here, the bars show number of genes in the category.

```
enrichplot::upsetplot(kegg_results)
```

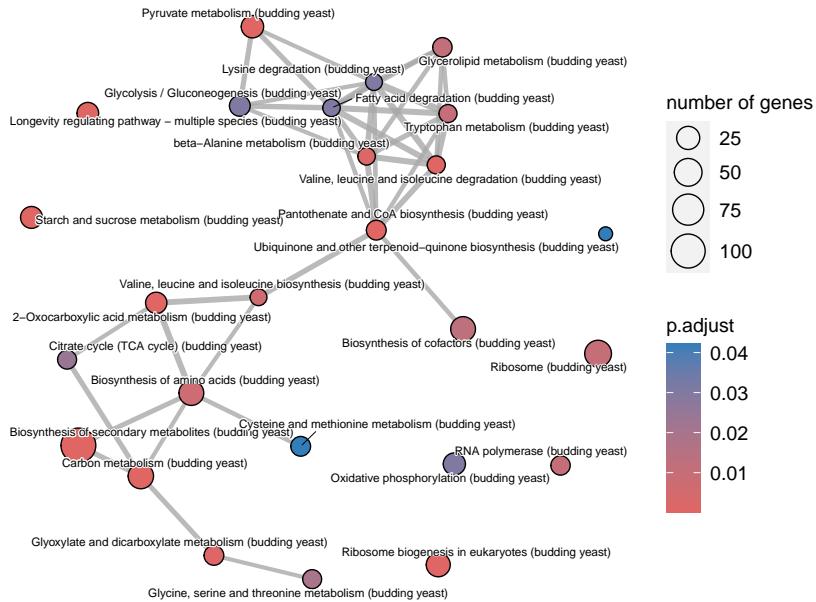


## 11.12 emapplot

The `emapplot` enrichment map organizes enriched terms into a network with edges connecting overlapping gene sets. In this way, mutually overlapping gene sets are tend to cluster together, making it easy to identify functional modules.

The `emapplot` function supports results obtained from hypergeometric test and gene set enrichment analysis. We have to first use the `pairwise_termsim` function from the `enrichplot` package on the `kegg_results` object in order to get a similarity matrix.

```
x2 = enrichplot::pairwise_termsim(kegg_results)
emapplot(x2,
  showCategory = 30,
  cex.params = list(category_label = 0.4))
```



### 11.13 GSEA

KEGG pathway gene set enrichment analysis allows us to identify whether a particular set of genes associated with a KEGG pathway is enriched in a given gene list compared to what would be expected by chance.

```
# For gseKEGG, we need a gene list with na removed and sorted. let's do that now.
# omit any NA values
kegg_gene_list = na.omit(fold_change_geneList)

# sort the list in decreasing order by FOLD CHANGE (required for this analysis)
kegg_gene_list = sort(kegg_gene_list, decreasing = TRUE)

gse_kegg <- gseKEGG(geneList      = kegg_gene_list,
                      organism     = 'sce',
                      # minGSSize   = 120,
                      pvalueCutoff = 0.05,
                      verbose      = FALSE)

## Warning in fgseaMultilevel(pathways = pathways, stats = stats, minSize =
## minSize, : For some pathways, in reality P-values are less than 1e-10. You can
## set the `eps` argument to zero for better estimation.
```

```
head(gse_kegg)
```

```
##          ID
## sce03010 sce03010
## sce03008 sce03008
## sce00500 sce00500
## sce03020 sce03020
## sce04213 sce04213
## sce00520 sce00520
##
##                                     Descri
## sce03010             Ribosome - Saccharomyces cerevisiae (budding y
## sce03008     Ribosome biogenesis in eukaryotes - Saccharomyces cerevisiae (budding y
## sce00500           Starch and sucrose metabolism - Saccharomyces cerevisiae (budding y
## sce03020           RNA polymerase - Saccharomyces cerevisiae (budding y
## sce04213 Longevity regulating pathway - multiple species - Saccharomyces cerevisiae (budding y
## sce00520   Amino sugar and nucleotide sugar metabolism - Saccharomyces cerevisiae (budding y
##          setSize enrichmentScore    NES    pvalue p.adjust    qvalue rank
## sce03010      174        0.620  2.64 1.00e-10 1.01e-08 8.00e-09 1481
## sce03008       74        0.660  2.53 5.27e-10 2.47e-08 1.95e-08 1094
## sce00500       39       -0.857 -2.21 7.33e-10 2.47e-08 1.95e-08 319
## sce03020       31        0.723  2.35 1.96e-06 4.94e-05 3.92e-05 1091
## sce04213       38       -0.770 -1.98 1.29e-05 2.61e-04 2.07e-04 615
## sce00520       31       -0.796 -2.00 1.65e-05 2.77e-04 2.20e-04 166
##          leading_edge
## sce03010 tags=68%, list=26%, signal=52%
## sce03008 tags=68%, list=19%, signal=55%
## sce00500 tags=49%, list=6%, signal=46%
## sce03020 tags=68%, list=19%, signal=55%
## sce04213 tags=50%, list=11%, signal=45%
## sce00520 tags=29%, list=3%, signal=28%
##
## sce03010 YLR406C/YIL069C/YKL156W/YNL002C/YDR471W/YMR194W/YOR234C/YGR085C/YBR189W/YPL198W/YGR21
## sce03008
## sce00500
## sce03020
## sce04213
## sce00520
```

```
# let's turn this into a searchable table for the knit file.
data.frame(gse_kegg) %>% reactable()
```

ID	Description	setSize	enrichment Score	NES
sce03010	Ribosome - Saccharomyces cerevisiae (budding yeast)	174 488332	0.619768515	2.637491947 329

```
# shorten the gse_kegg descriptions that are too long
gse_kegg$result$Description <- gse_kegg$result$Description %>%
  str_replace_all(., fixed(" - Saccharomyces cerevisiae"), "")
```

### 11.13.1 `gseKEGG` vs `enrichKEGG`

What is the difference, and when to use each?

Both `gseKEGG` and `enrichKEGG` functions are used for gene set enrichment analysis (GSEA) focusing on KEGG pathways, but they have different underlying methodologies and purposes.

#### 1. `gseKEGG`:

Methodology: Gene Set Test (GSE): `gseKEGG` employs a gene set test approach. It calculates an enrichment score for each KEGG pathway based on the distribution of genes within the pathway in the ranked list of genes (usually by their differential expression values). Permutations: The method involves permutations to assess the statistical significance of the enrichment scores.

Output: The output is a `GSEAResult` object containing information about the enriched KEGG pathways, including their names, enrichment scores, and p-values.

Use Case: `gseKEGG` is suitable when you have a ranked list of genes based on some criteria (e.g., differential expression) and want to perform GSEA to identify KEGG pathways associated with these genes.

#### 2. `enrichKEGG`:

Methodology: Over-Representation Analysis (ORA): `enrichKEGG` uses an over-representation analysis approach. It tests whether a predefined set of genes associated with a KEGG pathway is overrepresented in a given gene list compared to what would be expected by chance. Hypergeometric Test: The statistical significance is often assessed using a hypergeometric test.

Output: The output is a data frame containing information about the enriched KEGG pathways, including names, p-values, and other statistics.

Use Case: `enrichKEGG` is appropriate when you have a gene list and want to identify KEGG pathways that are significantly enriched in that list. It doesn't require a ranked list of genes.

Which to Choose:

If you have a ranked list of genes (e.g., based on differential expression) and want to perform GSEA, use `gseKEGG`.

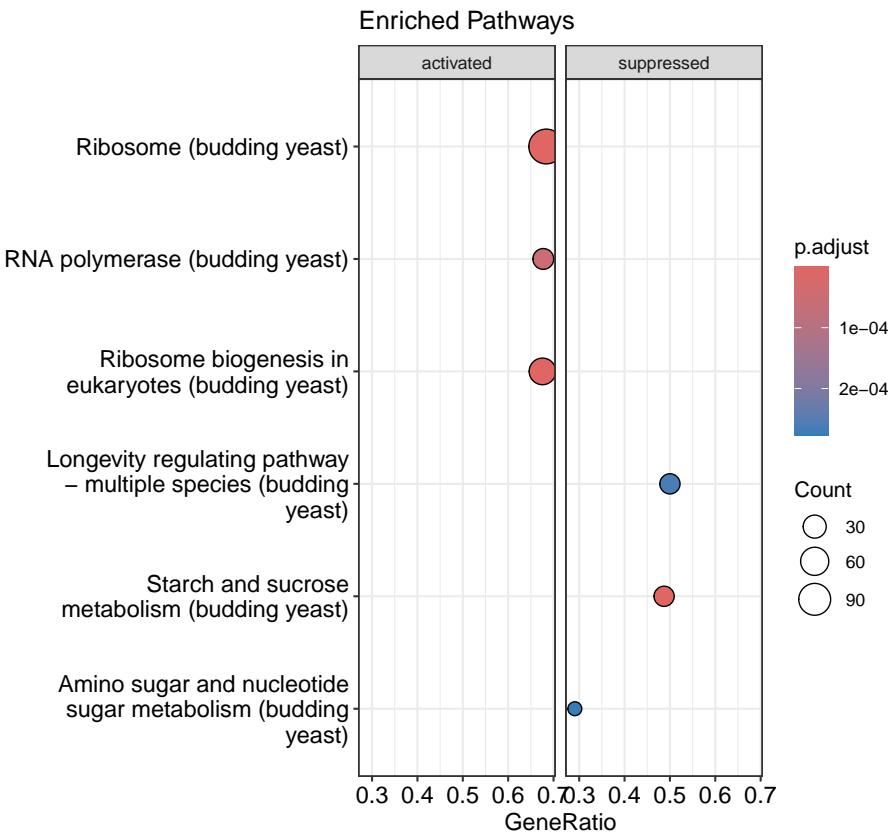
If you have a gene list and want to identify overrepresented pathways without ranking genes, use `enrichKEGG`.

In summary, the choice between `gseKEGG` and `enrichKEGG` depends on your data and the analysis you want to perform. Both methods can provide insights into the functional enrichment of gene sets (KEGG pathways, in this case) in the context of different experimental conditions.

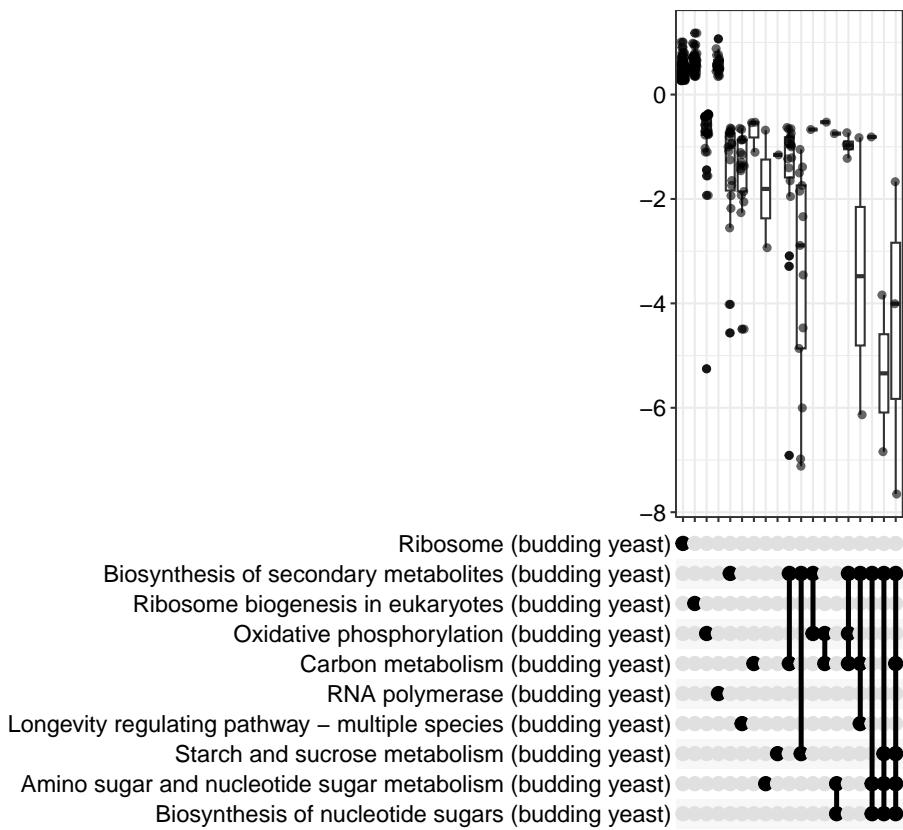
### 11.13.2 GSEA visualization

We can also visualize the outputs of `gseKEGG` using the same functions as above.

```
dotplot(gse_kegg, showCategory = 3, title = "Enriched Pathways" , split=".sign") + face
```

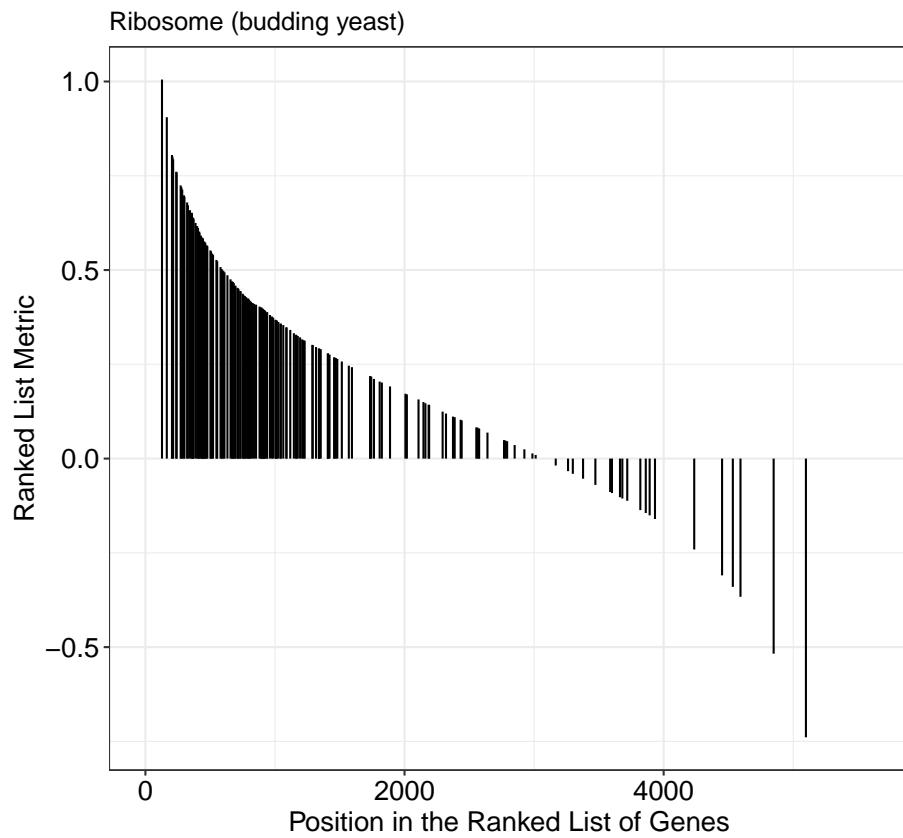


```
# enrichplots of the gseaResult object
enrichplot::upsetplot(gse_kegg)
```

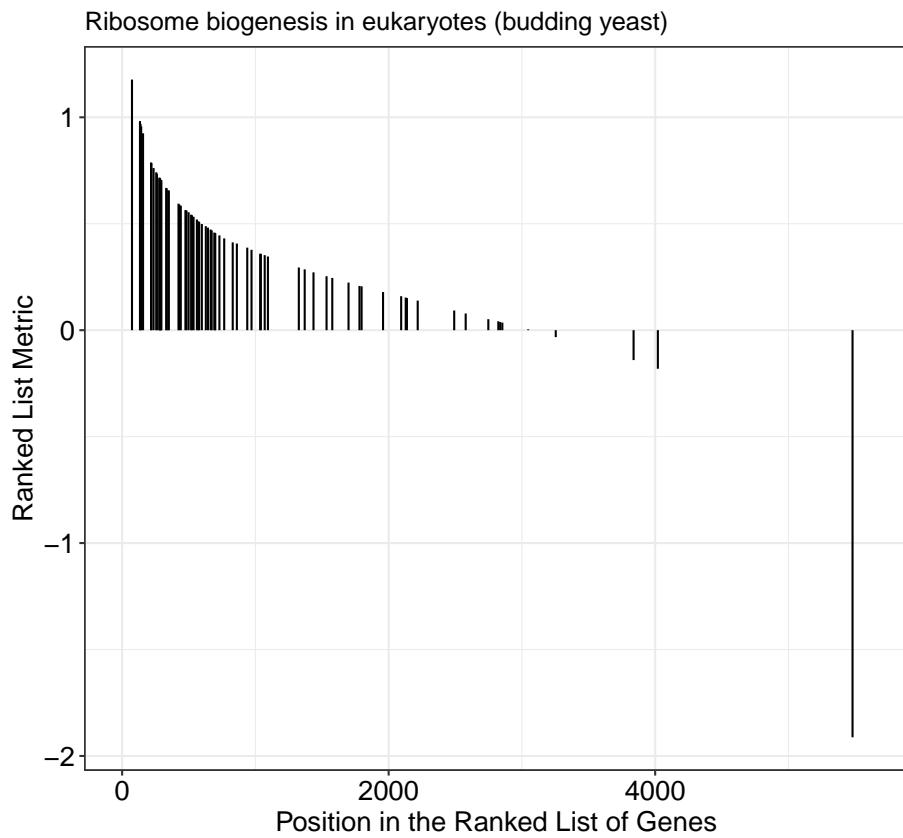


There are also visualizations specific to the `gseaResult` output of `gseKEGG`, shown below. We can see where in the rankings the genes belonging to a given KEGG group are found relative to the entire ranked list. We can change the `geneSetID` value to get other KEGG pathways.

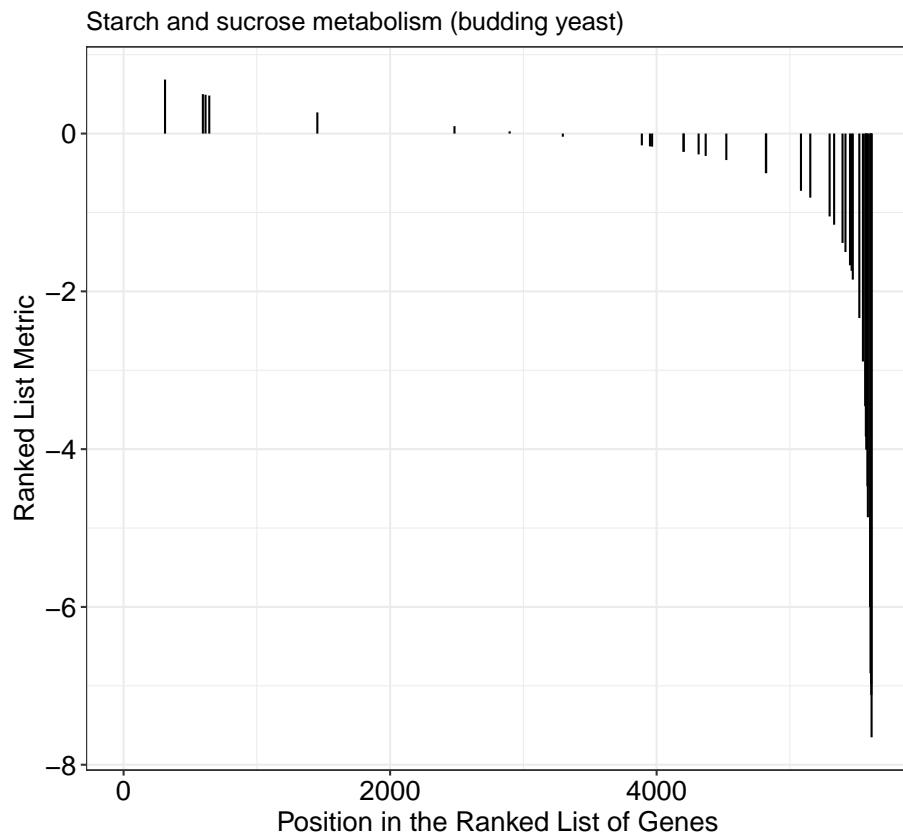
```
# Visualize the ranking of the genes by set
gseaplot(gse_kegg, geneSetID = 1, by = "preranked", title = gse_kegg$Description[1])
```



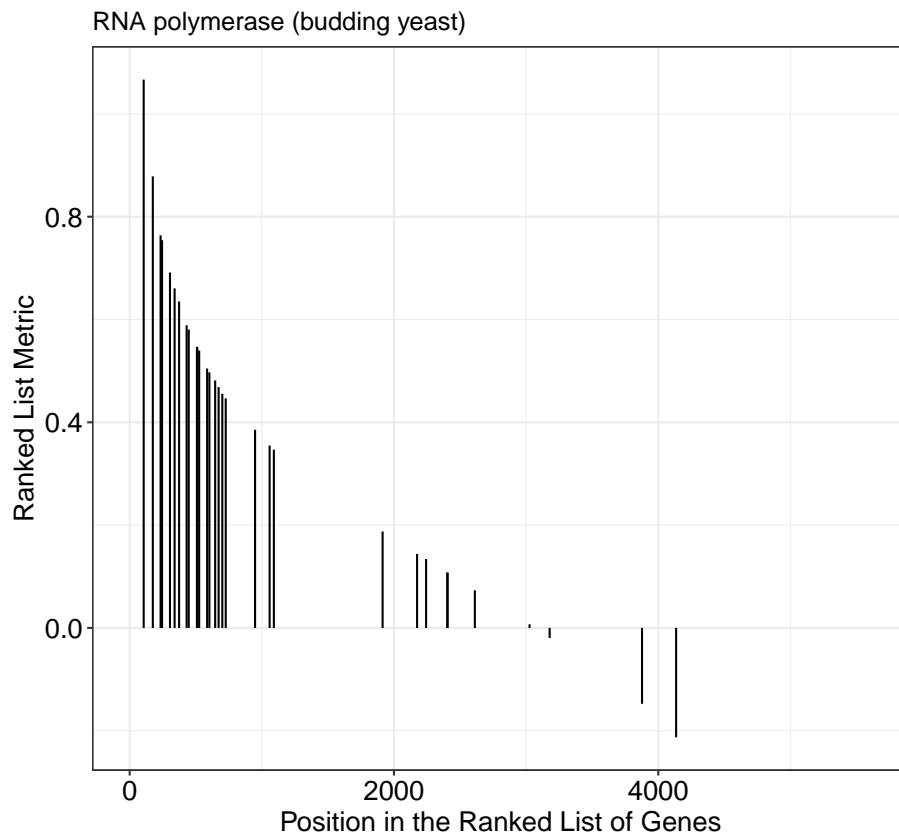
```
gseaplot(gse_kegg, geneSetID = 2, by = "preranked", title = gse_kegg$Description[2])
```



```
gseaplot(gse_kegg, geneSetID = 3, by = "preranked", title = gse_kegg$Description[3])
```

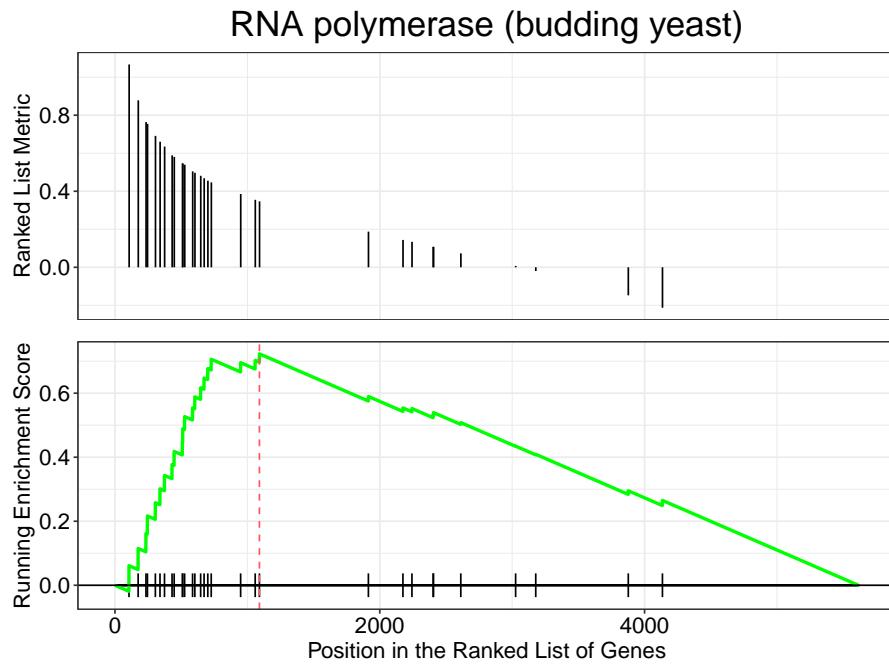


```
gseaplot(gse_kegg, geneSetID = 4, by = "preranked", title = gse_kegg$Description[4])
```



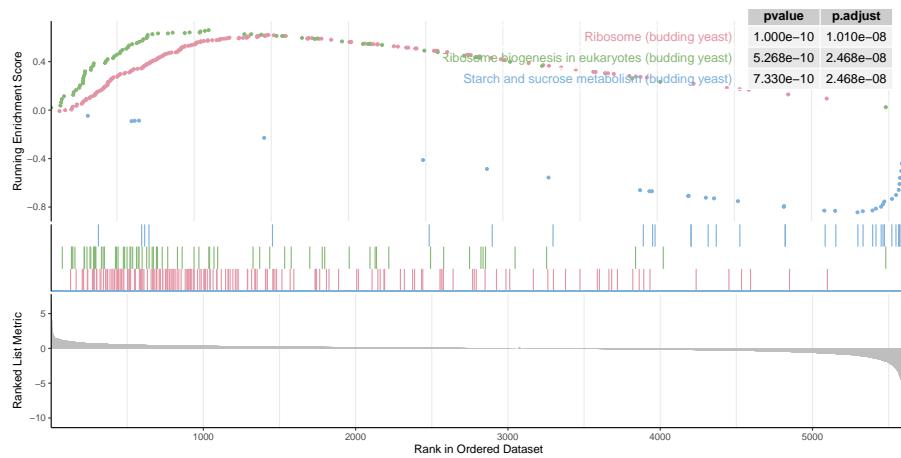
In addition to the plot above, we can create plots with more details by not included the `by="preranked"` argument. We can now see the calculated enrichment score across the list for this gene set.

```
# take a closer look at the RNA polymerase genes
gseaplot(gse_kegg, geneSetID = 4, title = gse_kegg$Description[4])
```



What if we want to look at multiple together KEGG pathways simultaneously? We can do that with `gseaplot2` from the `enrichplot` package. Here we look at the first 3 KEGG pathways together.

```
# gseaplot2 multiple KEGG pathway enrichments
enrichplot::gseaplot2(gse_kegg, geneSetID = 1:3, pvalue_table = TRUE,
                      color = c("#E495A5", "#86B875", "#7DB0DD"), ES_geom = "dot")
```



## 11.14 Questions

1. Can you explain the concept of gene set enrichment analysis and its relevance in functional genomics?
2. How would you interpret the results obtained from clusterProfiler, specifically focusing on enriched terms and associated p-values?
3. Compare and contrast KEGG enrichment with other tools or methods used for functional enrichment analysis.
4. Can you find the KEGG organism for the organism that you are interested in?

Be sure to knit this file into a pdf or html file once you're finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

**R version 4.3.1 (2023-06-16)**

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8|en\_US.UTF-8|C|en\_US.UTF-8|en\_US.UTF-8

**attached base packages:** stats4, stats, graphics, grDevices, utils, datasets, methods and base

**other attached packages:** ggfx(v.1.0.1), patchwork(v.1.1.3), ggkegg(v.0.99.4), tidygraph(v.1.2.3), XML(v.3.99-0.14), ggraph(v.2.1.0), KEGGgraph(v.1.62.0), ggupset(v.0.3.0), pathview(v.1.42.0), BiocFileCache(v.2.10.0), dbplyr(v.2.3.4), DOSE(v.3.28.0), NbClust(v.3.0.1), factoextra(v.1.0.7), ggrepel(v.0.9.4), viridis(v.0.6.4), viridisLite(v.0.4.2), scales(v.1.2.1), Glimma(v.2.12.0), DESeq2(v.1.41.12), edgeR(v.3.99.5), limma(v.3.58.0), reactable(v.0.4.4), webshot2(v.0.1.1), statmod(v.1.5.0), Rsubread(v.2.16.0), ShortRead(v.1.60.0), GenomicAlignments(v.1.38.0), SummarizedExperiment(v.1.32.0), MatrixGenerics(v.1.14.0), matrixStats(v.1.0.0), Rsamtools(v.2.18.0), GenomicRanges(v.1.54.0), Biostrings(v.2.70.1), GenomeInfoDb(v.1.38.0), XVector(v.0.42.0), BiocParallel(v.1.36.0), Rfastp(v.1.12.0), org.Sc.sgd.db(v.3.18.0), AnnotationDbi(v.1.64.0), IRanges(v.2.36.0), S4Vectors(v.0.40.0), Biobase(v.2.62.0), BiocGenerics(v.0.48.0), clusterProfiler(v.4.10.0), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pander(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3), forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyverse(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)

**loaded via a namespace (and not attached):** *fs(v.1.6.3), bitops(v.1.0-7), enrichplot(v.1.22.0), webshot(v.0.5.5), HDO.db(v.0.99.1), httr(v.1.4.7), RColorBrewer(v.1.1-3), Rgraphviz(v.2.46.0), tools(v.4.3.1), utf8(v.1.2.4), R6(v.2.5.1), lazyeval(v.0.2.2), GetoptLong(v.1.0.5), withr(v.2.5.1), gridExtra(v.2.3), textshaping(v.0.3.7), cli(v.3.6.1), Cairo(v.1.6-1), scatterpie(v.0.2.1), labeling(v.0.4.3), systemfonts(v.1.0.5), yulab.utils(v.0.1.0), gson(v.0.1.0), rstudioapi(v.0.15.0), RSQLite(v.2.3.1), generics(v.0.1.3), gridGraphics(v.0.5-1), hwriter(v.1.3.2.1), crosstalk(v.1.2.0), GO.db(v.3.18.0), Matrix(v.1.6-1.1), interp(v.1.1-4), fansi(v.1.0.5), abind(v.1.4-5), lifecycle(v.1.0.3), yaml(v.2.3.7), snakecase(v.0.11.1), qvalue(v.2.34.0), SparseArray(v.1.2.0), grid(v.4.3.1), blob(v.1.2.4), promises(v.1.2.1), crayon(v.1.5.2), lattice(v.0.22-5), cowplot(v.1.1.1), chromote(v.0.1.2), KEGGREST(v.1.42.0), magick(v.2.8.1), pillar(v.1.9.0), fgsea(v.1.28.0), rjson(v.0.2.21), codetools(v.0.2-19), fastmatch(v.1.1-4), glue(v.1.6.2), ggrepel(v.0.1.3), data.table(v.1.14.8), revolectives(v.2.4.2.1), vctrs(v.0.6.4), png(v.0.1-8), treeio(v.1.26.0), gtable(v.0.3.4), reactR(v.0.5.0), cachem(v.1.0.8), xfun(v.0.40), S4Arrays(v.1.2.0), mime(v.0.12), RVenn(v.1.1.0), interactiveDisplayBase(v.1.40.0), ellipsis(v.0.3.2), nlme(v.3.1-163), ggtree(v.3.10.0), bit64(v.4.0.5), filelock(v.1.0.2), rprojroot(v.2.0.3), colorspace(v.2.1-0), DBI(v.1.1.3), tidyselect(v.1.2.0), processx(v.3.8.2), bit(v.4.0.5), compiler(v.4.3.1), curl(v.5.1.0), graph(v.1.80.0), DelayedArray(v.0.28.0), bookdown(v.0.36), shadowtext(v.0.1.2), rappdirs(v.0.3.3), digest(v.0.6.33), rmarkdown(v.2.25), htmltools(v.0.5.6.1), pkgconfig(v.2.0.3), jpeg(v.0.1-10), fastmap(v.1.1.1), rlang(v.1.1.1), GlobalOptions(v.0.1.2), htmlwidgets(v.1.6.2), shiny(v.1.7.5.1), farver(v.2.1.1), jsonlite(v.1.8.7), GOSemSim(v.2.28.0), RCurl(v.1.98-1.12), magrittr(v.2.0.3), GenomeInfoDb-Data(v.1.2.11), ggplotify(v.0.1.2), munsell(v.0.5.0), Rcpp(v.1.0.11), ggnewscale(v.0.4.9), ape(v.5.7-1), stringi(v.1.7.12), zlibbioc(v.1.48.0), MASS(v.7.3-60), AnnotationHub(v.3.10.0), plyr(v.1.8.9), org.Hs.eg.db(v.3.18.0), parallel(v.4.3.1), HPO.db(v.0.99.2), deldir(v.1.0-9), graphlayouts(v.1.0.1), splines(v.4.3.1), hms(v.1.1.3), locfit(v.1.5-9.8), ps(v.1.7.5), reshape2(v.1.4.4), BiocVersion(v.3.18.0), evaluate(v.0.22), latticeExtra(v.0.6-30), tzdb(v.0.4.0), tweenr(v.2.0.2), httpuv(v.1.6.12), polyclip(v.1.10-6), ggforce(v.0.4.1), xtable(v.1.8-4), MPO.db(v.0.99.7), later(v.1.3.1), ragg(v.1.2.6), websocket(v.1.4.1), applot(v.0.2.2), memoise(v.2.0.1) and timechange(v.0.2.0)*

# Chapter 12

# Motif Analysis: MEME Suite

last updated: 2023-10-27

## 12.1 Description

In this class exercise, we will explore the use of the MEME suite for motif analysis.

In this exercise, we will learn to perform motif analysis using the MEME suite in R, covering tasks such as upstream sequence retrieval, motif identification, and comparison with external motif databases. Working with real yeast stress response data, we will gain proficiency in utilizing bioinformatics tools, and interpreting motif analysis results.

## 12.2 Learning Objectives

At the end of this exercise, you should be able to:

- **Bioinformatics Libraries:** load and utilize key bioinformatics libraries such as `biomaRt`, `memes`, and `Biostrings`.
- **Data Retrieval:** Retrieve upstream DNA sequences for specific genes from the Ensembl database using `biomaRt`.
- **Motif Analysis:**
  - Perform motif analysis using the MEME suite tools (`runMeme` and `runStreme`)

- Interpret results from motif analysis, including motif width and significance
- **TomTom Analysis:** Use TomTom to compare identified motifs with known motifs in databases & interpret results

### Install Packages

```
# Ensure required packages are installed
if (!require("pacman")) install.packages("pacman"); library(pacman)

# Load necessary packages
p_load("tidyverse", "knitr", "readr", "pander", "BiocManager",
       "dplyr", "stringr", "data.table",
       "biomaRt", "memes", "Biostrings", "curl", "universalmotif")

library(biomaRt)
library(memes)
library(curl)
library(universalmotif)
```

### 12.3 Install MEME suite

You will need to install MEME software on your computer if you don't already have it.

The code below downloads & installs the software for MacOS, if it not on your computer. The output when installing is VERY long, you can close the output by clicking the small x in the top right corner of the output box.

The MEME suite does not currently support Windows OS, although it can be done with WSL. We will be using Mac for this analysis.

```
# Check for XCode on Mac or prompt installation
xcode-select --install
cd ~

# Define MEME version
# Latest version as of 23 Oct 2023,
version="5.5.4"

# Install MEME if not already installed
if ! command -v $HOME/meme/bin/meme &> /dev/null; then
  curl -o $HOME/meme-$version.tar.gz https://meme-suite.org/meme/meme-software/$version
  tar zxf meme-$version.tar.gz
  cd meme-$version
```

## 12.4. ANALYSIS: MOTIF DISCOVERY FOR MSN2/4 VS WT RESPONSE TO ETOH295

```
./configure --prefix=$HOME/meme --with-url=http://meme-suite.org/ --enable-build-libxml2 --enable-build-libcurl  
make  
make test  
make install  
fi
```

### 12.3.1 Verify MEME Installation

```
# Check MEME installation  
check_meme_install()  
  
## checking main install  
  
## v /Users/clstacy/meme/bin  
## checking util installs  
##  
## v /Users/clstacy/meme/bin/dreme  
## v /Users/clstacy/meme/bin/ame  
## v /Users/clstacy/meme/bin/fimo  
## v /Users/clstacy/meme/bin/tomtom  
## v /Users/clstacy/meme/bin/meme  
## v /Users/clstacy/meme/bin/streme
```

If install didn't work, troubleshooting is needed.

## 12.4 Analysis: Motif Discovery for *msn2/4* vs WT Response to EtOH

### 12.4.1 Retrieve LogFC and FDR values

Let's load in the `DE_yeast_TF_stress.txt` file containing logFC and FDR values for a variety of yeast strains and stress conditions.

```
# Load gene file used in clustering  
FC_list <- data.table::fread(  
  "https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/data/DE_yeast_TF_stress.txt.gz"  
  as_tibble(.name_repair = "universal")  
)
```

```

## New names:
## * `logFC: WT NaCl response` -> `logFC..WT.NaCl.response`
## * `FDR: WT NaCl response` -> `FDR..WT.NaCl.response`
## * `logFC: msn24 mutant NaCl response` -> `logFC..msn24.mutant.NaCl.response`
## * `FDR: msn24 mutant NaCl response` -> `FDR..msn24.mutant.NaCl.response`
## * `logFC: yap1 mutant NaCl response` -> `logFC..yap1.mutant.NaCl.response`
## * `FDR: yap1 mutant NaCl response` -> `FDR..yap1.mutant.NaCl.response`
## * `logFC: skn7 mutant NaCl response` -> `logFC..skn7.mutant.NaCl.response`
## * `FDR: skn7 mutant NaCl response` -> `FDR..skn7.mutant.NaCl.response`
## * `logFC: WT EtOH response` -> `logFC..WT.EtOH.response`
## * `FDR: WT EtOH response` -> `FDR..WT.EtOH.response`
## * `logFC: msn24 mutant EtOH response` -> `logFC..msn24.mutant.EtOH.response`
## * `FDR: msn24 mutant EtOH response` -> `FDR..msn24.mutant.EtOH.response`
## * `logFC: yap1 mutant EtOH response` -> `logFC..yap1.mutant.EtOH.response`
## * `FDR: yap1 mutant EtOH response` -> `FDR..yap1.mutant.EtOH.response`
## * `logFC: skn7 mutant EtOH response` -> `logFC..skn7.mutant.EtOH.response`
## * `FDR: skn7 mutant EtOH response` -> `FDR..skn7.mutant.EtOH.response`
## * `logFC: WT v msn24 mutant: NaCl response` ->
##   `logFC..WT.v.msn24.mutant..NaCl.response`
## * `FDR: WT v msn24 mutant: NaCl response` ->
##   `FDR..WT.v.msn24.mutant..NaCl.response`
## * `logFC: WT v yap1 mutant: NaCl response` ->
##   `logFC..WT.v.yap1.mutant..NaCl.response`
## * `FDR: WT v yap1 mutant: NaCl response` ->
##   `FDR..WT.v.yap1.mutant..NaCl.response`
## * `logFC: WT v skn7 mutant: NaCl response` ->
##   `logFC..WT.v.skn7.mutant..NaCl.response`
## * `FDR: WT v skn7 mutant NaCl response` ->
##   `FDR..WT.v.skn7.mutant.NaCl.response`
## * `logFC: WT v msn24 mutant: EtOH response` ->
##   `logFC..WT.v.msn24.mutant..EtOH.response`
## * `FDR: WT v msn24 mutant: EtOH response` ->
##   `FDR..WT.v.msn24.mutant..EtOH.response`
## * `logFC: WT v yap1 mutant: EtOH response` ->
##   `logFC..WT.v.yap1.mutant..EtOH.response`
## * `FDR: WT v yap1 mutant: EtOH response` ->
##   `FDR..WT.v.yap1.mutant..EtOH.response`
## * `logFC: WT v skn7 mutant: EtOH response` ->
##   `logFC..WT.v.skn7.mutant..EtOH.response`
## * `FDR: WT v skn7 mutant: EtOH response` ->
##   `FDR..WT.v.skn7.mutant..EtOH.response`

```

## 12.4. ANALYSIS: MOTIF DISCOVERY FOR MSN2/4 VS WT RESPONSE TO ETOH297

### 12.4.2 Selecting an Ensembl BioMart database and dataset

```
ensembl <- useEnsembl(backend = "biomart")
ensembl

## Object of class 'Mart':
##   Using the ENSEMBL_MART_ENSEMBL BioMart database
##   No dataset selected.
```

Not we have loaded the “gene” database, but haven’t selected a dataset yet. Let’s find the one we want:

```
searchDatasets(mart = ensembl,
                pattern = "scerevisiae")

##                                     dataset                  description version
## 173 scerevisiae_gene_ensembl Saccharomyces cerevisiae genes (R64-1-1) R64-1-1
```

So the dataset needs to be “scerevisiae\_gene\_ensembl” for yeast. Can you find it for your organism?

```
# Assign ensembl with the desired dataset
ensembl <- useEnsembl(backend = "biomart",
                       dataset = "scerevisiae_gene_ensembl")
```

### 12.4.3 Retrieve Upstream Sequences

```
# Retrieve the ORF IDs included in the dataset
ORFs_in_analysis = FC_list$ID

# Get the upstream sequences for each gene and save in data.frame
seq <- getSequence(id = ORFs_in_analysis,
                    type = "ensembl_gene_id",
                    seqType = "coding_gene_flank",
                    upstream = 500,
                    mart = ensembl,
                    useCache = TRUE)

# Filter out rows with no sequence found
```

```

seq <- seq |> filter(coding_gene_flank != "Sequence unavailable")

# Display obtained sequences
glimpse(seq)

## Rows: 5,717
## Columns: 2
## $ coding_gene_flank <chr> "AATTGAATCTCATTGTGCATTCTCCACGCTACTTCTAGCAATTCCGCCT"
## $ ensembl_gene_id   <chr> "YBL091C", "YAL033W", "YAL064W", "YBL066C", "YBL089W"

```

Now we have the 500bp upstream sequences for (almost) all of the genes in the genome. Note that for yeast, this isn't too much data for a laptop. If you're working with a larger genome, you might need to subset the gene list down just to the genes in your gene list.

## 12.5 Motif Analysis for Genes Downregulated in EtOH Response

In order to continue, we need a list of genes.

What gene list do you think has the most interesting biological meaning for motif analysis? One comparison to consider is the genes that are most different in the EtOH response between WT and *msn2/4* samples. Let's get that list of genes with large magnitude and statistically significant negative logFC values now:

```

# Create gene list for downregulated genes
msn24_EtOH_down <- FC_list |>
  dplyr::filter(logFC..WT.v.msn24.mutant..EtOH.response < -2) |>
  dplyr::filter(FDR..WT.v.msn24.mutant..EtOH.response < 0.01) |>
  dplyr::select(ID) %>%
    # add the upstream sequences as a new column
    left_join(seq, by=c("ID" = "ensembl_gene_id")) |>
    drop_na("coding_gene_flank")

# Glimpse at the genes identified.
glimpse(msn24_EtOH_down)

## Rows: 107
## Columns: 2
## $ ID           <chr> "YAL061W", "YBL015W", "YBL049W", "YBL064C", "YBR054W"
## $ coding_gene_flank <chr> "CGCTATTTCTTTGTTCGTAACATCTGTGTATGTAGTAGTGTAAATCTA"

```

For MEME analysis, the gene list needs to be formatted like a fasta file. Here is how we can do that:

```
# create Biostrings object
msn24_EtOH_down_fa <- Biostrings::DNAStringSet(msn24_EtOH_down$coding_gene_flank)
# Add gene names
names(msn24_EtOH_down_fa) <- msn24_EtOH_down$ID
```

Let's create a folder to which we can save output files.

```
# Choose output directory for the output files to be saved
out_dir <- path.expand("~/Desktop/Genomic_Data_Analysis/Analysis/memes/")

# Create out_dir directory if doesn't already exist
if (!dir.exists(out_dir)) {
  dir.create(out_dir, recursive = TRUE)
}
```

### 12.5.1 Run Meme enrichment

This was the first algorithm developed in the MEME-suite, and is still widely used. However, it is not always the best approach. MEME is recommended when you have fewer than 50 sequences, while STREME is recommended when you have more. Also, the default parameter settings are often **not** the best options depending on your organism (e.g, bacteria tend to have longer motifs). We can adjust these settings in class to parameters better suited to our data (based on biological domain knowledge). See all of the parameter options at: <https://meme-suite.org/meme/doc/meme.html>

```
# Run Meme
meme_msn24_EtOH_down <- runMeme(msn24_EtOH_down_fa,
  minw = 8, # default is 8, for yeast I use 5
  maxw= 50, #default is 50, for yeast I use 20
  mod= "zoops", #zero or one occurrence per sequence
  parse_genomic_coord=FALSE,
  silent=F,
  outdir = path.expand(paste0(out_dir, "meme_msn24_EtOH_down"))
)

##

# Display Meme results
meme_msn24_EtOH_down
```

```

##          motif      name altname      consensus alphabet
## 1 <mot:TTYT..> TTYTTTTWYTTTTTYYTTBTT  MEME-1 YYYTTTTWYTTTYNTYTT      DNA
##   strand  icscore nsites    eval type pseudocount      bkg width
## 1       + 17.79857     64 1.4e-61   PPM           1 0.311, 0....    21
##   sites_hits
## 1 c("YOR34....
##
## [Hidden empty columns: family, organism, bkgsites, pval, qval.]

```

### 12.5.2 Run STEME enrichment

STEME is a newer MEME suite tool, that is better suited to looking for shorter motifs which are common in Eukaryotes.

For `runSteme`, properly setting the `control` parameter is key to discovering biologically relevant motifs. Often, using `control = "shuffle"` will produce a suboptimal set of motifs; however, some discriminative analysis designs don't have proper "control" regions other than to shuffle.

For our analysis, we can use the promoter sequences from the entire genome as a background to model the null distribution, let's create an object in R with all of the 500bp upstream sequences from `seq` to use as a control.

```

# create a DNAStringSet object from our FC_list we created above.
background_fa <- DNAStringSet(left_join(FC_list,seq, by=c("ID" = "ensembl_gene_id")) |>
                                drop_na("coding_gene_flank") |> pull(coding_gene_id))

# add gene names
names(background_fa) <- left_join(FC_list,seq, by=c("ID" = "ensembl_gene_id")) |>
                                drop_na("coding_gene_flank") |> pull(ID)

# Run Steme
steme_msn24_EtOH_down <- runSteme(msn24_EtOH_down_fa,
                                       control= background_fa,
                                       minw = 8, # default is 8, for yeast I use 5
                                       maxw= 15, #default is 15, for yeast i use 20
                                       parse_genomic_coord=FALSE,
                                       silent=TRUE,
                                       outdir = path.expand(paste0(out_dir, "steme_msn24_EtOH_down")))
                                       )

# Display Steme results
steme_msn24_EtOH_down

```

	motif	name	altname	consensus
##				
## m1_RCCCCCTTACC	<mot:m1_R..>	m1_RCCCCCTTACC	STREME-1	RCCCCCTTACC

12.5. MOTIF ANALYSIS FOR GENES DOWNREGULATED IN ETOH RESPONSE301

```

## m2_AAAGAACAGGAAGH <mot:m2_A..> m2_AAAGAACAGGAAGH STREME-2 AAAGAWGYAKGRAGH
## m3_AAGGGGAT <mot:m3_A..> m3_AAGGGGAT STREME-3 AAGGGGAT
## m4_CAAACAAGG <mot:m4_C..> m4_CAAACAAGG STREME-4 CAACAAGG
## m5_CCTTATATAD <mot:m5_C..> m5_CCTTATATAD STREME-5 CCKTATATAN
## m6_AYAGGGGT <mot:m6_A..> m6_AYAGGGGT STREME-6 AYAGGGGT
## m7_AAGGGAGAAAB <mot:m7_A..> m7_AAGGGAGAAAB STREME-7 AAGGGAGAAAN
## m8_MAGGGGCWGRANA <mot:m8_M..> m8_MAGGGGCWGRANA STREME-8 MAGGGGCWGRANA
## m9_CCTTTTCCC <mot:m9_C..> m9_CCTTTTCCC STREME-9 CCTTTTCCC
## alphabet strand icscore nsites pval type pseudocount
## m1_RCCCCTTACC DNA +- 11.03072 84 7.9e-06 PCM 0
## m2_AAAGAACAGGAAGH DNA +- 21.39292 9 1.7e-02 PCM 0
## m3_AAGGGGAT DNA +- 12.59474 27 2.1e-02 PCM 0
## m4_CAAACAAGG DNA +- 13.63451 16 3.9e-02 PCM 0
## m5_CCTTATATAD DNA +- 15.67707 17 4.3e-02 PCM 0
## m6_AYAGGGGT DNA +- 11.95501 21 4.7e-02 PCM 0
## m7_AAGGGAGAAAB DNA +- 13.59987 26 1.2e-01 PCM 0
## m8_MAGGGGCWGRANA DNA +- 14.77500 24 1.0e+00 PCM 0
## m9_CCTTTTCCC DNA +- 12.59822 28 1.0e+00 PCM 0
## bkg width initial_width seed
## m1_RCCCCTTACC 0.315, 0..... 10 9 ACCCCTTACC
## m2_AAAGAACAGGAAGH 0.315, 0..... 15 12 AAAGAACAGGAAGC
## m3_AAGGGGAT 0.315, 0..... 8 8 AAGGGGAT
## m4_CAAACAAGG 0.315, 0..... 8 8 CAACAAGG
## m5_CCTTATATAD 0.315, 0..... 10 9 CCTTATATAG
## m6_AYAGGGGT 0.315, 0..... 8 6 ACAGGGGT
## m7_AAGGGAGAAAB 0.315, 0..... 11 7 AAGGGAGAAAT
## m8_MAGGGGCWGRANA 0.315, 0..... 13 7 CAGGGGCAGGACA
## m9_CCTTTTCCC 0.315, 0..... 10 10 CCTTTTCCC
## score_threshold pos_count neg_count log_pval log_value
## m1_RCCCCTTACC 9.70716 8 81 -5.10131 -4.14707
## m2_AAAGAACAGGAAGH 17.469 1 0 -1.76418 -0.809934
## m3_AAGGGGAT 11.0408 3 34 -1.67894 -0.724701
## m4_CAAACAAGG 13.4256 2 17 -1.40874 -0.4545
## m5_CCTTATATAD 13.0759 2 18 -1.36701 -0.412767
## m6_AYAGGGGT 11.4254 2 19 -1.32757 -0.373326
## m7_AAGGGAGAAAB 12.0575 2 33 -0.93161 0.0226326
## m8_MAGGGGCWGRANA 14.0348 0 8 0 0.954243
## m9_CCTTTTCCC 11.8246 0 31 0 0.954243
## evaluate dtc bernoulli train_pos_count train_neg_count
## m1_RCCCCTTACC 7.1e-005 -1 -1 76 857
## m2_AAAGAACAGGAAGH 1.5e-001 -1 -1 8 17
## m3_AAGGGGAT 1.9e-001 -1 -1 24 300
## m4_CAAACAAGG 3.5e-001 -1 -1 14 179
## m5_CCTTATATAD 3.9e-001 -1 -1 15 153
## m6_AYAGGGGT 4.2e-001 -1 -1 19 234
## m7_AAGGGAGAAAB 1.1e+000 -1 -1 24 254

```



### 12.5.3 Run TomTom analysis

Now, we can see if any of those identified motifs correspond to known motifs in databases. The TomTom algorithm, part of the MEME-suite, allows us to just that, comparing our identified motifs to known motifs in annotation databases.

First, we need to download the YEASTRACT database file called (YEASTRACT\_20130918.meme). We will pull this file from Github, but you can see and download all of the available databases at: <https://meme-suite.org/meme/doc/download.html>.

```
# Download YEASTRACT database .meme

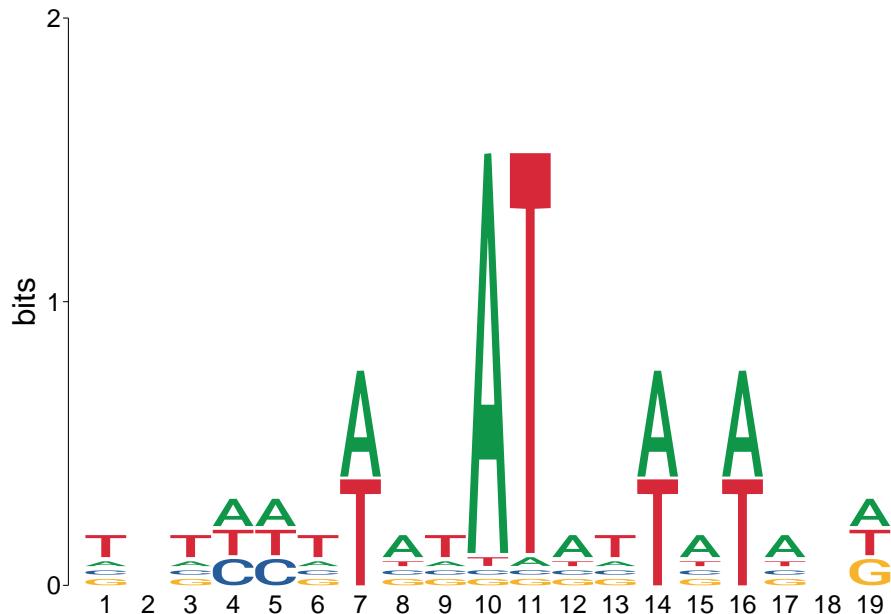
# Define URL where data is located
URL_to_download <-
  "https://github.com/clstacy/GenomicDataAnalysis_Fa23/raw/main/reference/YEAST/YEASTRACT_20130918.meme"

# Choose desired destination for the reference database file
db_destination <-
  path.expand("~/Desktop/Genomic_Data_Analysis/Reference/YEASTRACT_20130918.meme")

# Download the file and save to db_destination
curl::curl_download(
  url = URL_to_download,
  destfile = db_destination,
  quiet = FALSE,
  mode = "wb"
)

# Run TomTom on motifs found by runMeme()
meme_tomtom_msn24_EtOH_down <-
  runTomTom(
    input = meme_msn24_EtOH_down,
    norc = TRUE,
    thresh = 10,
    motif_pseudo = 0.1,
    database = db_destination,
    outdir = path.expand(paste0(out_dir, "tomtom_meme_msn24_EtOH_down"))
  )

# View Meme TomTom results
view_motifs(meme_tomtom_msn24_EtOH_down$best_match_motif)
```



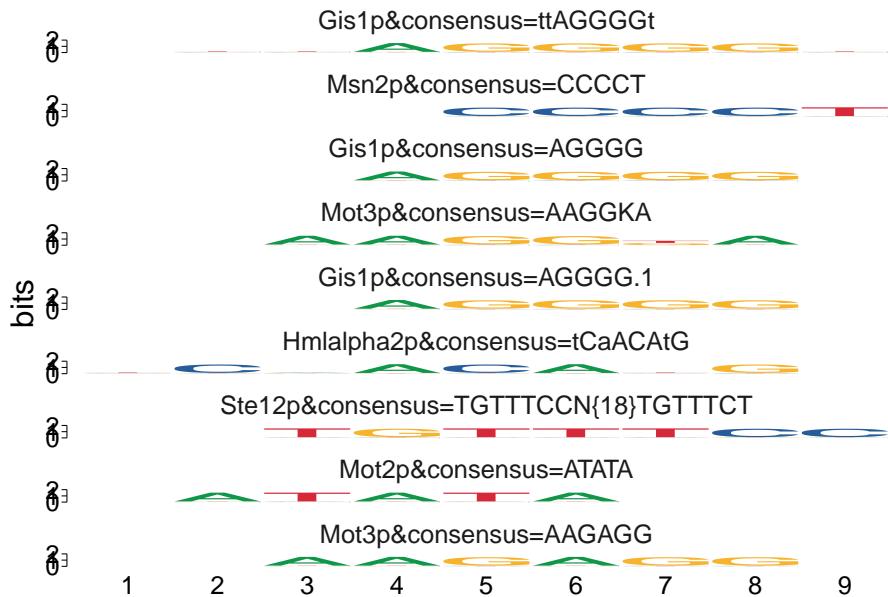
Go to the `out_dir` directory on your computer, and open the `tomtom.html` file. How convincing are those matches? With the default options, I'm not impressed.

Let's run TomTom on the output of the `runStreme()` command above, and see what we find.

```
# Run TomTom on motifs found by runStreme()
streme_tomtom_msn24_EtOH_down <-
  runTomTom(
    input = streme_msn24_EtOH_down,
    norc = TRUE,
    thresh = 10,
    motif_pseudo = 0.1,
    database = db_destination,
    outdir = path.expand(paste0(out_dir, "tomtom_streme_msn24_EtOH_down"))
  )

# View Streme TomTom results
view_motifs(streme_tomtom_msn24_EtOH_down$best_match_motif,
            relative_entropy = FALSE,
            normalise.scores = TRUE,
            use.type = "ICM",
            method = "WPCC",
            tryRC = FALSE)
```

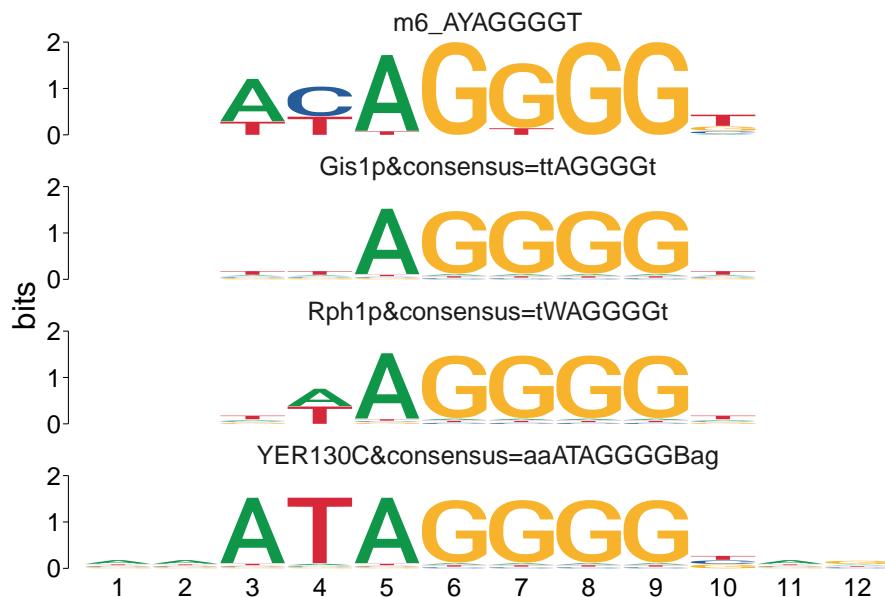
## 12.5. MOTIF ANALYSIS FOR GENES DOWNREGULATED IN ETOH RESPONSE



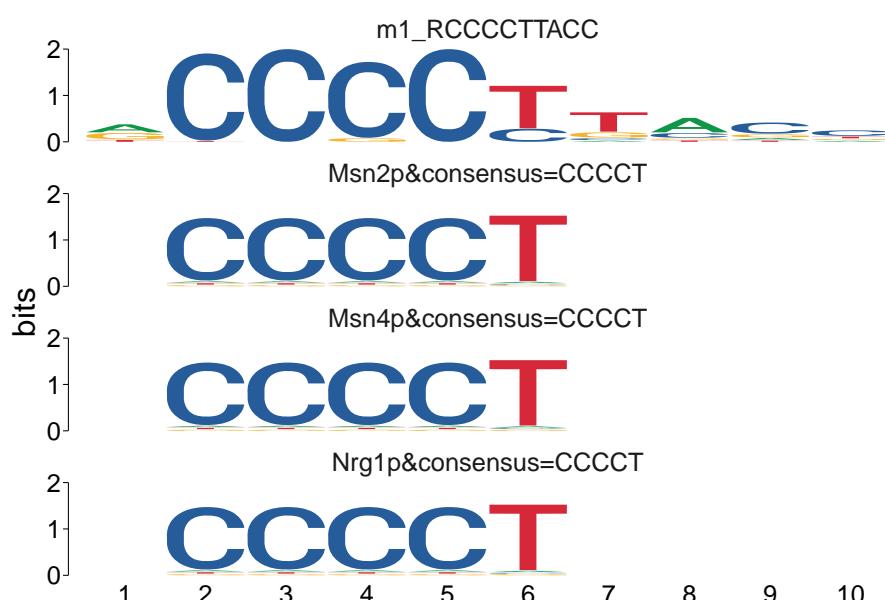
It can sometimes be helpful to manually look at the alignments to see if there's anything unexpected going on. We can use the command `view_tomtom_hits()` to do this. The figures aren't publication quality, but can be useful to see.

```
view_tomtom_hits(streme_tomtom_msn24_EtOH_down, top_n = 3)
```

```
## $m6_AYAGGGGT
```



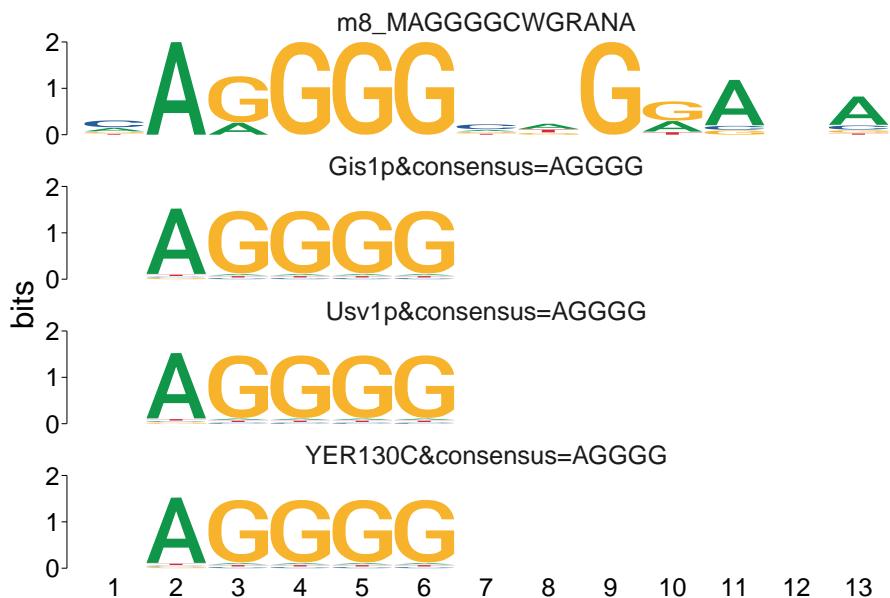
```
##  
## $m1_RCCCCTTACC
```



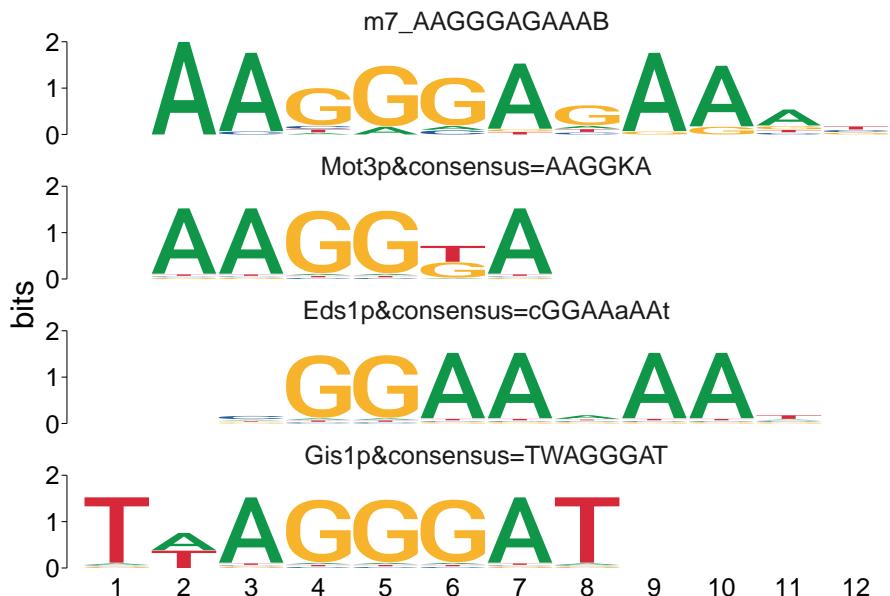
```
##
```

12.5. MOTIF ANALYSIS FOR GENES DOWNREGULATED IN ETOH RESPONSE307

```
## $m8_MAGGGGCWGRANA
```



```
##  
## $m7_AAGGGAGAAAB
```

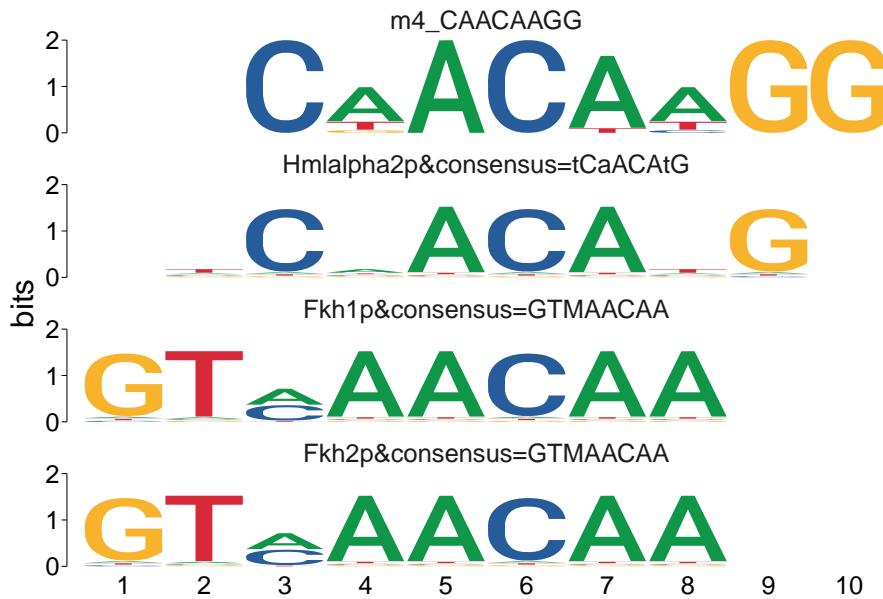


```
##  
## $m3_AAGGGGAT
```

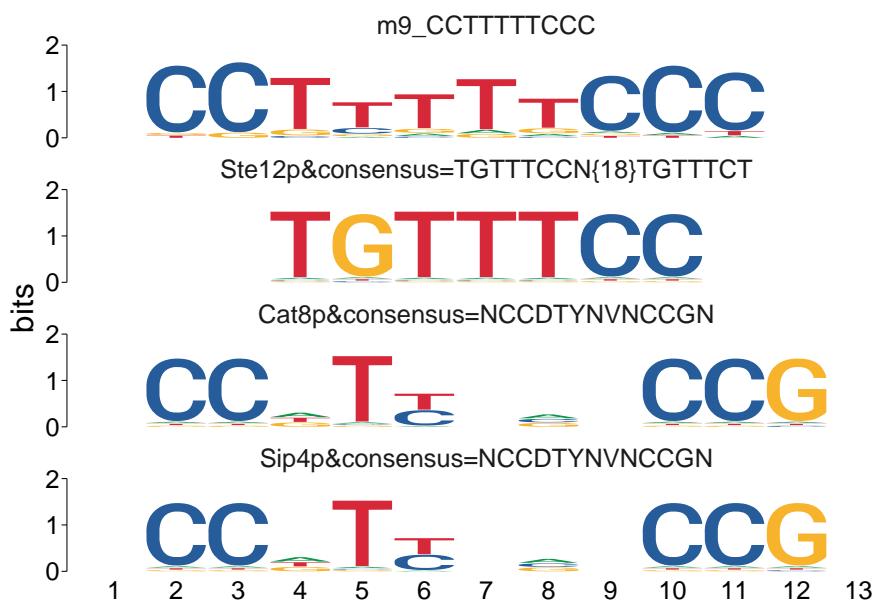


```
##  
## $m4_CAAACAAGG
```

12.5. MOTIF ANALYSIS FOR GENES DOWNREGULATED IN ETOH RESPONSE309

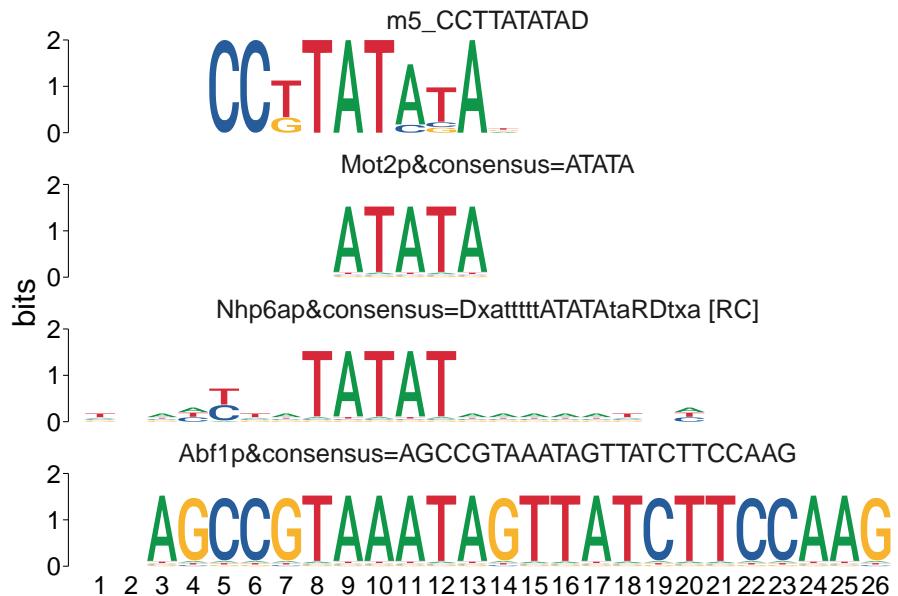


```
##  
## $m9_CCTTTTCCCC
```



```
##
```

```
## $m5_CCTTATATAD
```



```
##  
## $m2_AAAGAACAGGAAGH
```



Cool! We see Msn2/4 promoter sequence motif is strongly enriched in the promoter sequences of the genes that are lower in Msn2/4 vs WT response to EtOH.

Let's do a sanity check, and do the exact same thing but for genes that were *higher* in the same contrast, and see what enrichments if any are found. We can put just the code we need:

```
# Create gene list
msn24_EtOH_up <- FC_list |>
  # I want more than 8 genes, so I lowered the FC cutoff
  filter(logFC..WT.v.msn24.mutant..EtOH.response > 1) |>
  filter(FDR..WT.v.msn24.mutant..EtOH.response < 0.01) |>
  dplyr::select(ID) %>%
  # add the upstream seqs as a new column
  left_join(seq, by=c("ID" = "ensembl_gene_id")) |>
  drop_na("coding_gene_flank")

# create Biostrings object
msn24_EtOH_up_fa <- DNAStringSet(msn24_EtOH_up$coding_gene_flank)
# add gene names
names(msn24_EtOH_up_fa) <- msn24_EtOH_up$ID

# we already have background genes, so don't need to put that code again...

# Run Streme
streme_msn24_EtOH_up <- runStreme(msn24_EtOH_up_fa,
  control= background_fa,
  # control= "shuffle",
  minw = 5, # default is 8, for yeast I use 5
  maxw= 20, #default is 15, for yeast I use 20
  parse_genomic_coord=FALSE,
  silent=TRUE,
  outdir = path.expand(paste0(out_dir, "streme_msn24_EtOH_up"))
  )

# Run TomTom analysis
streme_tomtom_msn24_EtOH_up <- runTomTom(
  input = streme_msn24_EtOH_up,
  norc = TRUE,
  thresh = 10,
  motif_pseudo = 0.1,
  database = db_destination,
  outdir = path.expand(paste0(out_dir, "tomtom_streme_msn24_EtOH_up"))
)
```

Are Msn2/4 motifs enriched here? What biological meaning can we take away from this?

## 12.6 *skn7* exposed to salt.

Now, let's take a look another mutant from this study exposed to a different stressor.

```
# create gene list
skn7_NaCl_down <- FC_list |>
  filter(logFC..WT.v.skn7.mutant..NaCl.response < -2) |>
  filter(FDR..WT.v.skn7.mutant.NaCl.response<0.00001) |>
  dplyr::select(ID) %>%
  # add the upstream seqs as a new column
  left_join(seq, by=c("ID" = "ensembl_gene_id")) |>
  drop_na("coding_gene_flank")

# create Biostrings object
skn7_NaCl_down_fa <- DNAStringSet(skn7_NaCl_down$coding_gene_flank)
# add gene names
names(skn7_NaCl_down_fa) <- skn7_NaCl_down$ID

# we already have background genes, so don't need to put that code again...

# Run Streme
streme_skn7_NaCl_down <- runStreme(skn7_NaCl_down_fa,
  control= background_fa,
  # control= "shuffle",
  minw = 8, # default is 8, for yeast I use 5
  maxw= 15, #default is 15, for yeast I use 20
  evaluate=TRUE,
  parse_genomic_coord=FALSE,
  silent=TRUE,
  outdir = path.expand(paste0(out_dir, "streme_skn7_NaCl_down"))
)

## Warning: No hold-out set was created because the primary hold-out set
## would have had fewer than 5 sequences.

## Warning: Ignoring <thresh> (0.05) and setting <nmotifs> to 5.
```

```
# Run TomTom analysis
streme_tomtom_skn7_NaCl_down <- runTomTom(
  input = streme_skn7_NaCl_down,
  # norc = TRUE,
  thresh = 20,
  motif_pseudo = 0.1,
  database = db_destination,
  outdir = path.expand(paste0(out_dir, "tomtom_streme_skn7_NaCl_down"))
)
```

Notice we have fewer genes this time, let's try meme instead

```
# Run Meme
meme_skn7_NaCl_down <- runMeme(
  skn7_NaCl_down_fa,
  # control= "shuffle", #background_fa,
  # objfun="de",
  parse_genomic_coord = FALSE,
  minw = 5,
  maxw = 20,
  markov_order = 2,
  mod= "zoops", #zero or one occurence per sequence
  seed = 0,
  dna = T,
  revcomp = T,
  evt = 0.1,
  outdir = path.expand(paste0(out_dir, "meme_skn7_NaCl_down"))
)

# Run TomTom analysis
meme_tomtom_skn7_NaCl_down <- runTomTom(
  input = meme_skn7_NaCl_down,
  # norc = TRUE,
  thresh = 10,
  motif_pseudo = 0.1,
  min_overlap = 5,
  database = db_destination,
  outdir = path.expand(paste0(out_dir, "tomtom_meme_skn7_NaCl_down"))
)
```

## 12.7 Questions

1. What is the difference in motifs when you change the settings for analyzing the msn2/4 mutant vs wild-type ethanol response using “zoops” or “anr”

as the setting?

2. Perform MEME and STREME on the wild-type salt response and wild-type ethanol response (using a log2 FC of 3 and FDR < 0.01) to identify motifs in genes that are strongly induced for each stress, and then follow that analysis with TOMTOM. What TFs may be shared and what TFs may be different between the two stress responses?

Be sure to knit this file into a pdf or html file once you're finished.

System information for reproducibility:

```
pander::pander(sessionInfo())
```

### R version 4.3.1 (2023-06-16)

**Platform:** aarch64-apple-darwin20 (64-bit)

**locale:** en\_US.UTF-8|en\_US.UTF-8||en\_US.UTF-8||C||en\_US.UTF-8|en\_US.UTF-8

**attached base packages:** stats4, stats, graphics, grDevices, utils, datasets, methods and base

**other attached packages:** universalmotif(v.1.20.0), curl(v.5.1.0), memes(v.1.10.0), biomaRt(v.2.58.0), data.table(v.1.14.8), NbClust(v.3.0.1), factoextra(v.1.0.7), ggfx(v.1.0.1), patchwork(v.1.1.3), ggkegg(v.0.99.4), tidygraph(v.1.2.3), XML(v.3.99-0.14), ggraph(v.2.1.0), KEGGgraph(v.1.62.0), ggupset(v.0.3.0), pathview(v.1.42.0), BiocFileCache(v.2.10.0), dbplyr(v.2.3.4), DOSE(v.3.28.0), ggrepel(v.0.9.4), viridis(v.0.6.4), viridisLite(v.0.4.2), scales(v.1.2.1), Glimma(v.2.12.0), DESeq2(v.1.41.12), edgeR(v.3.99.5), limma(v.3.58.0), reactable(v.0.4.4), webshot2(v.0.1.1), statmod(v.1.5.0), Rsubread(v.2.16.0), ShortRead(v.1.60.0), GenomicAlignments(v.1.38.0), SummarizedExperiment(v.1.32.0), MatrixGenerics(v.1.14.0), matrixStats(v.1.0.0), Rsamtools(v.2.18.0), GenomicRanges(v.1.54.0), Biostrings(v.2.70.1), GenomeInfoDb(v.1.38.0), XVector(v.0.42.0), BiocParallel(v.1.36.0), Rfastp(v.1.12.0), org.Sc.sgd.db(v.3.18.0), AnnotationDbi(v.1.64.0), IRanges(v.2.36.0), S4Vectors(v.0.40.0), Biobase(v.2.62.0), BiocGenerics(v.0.48.0), clusterProfiler(v.4.10.0), ggVennDiagram(v.1.2.3), tidytree(v.0.4.5), igraph(v.1.5.1), janitor(v.2.2.0), BiocManager(v.1.30.22), pander(v.0.6.5), knitr(v.1.44), here(v.1.0.1), lubridate(v.1.9.3),forcats(v.1.0.0), stringr(v.1.5.0), dplyr(v.1.1.3), purrr(v.1.0.2), readr(v.2.1.4), tidyverse(v.1.3.0), tibble(v.3.2.1), ggplot2(v.3.4.4), tidyverse(v.2.0.0) and pacman(v.0.5.1)

**loaded via a namespace (and not attached):** fs(v.1.6.3), bitops(v.1.0-7), enrichplot(v.1.22.0), HDO.db(v.0.99.1), httr(v.1.4.7), RColorBrewer(v.1.1-3), Rgraphviz(v.2.46.0), tools(v.4.3.1), utf8(v.1.2.4), R6(v.2.5.1), lazyeval(v.0.2.2), GetoptLong(v.1.0.5), withr(v.2.5.1), prettyunits(v.1.2.0), gridExtra(v.2.3), cli(v.3.6.1), textshaping(v.0.3.7), Cairo(v.1.6-1), scatterpie(v.0.2.1), systemfonts(v.1.0.5), yulab.utils(v.0.1.0), gson(v.0.1.0),

*R.utils(v.2.12.2), rstudioapi(v.0.15.0), RSQLite(v.2.3.1), generics(v.0.1.3), gridGraphics(v.0.5-1), hwriter(v.1.3.2.1), GO.db(v.3.18.0), Matrix(v.1.6-1.1), interp(v.1.1-4), fansi(v.1.0.5), abind(v.1.4-5), R.methodsS3(v.1.8.2), lifecycle(v.1.0.3), yaml(v.2.3.7), snakecase(v.0.11.1), qvalue(v.2.34.0), SparseArray(v.1.2.0), grid(v.4.3.1), blob(v.1.2.4), promises(v.1.2.1), crayon(v.1.5.2), lattice(v.0.22-5), cowplot(v.1.1.1), chromote(v.0.1.2), KEGGREST(v.1.42.0), magick(v.2.8.1), pillar(v.1.9.0), fgsea(v.1.28.0), rjson(v.0.2.21), codetools(v.0.2-19), fastmatch(v.1.1-4), glue(v.1.6.2), ggrepel(v.0.1.3), remotes(v.2.4.2.1), vctrs(v.0.6.4), png(v.0.1-8), treeio(v.1.26.0), testthat(v.3.2.0), gtable(v.0.3.4), cachem(v.1.0.8), xfun(v.0.40), S4Arrays(v.1.2.0), mime(v.0.12), RVenn(v.1.1.0), interactiveDisplayBase(v.1.40.0), ellipsis(v.0.3.2), nlme(v.3.1-163), usethis(v.2.2.2), ggtree(v.3.10.0), bit64(v.4.0.5), progress(v.1.2.2), filelock(v.1.0.2), rprojroot(v.2.0.3), colorspace(v.2.1-0), DBI(v.1.1.3), tidyselect(v.1.2.0), processx(v.3.8.2), bit(v.4.0.5), compiler(v.4.3.1), graph(v.1.80.0), xml2(v.1.3.5), desc(v.1.4.2), DelayedArray(v.0.28.0), bookdown(v.0.36), shadowtext(v.0.1.2), rappdirs(v.0.3.3), digest(v.0.6.33), rmarkdown(v.2.25), htmltools(v.0.5.6.1), pkgconfig(v.2.0.3), jpeg(v.0.1-10), fastmap(v.1.1.1), rlang(v.1.1.1), GlobalOptions(v.0.1.2), htmlwidgets(v.1.6.2), shiny(v.1.7.5.1), farver(v.2.1.1), jsonlite(v.1.8.7), R.oo(v.1.25.0), GOSemSim(v.2.28.0), RCurl(v.1.98-1.12), magrittr(v.2.0.3), GenomeInfoDbData(v.1.2.11), ggplotify(v.0.1.2), munsell(v.0.5.0), Rcpp(v.1.0.11), ape(v.5.7-1), stringi(v.1.7.12), brio(v.1.1.3), zlibbioc(v.1.48.0), MASS(v.7.3-60), AnnotationHub(v.3.10.0), plyr(v.1.8.9), org.Hs.eg.db(v.3.18.0), ggseqlogo(v.0.1), parallel(v.4.3.1), HPO.db(v.0.99.2), deldir(v.1.0-9), graphlayouts(v.1.0.1), splines(v.4.3.1), hms(v.1.1.3), locfit(v.1.5-9.8), ps(v.1.7.5), pkgload(v.1.3.3), reshape2(v.1.4.4), BiocVersion(v.3.18.0), evaluate(v.0.22), latticeExtra(v.0.6-30), tzdb(v.0.4.0), tweenr(v.2.0.2), httpuv(v.1.6.12), polyclip(v.1.10-6), ggforce(v.0.4.1), xtable(v.1.8-4), MPO.db(v.0.99.7), later(v.1.3.1), ragg(v.1.2.6), websocket(v.1.4.1), aplot(v.0.2.2), memoise(v.2.0.1), timechange(v.0.2.0) and cmdfun(v.1.0.2)*