Project 1: Exploratory Data Analysis

Kenia E. Segura Aba

5/19/2020

Data Wrangling and Data Exploration

Introduction

Microalgae are attractive biofuel feedstock candidates due to their ability to accumulate a high percent of their dry weight as carbon storage compounds, such as starch and neutral lipids, when nutrient deprived. There have been many studies about nitrogen deprivation in *Chlamydomonas reinhardtii* and the resulting lipid accumulation, but recovery from this stress has not been elucidated (Miller et al.,2010, et al., Park et al., 2015). Understanding neutral lipid breakdown during stress recovery could lead to engineering microalgae that can retain high neutral lipid levels without sacrificing population growth rates. In this project, the relationship between lipid metabolism genes and recovery from nitrogen deprivation are examined. Tsai et al. (2018) conducted comparative transcriptomics on this alga to discern processes relevant to cellular quiescence, a reversible cell cycle arrest with drastic changes in metabolism allowing cells to remain viable, in the context of N deprivation and recovery following refeeding.

I chose this dataset because I want to learn how to process and interpret genomics datasets. I also want to delve into plant or microbial genetics in graduate school. The data set I will be working with has gene IDs row-wise and log2 fold change with corresponding p-values column-wise. RNA-Seq gene expression levels were measured by Illumina RNAseq. I presume that expression of genes that were upregulated during nitrogen deprivation should reverse upon re-introduction of nitrogen.

```
# upload data
chlamy_data <- read.csv("~/Downloads/chlamy_data.csv", header = T)
head(chlamy_data, 5)  # re-introduced nitrogen ; zero hours into N recovery
```

```
##         GENE.ID NminNR.L2FC NminNR.padj NR6Nmin.L2FC NR6Nmin.padj NR12Nmin.L2FC
## 1 Cre01.g000050        0.87     0.01750        -0.75     0.092600         -1.12
## 2 Cre01.g000100       -0.72     0.06100        -0.44     0.398000          0.42
## 3 Cre01.g000150       -2.79     0.00074         5.49     0.000475          6.06
## 4 Cre01.g000200       -0.16     0.78500         0.58     0.223000          0.17
## 5 Cre01.g000250        2.18     0.03900        -1.72     0.096800         -1.41
##   NR12Nmin.padj
## 1      5.27e-03
## 2      3.79e-01
## 3      3.98e-28
## 4      8.05e-01
## 5      1.57e-01
```

```
annotations <- read.csv("~/Downloads/annotations.csv", header = T)
head(annotations, 5)  # 6 hours and 12 hours into N recovery
```

```
##         GENE.ID PFAM.ID              PFAM.Description
## 1 Cre01.g000050 PF02042                 RWP-RK domain
## 2 Cre01.g000100
## 3 Cre01.g000150 PF02535           metal ion transport
```

```
## 4 Cre01.g000200
## 5 Cre01.g000250 PF09335 SNARE associated Golgi protein
```

```
# coerce numeric columns with type 'character' to a numeric
# double precision vector
chlamy_data$NR6Nmin.L2FC <- as.double(chlamy_data$NR6Nmin.L2FC)
chlamy_data$NR12Nmin.L2FC <- as.double(chlamy_data$NR12Nmin.L2FC)

head(chlamy_data, 5)
```

```
##         GENE.ID NminNR.L2FC NminNR.padj NR6Nmin.L2FC NR6Nmin.padj NR12Nmin.L2FC
## 1 Cre01.g000050        0.87     0.01750        -0.75     0.092600         -1.12
## 2 Cre01.g000100       -0.72     0.06100        -0.44     0.398000          0.42
## 3 Cre01.g000150       -2.79     0.00074         5.49     0.000475          6.06
## 4 Cre01.g000200       -0.16     0.78500         0.58     0.223000          0.17
## 5 Cre01.g000250        2.18     0.03900        -1.72     0.096800         -1.41
##   NR12Nmin.padj
## 1      5.27e-03
## 2      3.79e-01
## 3      3.98e-28
## 4      8.05e-01
## 5      1.57e-01
```

Tidying: Rearranging Wide/Long

```
library(tidyverse)

# untidy the data: pivot_longer(-1) is used to transpose
# every column except for the first one to a longer format in
# the full dataset.
untidy <- chlamy_data %>% pivot_longer(-1)
head(untidy)
```

```
## # A tibble: 6 x 3
##   GENE.ID       name             value
##   <chr>         <chr>            <dbl>
## 1 Cre01.g000050 NminNR.L2FC     0.87
## 2 Cre01.g000050 NminNR.padj     0.0175
## 3 Cre01.g000050 NR6Nmin.L2FC   -0.75
## 4 Cre01.g000050 NR6Nmin.padj    0.0926
## 5 Cre01.g000050 NR12Nmin.L2FC -1.12
## 6 Cre01.g000050 NR12Nmin.padj  0.00527
```

```
# tidy the data: pivot_wider() was used to reverse
# pivot_longer(-1). Rows with NAs were removed.
tidy <- untidy %>% separate(name, into = c("Condition", "Measurement")) %>%
    pivot_wider(names_from = "Measurement", values_from = "value") %>%
    na.omit()

head(tidy)
```

```
## # A tibble: 6 x 4
##   GENE.ID       Condition  L2FC    padj
##   <chr>         <chr>     <dbl>   <dbl>
## 1 Cre01.g000050 NminNR     0.87 0.0175
## 2 Cre01.g000050 NR6Nmin   -0.75 0.0926
```

```
## 3 Cre01.g000050 NR12Nmin  -1.12 0.00527
## 4 Cre01.g000100 NminNR    -0.72 0.061
## 5 Cre01.g000100 NR6Nmin   -0.44 0.398
## 6 Cre01.g000100 NR12Nmin   0.42 0.379
```

Joining

I used full_join to combine the datasets by the `GENE.ID` column in both datasets, therefore, no genes or attributes were dropped.

```
full <- annotations %>% full_join(tidy, by = "GENE.ID")

head(full)
```

```
##         GENE.ID PFAM.ID PFAM.Description Condition  L2FC    padj
## 1 Cre01.g000050 PF02042    RWP-RK domain    NminNR  0.87 0.01750
## 2 Cre01.g000050 PF02042    RWP-RK domain   NR6Nmin -0.75 0.09260
## 3 Cre01.g000050 PF02042    RWP-RK domain  NR12Nmin -1.12 0.00527
## 4 Cre01.g000100                            NminNR -0.72 0.06100
## 5 Cre01.g000100                           NR6Nmin -0.44 0.39800
## 6 Cre01.g000100                          NR12Nmin  0.42 0.37900
```

Data Wrangling

Compute summary statistics and determine which genes are up-regulated or down-regulation during N recovery. The threshold for "up" is 1 or more and the threshold for "down" is -1 or less.

```
# USE: filter, select, arrange, group_by, mutate, summarize
stats <- full %>% group_by(GENE.ID) %>% select_if(is.numeric) %>%
    summarize_at(vars(L2FC), list(mean, sd, min, max)) %>% rename(mean = fn1,
    sd = fn2, min = fn3, max = fn4) %>% arrange(desc(mean)) %>%
    na.omit()

# Suppose I only want a subset of the full data at zero hours
# into N recovery and figure out whether the gene was
# upregulated or downregulated:
zero <- full %>% filter(Condition == "NminNR") %>% mutate(NR.Category = ifelse(L2FC >
    1 & L2FC, "up", "down"))
head(zero)
```

```
##         GENE.ID PFAM.ID               PFAM.Description Condition  L2FC    padj
## 1 Cre01.g000050 PF02042                  RWP-RK domain    NminNR  0.87 0.01750
## 2 Cre01.g000100                                           NminNR -0.72 0.06100
## 3 Cre01.g000150 PF02535            metal ion transport    NminNR -2.79 0.00074
## 4 Cre01.g000200                                           NminNR -0.16 0.78500
## 5 Cre01.g000250 PF09335 SNARE associated Golgi protein    NminNR  2.18 0.03900
## 6 Cre01.g000300 PF00561      alpha/beta hydrolase fold    NminNR  3.82 0.01380
##   NR.Category
## 1        down
## 2        down
## 3        down
## 4        down
## 5          up
## 6          up
```

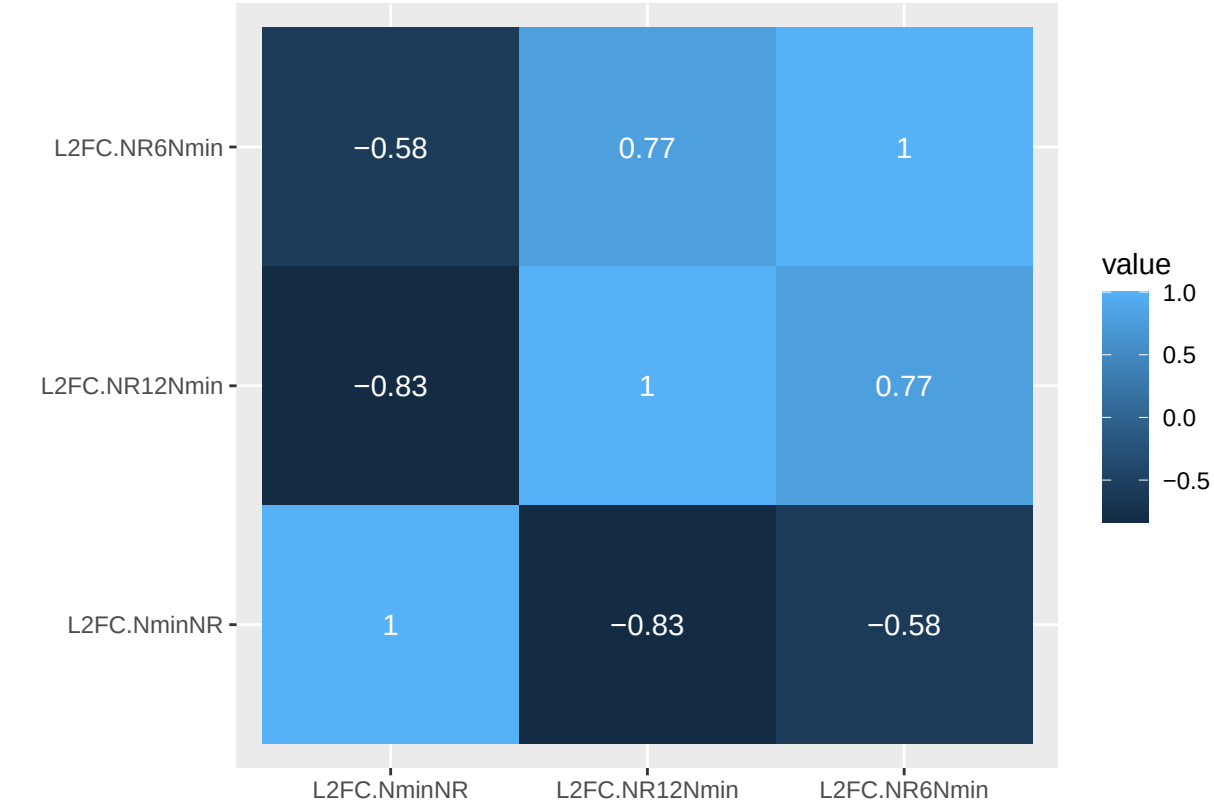
```
# Let's do this for the full dataset
full <- full %>% mutate(NR.Category = ifelse(L2FC < 1 & L2FC >
```

```
    -1, "same", ifelse(L2FC <= -1, "down", "up"))) %>% na.omit()
```

Visualizing Gene Expression

Figure 1: Correlation heatmap of *C. reinhardtii* gene expression in zero hours, 6 hours, and 12 hours into N recovery.

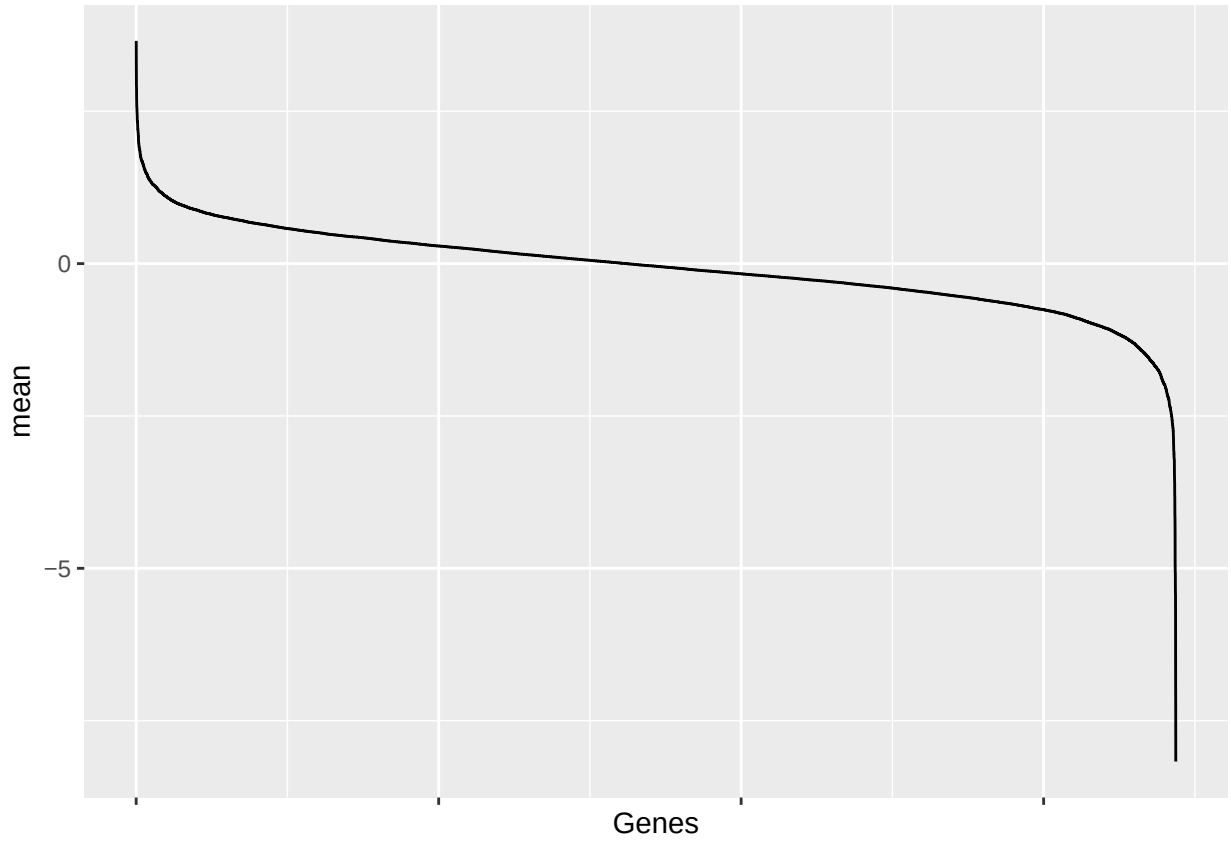
```
# rearrange full without NR category to calculate covariances
# between log 2 fold change values in the 3 conditions
full2 <- full[, 1:6] %>% pivot_wider(names_from = Condition,
    values_from = c(L2FC, padj), names_sep = ".") %>% na.omit

# set row names
rownames(full2) <- full2$GENE.ID

# correlation heatmap
full2 %>% select(4:6) %>% cor %>% as.data.frame %>% rownames_to_column %>%
    pivot_longer(-1) %>% ggplot(aes(rowname, name, fill = value,
    label = signif(value, 2))) + geom_tile() + xlab("") + ylab("") +
    geom_text(label.size = 4, colour = "white")
```



The variables NR6Nmin and NR12Nmin are positively correlated since they both measure gene expression during nitrogen recovery (NR) when compared to being deprived (Nmin for N minus). These two are negatively correlated with NminNR since this is the point in which cells are coming out of cellular quiescence during nitrogen deprivation. Thus, it is expected to see an opposite relationship because some genes take time to reverse their expression once nitrogen is reintroduced. Over time, this relatinoship becomes increasingly negative.

Figure 2: Line plot of mean gene expression.

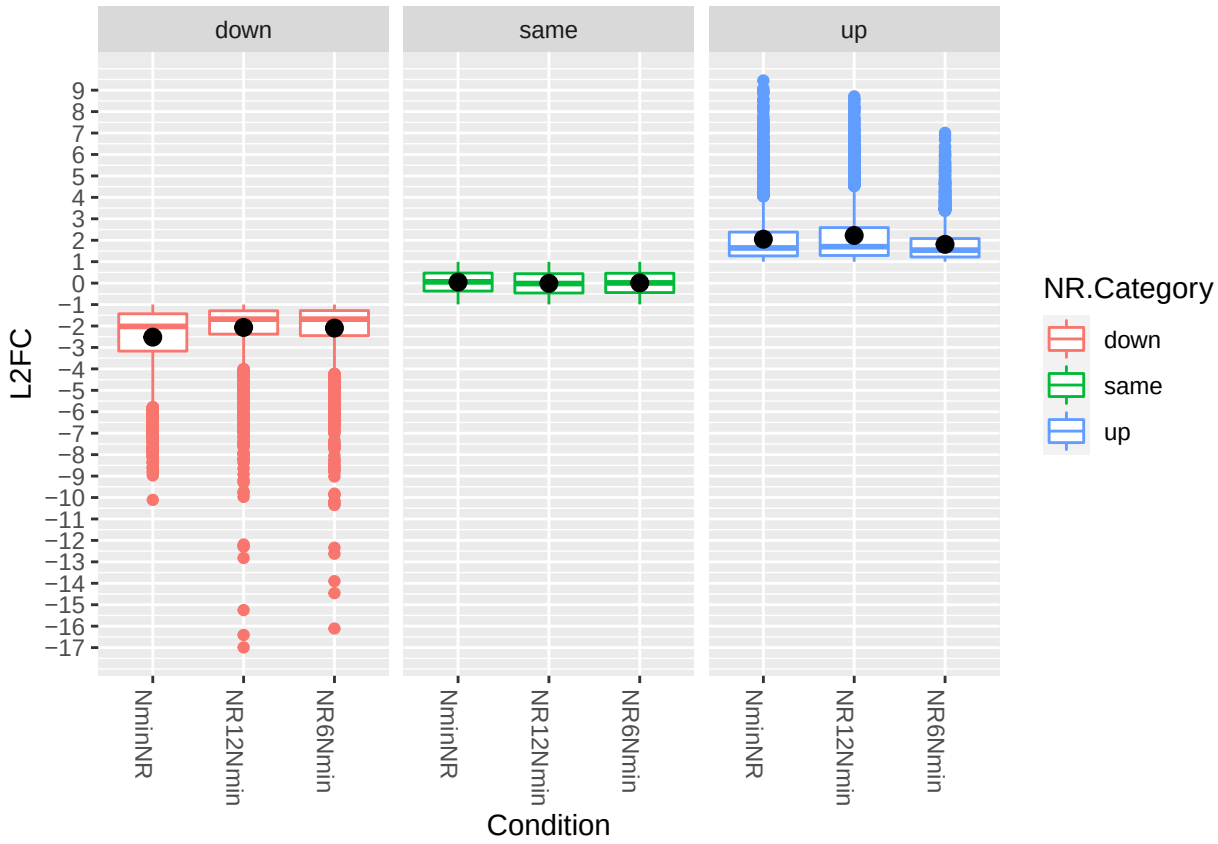
```
stats %>% ggplot(aes(x = 1:17184, y = mean)) + geom_line() +
    theme(axis.text.x = element_blank(), axis.title = element_text()) +
    labs(x = "Genes")
```



Genes were arranged from largest to smallest mean L2FC values and the means were plotted. From this plot, about half of the genes have log 2 fold change values above zero, and about half are below zero. For genes with a mean of about 2.5, it means that their gene expression increased 2.5 times when the initial state of the cell was nitrogen deprivation and the final state is nitrogen replete (cells are in nitrogen recovery mode). Down-regulationg of genes is more pronounced, with up to about a decrease of 7.5 times in gene expression.

Figure 3: Boxplots of log 2 fold change per condition - N minus/N replete, N replete at 6 hours / N minus, N replete at 12 hours / N minus.

```
full %>% ggplot(aes(x = Condition, y = L2FC)) + geom_boxplot(aes(colour = NR.Category)) +
    theme(axis.text.x = element_text(angle = -90, hjust = 0)) +
    scale_y_continuous(breaks = round(seq(min(full$L2FC), max(full$L2FC),
        by = 1), 1)) + stat_summary(fun.y = mean) + facet_wrap(~NR.Category)
```

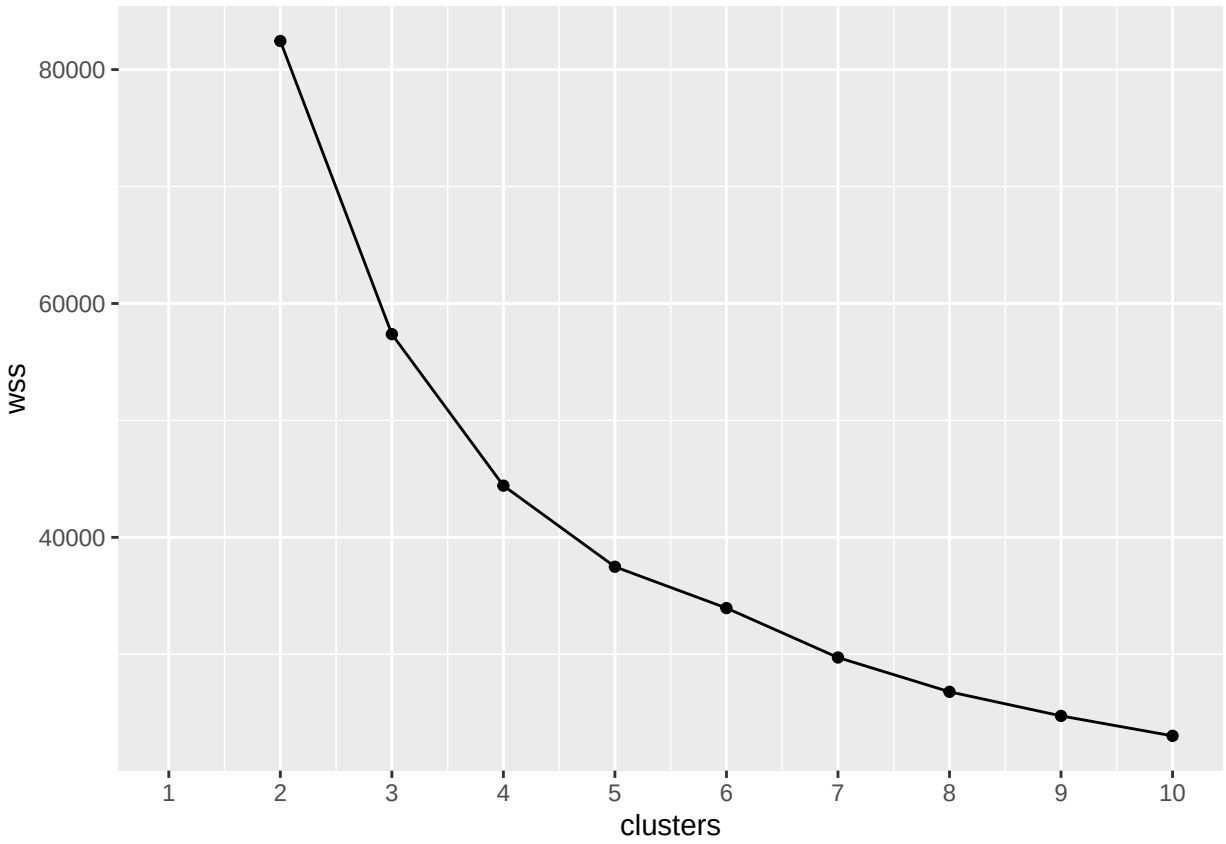


There is more upregulation of genes at zero hours into N recovery, then it decreases a little after 6 hours, and at 12 hours it goes up again. Perhaps, the cells are being restored to a healthier state after 12 hours. If we had more time points, then a more accurate understanding of how *C. reinhardtii* cells behave overall in terms of cell growth and how the genome fluctuates would be obtained.

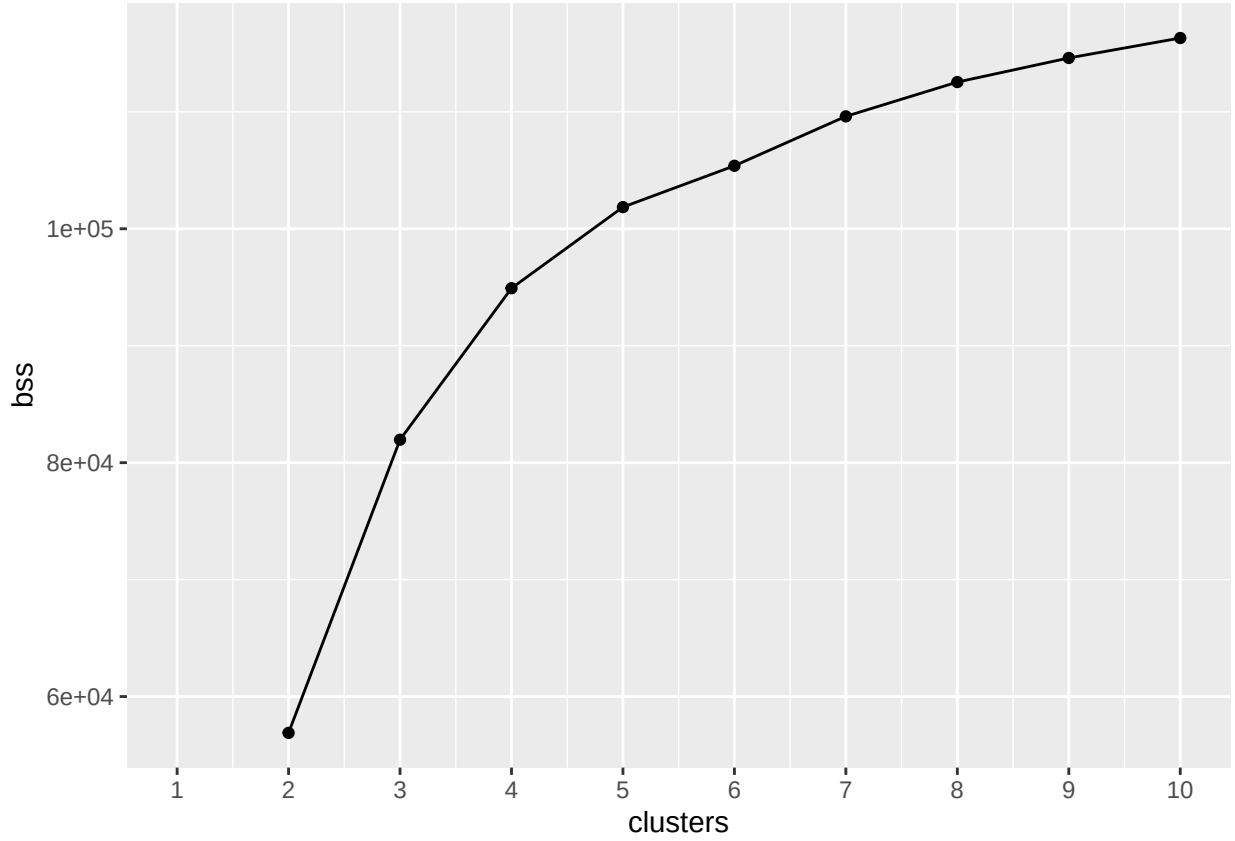
Dimensionality Reduction

Figure 6: K-means Clustering plot. The package `factoextra` plots a kmeans object using principal components for visualization.

```
# compute WSS (within cluster distances) for optimal k-value
wss <- vector()
bss <- vector()
for (i in 2:10) {
    temp <- full2 %>% select(4:6) %>% kmeans(i)
    wss[i] <- temp$tot.withinss
    bss[i] <- temp$betweenss
}

# plot is ambiguous, maybe 3 or 4 clusters is optimal
ggplot() + geom_point(aes(x = 1:10, y = wss)) + geom_path(aes(x = 1:10,
    y = wss)) + xlab("clusters") + scale_x_continuous(breaks = 1:10)
```




```
ggplot() + geom_point(aes(x = 1:10, y = bss)) + geom_path(aes(x = 1:10,
    y = bss)) + xlab("clusters") + scale_x_continuous(breaks = 1:10)
```



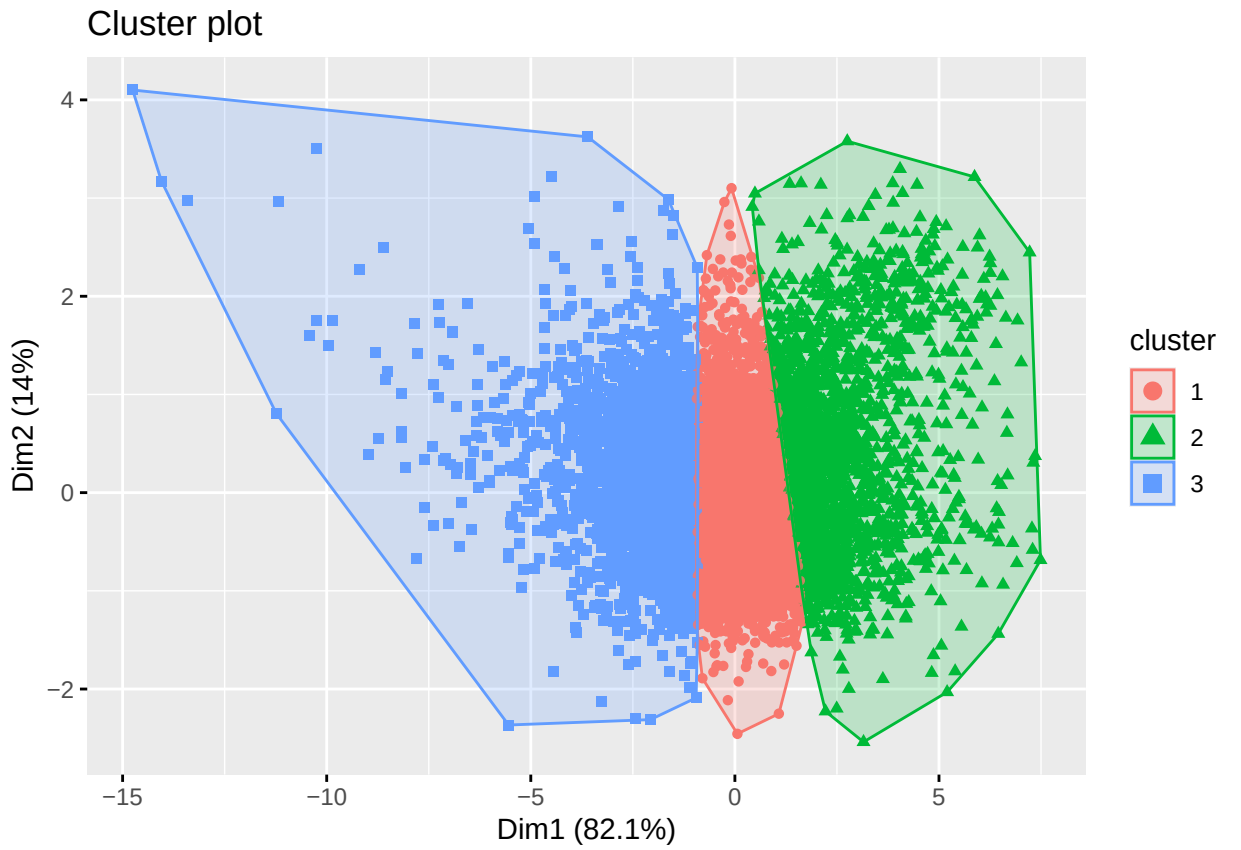

```
# couldn't compute silhouette widths due to error Error:
# vector memory exhausted (limit reached?)

# K-means clustering
kmeans <- full2[, 4:6] %>% kmeans(3)  #set number of clusters k
kmeans$size
```

```
## [1] 10765  2411  3590
```

```
kmeans$centers
```

```
##   L2FC.NminNR L2FC.NR6Nmin L2FC.NR12Nmin
## 1 -0.09672829    0.1591667     0.1129549
## 2 -3.01603069    1.5006387     2.6241767
## 3  1.70480223   -1.8882813    -2.0216602
```

```
# save K-means assignment as a column in dataset
kmeansclust <- full2 %>% mutate(cluster = as.factor(kmeans$cluster))

# plot data colored by final custer assignment
library(factoextra)
fviz_cluster(object = kmeans, data = full2[, 4:6], geom = "point")
```




```
# determine in which cluster LIP4 is
kmeansclust[10832, ]
```

```
## # A tibble: 1 x 10
##   GENE.ID PFAM.ID PFAM.Description L2FC.NminNR L2FC.NR6Nmin L2FC.NR12Nmin
##   <chr>   <chr>   <chr>                  <dbl>        <dbl>         <dbl>
## 1 Cre17.~ PF0173~ lipid metabolic~       -1.94         0.94           0.5
## # ... with 4 more variables: padj.NminNR <dbl>, padj.NR6Nmin <dbl>,
## #   padj.NR12Nmin <dbl>, cluster <fct>
```

Determining the k-value to use was difficult because there is no clear separation between clusters. Perhaps, if genes whose expression did not change were removed, we would see a clear separation between up-regulated and down-regulated genes. The majority of the variance, 82.1%, is explained by principal component 1. Genes were clustered based on gene expression, thus this separation is due to the log 2 fold change (up, down, same categories). There was a previous study conducted in 2019 that identified a triacylglycerol (TAG) lipase (LIP4), with the gene ID Cre17.g699100. This gene clustered in cluster 2. At zero hours of N recovery, it was downregulated 1.94 fold ($P < 9.83e-09$), and at 6 hours it was upregulated 0.94 fold ($P < 0.0248$). It is likely that other TAG lipases clustered together with LIP4.

References

Tsai, C. H., Uygun, S., Roston, R., Shiu, S. H., & Benning, C. (2018). Recovery from N deprivation is a transcriptionally and functionally distinct state in Chlamydomonas. Plant physiology, 176(3), 2007-2023.

Warakanont, J., Li-Beisson, Y., & Benning, C. (2019). LIP4 Is Involved in Triacylglycerol Degradation in Chlamydomonas reinhardtii. Plant and Cell Physiology, 60(6), 1250-1259.