Project 2 - Final

Code

Introduction

Scientific Question: How similar is the gene sequence of the Black Perigord truffle (*Tuber melanosporum*) of its fruit body stage compared to its free-living mycelium stage, and what can this tell us about the truffle's characteristics?

Background: The Black Perigord truffle (*T. melanosporum*) has a life cycle that includes the developmental stages of fruit body (FB) and free-living mycelium (FLM), among other stages. This project will just focus on these 2 stages. From scientific articles regarding T. melanosporum, it has been found that different elements/factors of the truffle's gene expression could be found. For instance, it was found that a specific enzyme tyrosinase was expressed during the black truffle's developing stages and cycle, and it changes in expression at different stages. So using this knowledge, I wanted to explore how this gene expression could perhaps allow for connections to be made between the developing stages and genes to form conclusions about the truffle.

Scientific Hypothesis: If there is a similar sequence found in both the FM and FLM stages that align with each other and is involved in the truffle's volatility, then the specific sequence identified has the most impact on the truffle's aroma.

Analyses: The analyses that will be done are RNAseq, RSCU analysis, and multiple sequence alignment (MSA). The results will be plotted as a volcano plot and principal component analysis (PCA).

Data: To obtain the data used for this project, txt files were obtained from Gene Expression Omnibus (GEO) from an experiment that involved performing whole-genome sequencing and RNA-sequencing on the different developmental stages of *T. melanosporum* (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE49700). The specific file that was used is named "GSE49700_FB_FLM.RNA.txt" and it lists values of each gene involved that represent their level of expression. The second set of data files used was obtained from the NCBI database (https://www.ncbi.nlm.nih.gov/nuccore/FN430075). Here specifically, a FASTA file of the truffle's nucleotide sequence was downloaded.

Loading in Packages

Package definitions:

1. BiocManager: This package is from Bioconductor and would be used for RNAseq. Packages from Bioconductor are generally used to perform statistical analyses related to gene expression.
2. DESeq2: This package is also part of Bioconductor, and it performs differential gene expression analysis for the RNAseq analysis method.
3. coRdon: This package is used for codon usage (CU) analysis & can predict gene expressivity in DNA sequences using calculated ratios of CU bias.
4. seqinr: This package is used to analyze data for DNA and protein sequences. For this project, it is used to read in FASTA files.
5. Biostrings: This package is part of Bioconductor and can analyze DNA and protein sequences using algorithms. Here, it's used to load the truffle's DNA sequence after the FASTA file has been read in.
6. msa: The msa package performs multiple sequence alignment using algorithms, which can then show how similar or different DNA or protein sequences are compared to each other.
7. ggplot2: This package is from CRAN and can be used to visualize data by creating plots and graphics in general based on the data. In this project, it is used in the volcano plot method.
8. devtools: The devtools package is part of CRAN and contains various package development tools that simplify functions in R. Here, this package is used to perform PCA.
9. ggbiplot: This package is involved in plotting PCA results. It includes more features in addition to plotting points; you can also label the groups and visualize correlations in the PCA plot.

Hide

```
#Install/load packages
library(BiocManager)
library(DESeq2)
library(coRdon)
library(seqinr)
library(Biostrings)
library(msa)
library(ggplot2)
library(devtools)
library(ggbiplot)
```

Performing Bioinformatics Analysis

RSCU analysis

Description: Relative synonymous codon usage (RSCU) analysis measures codon usage bias, which is when genes can be regulated to have certain preferences for specific codons for translation. This type of bioinformatics method can help us predict the relative gene expression level of T. melanosporum's protein. We can use this to make conclusions about whether or not there are certain sequences in the gene that may have a greater contribution to the truffle's aroma based on the codons that appear to have a higher preference. The value that results from this analysis represents the ratio of observed and expected codon frequency. The website that was used as reference for the code was https://www.bioconductor.org/packages/devel/bioc/vignettes/coRdon/inst/doc/coRdon.html.

Hide

```
truffle_sequence <- read.fasta(file = "protein1.fasta")
truffle_sequence
```

```
$`sp|B6VP39.1|4FTAS_STRCT`
  [1] "m" "p" "s" "s" "v" "n" "r" "t" "s" "r" "t" "e" "p" "a" "g" "h" "h" "r" "e" "f" "p"
 [22] "l" "s" "l" "a" "a" "i" "d" "e" "l" "v" "a" "e" "e" "e" "a" "e" "d" "a" "r" "v" "l"
 [43] "h" "l" "t" "a" "n" "e" "t" "v" "l" "s" "p" "r" "a" "r" "a" "v" "l" "a" "s" "p" "l"
 [64] "t" "s" "r" "y" "l" "l" "e" "h" "l" "d" "m" "r" "g" "p" "s" "p" "a" "r" "l" "g" "n"
 [85] "l" "l" "l" "r" "g" "l" "d" "r" "i" "g" "t" "i" "e" "e" "s" "a" "t" "e" "v" "c" "r"
[106] "r" "l" "f" "g" "a" "r" "y" "a" "e" "f" "r" "c" "l" "s" "g" "l" "h" "a" "m" "q" "t"
[127] "t" "f" "a" "a" "l" "s" "r" "p" "g" "d" "t" "v" "m" "r" "v" "a" "t" "k" "d" "g" "g"
[148] "h" "f" "l" "t" "e" "l" "i" "c" "r" "s" "f" "g" "r" "r" "s" "c" "t" "y" "v" "f" "d"
[169] "d" "t" "m" "t" "i" "d" "l" "e" "r" "t" "r" "e" "v" "v" "e" "k" "e" "r" "p" "s" "l"
[190] "l" "f" "v" "d" "a" "m" "n" "y" "l" "f" "p" "f" "p" "i" "a" "e" "l" "k" "a" "i" "a"
[211] "g" "d" "v" "p" "l" "v" "f" "d" "a" "s" "h" "t" "l" "g" "l" "i" "a" "g" "g" "r" "f"
[232] "q" "d" "p" "l" "r" "e" "g" "a" "d" "l" "l" "q" "a" "n" "t" "h" "k" "t" "f" "f" "g"
[253] "p" "q" "k" "g" "i" "i" "l" "g" "n" "d" "r" "s" "l" "m" "e" "e" "l" "g" "y" "t" "l"
[274] "s" "t" "g" "m" "v" "s" "s" "q" "h" "t" "a" "s" "t" "v" "a" "l" "l" "i" "a" "l" "h"
[295] "e" "m" "w" "y" "d" "g" "r" "e" "y" "a" "a" "q" "v" "i" "d" "n" "a" "r" "r" "l" "a"
[316] "g" "a" "l" "r" "d" "r" "g" "v" "p" "v" "v" "a" "e" "e" "r" "g" "f" "t" "a" "n" "h"
[337] "m" "f" "f" "v" "d" "t" "r" "p" "l" "g" "s" "g" "p" "a" "v" "i" "q" "r" "l" "v" "r"
[358] "a" "g" "v" "s" "a" "n" "r" "a" "v" "a" "f" "n" "h" "l" "d" "t" "i" "r" "f" "g" "v"
[379] "q" "e" "i" "t" "r" "r" "g" "y" "d" "h" "d" "d" "l" "d" "e" "a" "a" "d" "l" "v" "a"
[400] "a" "v" "l" "l" "e" "r" "q" "e" "p" "e" "r" "i" "r" "p" "r" "v" "a" "e" "l" "v" "g"
[421] "r" "r" "r" "t" "v" "r" "y" "t" "g" "d" "p" "a" "s" "a" "a" "g" "p" "p" "a" "r" "e"
[442] "r" "y" "a" "p" "p" "t" "a" "p" "a" "g" "h" "p" "a" "r" "p" "r" "w" "i" "g" "v" "r"
[463] "l" "t" "p" "l" "p" "e" "p" "v" "t" "e" "a" "e" "c" "a" "g" "a" "q" "r" "l" "g" "r"
[484] "l" "a" "g" "a" "f" "p" "h" "q" "i" "d" "s" "s" "g" "n" "v" "s" "f" "t" "s" "t" "d"
[505] "g" "r" "l" "f" "v" "t" "g" "s" "g" "t" "y" "i" "k" "d" "l" "a" "p" "g" "d" "f" "v"
[526] "e" "l" "t" "g" "a" "e" "g" "w" "t" "l" "h" "c" "r" "g" "d" "g" "p" "p" "s" "a" "e"
[547] "a" "y" "l" "h" "h" "l" "l" "r" "e" "r" "v" "g" "a" "r" "y" "v" "v" "h" "n" "h" "c"
[568] "i" "p" "g" "r" "a" "l" "e" "t" "s" "g" "a" "l" "v" "i" "p" "p" "k" "e" "y" "g" "s"
[589] "v" "a" "l" "a" "e" "a" "v" "a" "d" "a" "c" "q" "d" "s" "q" "v" "m" "y" "v" "r" "r"
[610] "h" "g" "l" "v" "f" "w" "a" "h" "s" "y" "d" "e" "c" "l" "a" "l" "i" "e" "d" "v" "r"
[631] "r" "i" "t" "g"
attr(,"name")
[1] "sp|B6VP39.1|4FTAS_STRCT"
attr(,"Annot")
[1] ">sp|B6VP39.1|4FTAS_STRCT RecName: Full=Fluorothreonine transaldolase; AltName: Full=4-fluorothreonine transa
ldolase; Short=4-FTase"
attr(,"class")
[1] "SeqFastadna"
```

Hide

```
#Load truffle sequence
truffle_codon <- readSet(file = "sequence.fasta")
truffle_codon_table <- codonTable(truffle_codon)
```

```
Warning in codonTable(truffle_codon) :

Length of sequence(s) at the following postion(s) is not divisible by 3:
1.
Discarding surplus nucleotides.
```

Hide

```
#Read codon counts
cc <- codonCounts(truffle_codon_table)
head(cc)
```

```
     AAA  AAC  AAG  AAT  ACA  ACC  ACG  ACT  AGA  AGC  AGG  AGT  ATA  ATC  ATG  ATT  CAA
ctb 9407 6217 7602 7234 6744 6077 2819 6366 7638 5032 6734 6386 7548 6315 6881 7290 7903
     CAC  CAG  CAT  CCA  CCC  CCG  CCT  CGA  CGC  CGG  CGT  CTA  CTC  CTG  CTT  GAA  GAC
ctb 5172 6018 6785 7158 5917 3224 6899 4238 2798 3380 3036 5584 6675 6120 8024 8026 4543
     GAG  GAT  GCA  GCC  GCG  GCT  GGA  GGC  GGG  GGT  GTA  GTC  GTG  GTT  TAA  TAC  TAG
ctb 6603 6699 5355 4411 2667 5125 6964 4590 5607 6197 5783 4572 5221 6478 5739 6048 5357
     TAT  TCA  TCC  TCG  TCT  TGA  TGC  TGG  TGT  TTA  TTC  TTG  TTT
ctb 7627 6937 6688 4023 7698 7290 5243 7130 6922 5541 7803 7891 9576
```

Hide

```
#Calculate CU bias
milc <- MILC(truffle_codon_table, ribosomal = TRUE)
head(milc)
```

```
     self
0.4998902
```

RNAseq

Description: The RNAseq method, or differential expression analysis, can help with analyzing data regarding the gene expression levels of certain proteins. In this case, it can show how much RNA is being expressed at the 2 developmental stages that I will be analyzing for the various proteins that are involved with the black truffle's volatile organic compounds and therefore its aroma. The example code is referenced from https://www.bioconductor.org/packages/devel/bioc/vignettes/coRdon/inst/doc/coRdon.html and the DataCamp 4 Answer Key from BIMM 143's Canvas.

Hide

```
#Read in truffle FB and FLM file
truffle_dataframe <- read.delim("GSE49700_FB_FLM.RNA.txt")
truffle_dataframe
```

gene <chr>	scaffold <chr>	length <int>	FB <int>	FLM <int>	RPKM_FB <dbl>	RPKM_FLM <dbl>	log2ratio <dbl>
GSTUMT00000002001	scaffold_101	2228	2383	1267	1.443237e+01	8.921426e+00	-0.693899768
GSTUMT00000003001	scaffold_101	1411	1264	1535	1.208784e+01	1.706688e+01	0.497608291
GSTUMT00000004001	scaffold_101	3039	3609	2449	1.602451e+01	1.264244e+01	-0.341981370
GSTUMT00000005001	scaffold_101	670	3393	3441	6.833414e+01	8.057174e+01	0.237664171
GSTUMT00000006001	scaffold_101	1658	5162	1512	4.201091e+01	1.430672e+01	-1.554004590
GSTUMT00000007001	scaffold_101	1383	4810	4678	4.693011e+01	5.306533e+01	0.177252330
GSTUMT00000009001	scaffold_101	81	0	0	0.000000e+00	0.000000e+00	0.000000000
GSTUMT00000012001	scaffold_101	988	3820	746	5.217171e+01	1.184554e+01	-2.138830052
GSTUMT00000013001	scaffold_101	846	102	224	1.626891e+00	4.153847e+00	1.351791262
GSTUMT00000014001	scaffold_101	2404	2589	3390	1.453203e+01	2.212270e+01	0.606257162

1-10 of 7,496 rows | 1-8 of 10 columns Previous 1 2 3 4 5 6 ... 100 Next

Hide

```
#Raw counts
truffle_subset <- subset(truffle_dataframe, select = c(gene, FB, FLM, RPKM_FB, RPKM_FLM))
truffle_subset
```

	gene <chr>	FB <int>	FLM <int>	RPKM_FB <dbl>	RPKM_FLM <dbl>
1	GSTUMT00000002001	2383	1267	1.443237e+01	8.921426e+00
2	GSTUMT00000003001	1264	1535	1.208784e+01	1.706688e+01
3	GSTUMT00000004001	3609	2449	1.602451e+01	1.264244e+01
4	GSTUMT00000005001	3393	3441	6.833414e+01	8.057174e+01
5	GSTUMT00000006001	5162	1512	4.201091e+01	1.430672e+01
6	GSTUMT00000007001	4810	4678	4.693011e+01	5.306533e+01
7	GSTUMT00000009001	0	0	0.000000e+00	0.000000e+00
8	GSTUMT00000012001	3820	746	5.217171e+01	1.184554e+01
9	GSTUMT00000013001	102	224	1.626891e+00	4.153847e+00
10	GSTUMT00000014001	2589	3390	1.453203e+01	2.212270e+01

1-10 of 7,496 rows Previous 1 2 3 4 5 6 ... 100 Next

Hide

```
#extract metadata columns by extracting column names into a vector
row_name <- c(names(truffle_dataframe))
print(row_name)
```

```
 [1] "gene"      "scaffold"  "length"    "FB"        "FLM"       "RPKM_FB"   "RPKM_FLM"
 [8] "log2ratio" "pval"      "padj"
```

Hide

```
#combine into metadata columns to describe condition of columns
FB_data <- c("FB", "FB")
FLM_data <- "FLM"

metadata_matrix <- data.frame(FB_data, FLM_data)
#rownames(metadata_matrix) <- row_name
print(metadata_matrix)
```

FB_data <chr>	FLM_data <chr>
FB	FLM
FB	FLM

2 rows

Hide

```
#Create matrix from truffle FB/FLM data
full_matrix <- as.matrix(truffle_dataframe)
head(full_matrix)
```

```
     gene                 scaffold       length  FB       FLM      RPKM_FB
[1,] "GSTUMT00000002001 " "scaffold_101" " 2228" "  2383" "  1267" "1.443237e+01"
[2,] "GSTUMT00000003001 " "scaffold_101" " 1411" "  1264" "  1535" "1.208784e+01"
[3,] "GSTUMT00000004001 " "scaffold_101" " 3039" "  3609" "  2449" "1.602451e+01"
[4,] "GSTUMT00000005001 " "scaffold_101" "  670" "  3393" "  3441" "6.833414e+01"
[5,] "GSTUMT00000006001 " "scaffold_101" " 1658" "  5162" "  1512" "4.201091e+01"
[6,] "GSTUMT00000007001 " "scaffold_101" " 1383" "  4810" "  4678" "4.693011e+01"
     RPKM_FLM       log2ratio       pval         padj
[1,] "8.921426e+00" " -0.693899768" "0.89149792" " 1"
[2,] "1.706688e+01" "  0.497608291" "0.85475586" " 1"
[3,] "1.264244e+01" " -0.341981370" "0.96650885" " 1"
[4,] "8.057174e+01" "  0.237664171" "0.90925283" " 1"
[5,] "1.430672e+01" " -1.554004590" "0.71533689" " 1"
[6,] "5.306533e+01" "  0.177252330" "0.92207869" " 1"
```

Hide

```
#Print structure of dataframe
print(str(truffle_dataframe))
```

```
'data.frame':   7496 obs. of  10 variables:
 $ gene     : chr  "GSTUMT00000002001 " "GSTUMT00000003001 " "GSTUMT00000004001 " "GSTUMT00000005001 " ...
 $ scaffold : chr  "scaffold_101" "scaffold_101" "scaffold_101" "scaffold_101" ...
 $ length   : int  2228 1411 3039 670 1658 1383 81 988 846 2404 ...
 $ FB       : int  2383 1264 3609 3393 5162 4810 0 3820 102 2589 ...
 $ FLM      : int  1267 1535 2449 3441 1512 4678 0 746 224 3390 ...
 $ RPKM_FB  : num  14.4 12.1 16 68.3 42 ...
 $ RPKM_FLM : num  8.92 17.07 12.64 80.57 14.31 ...
 $ log2ratio: num  -0.694 0.498 -0.342 0.238 -1.554 ...
 $ pval     : num  0.891 0.855 0.967 0.909 0.715 ...
 $ padj     : int  1 1 1 1 1 1 NA 1 1 1 ...
NULL
```

Hide

```
#Create metadata columns by extracting column names into a vector
gene_FB_FLM <- full_matrix[,c(4,5)]
```

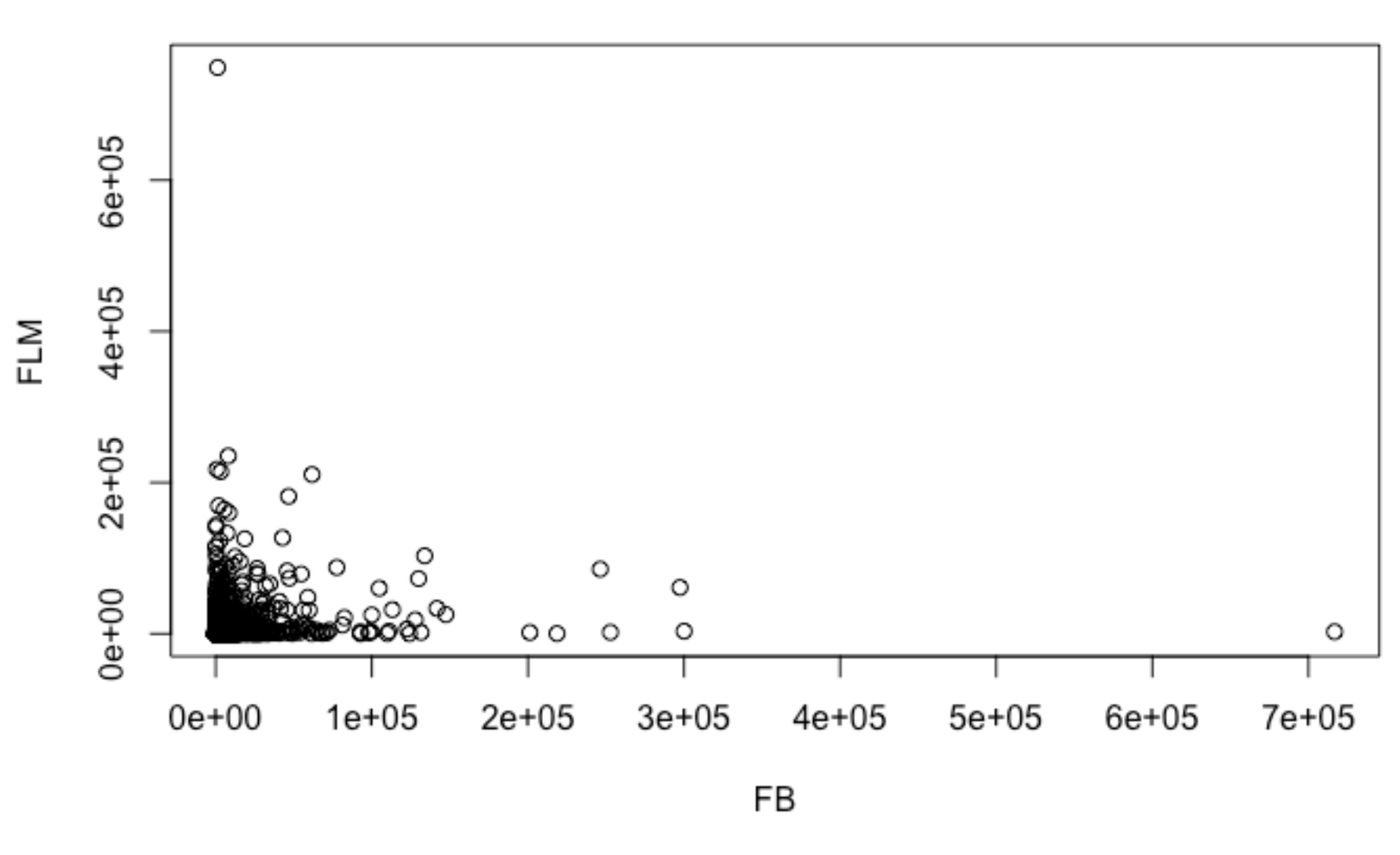
Hide

```
#Code for RNAseq

#Read in raw counts
#Create metadata
#Combine counts & metadata matrices
#Create DESeq object
dds <- DESeqDataSetFromMatrix(countData = _, colData = _, design = ~ condition)
```

```
Error: unexpected input in "dds <- DESeqDataSetFromMatrix(countData = _"
```

Hide

```
# Visualize FB gene expression vs. FLM gene expression
FB <- truffle_dataframe$FB
FLM <- truffle_dataframe$FLM

plot(x = FB, y = FLM)
```



Multiple sequence alignment (MSA)

Description: This bioinformatics method using an algorithm to align sequences, which can be used to compare the sequences for similarities in regions of the sequences. This can be helpful for finding similarities in regions of the sequences and identifying any significant differences.

Hide

```
#Read in sequences
library(seqinr)
truffle_18_sequence <- read.fasta("sequence.fasta")
```

Hide

```
#Perform MSA

#Read nucleotide FASTA file as AA sequence
protein_sequence <- readAAStringSet("sequence.fasta")
head(protein_sequence)
```

```
AAStringSet object of length 1:
      width seq                                                      names
[1] 1180472 TGTGACAGGGGCTGGTCAGCGAGCTTCT...AGTACGAGACTCTTAAGGCGGAGTGGTT FN430075.1 Tuber ...
```

Hide

```
#Run MSA
alignment <- msa(truffle_18_sequence)
```

```
Error in checkInputSeq(inputSeqs) : The parameter inputSeq is not valid!
Possible inputs are <character>, <XStringSet>, or a file.
```

Plotting the Results

Volcano plot

Description: This visualization method is a scatterplot that displays points representing the p-value (which represents statistical significance) along the y-axis and fold change (the magnitude of change) of the data along the x-axis. Here, the p-value and fold change values were already provided in the truffle FB & FLM dataset, so those values were plotted. The code was references from this website: https://biocorecrg.github.io/CRG_RIntroduction/volcano-plots.html.

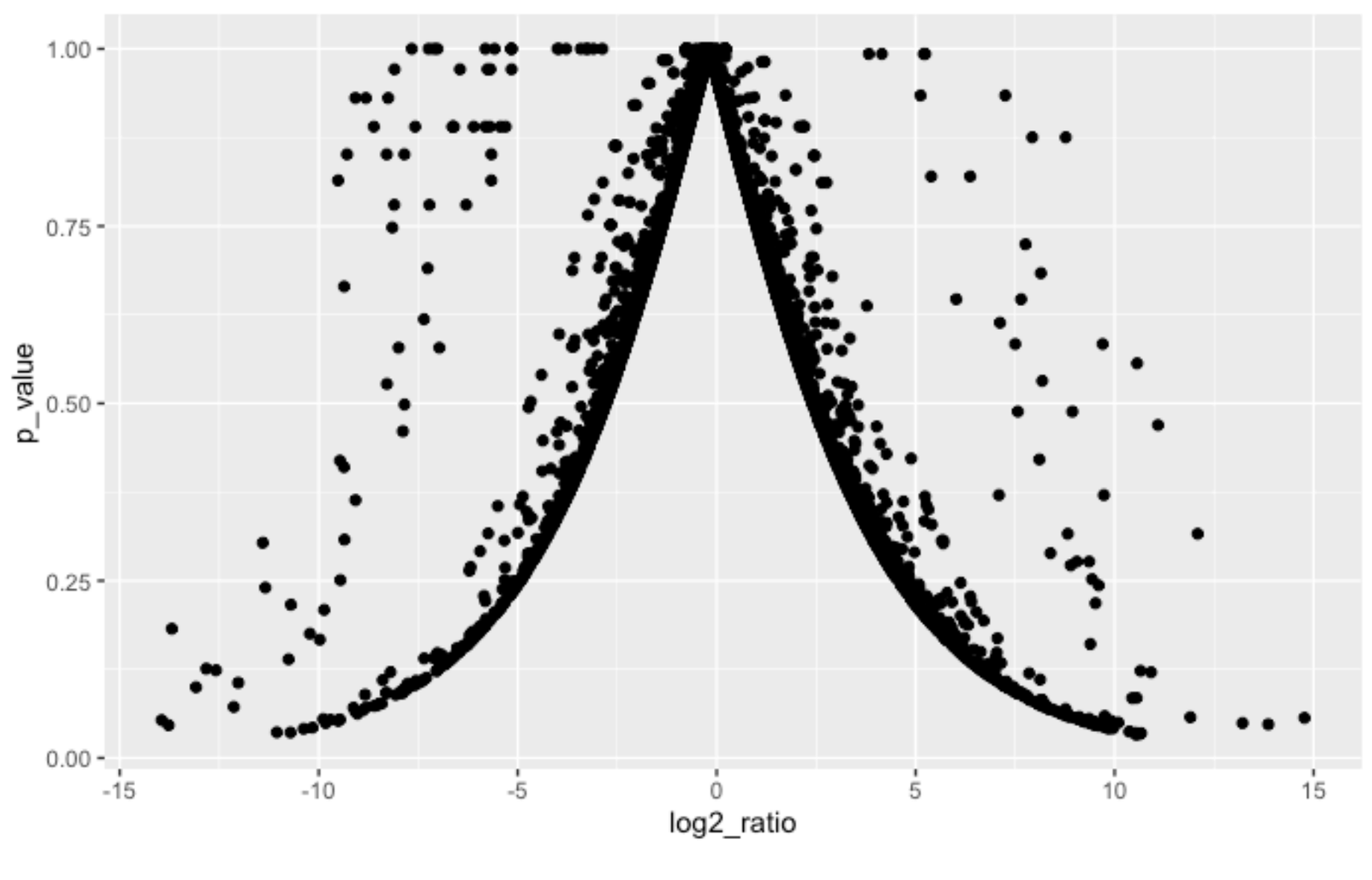
Hide

```
#Extract objects for volcano plot from truffle_dataframe
log2_ratio <- truffle_dataframe$log2ratio
p_value <- truffle_dataframe$pval
```

Hide

```
#Create plot
ggplot(data = truffle_dataframe, aes(x = log2_ratio, y = p_value)) + geom_point()
```

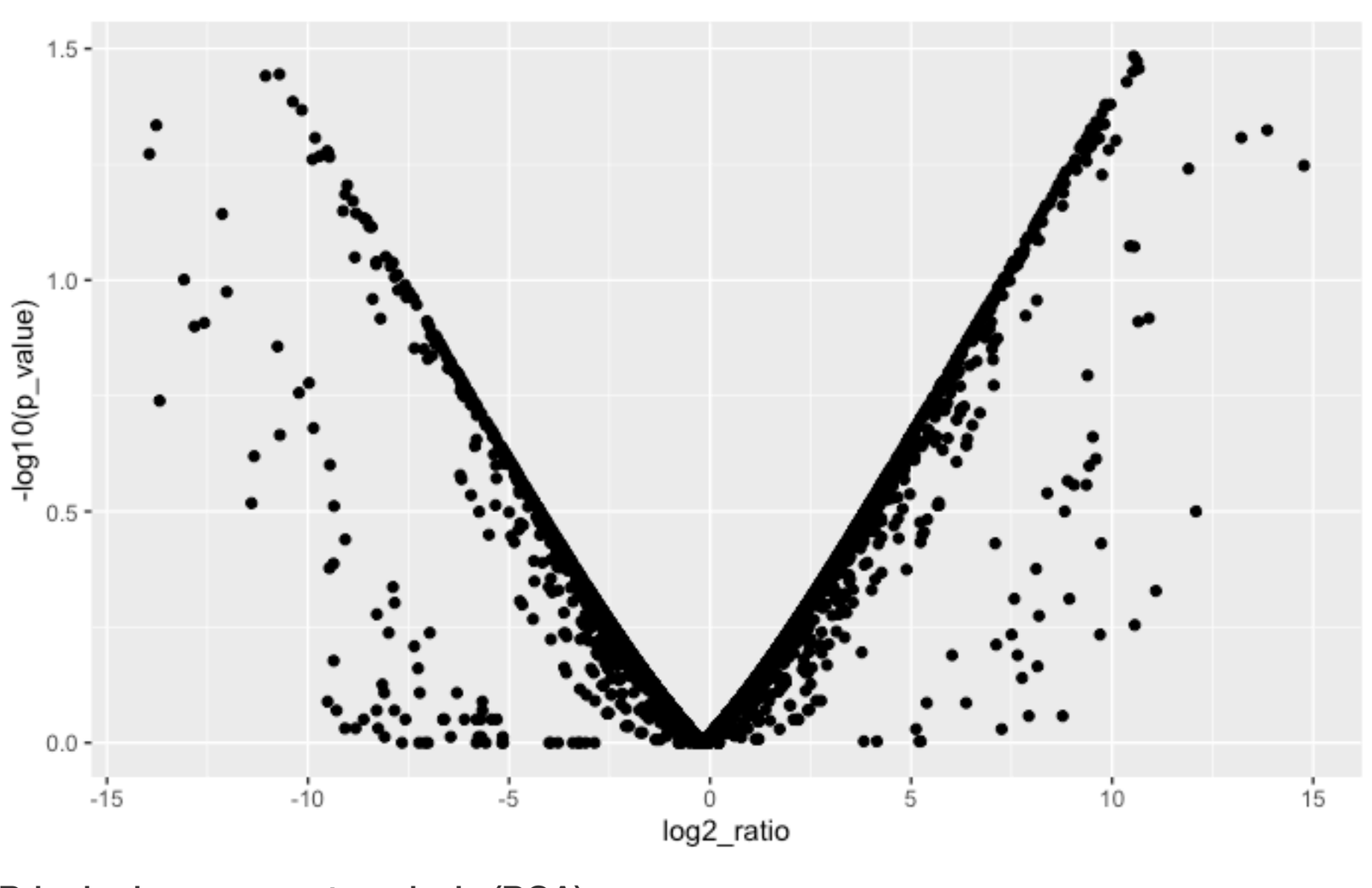
```
Warning: Removed 122 rows containing missing values (geom_point).
```



Hide

```
#Convert p-value to -log10(p-value) to get volcano plot to get a volcano plot
ggplot(data = truffle_dataframe, aes(x = log2_ratio, y = -log10(p_value))) + geom_point()
```

```
Warning: Removed 122 rows containing missing values (geom_point).
```



Principal component analysis (PCA)

Description: PCA is a bioinformatics visualization method that uses principal components to summarize a large amount of data into a smaller data set. This makes it easier to see and analyze the data and to draw comparisons between between various variables. The code used here was referenced from https://www.datacamp.com/tutorial/pca-analysis-r#simple.

Hide

```
#PCA code
truffle.pca <- prcomp(truffle_dataframe[,c(3,4,5,6,7)], center = TRUE, scale = TRUE)
summary(truffle.pca)
```

```
Importance of components:
                          PC1    PC2    PC3     PC4    PC5
Standard deviation     1.5007 1.2327 1.0047 0.37227 0.2837
Proportion of Variance 0.4504 0.3039 0.2019 0.02772 0.0161
Cumulative Proportion  0.4504 0.7543 0.9562 0.98390 1.0000
```

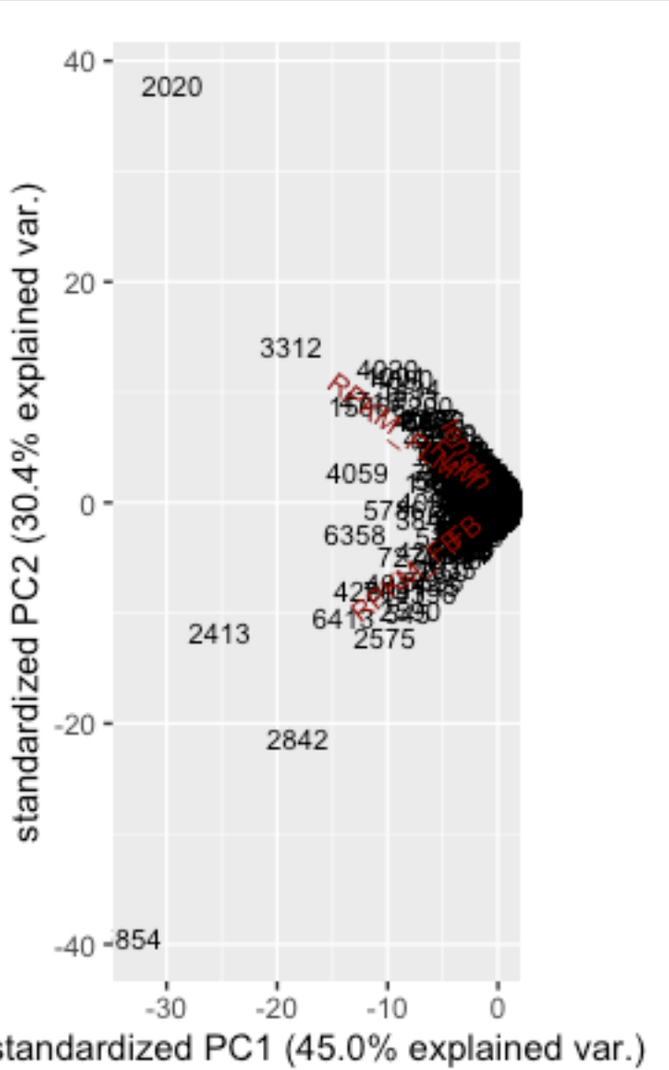
Hide

```
str(truffle.pca)
```

```
List of 5
 $ sdev    : num [1:5] 1.501 1.233 1.005 0.372 0.284
 $ rotation: num [1:5, 1:5] -0.0294 -0.5003 -0.4718 -0.5112 -0.5147 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:5] "length" "FB" "FLM" "RPKM_FB" ...
  .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
 $ center  : Named num [1:5] 1746.5 3121.3 3177.2 28.7 30.3
  ..- attr(*, "names")= chr [1:5] "length" "FB" "FLM" "RPKM_FB" ...
 $ scale   : Named num [1:5] 1185 13495 13667 157 126
  ..- attr(*, "names")= chr [1:5] "length" "FB" "FLM" "RPKM_FB" ...
 $ x       : num [1:7496, 1:5] 0.21489 0.24179 0.08814 -0.32738 0.00552 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
 - attr(*, "class")= chr "prcomp"
```

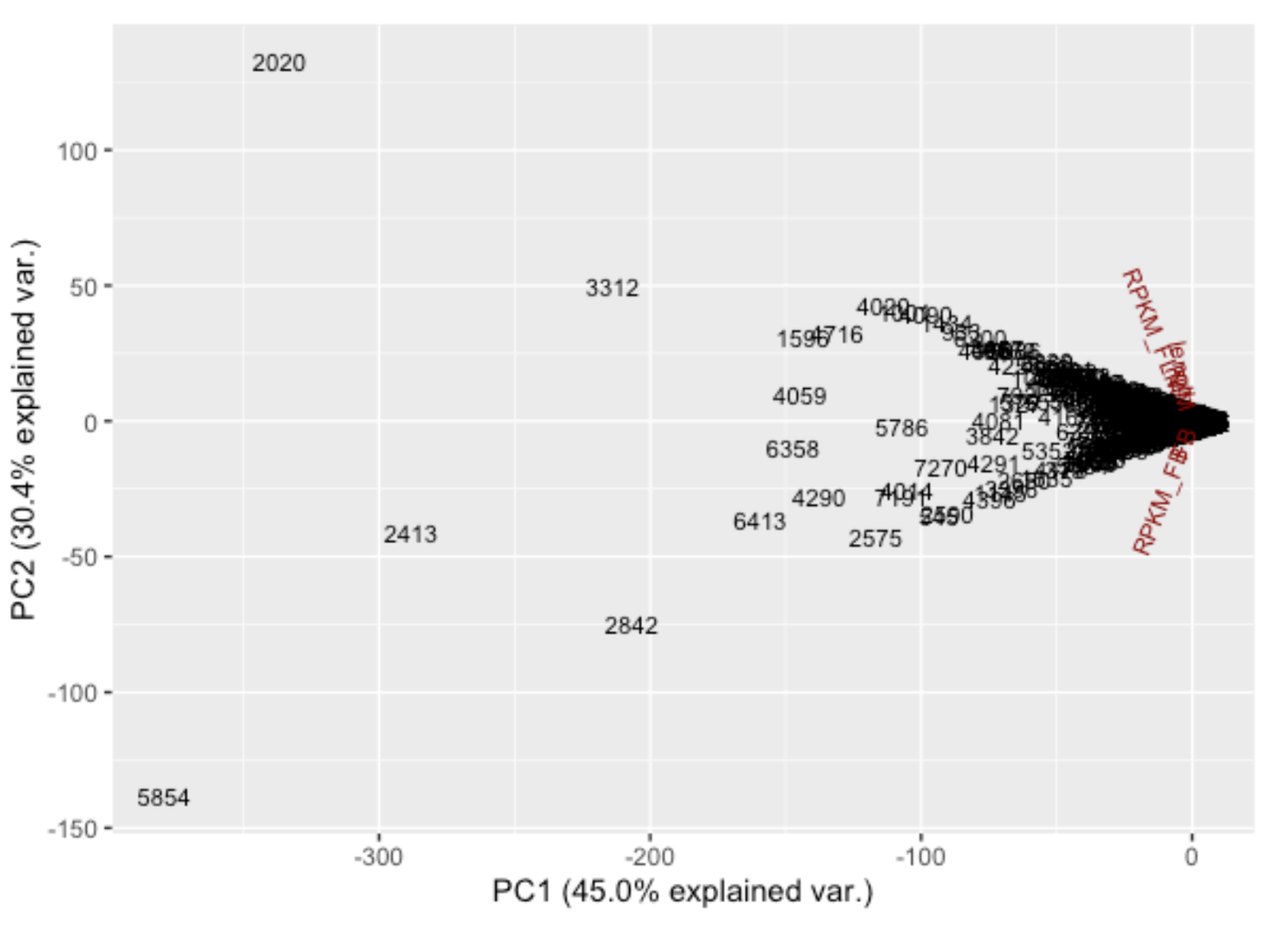
Hide

```
#plot PCA
ggbiplot(truffle.pca, labels = rownames(truffle_dataframe))
```



Hide

```
#Plot but scaled
ggbiplot(truffle.pca, labels = rownames(truffle_dataframe), scale = -5)
```



Analyzing the Results

The RSCU analysis results in a codon usage bias ratio of 0.4998902, which is the ratio that compares the observed to the expected codon frequency. The ratio indicates that the same codons are used almost half of the time in the T. melanosporum DNA sequence. This suggests that there could be certain regions in the sequence that are repeated and could contribute to specific characteristics of the truffle. For RNAseq, I wasn't able to get the code to work because I couldn't find any raw counts data files from any database. However, I used the gene expression reads provided by one of the data files I found (read in the "truffle_dataframe" variable), and I used those given values to make a plot to see how the expression levels in the FB stage compared to the FLM stage for each gene. It appears that they mostly have similar levels of expression, as the data points are clustered towards the lower left corner of the plot. For MSA, I also couldn't find data files from any database that would help me compare the truffle's gene sequence between different developmental stages. But if I were to perform MSA, the resulting alignment would reveal differences between the sequences if there are any present. For plotting the results, the volcano plot displays the relationship between the p-values and fold changes that were provided by one of the data files. According to the plot, the "V"-shape appears to be fairly spread out, which suggests that there isn't a strong correlation between the 2 variables. From the PCA plot, the results show that there was a variance of 45.0% for PC1 and 30.4% for PC2. The data points are clustered towards 0 of the x-axis, fanning out towards the left. The variables that were compared here were gene expression levels and length of the gene, so it can be concluded that there is not much similarity between these variables due to the relatively low percentages.