# Jackson Laboratory coding assessment

Sanjana Gorlla

2022-10-21

## TASKS

*1. Please make barplots for each gene (row) and plot them in one pdf file (you can find the way it needs to be done in the attached pdf).*

*2. It doesn't have to look exactly the same, this is just an example. But it would be better if the plotted values are sorted by groups (mock – WT – delta – SA).*

## SOLUTION

```
#loading required libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(ggplot2)
library(RColorBrewer)
```

## Loading the dataset

- read.delim() - for a text file

```
# Importing the data from text file
data <- read.delim("/Users/sanjanagorlla/Desktop/my projects/gayathri/test_R_10182022.txt")
```

## Exploring the dataset

1. Displaying values: head() and tail() - displays first 6 values and last 6 values
2. Evaluating the dimensions : dim() - Gives count of rows and columns
3. Checking the column names : colnames() - displays the names of each columns in the dataset

```
## 1. Exploring the dataset
head(data)
```

```
##           gene_id gene_name log2FoldChange     lfcSE      stat       pvalue
## 1 ENSG00000005108    THSD7A      1.6491466 0.3313305  4.977346 6.450000e-07
## 2 ENSG00000007171      NOS2     -0.8445846 0.5008484 -1.686308 9.173651e-02
## 3 ENSG00000007952      NOX1     -1.4880897 0.2168323 -6.862861 6.750000e-12
## 4 ENSG00000023445     BIRC3     -1.4455281 0.2535667 -5.700780 1.190000e-08
## 5 ENSG00000026950    BTN3A1     -0.9725725 0.1856663 -5.238281 1.620000e-07
## 6 ENSG00000029534      ANK1     -1.4670782 0.3400107 -4.314801 1.600000e-05
##           padj      mock   SA.1dpi   SA.2dpi   SA.3dpi   SA.4dpi   SA.5dpi
## 1 2.430000e-05  6.319705  4.612007  5.847734  5.161231  4.826943  6.650365
## 2 2.103687e-01  8.507822  4.526623  6.237764  6.684862  8.253633  8.293892
## 3 4.190000e-09  4.541318  5.243995  7.155892  6.191989  6.536108  7.027126
## 4 1.090000e-06 10.904035 12.391733 12.567441 12.078737 11.610626 13.923424
## 5 8.550000e-06 10.509430 11.260642 11.296825 11.522431 10.618283 12.152273
## 6 2.982180e-04  4.411515  4.612521  7.230741  7.032614  7.464113  9.012762
##     SA.6dpi   WT.1dpi   WT.2dpi   WT.3dpi   WT.4dpi   WT.5dpi   WT.6dpi
## 1  3.743060  4.657728  6.039845  5.548831  4.892006  5.450696  7.069145
## 2  8.043637  3.107852  6.786949  7.944369  8.539625  7.084391  7.472533
## 3  7.387362  5.062840  6.514285  6.317048  6.511500  6.917880  6.101796
## 4 14.287703 13.533894 13.238614 13.041324 12.286468 12.195527 13.567289
## 5 12.874917 11.477572 11.322574 11.293434 10.716591 10.355934 11.934040
## 6  8.968024  5.089663  7.837964  7.487751  7.557942  7.749232  7.943019
##   delta.1dpi delta.2dpi delta.3dpi delta.4dpi delta.5dpi delta.6dpi    mock.1
## 1   2.252859   4.963272   3.974997   5.343761   5.712090   5.259367  5.322849
## 2   4.283263   8.218424   8.318671   9.470920   8.623934   8.626607  8.086876
## 3   5.821705   7.885800   8.131115   7.968231   8.604869   7.632375  4.267958
## 4  13.270931  13.772943  13.808790  14.666180  15.844019  15.740498 11.033612
## 5  12.129613  11.787455  12.072206  13.236500  13.581055  13.577014 10.736909
## 6   7.480097   9.227157   8.255055   8.947177   9.489890   9.502637  5.615292
##   WT.1dpi.1 WT.2dpi.1 WT.3dpi.1 WT.4dpi.1 WT.5dpi.1 WT.6dpi.1 delta.3dpi.1
## 1  4.632090  5.050575  5.618554  7.283396  6.977791  6.145189     3.192305
## 2  4.468546  6.121611  7.363864  9.205262  7.163394  6.301495     9.498788
## 3  5.765481  7.069524  7.064838  6.945025  7.069299  7.067498     7.381348
## 4 12.668607 12.592365 12.341970 13.173655 14.143776 14.699631    13.240614
## 5 11.690312 11.576448 11.043384 11.591387 12.355550 12.901616    11.546382
## 6  9.340853  7.843515  7.733669  8.275166  8.844348  8.704938     8.618065
```

```
##   delta.4dpi.1 delta.5dpi.1 delta.6dpi.1    mock.2 delta.3dpi.2 delta.4dpi.2
## 1     4.756490     2.792960     3.406367  4.045984     2.903663     2.844605
## 2     4.008220     2.746298     8.939120  5.944187     7.246122     8.849181
## 3     7.425367     7.745312     7.878059  5.958884     7.556359     8.106034
## 4    13.702179    12.269149    14.232158 12.778121    12.978639    13.716507
## 5    11.892956    11.362925    13.396124 10.719719    11.068049    11.599127
## 6     9.880362    10.843059    10.045459  9.264298     7.262565    10.080268
##   delta.5dpi.2 delta.6dpi.2    mock.3
## 1     5.184874     3.996630  5.961407
## 2    11.417139    10.927880  8.936221
## 3     8.172117     6.992556  4.710046
## 4    14.481782    13.795091 10.910219
## 5    13.476895    13.463633 10.713588
## 6     9.197276     8.507685  4.033633
```

```
tail(data)
```

```
##            gene_id gene_name log2FoldChange     lfcSE      stat       pvalue
## 45 ENSG00000130487   KLHDC7B     -2.0974649 0.4226804 -4.962295 6.970000e-07
## 46 ENSG00000133328   HRASLS2     -1.6521805 0.3136189 -5.268115 1.380000e-07
## 47 ENSG00000134339      SAA2     -2.1275035 0.2402513 -8.855326 8.340000e-19
## 48 ENSG00000134532      SOX5      1.2285540 0.2284885  5.376876 7.580000e-08
## 49 ENSG00000136155      SCEL     -0.5072893 0.2172686 -2.334849 1.955131e-02
## 50 ENSG00000136872     ALDOB     -1.3375633 0.1960647 -6.822051 8.970000e-12
##             padj      mock   SA.1dpi   SA.2dpi   SA.3dpi   SA.4dpi   SA.5dpi
## 45 2.560000e-05  3.855660  6.051544  8.261620  7.663694  7.600280 10.825845
## 46 7.590000e-06  9.098714 12.083589 10.550501 10.165891 10.118433 13.497416
## 47 8.800000e-15  9.991672 13.260140 12.395962 12.826445 13.057030 14.183210
## 48 4.800000e-06 10.577872  8.425994  8.032489  9.011597  9.031590  8.147309
## 49 6.988054e-02 11.269139 10.294914 12.204504 11.633353 11.268594 10.755082
## 50 4.920000e-09  4.836280  6.272361  5.637342  5.593591  5.081141  5.858610
##      SA.6dpi   WT.1dpi   WT.2dpi   WT.3dpi   WT.4dpi   WT.5dpi   WT.6dpi
## 45 10.852622  5.945161  8.480294  7.686723  6.999482  7.554596  9.524448
## 46 14.146756 12.110300 11.011030 10.984521 10.622754 10.701291 12.962407
## 47 14.911123 13.625131 13.886249 14.131226 12.959700 13.590858 13.551398
## 48  8.017651  8.212468  7.724054  8.730696  9.289458  8.705446  9.141045
## 49 11.280813 11.249565 11.608141 11.793120 10.610960 10.907935 11.136502
## 50  6.221889  4.084270  5.633192  5.754486  4.634724  5.409601  5.733729
##    delta.1dpi delta.2dpi delta.3dpi delta.4dpi delta.5dpi delta.6dpi    mock.1
## 45   7.385332   9.402750  10.856842  12.416864  12.607173  12.319994  4.695869
## 46  13.040079  11.429744  13.379653  14.537524  15.101396  14.896054 10.011942
## 47  15.170972  14.980769  15.791035  15.688696  16.003540  16.118878 10.366956
## 48   6.362564   7.063906   8.390592   8.350084   7.566921   6.969231 10.359634
## 49  10.536231  11.787182  10.712570  11.162140  12.725175  12.425460 10.931402
## 50   5.373232   6.929321   5.680939   6.764557   7.507909   7.300729  3.625483
##    WT.1dpi.1 WT.2dpi.1 WT.3dpi.1 WT.4dpi.1 WT.5dpi.1 WT.6dpi.1 delta.3dpi.1
## 45  8.502148  8.810441  9.309306 10.655489 11.384670 11.077464     9.033251
## 46 11.624423 11.251624 11.552175 11.937660 13.620045 14.000334    14.029854
## 47 13.370867 12.913518 13.713724 13.989366 14.315604 14.514458    15.831078
## 48  7.032720  7.991414  8.930263  8.803211  8.422089  8.336163     7.779620
## 49 12.213747 12.025204 10.657247 10.861935 11.335556 11.956632    10.777446
## 50  5.731567  4.854604  5.692246  5.575940  5.669159  6.391987     5.700085
##    delta.4dpi.1 delta.5dpi.1 delta.6dpi.1    mock.2 delta.3dpi.2 delta.4dpi.2
## 45     9.716705    11.193496    11.710300  6.753930     8.788412    10.727363
```

```
## 46    11.986112    11.904405    14.019862 11.069771    10.989680    11.763686
## 47    14.543024    16.692777    15.415667 12.880944    14.787382    14.773727
## 48     7.811005     5.136820     6.633603  8.605037     9.258479     6.319789
## 49    12.318884    11.290444    13.643332 11.587527    10.596286    11.470359
## 50     5.811276     6.955968     6.576872  5.092188     5.027382     6.009547
##    delta.5dpi.2 delta.6dpi.2    mock.3
## 45    12.600008    11.699959  4.649869
## 46    14.232437    13.791577  8.770908
## 47    15.583856    15.455148 10.974630
## 48     7.881874     7.736876 10.559813
## 49    12.287132    12.241170 11.477309
## 50     7.209126     6.944196  2.511962
```

```
## 2.Dimension
dim(data)# 50 rows and 43 columns
```

```
## [1] 50 43
```

```
# Therefore, there are 50 genes in the datatset
## 3. Displaying column names
colnames(data)
```

```
##  [1] "gene_id"        "gene_name"      "log2FoldChange" "lfcSE"
##  [5] "stat"           "pvalue"         "padj"           "mock"
##  [9] "SA.1dpi"        "SA.2dpi"        "SA.3dpi"        "SA.4dpi"
## [13] "SA.5dpi"        "SA.6dpi"        "WT.1dpi"        "WT.2dpi"
## [17] "WT.3dpi"        "WT.4dpi"        "WT.5dpi"        "WT.6dpi"
## [21] "delta.1dpi"     "delta.2dpi"     "delta.3dpi"     "delta.4dpi"
## [25] "delta.5dpi"     "delta.6dpi"     "mock.1"         "WT.1dpi.1"
## [29] "WT.2dpi.1"      "WT.3dpi.1"      "WT.4dpi.1"      "WT.5dpi.1"
## [33] "WT.6dpi.1"      "delta.3dpi.1"   "delta.4dpi.1"   "delta.5dpi.1"
## [37] "delta.6dpi.1"   "mock.2"         "delta.3dpi.2"   "delta.4dpi.2"
## [41] "delta.5dpi.2"   "delta.6dpi.2"   "mock.3"
```

```
# The data set has 50 genes along with their ID's,
#gene name, log2Fold changes, pvalue, padj along with mock - WT - delta - SA groups
```

## Dropping the unwanted columns for the analysis

1. Dropping the values: select() - deselected unwanted variables and displaying the gene names

```
# Dropping columns which are unwanted for the analysis
df <-select (data,-c(gene_name, gene_id, log2FoldChange, lfcSE, stat, pvalue, padj))
# creating a vector for gene names
gene_name <- data.frame(select (data,gene_name))
gene_name
```

```
##    gene_name
## 1     THSD7A
## 2       NOS2
```

```
## 3       NOX1
## 4      BIRC3
## 5     BTN3A1
## 6       ANK1
## 7     NPFFR2
## 8        ME1
## 9       PYGM
## 10    GUCY2C
## 11     GSDMB
## 12     SP140
## 13      LAG3
## 14     IRAK3
## 15   SLC26A4
## 16      VNN3
## 17   SLC52A3
## 18     RENBP
## 19     TIMP1
## 20  TNFSF13B
## 21      CRYM
## 22      LFNG
## 23   CYP2C18
## 24    SLC1A2
## 25    BTN3A3
## 26     ADTRP
## 27      LIFR
## 28    LRRC31
## 29    IL18R1
## 30     PLCL1
## 31      KYNU
## 32    KIF21B
## 33    ATP10B
## 34     IFIT3
## 35     IFIT2
## 36   BCL2L14
## 37   TNFSF10
## 38      ZBP1
## 39     BEST3
## 40    STEAP4
## 41      ADM2
## 42     MGAT3
## 43    STRIP2
## 44      ACE2
## 45   KLHDC7B
## 46   HRASLS2
## 47      SAA2
## 48      SOX5
## 49      SCEL
## 50     ALDOB
```

## Transposing the rows(genes) and columns(groups)

1. Transposing: transpose() - transpose the gene names with the columns(groups) for further analysis The gene names on the column makes it easier to plot the graph

```
# transpose the gene names with the columns for further analysis
# This transpose will help to plot each gene
#according to sort it out by groups (mock - WT - delta - SA)
df_t <- transpose(df)
#redefine row and column names
rownames(df_t) <- colnames(df)
colnames(df_t) <- rownames(df)
# rownames
names <- rownames(df_t)
# The goups
names
```

```
##  [1] "mock"         "SA.1dpi"      "SA.2dpi"      "SA.3dpi"      "SA.4dpi"
##  [6] "SA.5dpi"      "SA.6dpi"      "WT.1dpi"      "WT.2dpi"      "WT.3dpi"
## [11] "WT.4dpi"      "WT.5dpi"      "WT.6dpi"      "delta.1dpi"   "delta.2dpi"
## [16] "delta.3dpi"   "delta.4dpi"   "delta.5dpi"   "delta.6dpi"   "mock.1"
## [21] "WT.1dpi.1"    "WT.2dpi.1"    "WT.3dpi.1"    "WT.4dpi.1"    "WT.5dpi.1"
## [26] "WT.6dpi.1"    "delta.3dpi.1" "delta.4dpi.1" "delta.5dpi.1" "delta.6dpi.1"
## [31] "mock.2"       "delta.3dpi.2" "delta.4dpi.2" "delta.5dpi.2" "delta.6dpi.2"
## [36] "mock.3"
```

## Assigning the column names and row names to the transposed dataframe

```
rownames(df_t) <- NULL # assigning the null values to the row values
newdf <- df_t
# assigning column names and row names
rownames(gene_name) <- data[,"gene_name"]
colnames(df_t) <- row.names(gene_name)
#gene names
df_t$name <- names
# final dataframe
final_df <- df_t %>%
  select(name, everything())
```

## Sorting the group names based on alphabetical order

```
# sorting the values and creating new data frame
final_new <- final_df[order(final_df$name),]
# creating a new column  for Source
final_new$Source <- c(replicate(14,"delta"),
                      replicate(4,"mock"), replicate(6,"SA"), replicate(12,"WT"))
#assigning row names after sorting
rownames(final_new) <- NULL
rownames_finaldf <- c(1:36)
rownames(final_new) <- rownames_finaldf
```

## Plotting barplots for each gene to plot them in one pdf file and and sort by groups (mock – WT – delta – SA)

1. opening a pdf file : pdf()
2. Adjusting the graphical parameters - par()
3. Adjusting the colour - brewer.pal()
4. Generating the barplots - barplot()

```
# assigning a null varaible
temp <-NULL
# opening a pdf file
pdf(file = "mainplot.pdf")

# par is  used to set graphical parameters.
par(mfrow = c(2,1), mar=c(6,4,6,1))

# adjusting the colour
palette <- RColorBrewer::brewer.pal(length(unique(final_new$Source)),name = 'Set1')
final_new$color <- palette[as.factor(final_new$Source)]

## plotting bar-plots using for loop- for each gene
for (i in 2:50) {
  tmp <- final_new[1:36,i]
  barplot(names=final_new$name,
          height=tmp,las=2, col=final_new$color,legend.text=T, main=colnames(final_new)[i])
}
```

## Output

**The mainplot.pdf file consists of the barplots for each of the 50 gene to plot which are sorted by groups (mock – WT – delta – SA)**

## Analysis

```
# Extract gene id, gene name, log2 fold changes and p-values from data  using subset()
data_subset<-subset(data, select = c("gene_id","gene_name",
                                     "log2FoldChange",
                                     "pvalue"))
# Filter the results to only significant changes
# using the typical cutoffs of pval < 0.05 and abs(log2fc) > 1
# keep rows with p-value<0.05  & |log2FoldChange|>1
exp1_df<- data.frame(data_subset[data_subset$pvalue<0.05 &
                                   abs(data_subset$log2FoldChange)>1,])

#rank tables high to low log2FoldChange_abs
exp1_final <- exp1_df %>%
  as.data.frame() %>%
```

```
  dplyr::arrange(dplyr::desc(log2FoldChange))

# Top 10 genes
head(exp1_final, 10)
```

```
##            gene_id gene_name log2FoldChange       pvalue
## 1  ENSG00000070019    GUCY2C       1.772129 0.0000121000
## 2  ENSG00000005108    THSD7A       1.649147 0.0000006450
## 3  ENSG00000068976      PYGM       1.537137 0.0001057710
## 4  ENSG00000110436    SLC1A2       1.346540 0.0001422660
## 5  ENSG00000056291    NPFFR2       1.242239 0.0061254150
## 6  ENSG00000134532      SOX5       1.228554 0.0000000758
## 7  ENSG00000118322    ATP10B      -1.013671 0.0000092200
## 8  ENSG00000111863     ADTRP      -1.033641 0.0000436000
## 9  ENSG00000128165      ADM2      -1.086260 0.0000589000
## 10 ENSG00000115604    IL18R1      -1.093068 0.0000050900
```

```
# down 10 genes
tail(exp1_final, 10)
```

```
##            gene_id gene_name log2FoldChange       pvalue
## 25 ENSG00000079263     SP140      -1.631839 1.960000e-10
## 26 ENSG00000133328   HRASLS2      -1.652180 1.380000e-07
## 27 ENSG00000124256      ZBP1      -1.866088 1.610503e-03
## 28 ENSG00000115896     PLCL1      -1.884295 1.535190e-04
## 29 ENSG00000091137   SLC26A4      -1.959640 2.440000e-08
## 30 ENSG00000102524  TNFSF13B      -2.034012 4.760000e-05
## 31 ENSG00000130487   KLHDC7B      -2.097465 6.970000e-07
## 32 ENSG00000134339      SAA2      -2.127504 8.340000e-19
## 33 ENSG00000128268     MGAT3      -2.194290 4.350000e-08
## 34 ENSG00000089692      LAG3      -2.652180 3.900000e-08
```