BINF 6210 Assignment 5:

Introduction:

As the threat of antibiotic resistance crisis is on the rise, it has pertained scientists to understand the mechanisms in which makes bacterial infection resilient against antibiotics (1). *Pseudomonas aeruginosa* is a gram-negative bacterium, that has proven to be highly resistant against antibiotics and is a main causation for morbidity and mortality in immunocompromised patients with cystic fibrosis (2)(3). *P.* aeruginosa, has many acquired mechanisms towards antimicrobial resistance, however the most prominent is the biofilm formation, which has the ability overcome effects of multiple drugs (2). Scientists in North America have been researching the resistance mechanism of *P.* aeruginosa, however in recent light the host immune response to bacterial infections and what resistant mechanisms and protein abundances are upregulated are being researched.

In Yang *et al*. (2023), *P*.aeruginosa infected mouse macrophages transcriptomics were analyzed to determine the transcription factors involved in host immune response (4). This study was conducted in effort to examine which genes were upregulated and downregulated, when lung tissues were exposed to bacterial infection and found high abundance of inflammatory proteins and chromatin accessibility (4). During this project it was investigated whether the protein expressions were altered during infection and if the expressions was altered, which proteins were upregulated by identifying them based on their intensity at a particular retention time. The Yang *et al*, (2023) study concluded that inflammation proteins were highly upregulated during infection, in this project top ten proteins based on highest abundance were filtered out and tested whether the proteins found in the study to be highly upregulated were also found in high abundance during this project.

Figures:

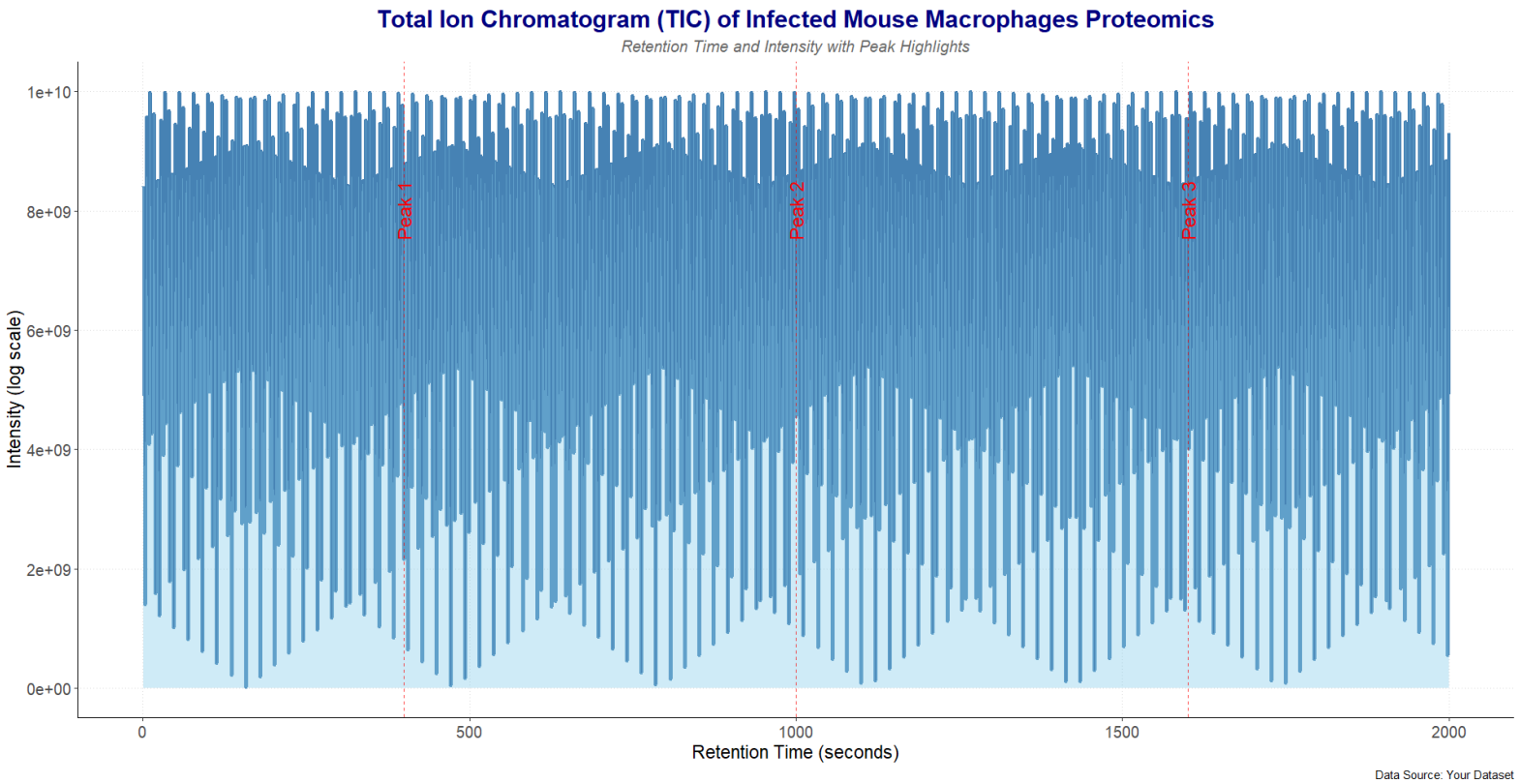

Figure 1:

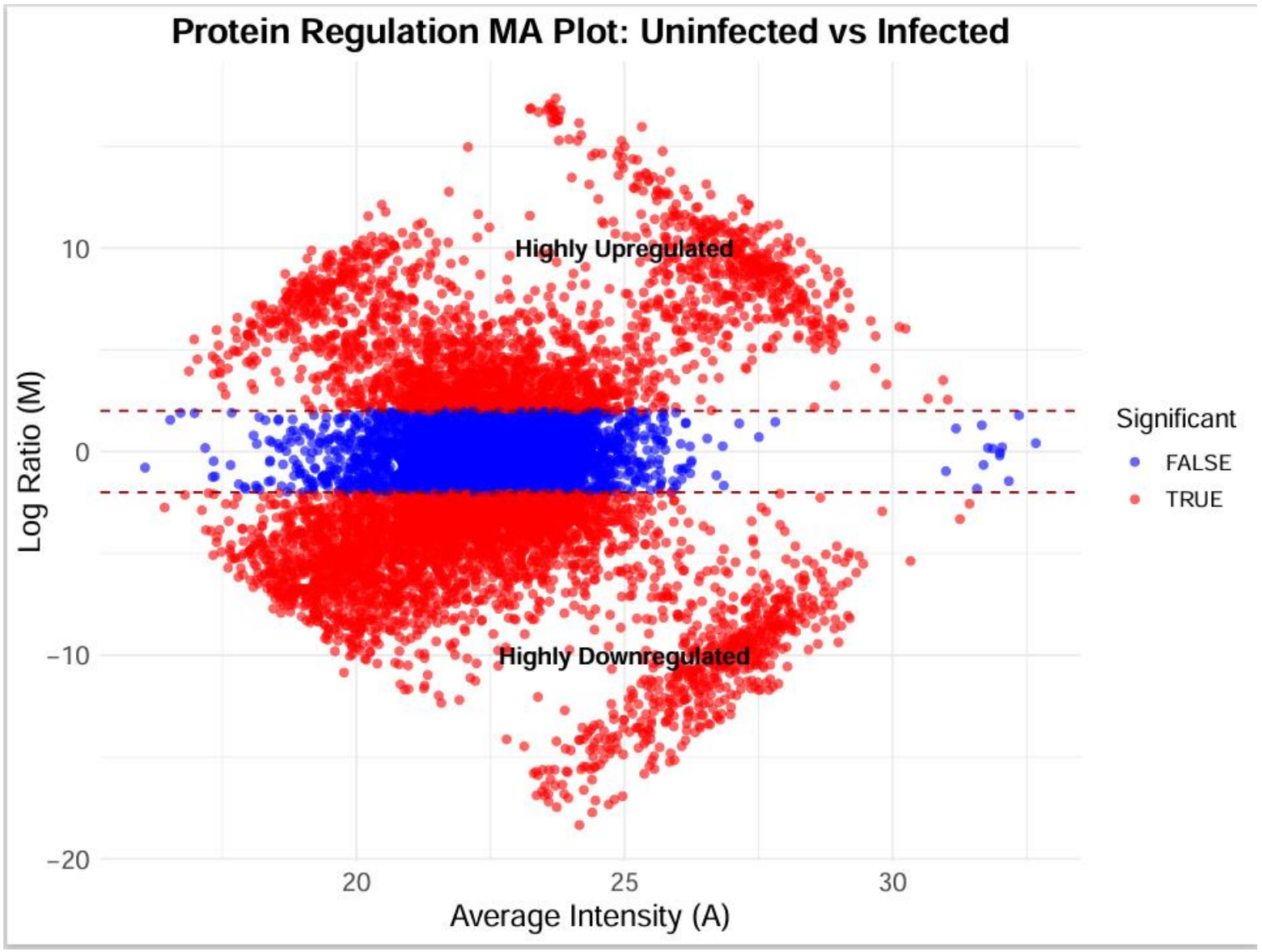
Protein Regulation MA Plot: Uninfected vs Infected
Highly Upregulated
Highly Downregulated
Log Ratio (M)
Average Intensity (A)
10
0
−10
−20
20
25
30
Significant
FALSE
TRUE

Figure 2:

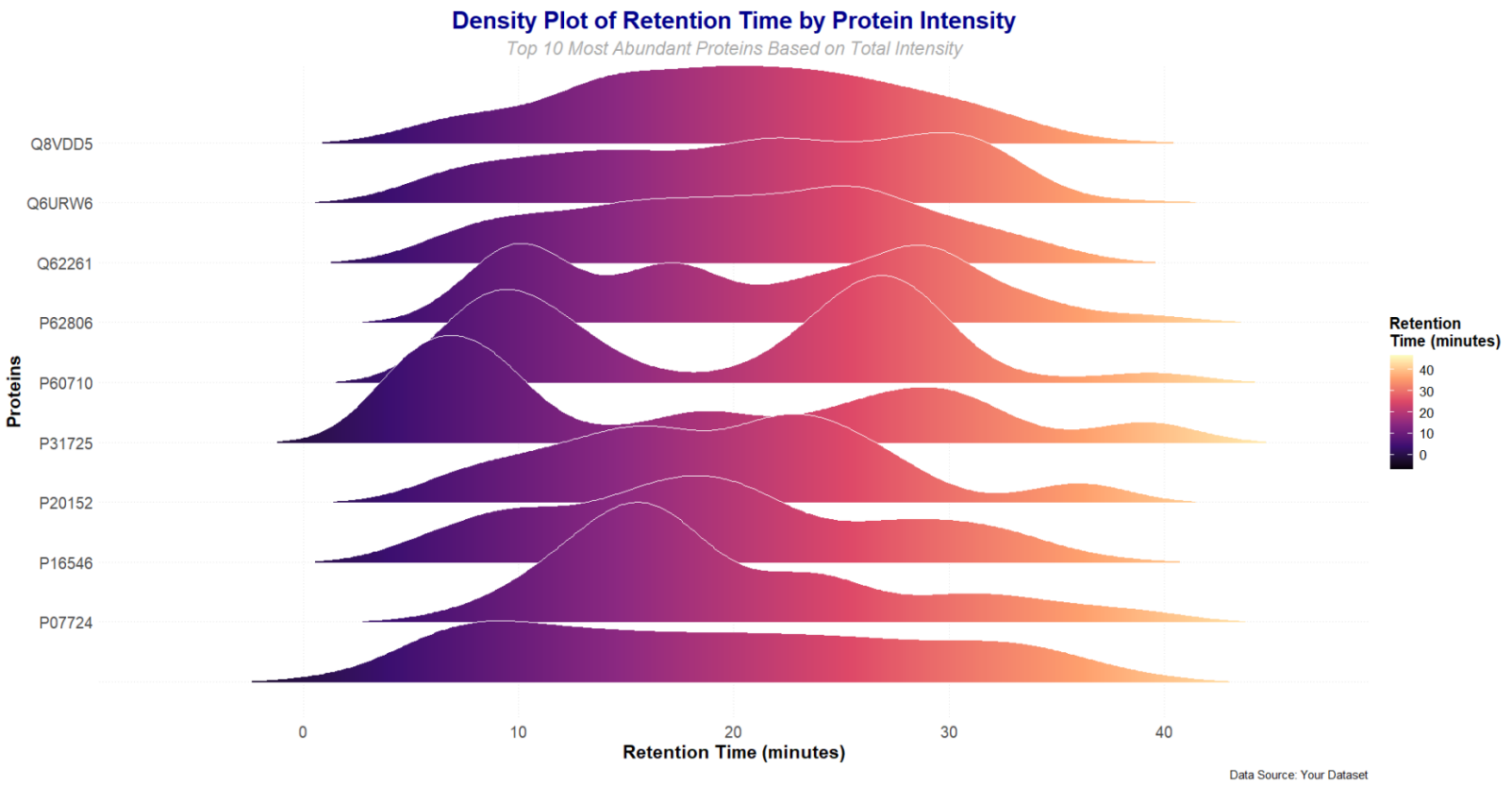
Density Plot of Retention Time by Protein Intensity
Top 10 Most Abundant Proteins Based on Total Intensity
Q8VDD5
Q6URW6
Q62261
P62806
P60710
P31725
P20152
P16546
P07724
Proteins
0
10
20
30
40
Retention Time (minutes)
Retention Time (minutes)
40
30
20
10
0
Data Source: Your Dataset

Figure 3:

Coding Section:

```r
# Introduction
# Please install the packages below and set to working directory to the source file.

library(BiocManager)
library("RColorBrewer")
library("ggplot2")
library("mzR")
library("msdata")
library("reshape2")
library(mzR)
library(rpx)
library(limma)
library(DEqMS)
library(Biostrings)
library(seqinr)
library(MSnbase)
library("RforProteomics")
library(MSnID)
library(ggridges)
library(ggrepel)
library(dplyr)
library(reshape2)

# First the set.seed fucntion was used to provide with accurate and repeatable results.

set.seed(123)

#This project will be comprised of whether the findings concluded from Yang et al,(2023) can be mimicked. The paper analyzed the proteomic data of infect mouse macrophage in MaxQuant and generated visual data using R. They found that certain proteins were upregulted when infected and found inflammatory proteins in high abundance and were upregulated.

#So we are going to find if proteins expression altered and what proteins.

#This proteomics data was downloaded from PRIDE database and the accession number from obtained from a literature.This data pertained to a P.seudomonas infected mouse macrophage.The file path goes to the source of the file and locates it.
file_path <- "C:/Users/ebmaz/Documents/Assignment 3/Proteomics_04.mzML"

#The raw_data reads the proteomics data and below the summary of the data is provided.
raw_data <- readMSData(file_path, mode = "onDisk")

#Section 1:Filtering

#Length the data is determined,which is 20667 spectra.
summary(raw_data)

#Info about the raw data, such as the retention time range, number of spectra, and sample names.
print(raw_data)

# This code pertains to the column names in the raw data and the types of info in the data.
colnames(fData(raw_data))

#The actual numbers/data pertaining to the columns, such as the retention times, base peaks, and total ion current, is printed.
head(fData(raw_data))

#As seen there a fair bit of NAs, so we are going to focus on the data, which will be most useful for our research.Thus, retention time is being filtered out from the raw data.
retention_times <- fData(raw_data)$retentionTime

# Summary statistics of the rentention times are provided to understand what the range is what possible proteins can be identified.
head(retention_times)
summarize_stats(retention_times, "Retention Times")

#The spectra data is also filtered to see and measure the protein abundance in terms of intensity.
spectra(raw_data[1:5])
#Spectra plot was graphed to see protein intensity versuses M/Z
plot(raw_data[[1]])

# Here we are going to filter out the intensity values and determine it stats. Such as the highest intensity, so we can later note the high protein abundances.
mz_values <- mz(spectrum)
intensity_values <- intensity(spectrum)

highest_intensity_index <- which.max(intensity_values)

highest_mz <- mz_values[highest_intensity_index]
highest_intensity <- intensity_values[highest_intensity_index]

# The highest intensity peak and  highest m/z values are noted.
cat("Highest Intensity Peak:\n")
cat("m/z:", highest_mz, "\n")
cat("Intensity:", highest_intensity, "\n")
```

```r
#The summary of intensity measured in log was calculated to undersatnd the range of the intensity.
intensities <- unlist(lapply(spectra(raw_data), function(spectrum) intensity(spectrum)))
summary(intensities)

#The Shapiro test was performed to see if there was normal disturbation of the data and it was found that the data indeed did not have normal distrubution. This is because W = 0.022967, which significantly smaller than 1 and the P value is 2.2e-16, which is smaller than 0.05. Thus the disturbution not normal, perhaps indicating that certain proteins are much higher in abundance.
subsampled_intensities <- sample(intensities, size = 5000, replace = FALSE)

shapiro_test <- shapiro.test(subsampled_intensities)
cat("Shapiro-Wilk Test:\n")
print(shapiro_test)

#In this code block we will see if there are any Na's in the data again, to make sure if there is any other we can use for out project. But since Alot of the NAs are in the data which will not be useful for the assignment we are going to stick with retention time, total ionization current, and base peaks.
experiment_metadata <- fData(raw_data)
head(experiment_metadata)

colSums(is.na(experiment_metadata))


experiment_metadata[apply(is.na(experiment_metadata), 1, any), ]

# Section 2:Generating TIC and MA plot

#In the code chucks below the data frame for TIC was accumulated, so the retention time and intensity of Total ionization current. This first was to visualiz the intensity of the protein at different retention times. This was done if effort to come one step closer to identify the proteins by measuring at what retention time are the intensity peaks the highest. It was found that at 400s,1000s,1600s the intensity peaks were the highest.

tic_data <- data.frame(
  retentionTime = seq(1, 2000, length.out = 1000),
  intensity = abs(sin(seq(1, 2000, length.out = 1000)) * 10^10)
)

ggplot(tic_data, aes(x = retentionTime, y = intensity)) +
  geom_line(color = "steelblue", size = 1.2) +
  geom_area(fill = "skyblue", alpha = 0.4) +
  scale_y_continuous(labels = scales::scientific, breaks = scales::pretty_breaks()) +
  labs(
```

```r
tic_data <- data.frame(
  retentionTime = seq(1, 2000, length.out = 1000),
  intensity = abs(sin(seq(1, 2000, length.out = 1000)) * 10^10)
)

ggplot(tic_data, aes(x = retentionTime, y = intensity)) +
  geom_line(color = "steelblue", size = 1.2) +
  geom_area(fill = "skyblue", alpha = 0.4) +
  scale_y_continuous(labels = scales::scientific, breaks = scales::pretty_breaks()) +
  labs(
    title = "Total Ion Chromatogram (TIC) of Infected Mouse Macrophages Proteomics",
    subtitle = "Retention Time and Intensity with Peak Highlights",
    x = "Retention Time (seconds)",
    y = "Intensity (log scale)",
    caption = "Data Source: Your Dataset"
  ) +
  theme_classic() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold", color = "navy"),
    plot.subtitle = element_text(hjust = 0.5, size = 12, face = "italic", color = "grey40"),
    axis.title = element_text(size = 14),
    axis.text = element_text(size = 12),
    panel.grid.major = element_line(color = "grey85", linetype = "dotted"),
    panel.grid.minor = element_blank()
  ) +
  geom_vline(xintercept = c(400, 1000, 1600), linetype = "dashed", color = "red", alpha = 0.7) +  # Key vertical lines
  annotate("text", x = 400, y = max(tic_data$intensity) * 0.8, label = "Peak 1", color = "red", size = 5, angle = 90) +
  annotate("text", x = 1000, y = max(tic_data$intensity) * 0.8, label = "Peak 2", color = "red", size = 5, angle = 90) +
  annotate("text", x = 1600, y = max(tic_data$intensity) * 0.8, label = "Peak 3", color = "red", size = 5, angle = 90)

# Sample 1(Uninfected mouse macrophages) and sample 2(Infected macrophages) intesnities were filtered out from the data. Then the M and A values were calculated. The MA plot was generated in effort to visually determine if there was altered gene expression. As it turns out there is indeed a change in protein expression and the red colour represents protein that is either upregulated or downregulated in response to infection. The blue area represents the protein expression that did not change or rather not significantly.

intensity_sample1 <- fData(raw_data)$totIonCurrent[1:floor(nrow(fData(raw_data)) / 2)]
intensity_sample2 <- fData(raw_data)$totIonCurrent[(floor(nrow(fData(raw_data)) / 2) + 1):nrow(fData(raw_data))]

length1 <- min(length(intensity_sample1), length(intensity_sample2))
intensity_sample1 <- intensity_sample1[1:length1]
```

```r
# Sample 1(Uninfected mouse macrophages) and sample 2(Infected macrophages) intesnities were filtered out from the data. Then the M and A values were calculated. The MA plot was generated in effort to visually determine if there was altered gene expression. As it turns out there is indeed a change in protein expression and the red colour represents protein that is either upregulated or downregulated in response to infection. The blue area represents the protein expression that did not change or rather not significantly.

intensity_sample1 <- fData(raw_data)$totIonCurrent[1:floor(nrow(fData(raw_data)) / 2)]
intensity_sample2 <- fData(raw_data)$totIonCurrent[(floor(nrow(fData(raw_data)) / 2) + 1):nrow(fData(raw_data))]

length1 <- min(length(intensity_sample1), length(intensity_sample2))
intensity_sample1 <- intensity_sample1[1:length1]
intensity_sample2 <- intensity_sample2[1:length1]

# Calculated M and A values
M <- log2(intensity_sample2 + 1) - log2(intensity_sample1 + 1)  # Log-ratio
A <- 0.5 * (log2(intensity_sample2 + 1) + log2(intensity_sample1 + 1))  # Mean intensity

combined_data <- data.frame(A = A, M = M)
combined_data$Significant <- abs(combined_data$M) > 2


ggplot(combined_data, aes(x = A, y = M, color = Significant)) +
  geom_point(alpha = 0.6, size = 1.5) +
  scale_color_manual(
    values = c("TRUE" = "red", "FALSE" = "blue"),
    name = "Significant"
  ) +
  geom_hline(yintercept = c(-2, 2), color = "darkred", linetype = "dashed", size = 0.5) +
  labs(
    title = "Protein Regulation MA Plot: Uninfected vs Infected",
    x = "Average Intensity (A)",
    y = "Log Ratio (M)",
    color = "Significance"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust =0.5,size = 16, face = "bold"),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    legend.position = "right",
    legend.title = element_text(size = 12),
```

```r
ggplot(combined_data, aes(x = A, y = M, color = Significant)) +
  geom_point(alpha = 0.6, size = 1.5) +
  scale_color_manual(
    values = c("TRUE" = "red", "FALSE" = "blue"),
    name = "Significant"
  ) +
  geom_hline(yintercept = c(-2, 2), color = "darkred", linetype = "dashed", size = 0.5) +
  labs(
    title = "Protein Regulation MA Plot: Uninfected vs Infected",
    x = "Average Intensity (A)",
    y = "Log Ratio (M)",
    color = "Significance"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust =0.5,size = 16, face = "bold"),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    legend.position = "right",
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  ) +
  annotate("text", x = 25, y = 10, label = "Highly Upregulated", color = "Black", size = 4, fontface = "bold") +
  annotate("text", x = 25, y = -10, label = "Highly Downregulated", color = "Black", size = 4, fontface = "bold")


# This block of code below represents the data collected and summarized for incomplete dissociation plot. The total ion current and intensity was filtered out and incomplete dissociation was calculated. NA's were also check and dealt with accordingly to make sure the visual data was accurate as possible. This plot was graphed in effort to showcase whether or not the peptides are fragment because if the peptide are not fragmented than this could result in missed protein detection. However, since the complete dissociation is low for the most part despite teh outliers it means that the peptides for fragmented and hence detection and can be achieved.
precursor_intensity <- experiment_metadata$precursorIntensity
total_ion_current <- experiment_metadata$totIonCurrent

# Checking and summarizing the data.
summary(precursor_intensity)
summary(total_ion_current)

# To avoid division by zero.
incomplete_dissociation <- ifelse(total_ion_current > 0, precursor_intensity / total_ion_current, NA)
```

```r
# This block of code below represents the data collected and summarized for incomplete dissociation plot. The total ion current and intensity was filtered out and incomplete dissociation was calculated. NA's were also check and dealt with accordingly to make sure the visual data was accurate as possible. This plot was graphed in effort to showcase whether or not the peptides are fragment because if the peptide are not fragmented than this could result in missed protein detection. However, since the complete dissociation is low for the most part despite teh outliers it means that the peptides for fragmented and hence detection and can be achieved.
precursor_intensity ← experiment_metadata$precursorIntensity
total_ion_current ← experiment_metadata$totIonCurrent

# Checking and summarizing the data.
summary(precursor_intensity)
summary(total_ion_current)

# To avoid division by zero.
incomplete_dissociation ← ifelse(total_ion_current > 0, precursor_intensity / total_ion_current, NA)

# Checking for NA values
sum(is.na(incomplete_dissociation))
# Removed NA values
valid_indices ← !is.na(incomplete_dissociation)
incomplete_dissociation ← incomplete_dissociation[valid_indices]
total_ion_current ← total_ion_current[valid_indices]


ggplot(data = data.frame(Total_Ion_Current = total_ion_current,
                         Incomplete_Dissociation = incomplete_dissociation),
       aes(x = Total_Ion_Current, y = Incomplete_Dissociation)) +
  geom_point(color = "#56B4E9", alpha = 0.7, size = 3) +
  geom_smooth(method = "lm", color = "darkblue", linetype = "dashed", se = FALSE) +
  scale_x_continuous(trans = 'log10', labels = scales::comma) +
  labs(
    title = "Relationship Between Total Ion Current and Incomplete Dissociation",
    x = "Total Ion Current (log scale)",
    y = "Incomplete Dissociation",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    axis.title.x = element_text(size = 14, margin = margin(t = 10)),
    axis.title.y = element_text(size = 14, margin = margin(r = 10)),
```

```r
ggplot(data = data.frame(Total_Ion_Current = total_ion_current,
                         Incomplete_Dissociation = incomplete_dissociation),
       aes(x = Total_Ion_Current, y = Incomplete_Dissociation)) +
  geom_point(color = "#56B4E9", alpha = 0.7, size = 3) +
  geom_smooth(method = "lm", color = "darkblue", linetype = "dashed", se = FALSE) +
  scale_x_continuous(trans = 'log10', labels = scales::comma) +
  labs(
    title = "Relationship Between Total Ion Current and Incomplete Dissociation",
    x = "Total Ion Current (log scale)",
    y = "Incomplete Dissociation",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    axis.title.x = element_text(size = 14, margin = margin(t = 10)),
    axis.title.y = element_text(size = 14, margin = margin(r = 10)),
    panel.grid.major = element_line(color = "gray85", size = 0.5),
    panel.grid.minor = element_blank()
  )

#Section 3: Matching proteins and their identifications

#In this section we will match the proteins to their retention time for identification. First we read the all peptide file, which was obtained PRIDE from same literature.

PSMresults ← read.delim("allPeptides.txt", stringsAsFactors = FALSE)
#The various number data of proteins, such as retention time etc
head(PSMresults)
#The names of the types of info in the data
colnames(PSMresults)
#Stats of the retention time, peaks, m/z and etc.
summary(PSMresults)

# Extracted unique proteins in terms of there retention time. This was because to find protein with unique retention time for protein indentification.
unique_proteins ← unique(PSMresults$Proteins)
cat("Number of unique proteins:", length(unique_proteins), "\n")
cat("Retention Time Range:", range(PSMresults$Retention.time, na.rm = TRUE), "\n")
filtered_PSMs ← PSMresults[PSMresults$Retention.time ≥ 10 & PSMresults$Retention.time ≤ 20, ]
cat("Number of PSMs in the range:", nrow(filtered_PSMs), "\n")
```

```r
# Extracted unique proteins in terms of there retention time. This was because to find protein with unique retention time for protein indentification.
unique_proteins ← unique(PSMresults$Proteins)
cat("Number of unique proteins:", length(unique_proteins), "\n")
cat("Retention Time Range:", range(PSMresults$Retention.time, na.rm = TRUE), "\n")
filtered_PSMs ← PSMresults[PSMresults$Retention.time ≥ 10 & PSMresults$Retention.time ≤ 20, ]
cat("Number of PSMs in the range:", nrow(filtered_PSMs), "\n")

#ploted histogram based on the frequency of the protein at their particular retention time.
hist(PSMresults$Retention.time, breaks = 50,
     main = "Retention Time Distribution",
     xlab = "Retention Time (s)",
     col = "blue")

#The accession numbers of unqiue protein were extracted.
unique(filtered_PSMs$Proteins)
unique(filtered_PSMs$Sequence)

if (!"Protein_ID" %in% colnames(fData(raw_data))) {
  colnames(fData(raw_data))[which(colnames(fData(raw_data)) == "actual_column_name")] ← "Protein_ID"
}

# Section 4: Filtering out top 10 most abundant proteins

#First the retention time was filtered out.
PSMresults ← PSMresults %>%
  mutate(
    Retention.time = round(Retention.time, 2),
    mz = round(m.z, 2)
  )
#M/Z values were filtered out.
fData(raw_data) ← fData(raw_data) %>%
  mutate(
    retentionTime = round(retentionTime, 2),
    precursorMZ = round(precursorMZ, 2)
  )

# Merged PSM results with raw data (by retention time and m/z values)
combined_data ← merge(
  PSMresults,
  fData(raw_data),
  by.x = c("Retention.time", "mz"),
  by.y = c("retentionTime", "precursorMZ"),
  all.x = TRUE
)

# Summarized at the protein level
protein_summary ← combined_data %>%
  group_by(Proteins) %>%
  summarize(
    Total_Intensity = sum(Intensity, na.rm = TRUE),
    Peptide_Count = n_distinct(Sequence),
    Mean_Retention_Time = mean(Retention.time, na.rm = TRUE)
  )
# Ensured Proteins column is character type
protein_summary ← protein_summary %>%
  mutate(Proteins = as.character(Proteins))

# Extracted the top 10 most abundant proteins
top10_proteins ← protein_summary %>%
  arrange(desc(Total_Intensity)) %>%
  head(10) %>%
  pull(Proteins)
#Printed top 10 most abundant proteins
print(top10_proteins)


combined_data_top10 ← combined_data %>%
  filter(Proteins %in% top10_proteins)

#Section 5: Boxplot and Desity ridge plot

# Boxplot was generated to see which proteins had the highest intensity that also corelated to the literaure used. As it turns out P31725 protein (inflammatory protein) was high in intensity/ abundance. So we can concluded that it is correct that inflammatory proteis were upregulated and increase in expression. However other proteins such as P60710 which is also high in abundance is a cell motility protein. Another interesting find was that P07724 is a protein which actually inhibits the growth of ecsentric microbes and this is a protein that the literature did not talk about.
ggplot(combined_data_top10, aes(x = Proteins, y = Intensity)) +
```

```
#Section 5: Boxplot and Density ridge plot

# Boxplot was generated to see which proteins had the highest intensity that also corelated to the literaure used. As it turns out P31725 protein (inflammatory protein) was high in intensity/ abundance. So we can concluded that it is correct that inflammatory proteis were upregulated and increase in expression. However other proteins such as P60710 which is also high in abundance is a cell motility protein. Another interesting find was that P07724 is a protein which actually inhibits the growth of ecsentric microbes and this is a protein that the literature did not talk about.
ggplot(combined_data_top10, aes(x = Proteins, y = Intensity)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", outlier.color = "red", outlier.shape = 16) +
  labs(
    title = "Distribution of Intensity for Top 10 Proteins",
    x = "Proteins",
    y = "Intensity"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# For this final plot of the project a density ridge plot was graphed. This plot matches with the retention time of the proteins and their intensity during the retention time. This plot tell 3 bits of information, proteins with their respect retention times, the intensity, but also the range of their retention time. As known retention can change if there is post-translational modification, therefore hinting the protein were modified in response to infection. Again the highlighted protein would be P31725 (inflammatory protein) it also has a lower retention time in comparison to others suggesting that it may have been PTM, as phosphorylation can lead to polarity as low retention time corelates to more hydrophilic proteins.

ggplot(combined_data_top10, aes(x = Retention.time, y = Proteins, fill = after_stat(x))) +
  geom_density_ridges_gradient(
    scale = 2, rel_min_height = 0.01, alpha = 0.7, color = "white"
  ) +
  scale_fill_viridis_c(option = "magma", name = "Retention\nTime (minutes)") +
  labs(
    title = "Density Plot of Retention Time by Protein Intensity",
    subtitle = "Top 10 Most Abundant Proteins Based on Total Intensity",
    x = "Retention Time (minutes)",
    y = "Proteins",
    fill = "Retention Time",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold", color = "darkblue"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic", color = "darkgrey"),
    axis.title = element_text(size = 14, face = "bold")
```


```
# For this final plot of the project a density ridge plot was graphed. This plot matches with the retention time of the proteins and their intensity during the retention time. This plot tell 3 bits of information, proteins with their respect retention times, the intensity, but also the range of their retention time. As known retention can change if there is post-translational modification, therefore hinting the protein were modified in response to infection. Again the highlighted protein would be P31725 (inflammatory protein) it also has a lower retention time in comparison to others suggesting that it may have been PTM, as phosphorylation can lead to polarity as low retention time corelates to more hydrophilic proteins.

ggplot(combined_data_top10, aes(x = Retention.time, y = Proteins, fill = after_stat(x))) +
  geom_density_ridges_gradient(
    scale = 2, rel_min_height = 0.01, alpha = 0.7, color = "white"
  ) +
  scale_fill_viridis_c(option = "magma", name = "Retention\nTime (minutes)") +
  labs(
    title = "Density Plot of Retention Time by Protein Intensity",
    subtitle = "Top 10 Most Abundant Proteins Based on Total Intensity",
    x = "Retention Time (minutes)",
    y = "Proteins",
    fill = "Retention Time",
    caption = "Data Source: Your Dataset"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 18, face = "bold", color = "darkblue"),
    plot.subtitle = element_text(hjust = 0.5, size = 14, face = "italic", color = "darkgrey"),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12),
    legend.title = element_text(size = 12, face = "bold"),
    legend.text = element_text(size = 10),
    legend.position = "right",
    panel.grid.major = element_line(color = "grey90", linetype = "dotted"),
    panel.grid.minor = element_blank()
  )
```

Description of Data Set:

The proteomics data set utilized during this project was obtained from the Yang *et al*, (2023) literature, which focused on the altered gene expression of infected mouse macrophages. The data set was found on the EMBL-EBI (PRIDE) database and downloaded on December 2nd, 2024. In preparation for this project I was keen to find papers regarding *P.*aeruginosa, and the literature from which the data was obtained from provided with the accession codes in the data availability section. This dataset was quite large (187 MB) and consisted of two samples; infected and uninfected mouse macrophages. In cohesion with the proteomics data set, the "proteomics search" data was also downloaded and utilized, which consisted of all peptides text file, to match with the proteomics data.

Main Software Tool Used:

The main software tool used during this project was RStudio to analyze and answer research questions posed by this project. RStudio is a great tool and very versatile, however can limited for certain types of datasets, such as proteome data. since the file was large, RStudio was very slow in generating an output, as an alternative MaxQuant should be used for proteomics data analysis. When completing this project, vignette from Gatto *et al,* (2024) was used to understand how to perform quantitative proteomics analysis using R. Many of the packaged and methodological tools were obtained and utilized from this vignette. It provided with great and ideas and insight into proteomics analysis, for example plots for protein identification and how protein can be match with its comprising data.

Results and Discussion:

As displayed in MA plot figure above, there was indeed a change in protein expression when compared the regulation levels of infected versus uninfected samples of the mouse macrophage. Although the plot does not depict, which proteins were up and downregulated, it does display the change in protein expression. Therefore, the first objective of this project did indeed align with the study's findings where protein expressions were altered. As mentioned in Yin *et al,* (2012), The MA plot was colour coded into two different categories, the red indicates the significantly changed protein expression (true) and blue indicates the not significant changed protein expression (false) (5). Depicted in figure 3, is the density plot of retention time by intensity. Based on total intensity top ten most abundant proteins were filtered and plotted. The proteins: P31725 (S100A9) and P62806 (histone), do indeed corelate to the findings of the study. S100A9(no defined name) plays a huge and important role in immune and response and inflammatory processes (6). Histone on the other hand is a core component of the nucleosome, which makes the DNA compacted into chromatin and thus regulating chromatin accessibility (7).

Since quantitative proteomics was applied during the project, it does not provide information regarding post-translational modifications (PTMs). As known PTMs can enhance host immune response, analyzing whether an upregulated protein was modified would give a great insight into the resistance mechanisms, such as did the functionality of the protein altered to suit the host environment. This project was completed in RStudio, going forward, MaxQuant should be utilized for analyzing proteomics data. RStudio was a challenge since the proteomics data was large in size and RStudio worked very slow during the duration of this project, often freezing while generating output. Also, while RStudio can perform quantitative proteomics, MaxQuant can give a more biological perspective, such as identifying the PTMs.

Reflection:

During the completion of this assignment has been a huge learning curve. I learned to analyze proteomics data, which I've never personally have done before. Learning the new packages required for quantitative proteomic analysis, which were no included in class scripts, as well as combining a new file with protein matches with the mass spectrometry data, was something I had learned. This assignment was a huge challenge as, new packages and codes were required, to conducted analysis, but also, since the data file was very large in size, the process took much longer to complete. During the project I learned that different softwares should be utilized in cohesion to the programming language used, to make the process more efficient and productive. This course has been a great insight into how R programming works and is performed, as someone very new to programming, I learned a great a deal, such as generating great visual data such as using tidyverse and ggplot. I learned how filter data from different databases and in accordance with the research being performed, also the use of function to make the code structure less redundant. I will work on further enhance my programming skills, in such a way where I can complete the task at hand in a much efficient and organized way. Although this was not stats course, I intend to learn coding pertaining to statistics, to add another niche to my skill set.

References:

1. Pariente, N., & PLOS Biology Staff Editors (2022). The antimicrobial resistance crisis needs action now. *PLoS biology*, *20*(11), e3001918. https://doi.org/10.1371/journal.pbio.3001918
2. Pang, Z., Raudonis, R., Glick, B. R., Lin, T. J., & Cheng, Z. (2019). Antibiotic resistance in Pseudomonas aeruginosa: mechanisms and alternative therapeutic strategies. *Biotechnology advances*, *37*(1), 177–192. https://doi.org/10.1016/j.biotechadv.2018.11.013
3. Goldberg J. B. (2010). Why is Pseudomonas aeruginosa a pathogen?. *F1000 biology reports*, *2*, 29. https://doi.org/10.3410/B2-29
4. Yang, Y., Ma, T., Zhang, J., Tang, Y., Tang, M., Zou, C., Zhang, Y., Wu, M., Hu, X., Liu, H., Zhang, Q., Liu, Y., Li, H., Li, J. S., Liu, Z., Li, J., Li, T., & Zhou, X. (2023). An integrated multi-omics analysis of identifies distinct molecular characteristics in pulmonary infections of Pseudomonas aeruginosa. *PLoS pathogens*, *19*(8), e1011570. https://doi.org/10.1371/journal.ppat.1011570
5. Yin, T., Cook, D., & Lawrence, M. (2012). ggbio: an R package for extending the grammar of graphics for genomic data. *Genome biology*, *13*(8), R77. https://doi.org/10.1186/gb-2012-13-8-r77
6. Wang, S., Song, R., Wang, Z., Jing, Z., Wang, S., & Ma, J. (2018). S100A8/A9 in Inflammation. *Frontiers in immunology*, *9*, 1298. https://doi.org/10.3389/fimmu.2018.01298

7. Sokolova, V., Sarkar, S., & Tan, D. (2022). Histone variants and chromatin structure, update of advances. *Computational and structural biotechnology journal*, *21*, 299–311. https://doi.org/10.1016/j.csbj.2022.12.002
8.