Getting to know the Data

Filipe Russo

March 17, 2019

Introduction

From the experimental design we know we are dealing with Proteomics data from the microalgae Chrorella vulgaris. Prof. Flavia V. Winck and Dra. Annamria Doria Vidotti studied the transition between auto-mixo-heterotrophic growth conditions.

Loading the Data

First, we load the dataset and store it in a variable:

```
Proteins = read.csv("proteinGroups.txt", sep = "\t", header=TRUE)
```

Then, we load some libraries that we will help us manipulate the data:

```
library(dplyr) #for data manupilation
library(stringr) #for string manipulation
library(VennDiagram) #for the Venn Diagram
library(ggplot2) #for plotting the data
library(tidyr) #for gather function

# little nice function to check if an element IS NOT in a list
'%ni%' <- Negate('%in%')
```

Data Summary

The `Proteins` dataset is made of:

- 768 observations (rows), one for each protein.
- 106 variables (columns).

We need first to rename some columns. MixoTP4 represents the fourth time point in the Mixotrophic growth condition, so it is actually the TP6 (120hrs) seen in the experimental design. Analogously HeteTP4 represents the fourth time point in the Heterotrophic growth condition, so it is TP10 (140hrs).

```
names(Proteins) <- str_replace(names(Proteins),
                               pattern = "MixoTP4",
                               replacement = "MixoTP6")
names(Proteins) <- str_replace(names(Proteins),
                               pattern = "HeteTP4",
                               replacement = "HeteTP10")
```

From the 106 columns we are only interested in the ones that start with LFQ and the `Majority.protein.IDs` one. From the 768 proteins we are interested in the ones that are not reverse, contaminants or only identified by site.

```
# we define positive identifiers for the Reverse,
# Only.identified.by.site
# and Potential.contaminant columns
```

```
rev_posids = list("+")
site_posids = list("+")
cont_posids = list("+")

filtered_long <- Proteins %>%

  # filters out rows based on posids of rev, site and cont
  # filters out rows based on the protein IDs, names with CON or REV
  filter(Reverse %ni% rev_posids &
         Only.identified.by.site %ni% site_posids &
         Potential.contaminant %ni% cont_posids &
         !str_detect(Majority.protein.IDs, "CON|REV")) %>%

  # keep only the id and LFQ columns
  select(c("Majority.protein.IDs", str_subset(names(Proteins), "LFQ"))) %>%

  # wide to long
  gather(key = Condition, value = LFQIntensity, ... = -"Majority.protein.IDs") %>%

  separate(col = Condition, into = c("Growth", "TimeRep"), sep = "(TP)") %>%

  separate(col = TimeRep, into = c("Time", "Replicate"), sep = "R") %>%

  mutate(Growth = str_replace(Growth, pattern = "LFQ.intensity.", replacement = "")) %>%

  rename(ID = Majority.protein.IDs) %>%

  mutate(ID = as.character(ID),
         Growth = as.factor(Growth),
         Time = as.factor(Time),
         Replicate = as.factor(Replicate))

# lets take a look at the resulting filtered dataset
glimpse(filtered_long)
```

```
## Observations: 6,813
## Variables: 5
## $ ID           <chr> "g10015.t1", "g10033.t1", "g10079.t1", "g10080.t1...
## $ Growth       <fct> Auto, Auto, Auto, Auto, Auto, Auto, Auto, Auto, A...
## $ Time         <fct> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2...
## $ Replicate    <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ LFQIntensity <int> 0, 0, 386920000, 0, 687130, 197500, 1211500, 3373...
```

From the `filtered_long` dataset we create the `filtered_tidy` one, which we will be using to make the *Scatterplots*, *Venn Diagram* and *Line Plot*. We decide to take the median value among the triplicates as the best estimative of the true LFQIntensity value.

```
filtered_tidy <- filtered_long %>%
  group_by(ID, Growth, Time) %>%
  # we add the MedLFQ variable to the dataset
  # which is the median LFQIntensity grouped by ID, Growth and Time
  mutate(MedLFQ = median(LFQIntensity)) %>%
  ungroup() %>%
  filter(Replicate == 1) %>%
  select(-c(Replicate, LFQIntensity)) %>%
```

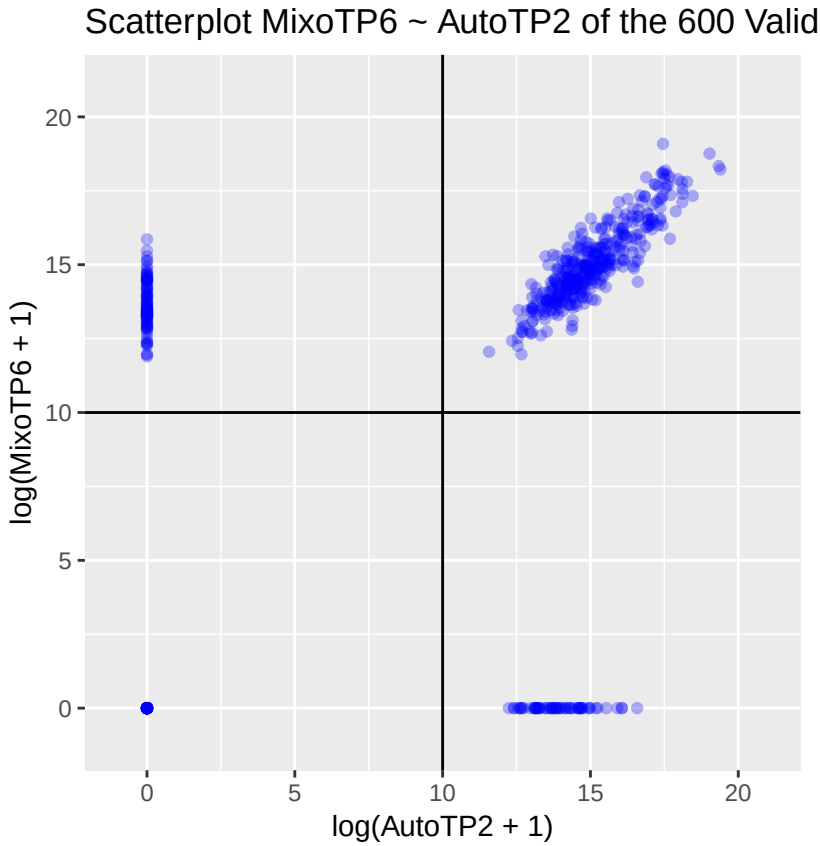
```
  unite(Condition, Growth, Time, sep = "TP", remove = TRUE) %>%
  spread(Condition, MedLFQ) %>%
  # we filter out proteins that have a null median value across all conditions
  filter(AutoTP2 + MixoTP6 + HeteTP10 > 0)

# lets take a look at the resulting filtered dataset
glimpse(filtered_tidy)
```

```
## Observations: 600
## Variables: 4
## $ ID       <chr> "g10033.t1", "g10079.t1", "g10080.t1", "g10108.t1", "...
## $ AutoTP2  <int> 0, 184340000, 0, 285140, 3102400, 43824000, 25921000,...
## $ HeteTP10 <int> 0, 85724000, 4732200, 0, 0, 65473000, 5581100, 555780...
## $ MixoTP6  <int> 1446700, 140210000, 5261800, 0, 0, 69094000, 10786000...
```

Scatterplots

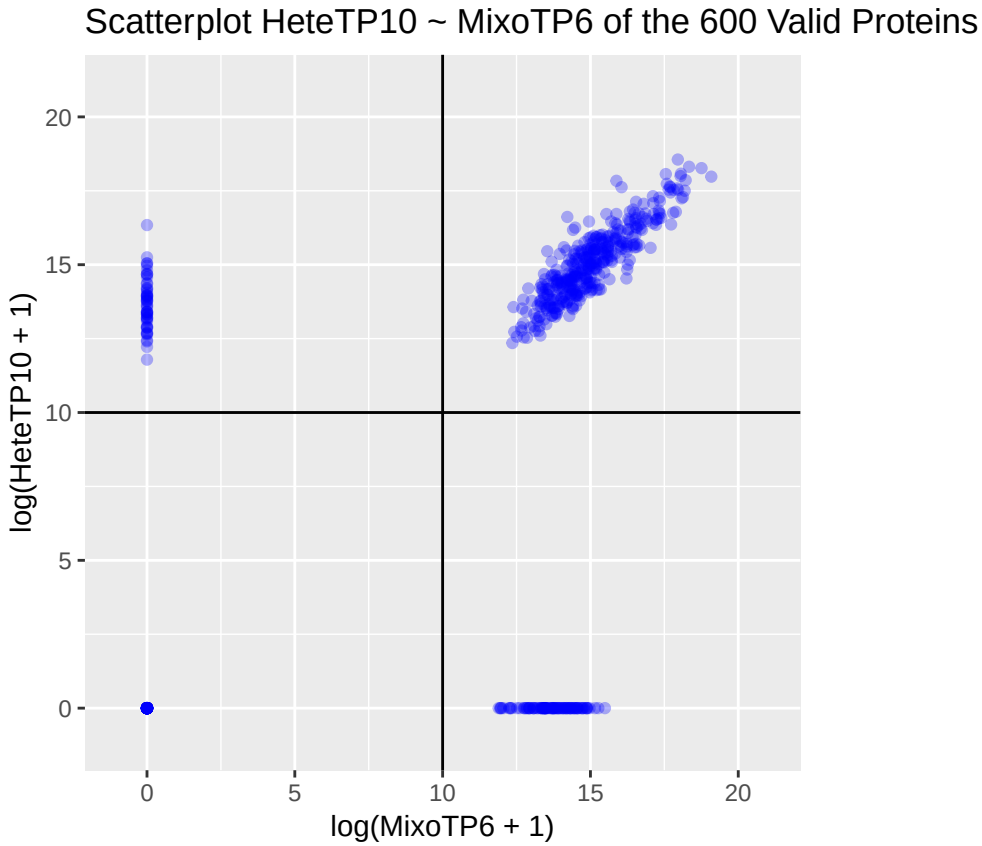
After all the filtering we end up with 600 Valid Proteins. We are considering valid a protein that is not reverse, contaminant, nor only identified by site and that have at least one positive median value among triplicates across the three growth conditions: AutoTP2, MixoTP6 and HeteTP10.

```
ggplot(filtered_tidy, aes(x = log(AutoTP2 + 1), y = log(MixoTP6 + 1))) +
  geom_point(alpha = 0.3, colour = "blue", fill = "blue", shape = 21) +
  geom_hline(yintercept = 10, color = "black") +
  geom_vline(xintercept = 10, color = "black") +
  coord_fixed(ratio = 1, xlim = c(-1, 21), ylim = c(-1, 21)) +
  labs(title="Scatterplot MixoTP6 ~ AutoTP2 of the 600 Valid Proteins")
```



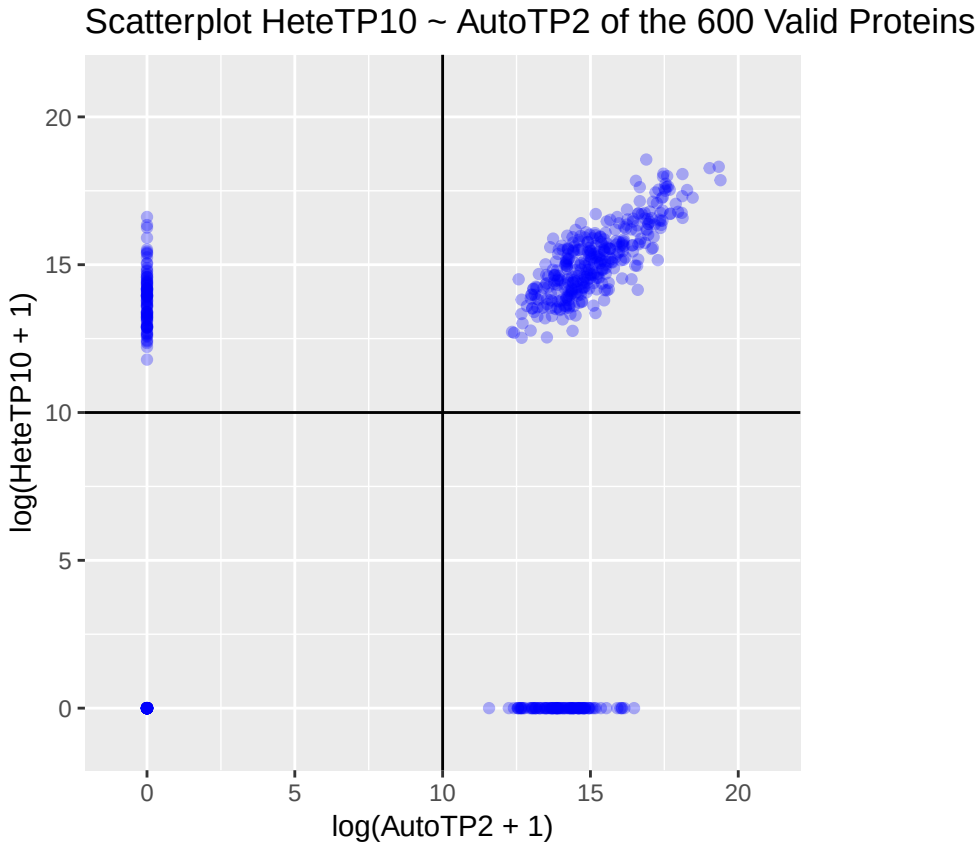
The *Scatterplots* allow us to visualize the proteins that are present in both conditions (top-right quadrant), absent in both conditions (bottom-left quadrant) or present in one and absent from the other (top-left and bottom-right quadrants).

```
ggplot(filtered_tidy, aes(x = log(MixoTP6 + 1), y = log(HeteTP10 + 1))) +
  geom_point(alpha = 0.3, colour = "blue", fill = "blue", shape = 21) +
  geom_hline(yintercept = 10, color = "black") +
  geom_vline(xintercept = 10, color = "black") +
  coord_fixed(ratio = 1, xlim = c(-1, 21), ylim = c(-1, 21)) +
  labs(title="Scatterplot HeteTP10 ~ MixoTP6 of the 600 Valid Proteins")
```



For the *Scatterplots* we added 1 to all median LFQIntensity values and then took the log, this way 0 still goes to 0, but the scale becomes more practical for the visualization.

```
ggplot(filtered_tidy, aes(x = log(AutoTP2 + 1), y = log(HeteTP10 + 1))) +
  geom_point(alpha = 0.3, colour = "blue", fill = "blue", shape = 21) +
  geom_hline(yintercept = 10, color = "black") +
  geom_vline(xintercept = 10, color = "black") +
  coord_fixed(ratio = 1, xlim = c(-1, 21), ylim = c(-1, 21)) +
  labs(title="Scatterplot HeteTP10 ~ AutoTP2 of the 600 Valid Proteins")
```



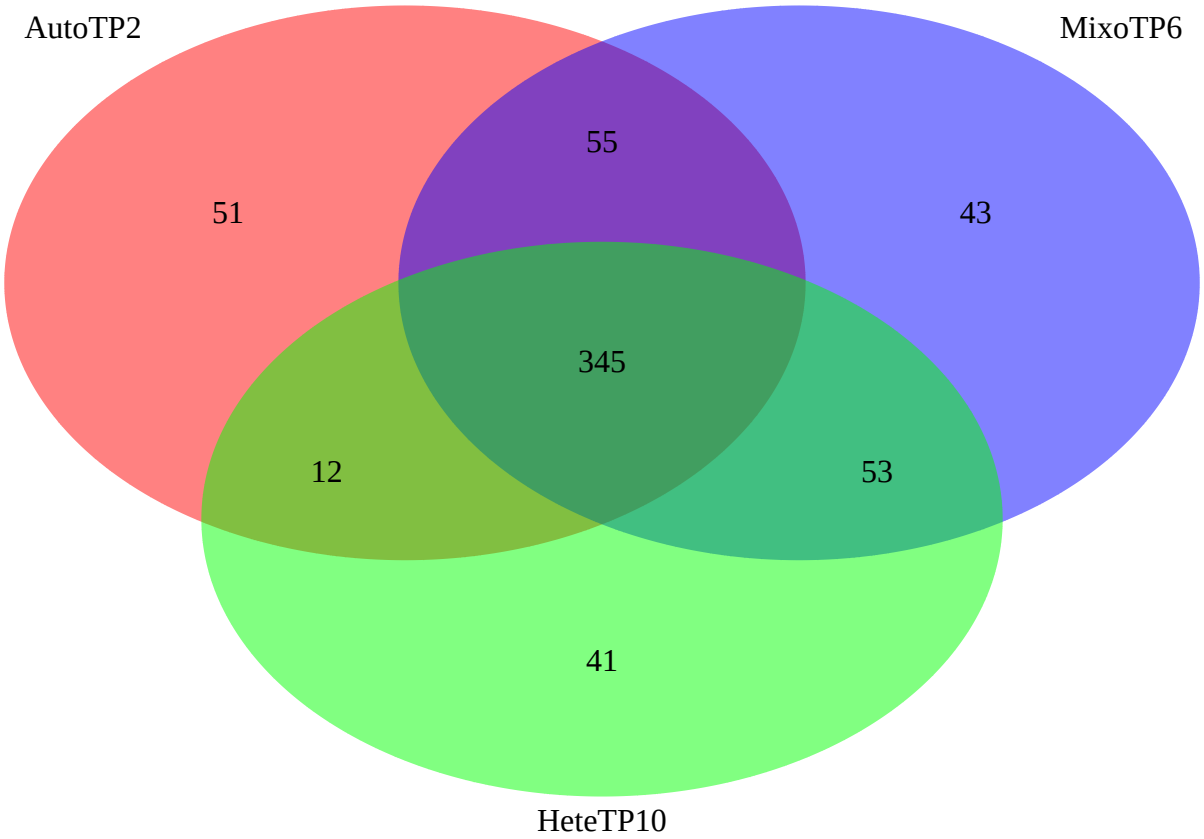
Venn Diagram

We slice the `filtered_tidy` dataset in subsets, where we can find the proteins present in AutoTP2, MixoTP6 or HeteTP10, for example.

```
AutoTP2Pos <- filtered_tidy %>% filter(AutoTP2 > 0)
MixoTP6Pos <- filtered_tidy %>% filter(MixoTP6 > 0)
HeteTP10Pos <- filtered_tidy %>% filter(HeteTP10 > 0)


AutoTP2Pos_MixoTP6Pos <- filtered_tidy %>% filter(AutoTP2 > 0 & MixoTP6 > 0)
AutoTP2Pos_HeteTP10Pos <- filtered_tidy %>% filter(AutoTP2 > 0 & HeteTP10 > 0)
MixoTP6Pos_HeteTP10Pos <- filtered_tidy %>% filter(MixoTP6 > 0 & HeteTP10 > 0)

AutoTP2Pos_MixoTP6Pos_HeteTP10Pos <- filtered_tidy %>% filter(AutoTP2 > 0 &
                                                              MixoTP6 > 0 &
                                                              HeteTP10 > 0)
```

The *Venn Diagram* allows us to visualize the intersections between the three growth conditions.

```
draw.triple.venn(area1 = 463, area2 = 496, area3 = 451, n12 = 400, n23 = 398, n13 = 357,
                 n123 = 345, category = c("AutoTP2", "MixoTP6", "HeteTP10"),
                 lty = "blank",
                 fill = c("red", "blue", "green"))
```

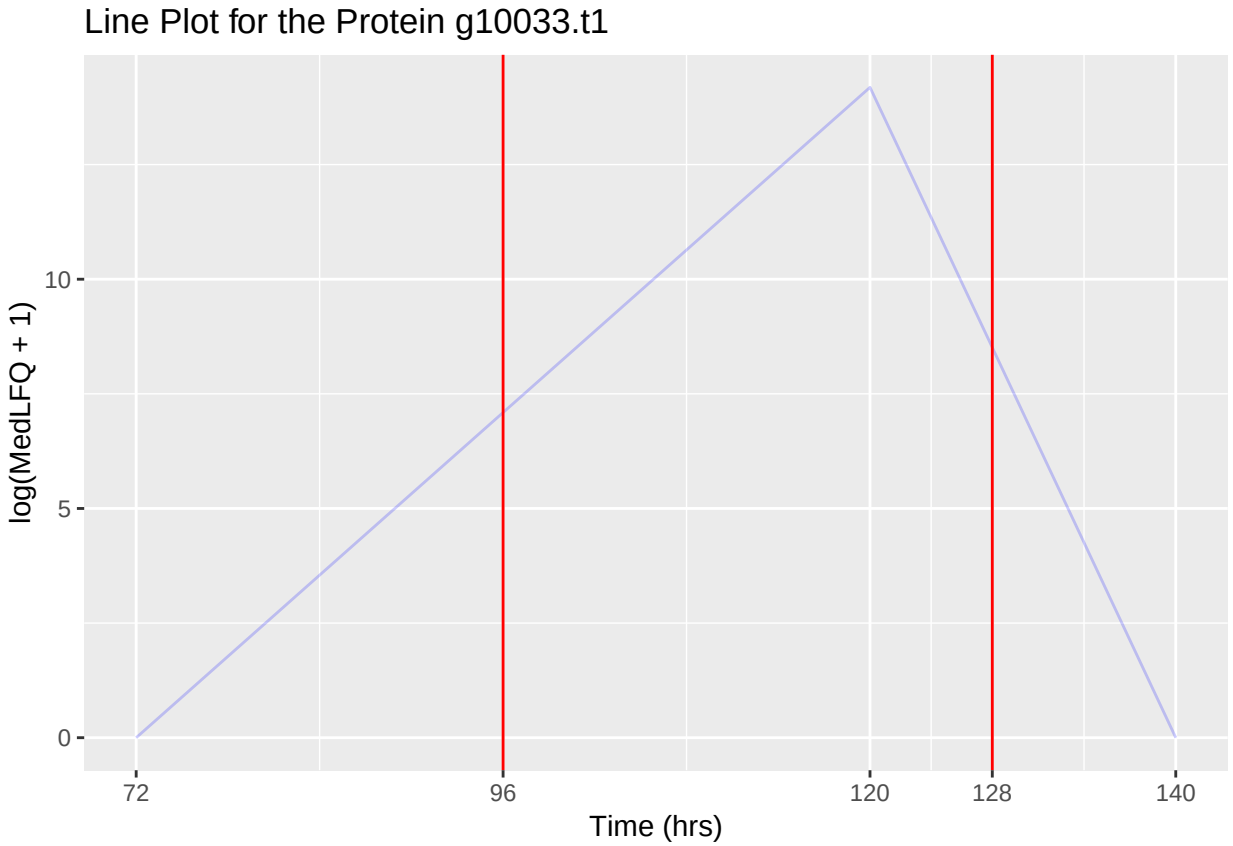


Do the 345 Valid Proteins present in the triple intersection represent the most vital proteins necessary to the maintenance of life? On the other hand we find 51, 43, 41 proteins present only in AutoTP2, MixoTP6, HeteTP10 respectively.

Line Plot

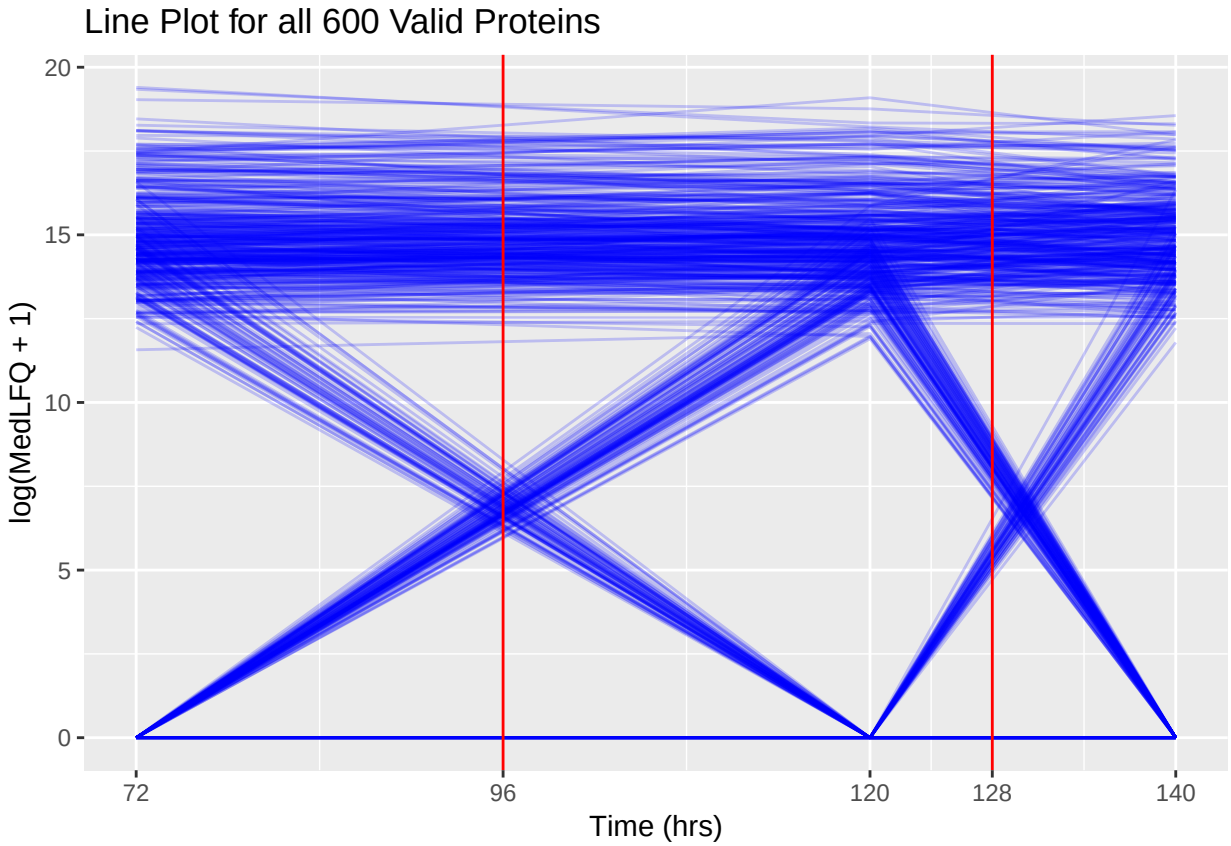
First, we create the `filtered_tidy_2` dataset out of `filtered_tidy`. Then we make a *Line Plot* for the Protein g10033.t1.

```
filtered_tidy_2 <- filtered_tidy
names(filtered_tidy_2) <- c("ID", "AutoTP72",  "HeteTP140", "MixoTP120")
filtered_tidy_2 <- filtered_tidy_2 %>%
  gather(key = Condition, value = MedLFQ, ... = -"ID") %>%
  separate(col = Condition, into = c("Growth", "Time"), sep = "(TP)") %>%
  mutate(Time = as.numeric(Time))

filtered_tidy_2  %>% filter(ID == "g10033.t1") %>%
    ggplot(aes(x = Time, y = log(MedLFQ + 1), group = ID)) +
      geom_line(alpha = 0.2, colour = "blue") +
      scale_x_continuous(breaks = c(72, 96, 120, 128, 140)) +
      geom_vline(xintercept = 96, color = "red") +
      geom_vline(xintercept = 128, color = "red") +
      labs(x = "Time (hrs)",
           title="Line Plot for the Protein g10033.t1")
```



We can see the Protein g10033.t1 is present at 120 hours in the Mixotrophic growth condition. But not at 72 hours, nor at 140 hours in the Autotrophic and Heterotrophic conditions respectively. In vertical red lines we show at 96 hours the shift from Autotrophic to Mixotrophic and at 128 hours the shift from Mixotrophic to Heterotrophic.

```
filtered_tidy_2  %>%
    ggplot(aes(x = Time, y = log(MedLFQ + 1), group = ID)) +
      geom_line(alpha = 0.2, colour = "blue") +
      scale_x_continuous(breaks = c(72, 96, 120, 128, 140)) +
      geom_vline(xintercept = 96, color = "red") +
      geom_vline(xintercept = 128, color = "red") +
      labs(x = "Time (hrs)",
           title="Line Plot for all 600 Valid Proteins")
```



Now we plotted all the 600 Valid Proteins, each represented by a single line and again we are able to visualize the proteins that are present or absent at a given condition. At the top of the image is also clear to see the 345 proteins present at all 3 conditions.

Saving our Filtered Dataset

Finally we save our `filtered_tidy_2` dataset for further analysis.

```
write.csv(filtered_tidy_2, file = "proteins.csv", row.names = FALSE)
valid_proteins <- read.csv("proteins.csv", sep = ",", header = TRUE)
```