USC Department Of Translational Genomics
Keck Medicine of USC

# TRGN 527: Applied Data Science and Bioinformatics

## UNIT I. Introduction and Basic Data Science

## Week 4 – Lecture 3 – Case Study Part 3

**Enrique I. Velazquez Villarreal, M.D., Ph.D., M.P.H., M.S.** | **Assistant Professor**

Dept. of Translational Genomics

USC | Keck School of Medicine | Norris Comprehensive Cancer Center

Leader of the USC Bioinformatics Core – *USC CaRE2 Health Equity Center*

**David W. Craig, Ph.D.** | **Professor and Vice Chair**

Dept. of Translational Genomics

USC | Keck School of Medicine | Norris Comprehensive Cancer Center

Co-Director, Institute of Translational Genomics

# Case Study – Proteomics Analysis

# Case Study – Proteomics Analysis

- Download the aa.mass.csv file from Blackboard.
- Read in R the monoisotopic mass of the amino acids that is stored in the downloaded file:

````
128
129 # Download the aa.mass.csv file from Blackboard
130 # Read the monoisotopic mass of the amino acids that is stored in the file:
131 ```{r}
132
133 aa.mass <-read.csv("aa_mass.csv", row.names = 1)
134
135 ```
````

# Case Study – Proteomics Analysis

- Creating a matrix with the same dimensions as the peakdiff matrix.
- This step is also storing the identified amino acids:

````
# Create a matrix with the same dimensions as the peakdiff matrix and fill it up with dashes.
# Storing the identified amino acids:
```{r}

peakdiff.aa <- matrix(rep('-', topnum^2), ncol=topnum, nrow=topnum)

```
````

# Case Study – Proteomics Analysis

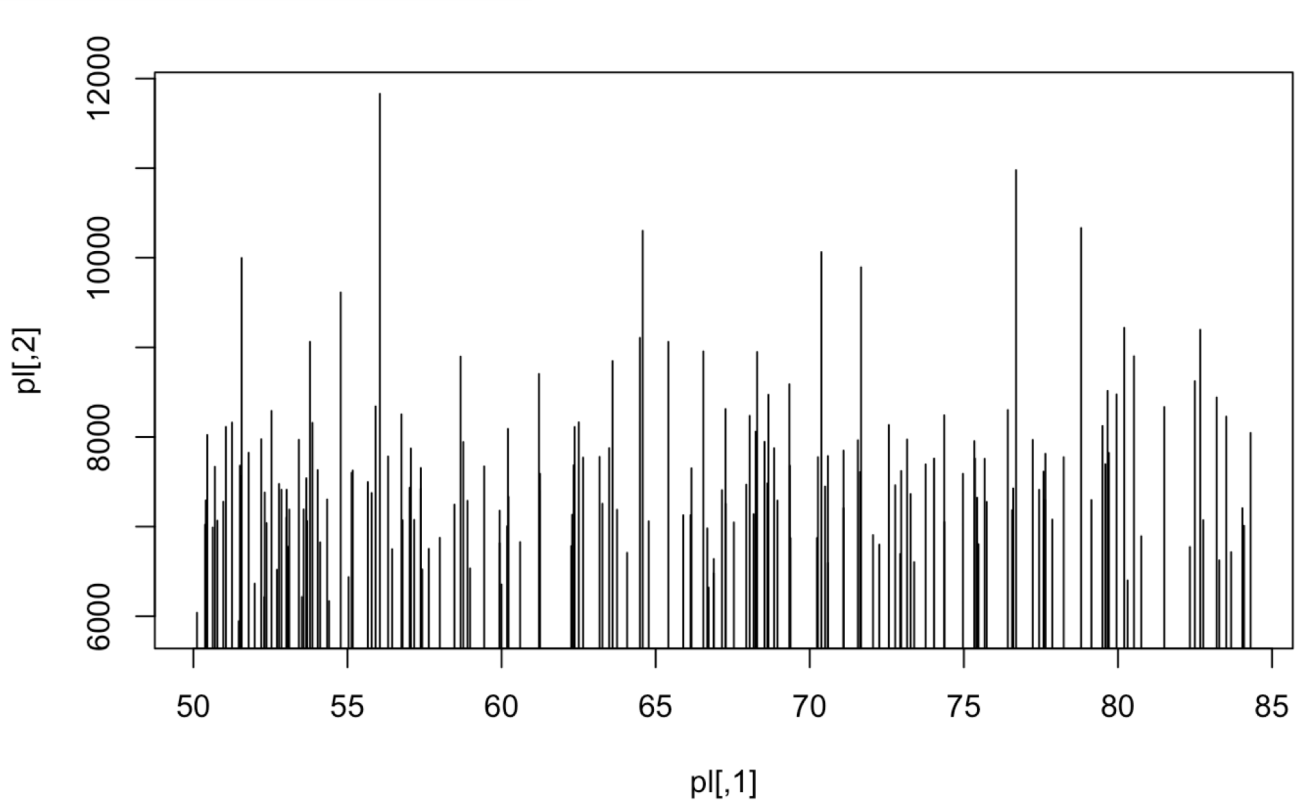
- Pick up the weight for each amino acids one by one and check the peakdiff matrix if we find the masses among the differences

```
# Take the weight for each aminoacids one by one and check the peakdiff matrix if we find the masses among the differences.

```{r}

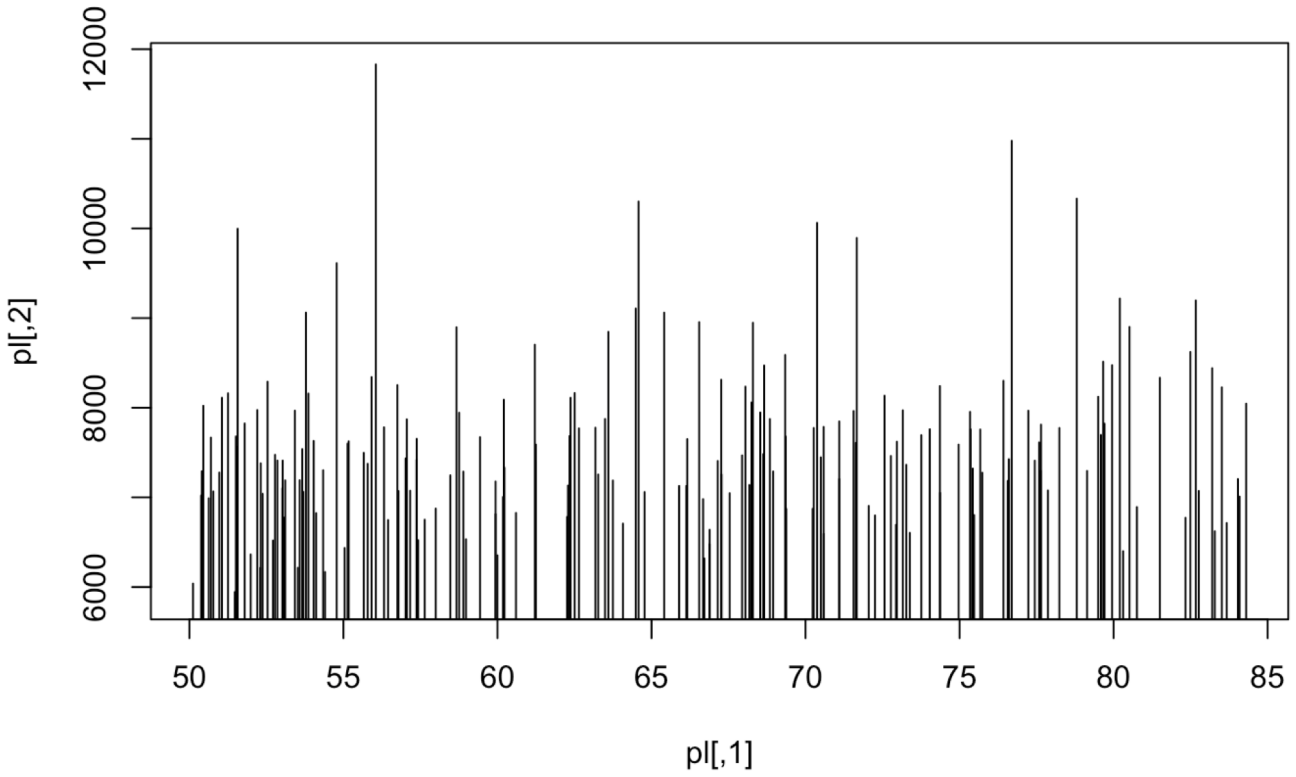
topnum <- 15
prec <- 0.04
pl.top <- pl[pl[,2]%in% head(sort(pl[,2], decreasing=T), topnum),1]
peakdiff <- outer(pl.top,pl.top,'-')
peakdiff.aa <- matrix(rep('-',topnum^2),ncol=topnum, nrow=topnum)
aa.mass <- read.csv("aa_mass.csv", row.names = 1)

ystep<-max(pl[,2])/25
ypos <- ystep*4
plot(pl,type="h")
for(aa in row.names(aa.mass))
  {
  peakdiff.aa[abs(peakdiff-aa.mass[aa,])<prec]<-aa
  aa.match <- which(abs(peakdiff-aa.mass[aa,])<prec,arr.ind=T)
  if(length(aa.match))
    {
    for(i in 1:dim(aa.match)[1])
        {
      segnebts(pl.top[aa.match[i,2]], ypos,
               pl.top[aa.match[i,1]],ypos)
      text(pl.top[aa.match[i,2]],ypos,aa)
        }
  }
  ypos <- ypos+ystep
}
```


```

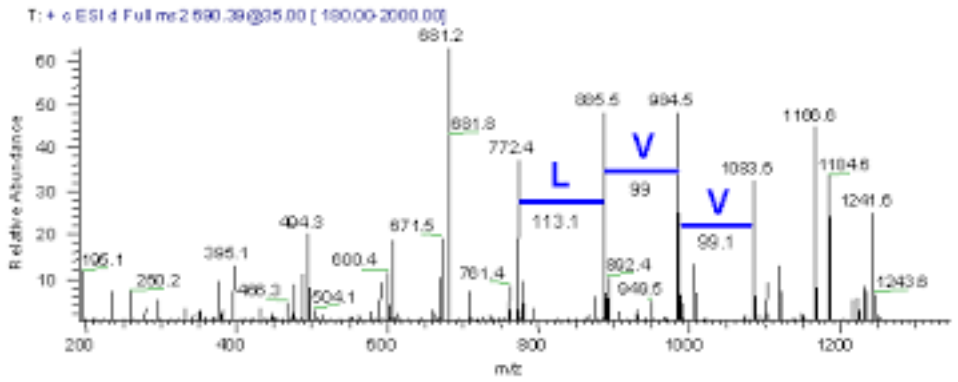
# Case Study – Proteomics Analysis

- The goal is to find segments representing the b (or y) ladder
- A sequence of amino acids next to each other.



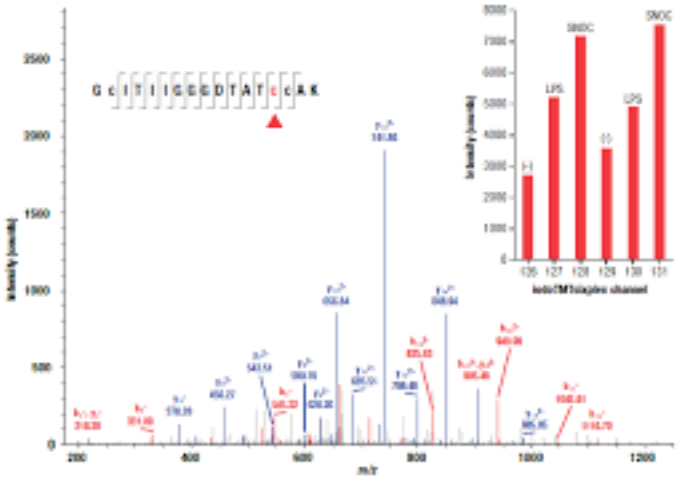
# Case Study – Proteomics Analysis

- Quantitative Proteomics:
- The relative amount of proteins is one possible research question in proteomics:
- Qualitative applications of MS have similar goals at the proteome level as differential gene expression studies have at the transcriptome level.



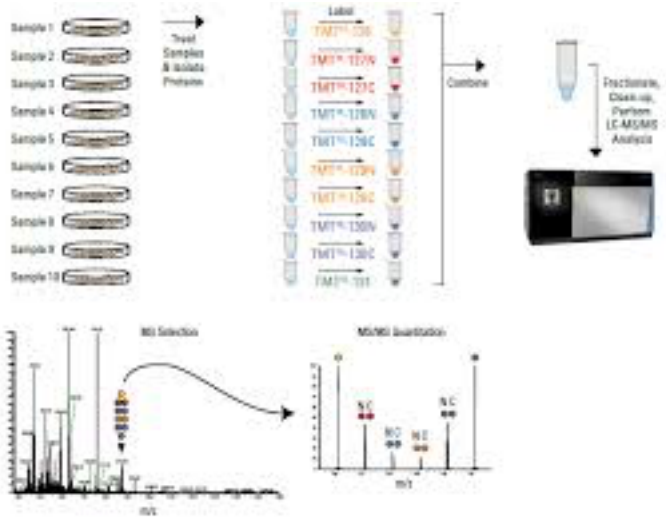
# Case Study – Proteomics Analysis

- One of the important uses of MS in proteomics is to measure changes in protein amounts between biological samples.
- Several methods are developed to measure the relative or absolute quantities.
- In practice there are two kinds of chemical groups, named tags, in use:
  - Tandem mass tags (TMT)
  - Isobaric tags for relative and absolute quantification (iTRAQ)



# Case Study – Proteomics Analysis

- TMT technology uses a pair or a set of six tag variants that can distinguish the samples in MS/MS measurements.
- The MSnbase package from Bioconductor is employed to show a possible analysis of experimental data where TMT 6-plex labeling was applied.



# Case Study – Proteomics Analysis

- Read the peptide and metadata section of the file:

````
188
189 # Rename the samples in the qnt object by applying the names of the TNT reporter ions:
190 ```{r}
191
192 sampleNames(qnt) <- c("TMT6.126", "TMT6.127", "TMT6.128", "TMT6.129", "TMT6.130", "TMT6.131")
193
194 ```
195
````

# Case Study – Proteomics Analysis

- Accessing the peptide abundance of the corresponding peaks in the six samples by employing the exprs() function

````
# Accessing the peptide abundance of the corresponding peaks in the six samples by employing the exprs() function
```{r}

head(exprs(qnt))

```
````

```
  TMT6.126 TMT6.127 TMT6.128 TMT6.129 TMT6.130 TMT6.131
1 10630132 11238708 12424917 10997763  9928972 10398534
2 11105690 12403253 13160903 12229367 11061660 10131218
3  1183431  1322371  1599088  1243715  1306602  1159064
4  5384958  5508454  6883086  6136023  5626680  5213771
5 18033537 17926487 21052620 19810368 17381162 17268329
6  9873585 10299931 11142071 10258214  9664315  9518271
```

# Case Study – Proteomics Analysis

- Accessing the peptide abundance of the corresponding peaks in the six samples by employing the exprs() function:

```
202
203 ```{r}
204
205 dim(exprs(qnt))
206
207 ```
```

```
[1] 1528    6
```

# Case Study – Proteomics Analysis

- Reaching the peptide abundances in the individual channels by using the index of the channel:

```
# Reaching the peptide abundances in the individual channels by using the index of the channel:
```{r}

exprs(qnt)[,1]

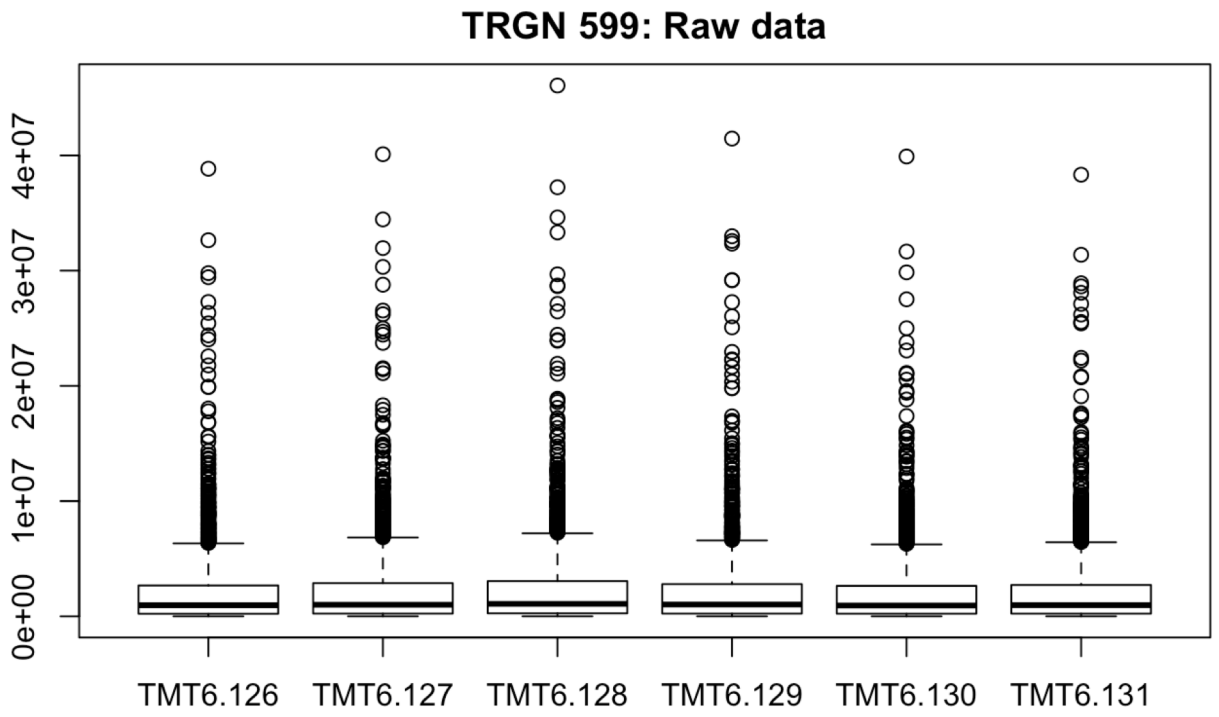
```
```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 10630132.000 | 11105690.027 | 1183431.000 | 5384958.500 | 18033537.324 | 9873585.085 | 7357824.338 | 27283373.297 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 11079607.500 | 1498217.750 | 10463639.000 | 11126745.840 | 15168991.156 | 9811475.459 | 2152136.828 | 9100719.938 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 60128.574 | 12445631.500 | 4895852.250 | 3566834.250 | 795269.602 | 2656228.125 | 1214927.336 | 34856.350 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 1941459.375 | 190413.923 | 10995.982 | 53441.062 | 7706883.000 | 5187416.500 | 1818689.750 | 26337004.321 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 4126961.750 | 12014520.050 | 6737835.176 | 17791696.000 | 5372948.500 | 7279512.194 | 4901220.500 | 1447231.250 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 633333.000 | 5967910.623 | 40153.027 | 4840065.328 | 9001770.734 | 150736.207 | 2126786.250 | 621263.875 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| 159293.797 | 2222457.906 | 1857580.840 | 6894593.344 | 467724.906 | 13618910.000 | 1530024.688 | 108055.711 |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |
| 1851247.500 | 1020487.700 | 3239931.413 | 9824023.000 | 2564436.500 | 4919526.703 | 3981098.151 | 4684487.000 |
| 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 |
| 5976195.250 | 1578169.875 | 4322170.500 | 6157737.854 | 1668680.750 | 2380747.750 | 1851065.750 | 1541885.250 |
| 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 837401.438 | 2583452.594 | 2203981.500 | 21243.061 | 42580.801 | 32303.850 | 394571.343 | 883539.250 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 |
| 418622.094 | 10835998.570 | 199696.188 | 2280394.000 | 1826664.043 | 5134070.000 | 151172.547 | 801033.188 |
| 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 |
| 143029.662 | 137236.281 | 2080865.125 | 1320006.500 | 374689.844 | 1116391.875 | 4849247.125 | 1477509.625 |
| 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 |
| 3540933.656 | 571517.438 | 88922.867 | 38845100.719 | 10566923.234 | 3811169.000 | 1675793.041 | 2975240.000 |
| 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 |
| 416909.031 | 5871582.910 | 371983.360 | 4310139.469 | 926682.156 | 6210380.295 | 2432723.912 | 1476854.750 |
| 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

# Case Study – Proteomics Analysis

- Looking the distribution of the measured values in the six channels:

```
215
216 # Looking the distribution of the measured values in the six channels:
217 ```{r}
218
219 boxplot(exprs(qnt), main="TRGN 599: Raw data")
220
221 ```
```



# Case Study – Proteomics Analysis

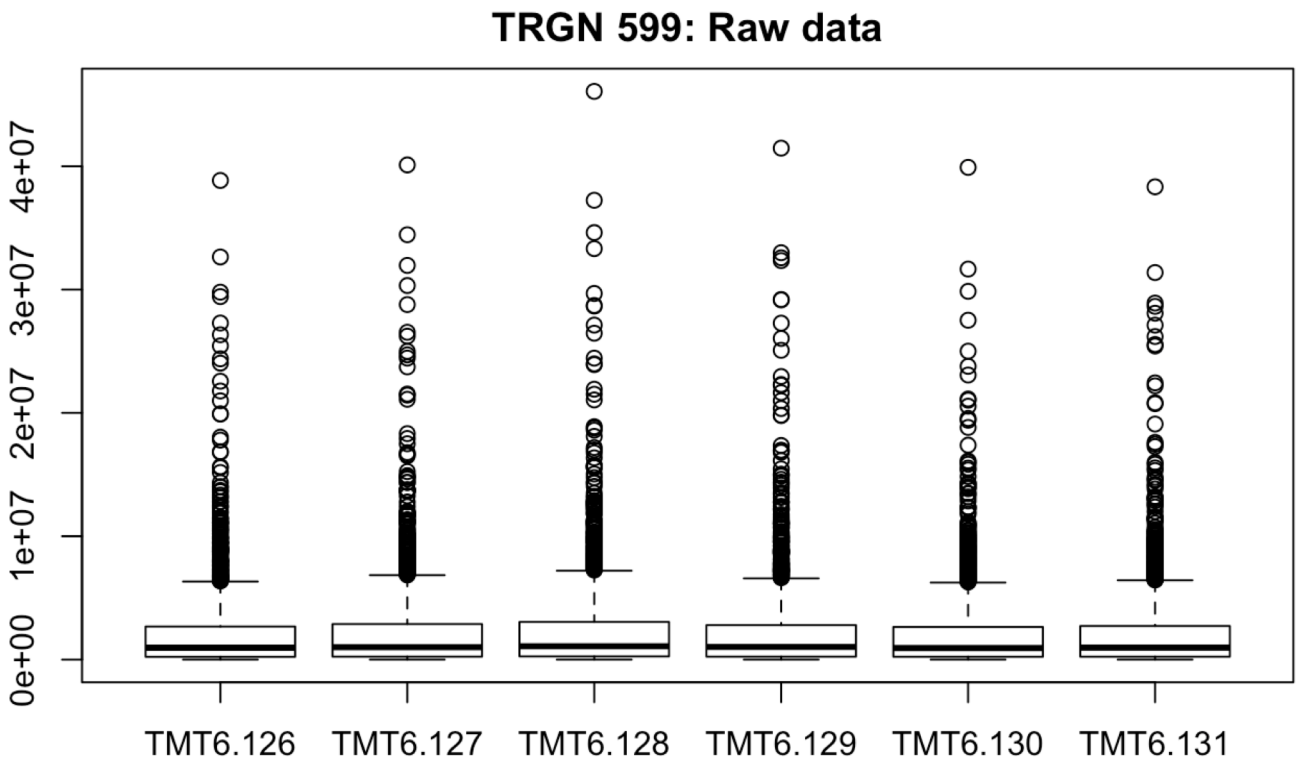
- Normalizing to make values comparable using the normalize() function:

```
222
223 # Normalizing to make values comparable using the normalize() function:
224 ```{r}
225
226 qntS <- normalise(qnt, "sum")
227 qntV <- normalise(qntS, "vsn")
228 qntV2 <- normalise(qnt, "vsn")
229
230 ```
```

# Case Study – Proteomics Analysis

- Generating boxplots:

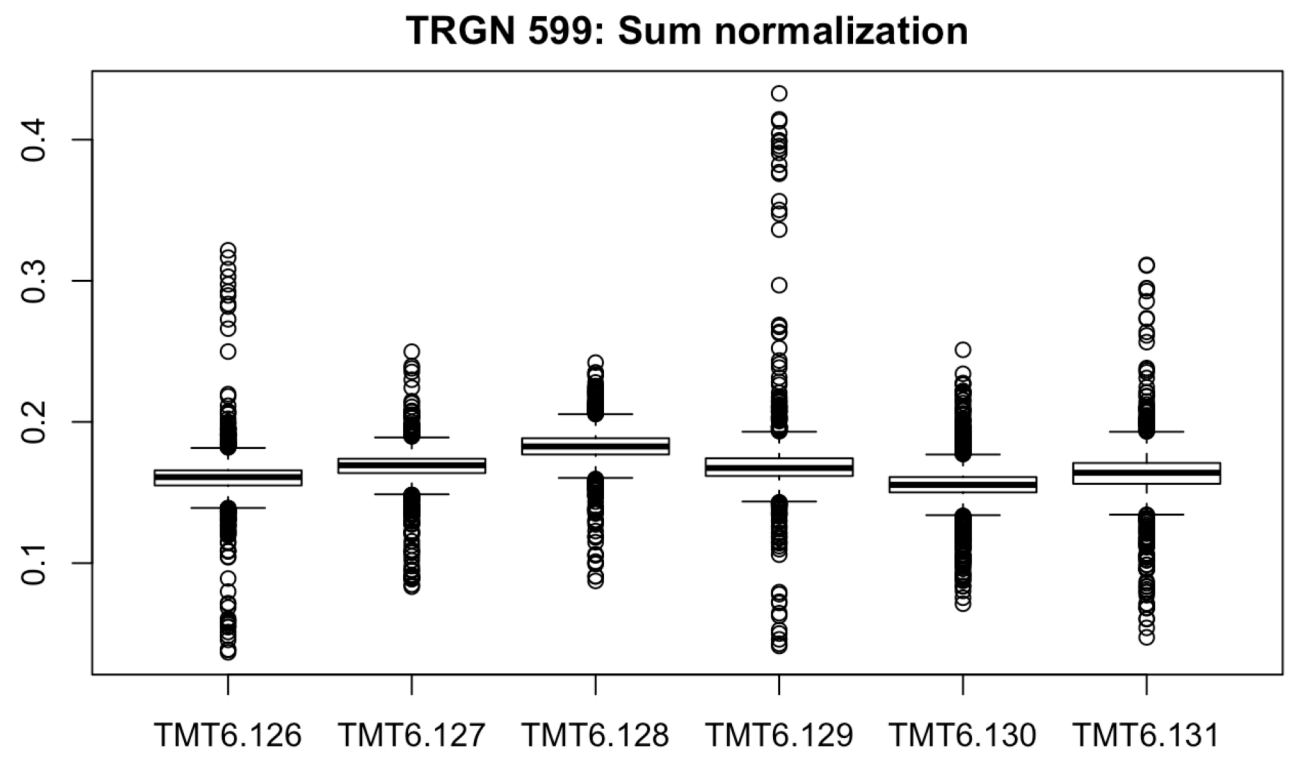
````
231
232 # Creating boxplots:
233 ```{r}
234
235 par(mfrow=c(2,2))
236 boxplot(exprs(qnt), main="TRGN 599: Raw data")
237
238 ```
````



# Case Study – Proteomics Analysis

- Generating boxplots:

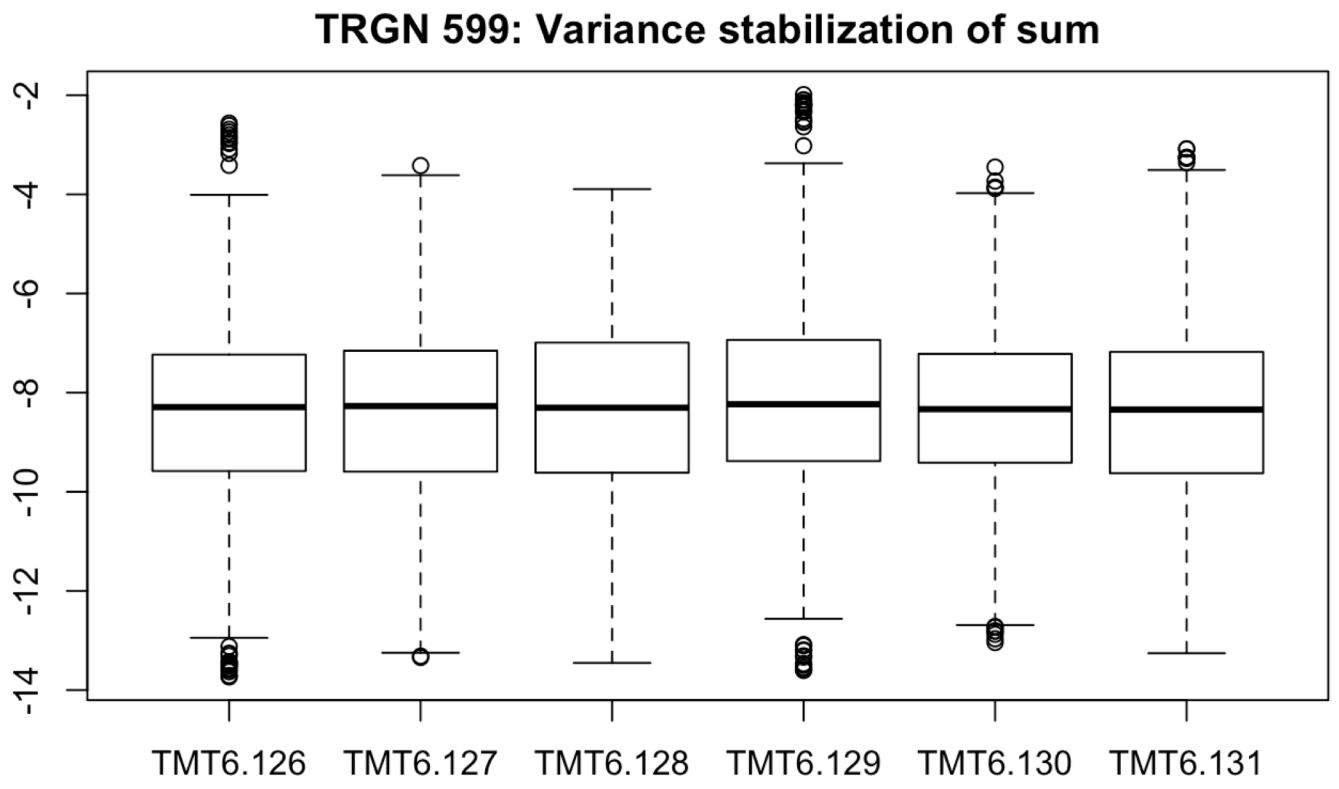
```
239
240 ```{r}
241
242 boxplot(exprs(qntS), main="TRGN 599: Sum normalization")
243
244 ```
```



# Case Study – Proteomics Analysis

- Generating boxplots:

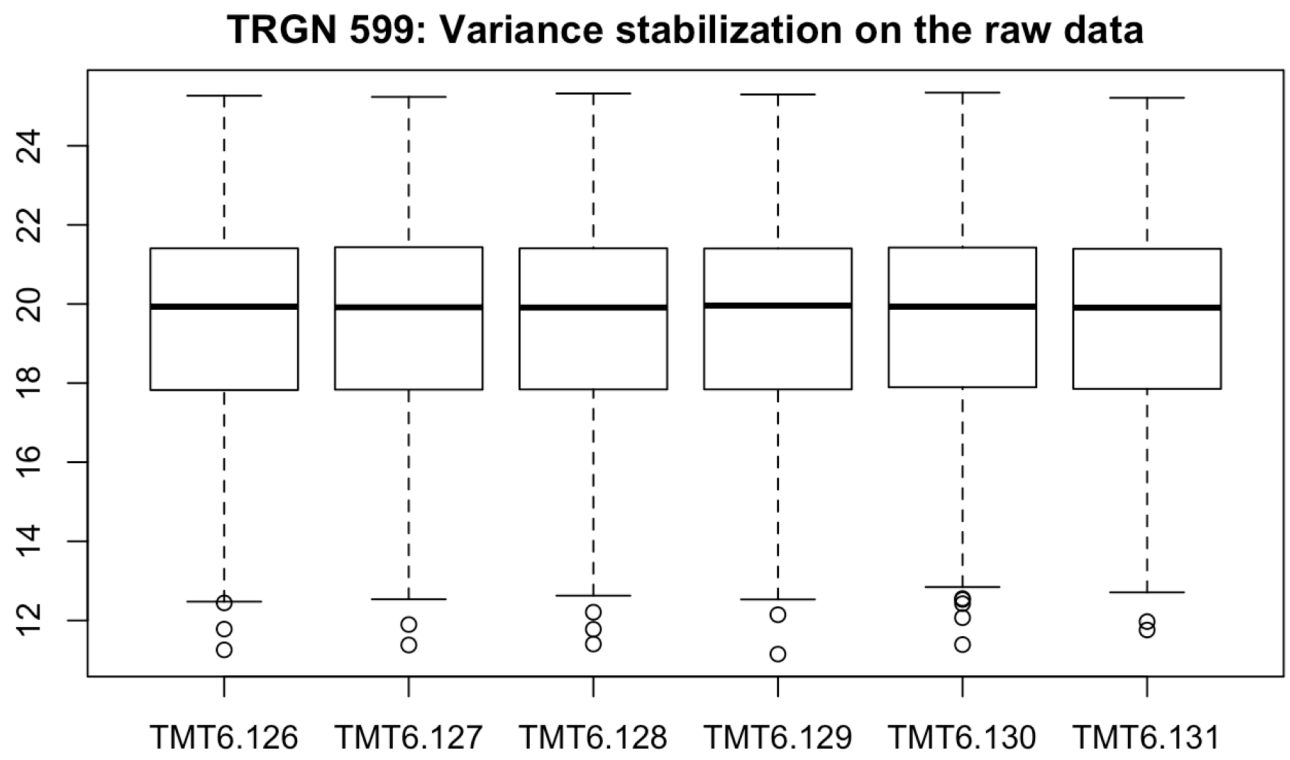
````
245
246 ```{r}
247
248 boxplot(exprs(qntV), main="TRGN 599: Variance stabilization of sum")
249
250 ```
````



# Case Study – Proteomics Analysis

- Generating boxplots:

```
252 ```{r}
253
254 boxplot(exprs(qntV2), main="TRGN 599: Variance stabilization on the raw data")
255
256 ```
```



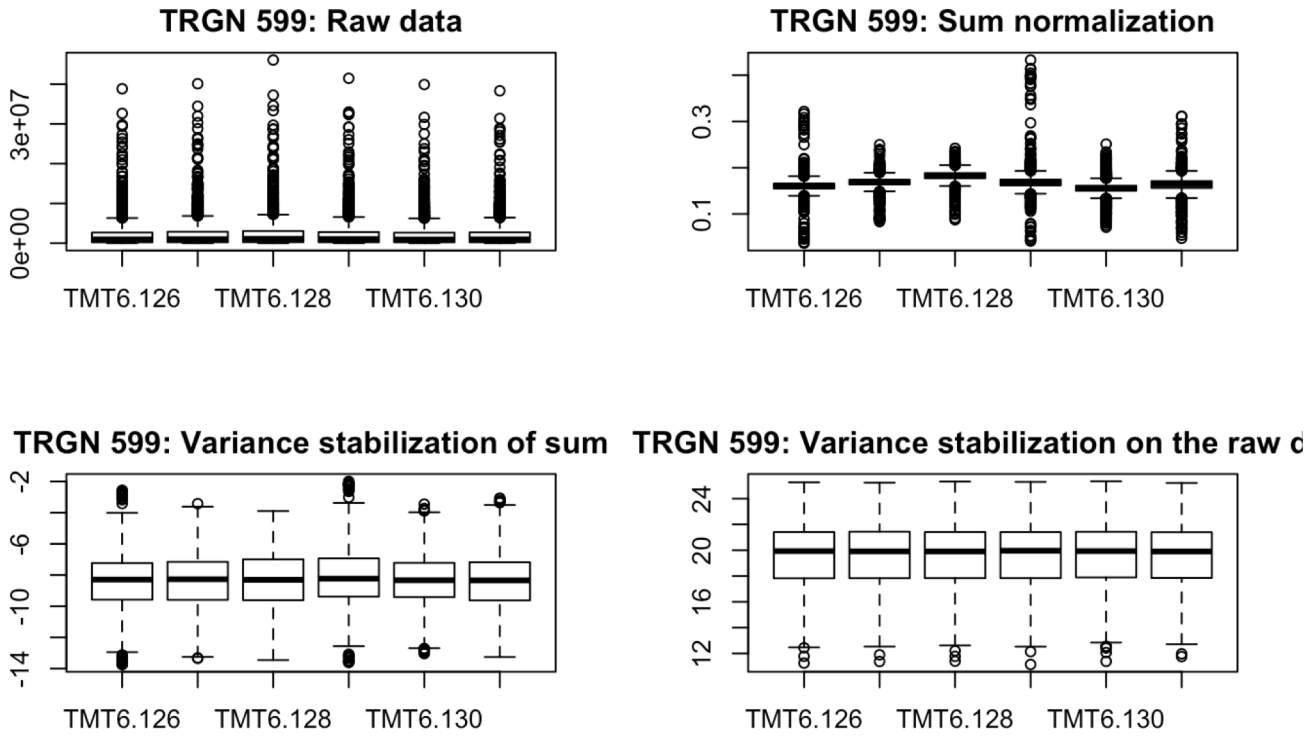
# Case Study – Proteomics Analysis

- Visualizing box pots together

```
# Visualizing box pots together
```{r}

par(mfrow=c(2,2))
boxplot(exprs(qnt), main="TRGN 599: Raw data")
boxplot(exprs(qntS), main="TRGN 599: Sum normalization")
boxplot(exprs(qntV), main="TRGN 599: Variance stabilization of sum")
boxplot(exprs(qntV2), main="TRGN 599: Variance stabilization on the raw data")
par(mfrow=c(1,1))

```
```



# Case Study – Proteomics Analysis

- # Note: VSN-normalized data can be used for comparative purposes.
- # Using the protein accession number to put the measures together for the different peptides of the same proteins
- # By employing the combineFeatures() function:

````
# Note: VSN-normalized data can be used for comparative puroses.
# Using the protein accession number to put the measures together for the different peptides of the same proteins
# By employing the combineFeatures() function:
```{r}

protqnt <-
 combineFeatures(qnt,
                 groupBy = fData(qnt)$accession, fun = sum)

```
````

# Case Study – Proteomics Analysis

- Same step that above but using normalized data:

````
280
281 # Same step that above but using normalized data
282 ```{r}
283
284 protqntV2 <-
285  combineFeatures(qntV2,
286                  groupBy = fData(qnt)$accession, fun = sum)
287
288 ```
````

# Case Study – Proteomics Analysis

- Replacing the cryptic names of the added proteins with their SwissProt ID:

````
289
290 # Replacing the cryptic names of the added proteins with their SwissProt ID
291 ```{r}
292
293 protexp <- exprs(protqnt)
294 row.names(protexp)[400:404] <- c("PYGM_RABIT", "TRYO_PIG", "ENO1_Y EAST", "ALBU_BOVIN", "CYC_BOVIN")
295
296 ```
````

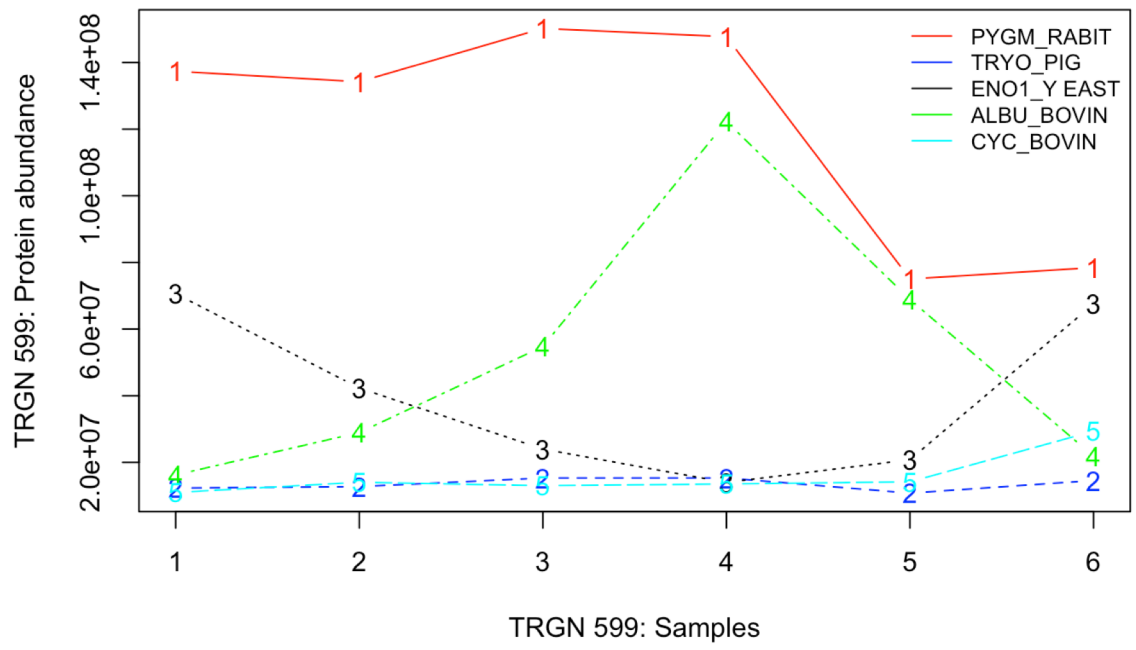
# Case Study – Proteomics Analysis

- Checking the relative protein amounts in the different samples for the added proteins:

```
# Checking the relative protein amounts in the different samples for the added proteins:
```{r}

matplot(t(protexp[400:404,]), type = "b", col=c("red", "blue", "black", "green", "cyan"), xlab="TRGN 599: Samples",
ylab="TRGN 599: Protein abundance")
legend("topright", row.names(protexp)[400:404], lty = 1,
      bty="n", cex = .8, col = c("red","blue","black","green","cyan"))


```
```



# Case Study – Proteomics Analysis

- Normalize data to visualize the similarity between proteins and samples by creating heatmaps:

```
307
308 # Normalize data to visualize the similarity between proteins and samples by creating heatmaps
309 ```{r}
310
311 wbcol <- colorRampPalette(c("white","darkblue"))(256)
312 heatmap(protexp[350:404,], col=wbcol)
313
314 ```
```

# Case Study – Proteomics Analysis

- Generating a Heat map:

```
# Normalize data to visualize the similarity between proteins and samples by creating heatmaps
```{r}

wbcol <- colorRampPalette(c("white","darkblue"))(256)
heatmap(protexp[350:404,], col=wbcol)

```
```