R_Notebook_ibrutinib_swath.Rmd

Code ▾

Somchai Chutipongtanate

January 11, 2020

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook version of ibrutinib_swath.R . In R Notebook, you can execute the code chunk by clicking the run button on the upper right corner of each chunk. The results will then appear beneath the code.

To use this script, please download and install R (version 3.4.4 or later) and RStudio (version 1.1.453 or later).

Once R and Rstuido installations finish, please open the file "R_Notebook_ibrutinib_swath.Rmd". Since this script needs functions from several R packages, the first step is to install all package dependencies below. This step can be skipped if all required packages (as shown below) have already been installed.

Hide

```
# Install package dependencies
install.packages(c("readxl", "dplyr", "tidyr", "ggplot2", "ggrepel", "reshape2", "
FactoMineR", "pheatmap"))
source("https://bioconductor.org/biocLite.R")
biocLite(c("biomaRt", "preprocessCore"))
```

Then, we load the required R packages.

Hide

```
# Load: R packages
library(readxl)
library(dplyr)
library(tidyr)
library(biomaRt)
library(preprocessCore)
library(ggplot2)
library(ggrepel)
library(reshape2)
library(FactoMineR)
library(pheatmap)
```

Data loading

The raw data (ibrutinib_SWATH.xlsx) is available via ProteomeXchange (PXD013402) and also downloadable as the supplementary dataset 1 once this dataset published. Please downlaod and place the dataset file on the desktop, so that it can be loaded into R.

Hide

```
# Load: ibrutinib-SWATH dataset (PXD013402)
setwd("~/Desktop")
data_path <- "~/Desktop/ibrutinib_SWATH.xlsx"
# Start: Data preprocess ---------------------------------------------------------
--------------------
## loading
group <- as.factor(c("WT", "WT", "WT", "WT+inh", "WT+inh", "WT+inh", "Q741x", "Q74
1x", "Q741x", "Q741x+inh", "Q741x+inh", "Q741x+inh"))
#group <- as.factor(c("W", "W", "W", "iW", "iW", "iW", "Q", "Q", "Q", "iQ", "iQ",
"iQ"))
group <- factor(group, ordered = TRUE,
                levels = c("Q741x+inh", "WT+inh", "Q741x", "WT"))
sample_label <- as.character(c("WT_1", "WT_2", "WT_3", "WT+inh_1", "WT+inh_2", "WT
+inh_3", "Q741x_1", "Q741x_2", "Q741x_3", "Q741x+inh_1", "Q741x+inh_2", "Q741x+inh
_3"))
#sample_label <- as.character(c("W1", "W2", "W3", "iW1", "iW2", "iW3", "Q1", "Q2",
"Q3", "iQ1", "iQ2", "iQ3"))
areaPept <- read_excel(data_path, sheet = "Area - peptides")
areaProt <- read_excel(data_path, sheet = "Area - proteins")
```

Now the SWATH data at peptide and protein levels are ready for downstream analyses.

Showing the first 10 rows of SWATH dataset at the peptide level;

Hide

```
head(areaPept, n = 10)
```

Protein <chr>	Peptide <chr>	Precursor MZ <dbl>	Precursor Charge <dbl>	RT <dbl> ▸
sp|Q8VDD5|MYH9_MOUSE	ALELDSNLYR	597	2	28.7
sp|Q8VDD5|MYH9_MOUSE	VSHLLGINVTDFTR	525	3	35.1
sp|Q8VDD5|MYH9_MOUSE	AGVLAHLEEER	409	3	22.2
sp|Q8VDD5|MYH9_MOUSE	LDPHLVLDQLR	440	3	33.9
sp|Q8VDD5|MYH9_MOUSE	VVFQEFR	463	2	25.2
sp|Q8VDD5|MYH9_MOUSE	LQQELDDLLVDLDHQR	651	3	43.4
sp|Q8VDD5|MYH9_MOUSE	SMEAEMIQLQEELAAAER	684	3	48.5
sp|Q8VDD5|MYH9_MOUSE	VIQYLAHVASSHK	364	4	18.5
sp|Q8VDD5|MYH9_MOUSE	YEILTPNSIPK	638	2	29.7
sp|P26039|TLN1_MOUSE	EQGVEEHETLLLR	518	3	22.7

1-10 of 10 rows | 1-5 of 17 columns

And at the protein level;

Hide

```
head(areaProt, n = 10)
```

Protein <chr>	020518 Somchai (Greis)_SWATH_W1 (Data020518_02.w
sp|Q8VDD5|MYH9_MOUSE	
sp|P26039|TLN1_MOUSE	
sp|Q68FD5|CLH1_MOUSE	
sp|P58252|EF2_MOUSE	
sp|P07901|HS90A_MOUSE	
sp|Q8BTM8|FLNA_MOUSE	
sp|P52480|KPYM_MOUSE	
sp|Q9JHU4|DYHC1_MOUSE	
sp|P20029|GRP78_MOUSE	
sp|P08113|ENPL_MOUSE	

1-10 of 10 rows | 1-2 of 13 columns

In this analysis, we use SWATH quantitative data at the protein level for downstream data processing.

Gene mapping

UniProt IDs can be mapped to gene names using useMart and getBM funcitons in BiomaRt package.

Hide

```
# Gene mapping using biomaRt package (ref#1)
df <- areaProt[ ,1] %>%
  tidyr::separate(Protein, c("sp", "uniProtID", "entry_name"), sep = "\\|") %>%
  tidyr::separate(entry_name, c("entry_names", "species"), sep = "_") %>%
  dplyr::select(uniProtID, entry_names, species)
ensembl <- useMart("ensembl", dataset="mmusculus_gene_ensembl",
                   host = "www.ensembl.org",
                   ensemblRedirect = FALSE)
tmp <- getBM(attributes = c('uniprotswissprot', 'external_gene_name'),
             filters = 'uniprotswissprot',
             values = df$uniProtID,
             mart = ensembl)
```

```
Batch submitting query [===============================================================
==>-------------------------------]  67% eta:  1s
Batch submitting query [===============================================================
=================================] 100% eta:  0s
```

Hide

```
colnames(tmp) <- c('uniProtID', "gene.SYMBOL")
df <- left_join(df, tmp[!duplicated(tmp$uniProtID), ], by = "uniProtID")
ind <- is.na(df$gene.SYMBOL)
df$gene.SYMBOL[ind] <- df$entry_names[ind]
id_all <- df
```

Hide

```
head(id_all, n = 10)
```

uniProtID	**entry_names**	**species**	**gene.SYMBOL**
<chr>	<chr>	<chr>	<chr>
Q8VDD5	MYH9	MOUSE	Myh9
P26039	TLN1	MOUSE	Tln1
Q68FD5	CLH1	MOUSE	Cltc
P58252	EF2	MOUSE	Eef2
P07901	HS90A	MOUSE	Hsp90aa1
Q8BTM8	FLNA	MOUSE	Flna
P52480	KPYM	MOUSE	Pkm
Q9JHU4	DYHC1	MOUSE	Dync1h1
P20029	GRP78	MOUSE	Hspa5
P08113	ENPL	MOUSE	Hsp90b1

1-10 of 10 rows

Quantile normalization and missing value handling

For data preprocessing, the normalize.quantiles function of preprocessCore package is applied, while missing values are replaced by zero.

Hide

```
## Quantile normalization using preprocessCore package (ref#2)
expr_raw <- areaProt[ , 2:length(areaProt)]
colnames(expr_raw) <- sample_label
Quantile <- as.data.frame(normalize.quantiles(log2(as.matrix(expr_raw))))
colnames(Quantile) <- sample_label
## Missing values replaced by zero
ind <- which(is.na(Quantile), arr.ind = TRUE)
Quantile[ind] <- 0
expr_processed <- Quantile
## Collect datasets
raw_ds <- cbind(id_all, expr_raw)
process_ds <- cbind(id_all, expr_processed)
df <- t(expr_processed)
colnames(df) <- id_all$gene.SYMBOL
log_ds <- data.frame(group, df)
# End: Data preprocess -----------------------------------------------------------
------------------
```

Once the preprocessing finished, we got the process dataset at the protein level, in which the quantitative data are expressed in log2 values.

Hide

```
head(process_ds, n=10)
```

	uniProtID	entry_names	species	gene.SYMB...	...	...	...	WT+in...	WT+in...
	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Q8VDD5	MYH9	MOUSE	Myh9	19.8	19.8	19.7	19.7	19.8
2	P26039	TLN1	MOUSE	Tln1	19.7	19.6	19.7	19.7	19.7
3	Q68FD5	CLH1	MOUSE	Cltc	19.9	20.4	20.0	20.4	20.4
4	P58252	EF2	MOUSE	Eef2	20.0	20.0	20.1	20.2	20.2
5	P07901	HS90A	MOUSE	Hsp90aa1	19.5	19.4	19.4	19.9	19.9
6	Q8BTM8	FLNA	MOUSE	Flna	17.4	17.1	17.3	17.5	17.1
7	P52480	KPYM	MOUSE	Pkm	20.7	20.6	20.7	21.1	20.9
8	Q9JHU4	DYHC1	MOUSE	Dync1h1	17.1	17.3	17.3	17.1	17.5
9	P20029	GRP78	MOUSE	Hspa5	19.3	19.0	19.4	19.3	19.2
10	P08113	ENPL	MOUSE	Hsp90b1	19.6	19.7	19.7	19.8	19.9

1-10 of 10 rows | 1-10 of 16 columns

Data quality check

The data quality is checked by several measures. The first one is %coefficient of variation (CV).

Hide

```
# Start: Data analysis and visualization -----------------------------------------
-----------------------------------
## Group average
tmp <- data.frame(group = log_ds[ , 1], 2^log_ds[ , 2:length(log_ds)]) %>%
  gather(gene.SYMBOL, expression, -group) %>%
  dplyr::group_by(group, gene.SYMBOL) %>%
  dplyr::summarize(group_mean = mean(expression)) %>%
  spread(gene.SYMBOL, group_mean)
gr_avr <- as.data.frame(tmp[ , 2:length(tmp)])
rownames(gr_avr) <- tmp$group
gr_pair <- combn(unique(tmp$group), 2)
fc <- (gr_avr[gr_pair[1, ], ] / gr_avr[gr_pair[2, ], ]) %>% log2()
rownames(fc) <- paste0('log2', '(', gr_pair[1, ], '/', gr_pair[2, ], ')')
log2fc_ds <- fc
## Group SD
tmp <-  data.frame(group = log_ds[ , 1], 2^log_ds[ , 2:length(log_ds)]) %>%
  gather(gene.SYMBOL, expression, -group) %>%
  dplyr::group_by(group, gene.SYMBOL) %>%
  dplyr::summarize(group_sd = sd(expression)) %>%
  spread(gene.SYMBOL, group_sd)
gr_sd <- as.data.frame(tmp[ , 2:length(tmp)])
rownames(gr_sd) <- tmp$group
## Coefficient of variation
qc <- 100 *gr_sd/gr_avr
qc <- data.frame(group = tmp$group, qc)
QC <- qc %>% gather(gene, CV, -group)
# Calculate median-CV of each group
medianCV <- QC %>% dplyr::group_by(group) %>% summarise(CV = round(median(CV), 1))
```

Median-CVs for each group;

Hide

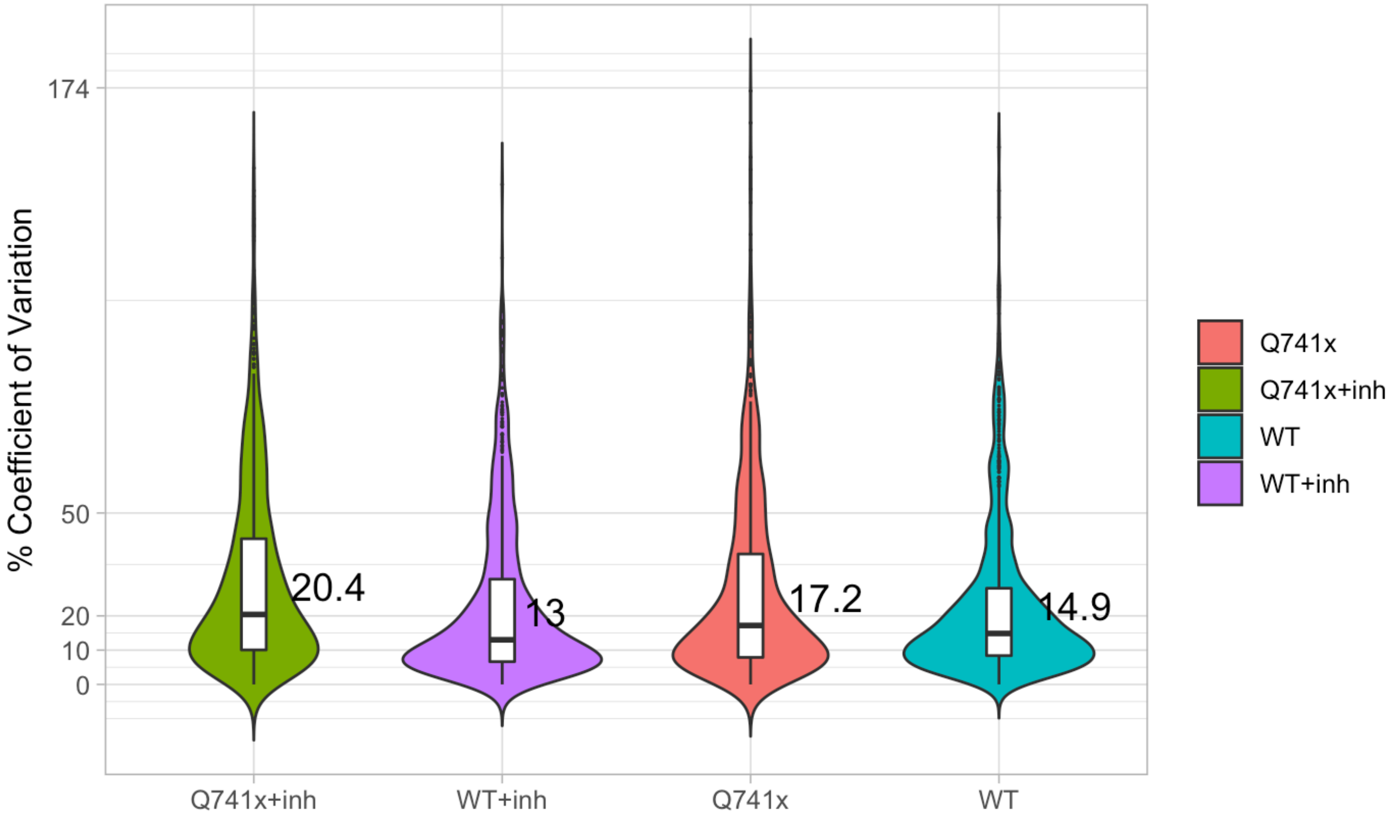
```
print(paste0("Median-CV: Q741x+inh, ", medianCV[1,2], "%; WT+inh, ", medianCV[2,2]
, "%; Q741x, ", medianCV[3,2], "%; WT, ", medianCV[4,2], "%"))
```

```
[1] "Median-CV: Q741x+inh, 20.4%; WT+inh, 13%; Q741x, 17.2%; WT, 14.9%"
```

Violin plot of inter-group CV

Hide

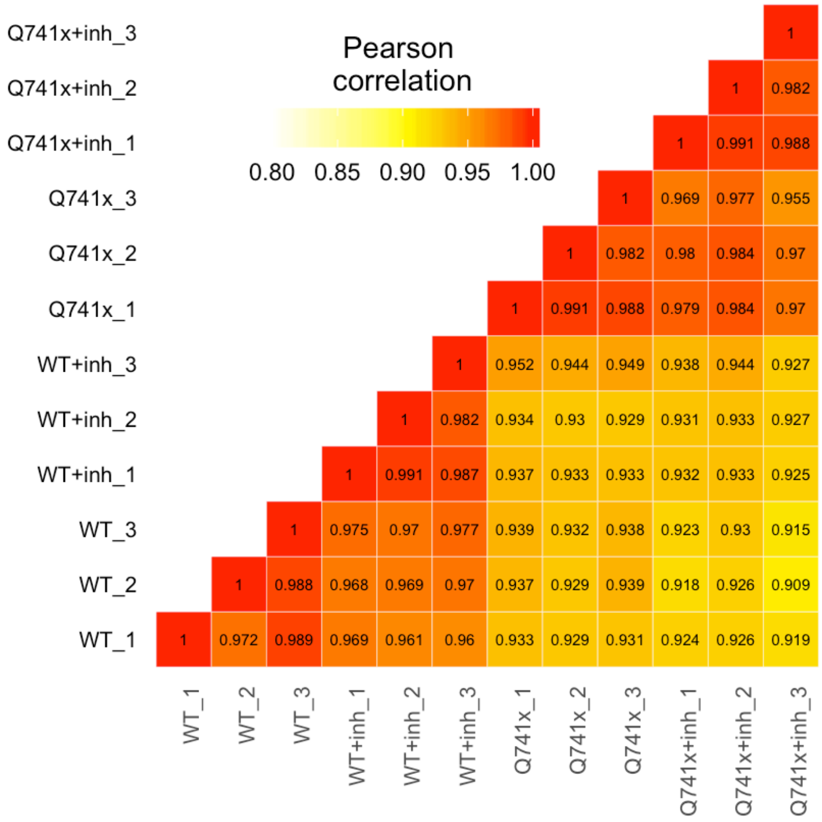
```
# Violin plot of inter-group CV
plot.qc <- ggplot(QC, aes(x=group, y=CV)) +
              geom_violin(aes(fill = as.character(group)), trim=FALSE, width = 0.8
, #aes(fill = group),
                          na.rm = TRUE, position = "dodge")+
              labs(fill = "") +
              geom_boxplot(width=0.1, fill = 'white', outlier.size = 0,
                          na.rm = TRUE, position = "dodge")+
              geom_text(data = medianCV, aes(label = CV), position = position_dodg
e(width = 1),
                          hjust = -0.5, vjust = -0.5, size = 5) +
              xlab("") + ylab("% Coefficient of Variation") +
              scale_y_continuous(breaks=c(0, 10, 20, 50, ceiling(max(QC$CV, na.rm=
TRUE)))) +
              theme_light(base_size = 12)
plot.qc
```



Correlation heatmap

Hide

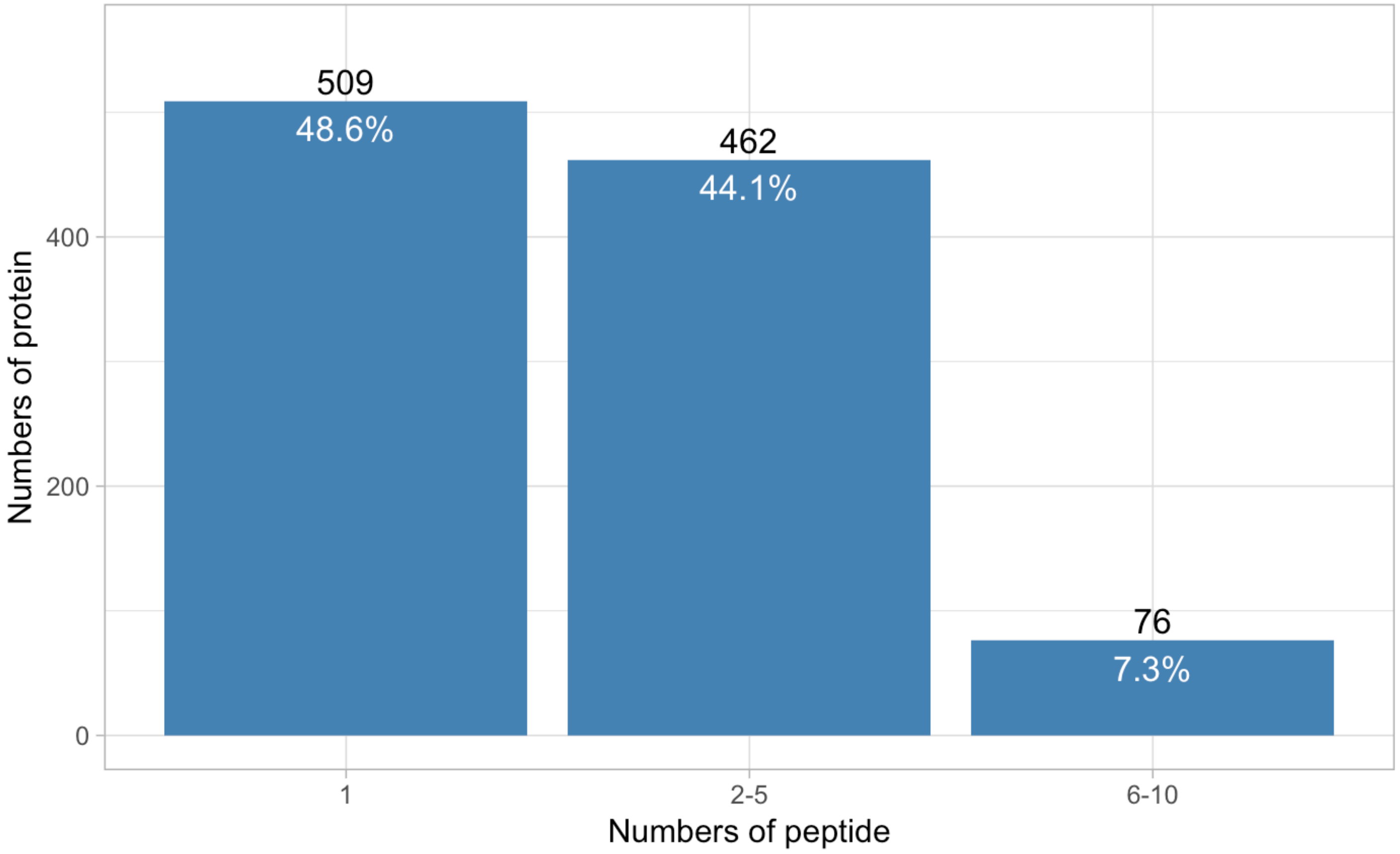
```
## Correlation heatmap
corr <- 2^expr_processed
corr <- round(cor(corr, method = "pearson"),3)
corr[lower.tri(corr)] <- NA
melted_corr <- melt(corr, na.rm = TRUE)
plot_corrHM <- ggplot(data = melted_corr, aes(x = Var2, y = Var1, fill = value))+
                  geom_tile(color = "white")+
                  scale_fill_gradient2(low = "white", high = "red", mid = "yellow"
,
                                       midpoint = 0.9, limit = c(0.8, 1), space =
"Lab",
                                       name= paste("Pearson", "\ncorrelation") ) +
                  labs(x = "", y = "") +
                  theme_minimal() +
                  theme(axis.text.x = element_text(angle = 90, vjust = 1,
                                                   size = 8, hjust = 1)) +
                  coord_fixed() +
                  geom_text(aes(label = value), color = "black", size = 2) +
                  theme(axis.text.y = element_text(color = "black", size=8),
                        panel.grid.major = element_blank(),
                        panel.border = element_blank(),
                        panel.background = element_blank(),
                        axis.ticks = element_blank(),
                        legend.justification = c(1, 0),
                        legend.position = c(0.6, 0.7),
                        legend.direction = "horizontal")+
                  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                               title.position = "top", title.hjust = 0.5))
plot_corrHM
```



The plot for the numbers of peptide per protein

Hide

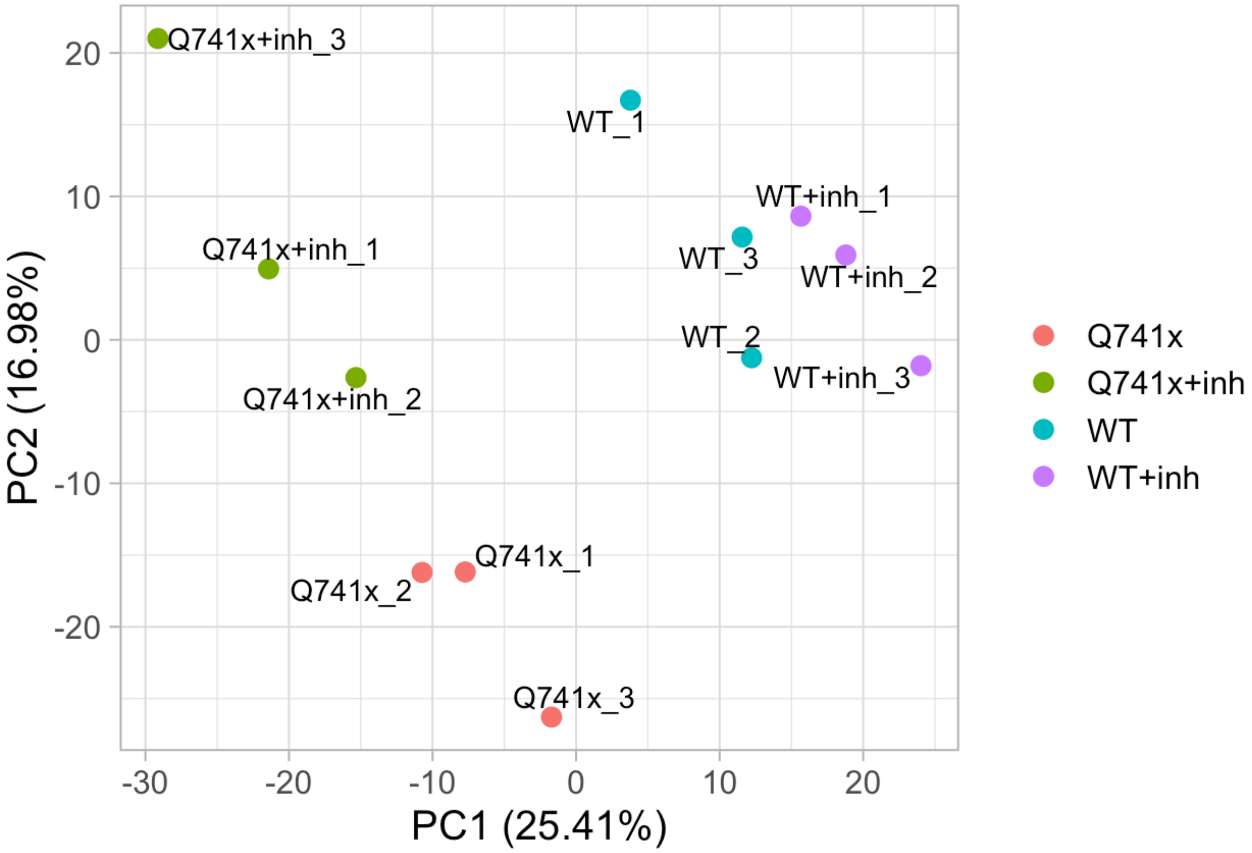
```
## nPP plot
n_pept_prot <- areaPept %>%
  dplyr::group_by(Protein) %>%
  dplyr::summarize(n_pept = n()) %>%
  arrange(desc(n_pept))
nPP <- data.frame(n_pept = c("1", "2-5", "6-10"),
                    n_prot = rbind(n_pept_prot %>% filter(n_pept ==1) %>% nrow(),
                                   n_pept_prot %>% filter(n_pept >=2 & n_pept <= 5
) %>% nrow(),
                                   n_pept_prot %>% filter(n_pept >=6) %>% nrow()))
nPP_plot <- ggplot(nPP, aes(x = n_pept, y= n_prot)) +
                    geom_bar(stat = "identity", fill = "steelblue") +
                    ylim(0, max(nPP$n_prot)+50) +
                    geom_text(aes(label= n_prot), vjust=-0.3, color="black", size=
4.5) +
                    geom_text(aes(label= paste0(round(100*n_prot/sum(n_prot), 1),
"%")), vjust=1.6, color="white", size=4.5) +
                    xlab("Numbers of peptide") + ylab("Numbers of protein") +
                    theme_light(base_size = 12)
nPP_plot
```



PCA individual plot

Hide

```
## PCA individual plot using FactorMineR package (ref#3)
fit_pca <- PCA(log_ds[ , 2:length(log_ds)], graph = FALSE, scale.unit = TRUE)
percentage <- fit_pca$eig[ , 2]
PCs <- data.frame(fit_pca$ind$coord)
PCs$group <- as.character(group)
plotPCA <- ggplot(data = PCs, aes(x = Dim.1, y = Dim.2)) +
              geom_point(aes(colour = group), size = 3) +
              labs(colour = '') +
              xlab(paste0('PC1', ' ', '(', round(percentage[1], 2), '%)')) +
              ylab(paste0('PC2', ' ', '(', round(percentage[2], 2), '%)')) +
              scale_fill_hue(l=40) +
              coord_fixed(ratio=1, xlim=range(PCs$Dim.1), ylim=range(PCs$Dim.2)) +
              geom_text_repel(label = rownames(PCs)) +
              theme_light(base_size = 15)
plotPCA
```



NOTE: The contributions of protein variables of each component can be extracted from the fit_pca object for in-depth biological interpretation.

Hide

```
head(fit_pca[["var"]][["contrib"]], n = 10)
```

```
          Dim.1   Dim.2   Dim.3    Dim.4    Dim.5
Myh9     0.0607 0.35234 0.00352 0.005935 0.029067
Tln1     0.2853 0.04184 0.00242 0.012616 0.015562
Cltc     0.0121 0.38603 0.11048 0.060218 0.053396
Eef2     0.3336 0.02708 0.00107 0.001344 0.009143
Hsp90aa1 0.0122 0.41772 0.10758 0.000423 0.005738
Flna     0.2242 0.02023 0.03532 0.170729 0.024465
Pkm      0.1944 0.08087 0.14988 0.008074 0.001413
Dync1h1  0.0131 0.01155 0.02371 0.354755 0.080413
Hspa5    0.0934 0.00388 0.00606 0.005852 0.601517
Hsp90b1  0.2801 0.09564 0.02236 0.001281 0.000225
```

Lastly, the protein abundance heatmap (values in the log10 scale) where the missing values are mapped in black color.

Hide

```
## Protein abundance heatmap by pheatmap package (ref#4)
qc_hm <- 2^expr_processed
rownames(qc_hm) <- process_ds$gene.SYMBOL
for(i in seq_along(qc_hm)){
  if(qc_hm[i] != 0){
    qc_hm[i] <- log10(qc_hm[i])
  } else {
    qc_hm[i] <- 0
  }}
n_missing <- sum(qc_hm == 0)
n_total <- dim(qc_hm)[1] * dim(qc_hm)[2]
```

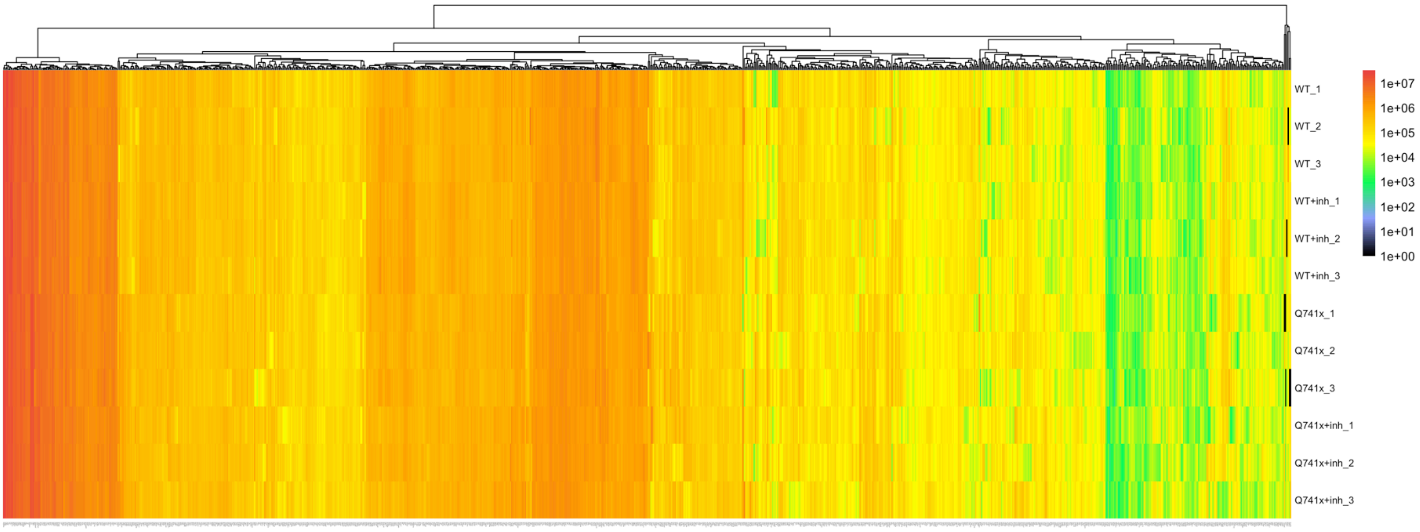
Hide

```
print(paste0("QC_heatmap: Total ", n_total, " data points; ", n_missing, " missing
values (", round(100*n_missing/n_total, 2), "%) showed in black)"))
```

```
[1] "QC_heatmap: Total 12564 data points; 7 missing values (0.06%) showed in black
)"
```

Hide

```
qc_hm_plot <- pheatmap(t(qc_hm),
                         breaks = seq(0, max(qc_hm), length.out=101),
                         legend_breaks = seq(0, round(max(qc_hm), 0), length.out=8
),
                         legend_labels = c("1e+00", "1e+01", "1e+02", "1e+03", "1e
+04", "1e+05", "1e+06", "1e+07"),
                         color = colorRampPalette(c("black", "#8ea1ff", "#14ff57",
"yellow",    "orange", "#ea4444"))(100),
                         border_color = "gray",
                         clustering_distance_cols = "maximum",
                         clustering_method_columns = "complete",
                         cluster_rows = FALSE,
                         fontsize_row = 8, fontsize_col = 1,
                         scale = "none")
```



Data analysis and visualization

Differntial expression analysis for multiple group comparison is performed by ANOVA with Tukey's post-hoc.

Hide

```
## ANOVA with Tukey's post-hoc
tmp <- as.matrix(log_ds[, 2:length(log_ds)])
fit.aov <- aov(tmp ~ group)
output.aov <- summary.aov(fit.aov)
anova.pVal <- numeric(length = ncol(tmp))
for (i in 1:length(output.aov)){
  anova.pVal[i] <- output.aov[[i]][1, 5]
}
adj.pVal <- matrix(nrow = ncol(tmp), ncol = nrow(log2fc_ds))
colnames(adj.pVal) <-  paste(gr_pair[1, ], " vs ", gr_pair[2, ])
rownames(adj.pVal) <- colnames(tmp)
for (i in 1:ncol(tmp)){
  adj.pVal[i, ] <- (TukeyHSD((aov(tmp[, i] ~ group))))[[1]][ ,4]
}
anova_ds <- cbind(anova.pVal, adj.pVal)
```

The ANOVA p-values (the first column) and the adjusted p-values from Tukey's posthoc for each pairwise comparison (as labelled in the column name) are ready for further use.

Hide

```
head(anova_ds, n=10)
```

```
         anova.pVal Q741x+inh  vs  WT+inh Q741x+inh  vs  Q741x Q741x+inh  vs  WT W
T+inh  vs  Q741x WT+inh  vs  WT Q741x  vs  WT
Myh9       2.75e-02              0.538046               0.2710           0.49571
3.95e-02         0.9998      0.035361
Tln1       2.33e-03              0.004109               0.6777           0.00979
1.76e-02         0.8966      0.045834
Cltc       1.50e-03              0.896792               0.0110           0.25091
2.80e-02         0.0970      0.000996
Eef2       6.85e-05              0.000107               0.5638           0.00106
3.54e-04         0.1473      0.004905
Hsp90aa1   1.60e-02              0.719928               0.0536           0.94154
1.25e-02         0.4196      0.117739
Flna       2.88e-02              0.161970               0.9625           0.10916
8.33e-02         0.9914      0.055835
Pkm        1.35e-03              0.002028               1.0000           0.36323
2.01e-03         0.0176      0.360103
Dync1h1    6.30e-01              0.999364               0.7075           0.95402
6.42e-01         0.9197      0.937132
Hspa5      4.80e-01              0.417546               0.7729           0.73844
9.08e-01         0.9302      0.999893
Hsp90b1    5.01e-05              0.000157               0.8479           0.00501
7.97e-05         0.0407      0.001931
```

Data including the fold changes and the adjusted p-values of proteins in each pairwise are ready for the volcano plots.

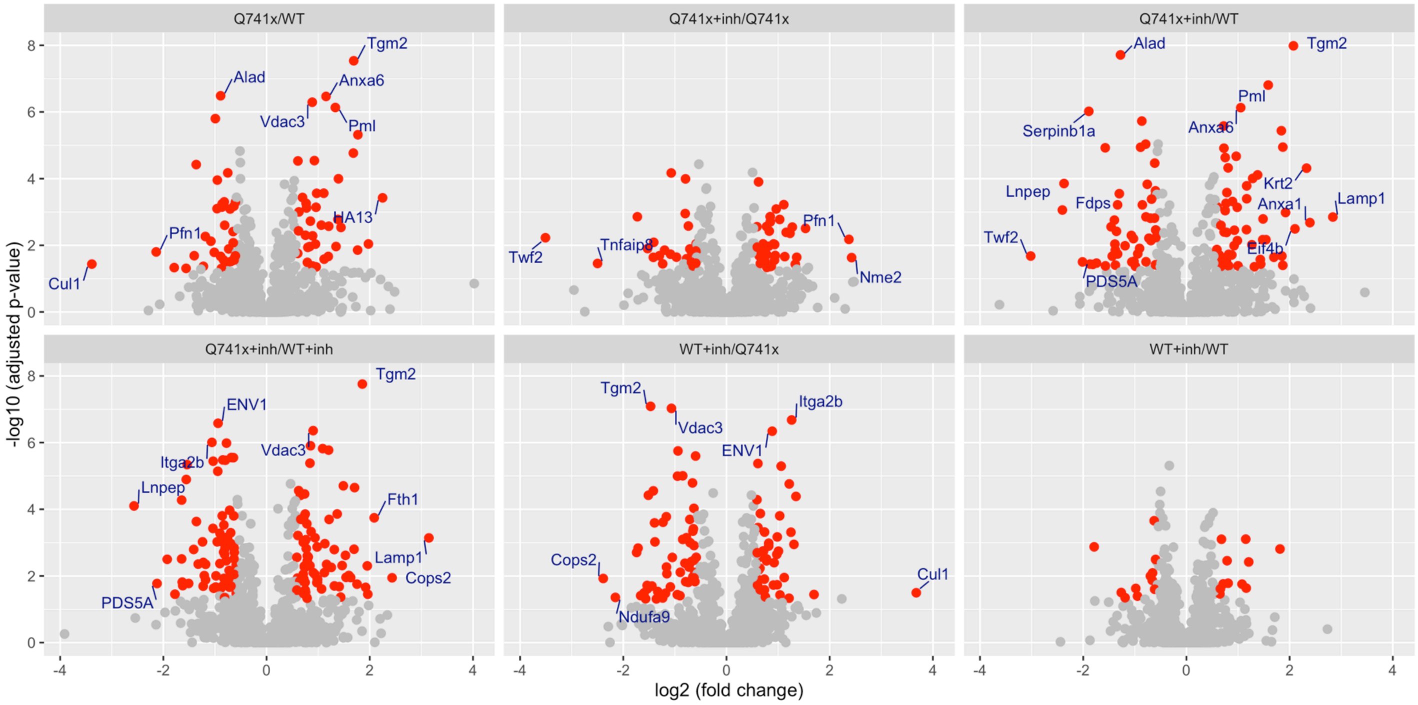
Hide

```
## Pairwise-Volcano plot
tmp <- data.frame(gene = rownames(anova_ds), anova_ds)
colnames(tmp) <- c("gene", "anova.pVal", paste0(gr_pair[1, ], "/", gr_pair[2, ]) )
long_ano <- gather(tmp, compare, adj_pVal, -gene, -anova.pVal)
fc.vp <- t(log2fc_ds)
fc.vp <- data.frame(gene = colnames(log2fc_ds), fc.vp)
colnames(fc.vp) <- c("gene", paste0(gr_pair[1, ], "/", gr_pair[2, ]) )
long_fc <- gather(fc.vp, compare, log2FC, -gene)
long_ano.fc <- long_ano %>%
  left_join(long_fc, by = c("gene", "compare"))
long_ano.fc$gene <- as.character(long_ano.fc$gene)
```

Here is the mulitple pairwise volcano plots, where the red dots represent the relevant proteins based on the thresholds of >1.5x fold change and the adjusted p-value <0.05;

Hide

```
volcano_all <- ggplot(data = long_ano.fc, aes(x= log2FC, y=-log10(adj_pVal))) +
                    geom_point(aes(color = as.factor(abs(log2FC) >= log2(1.5) & an
ova.pVal < 0.05 & adj_pVal < 0.05)), size = 3,  show.legend = FALSE) + #alpha = 0.
5,
                    scale_color_manual(values = c("grey", "red")) +
                    xlab("log2 (fold change)") + ylab("-log10 (adjusted p-value)")
+
                   # ggtitle(label = paste0("Volcano plot at ", 1.5,
                    #                        "x fold change and adjusted P-value <
", 0.05)) +
                    theme_grey(base_size = 15) +
                    geom_text_repel(data = (subset(long_ano.fc,
                                    abs(log2FC) > 2 & -log10(adj_pVal) > 1.33 | -l
og10(adj_pVal) > 6)),
                                    aes(label = gene, size = 0.1),
                                    show.legend = FALSE,
                                    colour = 'darkblue',
                                   # box.padding = unit(0.35, "lines"),
                                    point.padding = unit(0.5, "lines")
                                    ) +
                    facet_wrap(~ compare)
volcano_all
```



The lists of relevant proteins can be extracted from the long.ano.fc object;

Hide

```
head(long_ano.fc, n=10)
```

	gene <chr>	anova.pVal <dbl>	compare <chr>	adj_pVal <dbl>	log2FC <dbl>
1	Myh9	2.75e-02	Q741x+inh/WT+inh	0.538046	0.0935

2	Tln1	2.33e-03	Q741x+inh/WT+inh	0.004109	-0.5089
3	Cltc	1.50e-03	Q741x+inh/WT+inh	0.896792	-0.0690
4	Eef2	6.85e-05	Q741x+inh/WT+inh	0.000107	-0.7151
5	Hsp90aa1	1.60e-02	Q741x+inh/WT+inh	0.719928	-0.1858
6	Flna	2.88e-02	Q741x+inh/WT+inh	0.161970	0.4186
7	Pkm	1.35e-03	Q741x+inh/WT+inh	0.002028	-0.4366
8	Dync1h1	6.30e-01	Q741x+inh/WT+inh	0.999364	0.0309
9	Hspa5	4.80e-01	Q741x+inh/WT+inh	0.417546	-0.1744
10	Hsp90b1	5.01e-05	Q741x+inh/WT+inh	0.000157	-0.4749

1-10 of 10 rows

For example, a list of 61 relevent proteins (in gene names) can be extracted from the Q741x+inh/Q741x condition at the threshold of >1.5x fold changes and adj.pVal < 0.05. The protein list can be used for further biological interpretation;

Hide

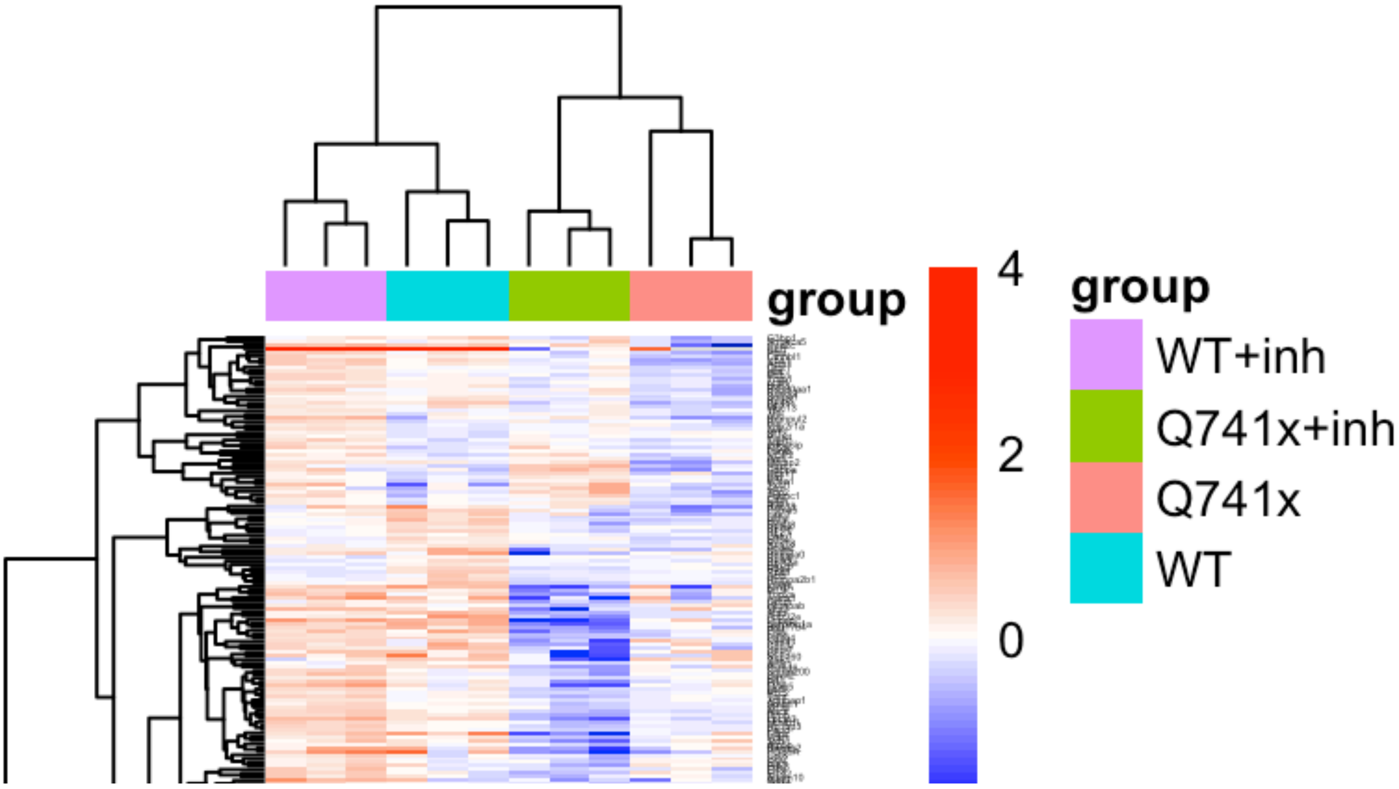
```
long_ano.fc %>% filter(compare == "Q741x+inh/Q741x") %>% filter(abs(log2FC) >= log
2(1.5)) %>% filter(adj_pVal < 0.05) %>% .$gene
```

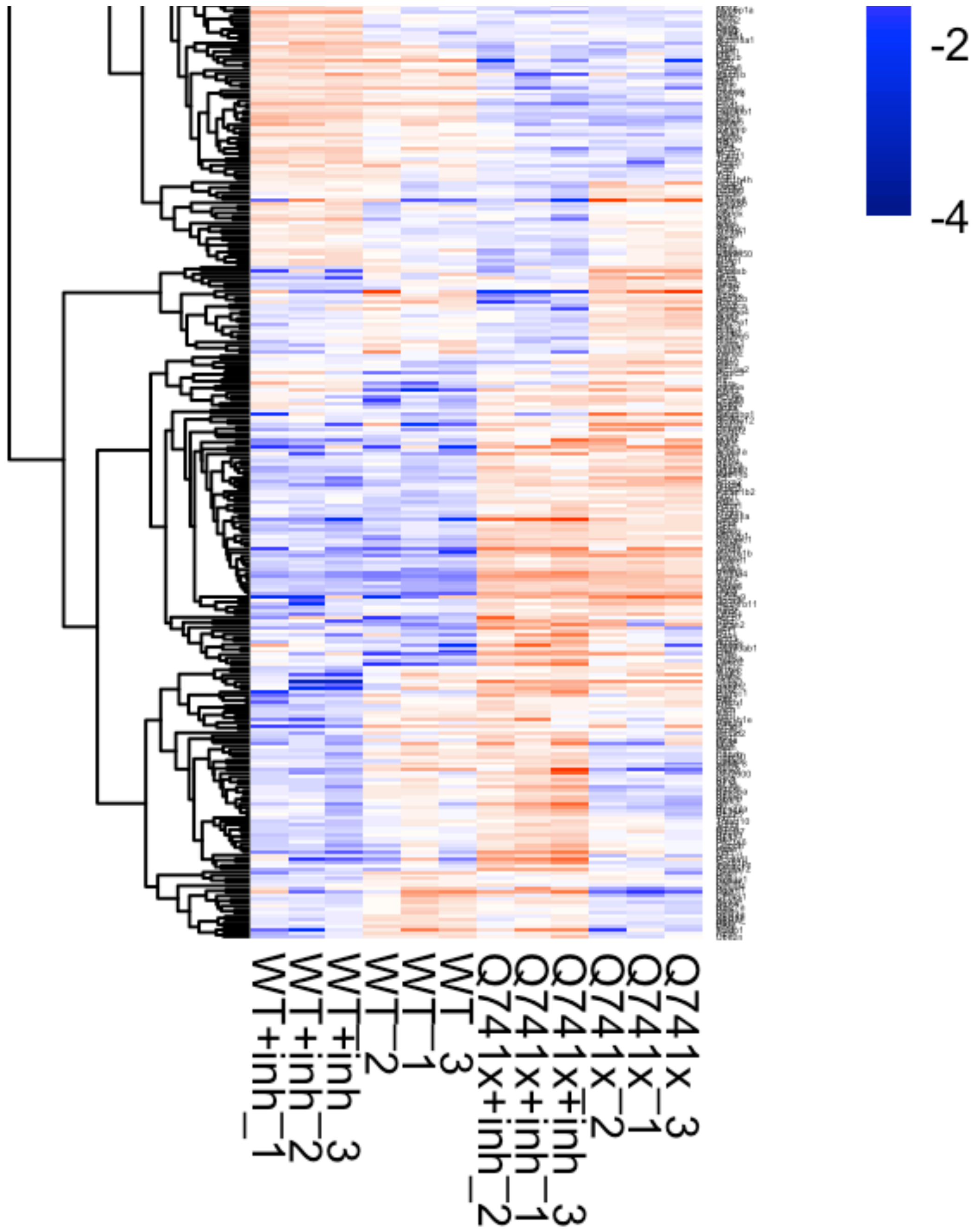
```
 [1] "Uba1"       "Iqgap1"     "Pabpc1"     "Serpinb1a" "Rps4x"      "Pgam1"      "Mdh2
"       "Phb2"       "Copb2"      "Gsn"        "Etfa"      
[12] "Pfn1"       "Capzb"      "RL10A"      "Psmd6"      "Hsd17b4"    "Uqcrc1"     "Ssb"        
"Hnrnpab"    "Rps27a"     "Gcn1"       "Rps15a"    
[23] "Fdps"       "Rps19"      "Hars"       "Sri"        "Smarca5"    "Sae1"       "Dcps
"       "Tpp2"       "Fth1"       "Hnrnpul2"   "Gm2000"    
[34] "Rps25"      "Dnpep"      "Grb2"       "Nme2"       "Ahcyl1"     "Lamp2"      "Twf2
"       "Anp32b"     "Cox6c"      "Lpcat3"     "Lnpep"     
[45] "Metap2"     "Rtn3"       "Atp5d"      "Ndufa12"    "SNX3"       "Krt2"       "Tnfa
ip8"    "Uqcr10"     "Srp9"       "VAMP8"      "CPNS1"     
[56] "Atp5k"      "Nmt1"       "Pdk3"       "Rrm2"       "Gabpa"      "Sec61b"    
```

Finally, the significant protein heatmap demonstrated several protein clusters distinct to each treatment conditions which can be used later for in-depth biological interpretation. The heatmap is plotted using the pheatmap function of pheatmap package.

Hide

```
## Significant protein heatmap by pheatmap (ref#4)
tmp <- as.matrix(log_ds[ , 2:length(log_ds)])
med <- apply(t(tmp), 1, mean)
medScale <- (t(tmp) - med)
tmp <- anova_ds[, 1]
medScale <- data.frame(medScale,
                      anova_pVal = tmp,
                      gene = rownames(medScale))
colnames(medScale) <- c("WT_1", "WT_2", "WT_3", "WT+inh_1", "WT+inh_2", "WT+inh_3"
, "Q741x_1", "Q741x_2", "Q741x_3", "Q741x+inh_1", "Q741x+inh_2", "Q741x+inh_3", "a
nova_pVal", "gene")
medScale_sig <- medScale %>% filter(anova_pVal < 0.05)
rownames(medScale_sig) <- medScale_sig$gene
medScale_sig <- medScale_sig[, 1: (length(medScale_sig) - 2)]
nprot_sig <- nrow(medScale_sig)
group <- factor(group, ordered = TRUE,
                levels = c("WT+inh", "Q741x+inh", "Q741x", "WT"))
hm_sig <- pheatmap(medScale_sig, silent = FALSE,
                     breaks = seq(-(max(round(medScale_sig, 0))), max(round(medSca
le_sig, 0)), length.out=101),
                     legend_breaks = seq(-(max(round(medScale_sig, 0))), max(round
(medScale_sig, 0)), length.out=5),
                     color = colorRampPalette(c("darkblue", "blue", "white", "oran
gered", "red"))(100),
                     border_color = NA,
                     annotation_col = data.frame(group = group, #factor(group),
                                                 row.names = sample_label),
                     clustering_distance_rows = "correlation",
                     clustering_distance_cols = "correlation",
                     clustering_method = "average",
                     fontsize_row = 2, fontsize_col = 10,
                     scale = "none")
```





Hide

```
NA
```

The heatmap parameters are provided here for a reproducibility purpose.

Hide

```
print(paste0("Significant protein heatmap:", nprot_sig, " significant proteins (AN
OVA p-value < ", 0.05, ")", "; Scale: Log2(fold change) with mean center", "; Clus
tering: Correlation distance and average linkage"))
```

```
[1] "Significant protein heatmap:397 significant proteins (ANOVA p-value < 0.05);
Scale: Log2(fold change) with mean center; Clustering: Correlation distance and av
erage linkage"
```

Hide

```
# End: Data analysis and visualization -----------------------------------------
----------------------------------
# References
## 1. Durinck S, Spellman P, Birney E, Huber W (2009). “Mapping identifiers for th
e integration of genomic datasets with the R/Bioconductor package biomaRt.” Nature
Protocols, 4, 1184–1191.
## 2. Bolstad B (2018). preprocessCore: A collection of pre-processing functions.
R package version 1.44.0,
## 3. Lê, S., Josse, J. & Husson, F. (2008). FactoMineR: An R Package for Multivar
iate Analysis. Journal of Statistical Software. 25(1). pp. 1-18.
## 4. Raivo Kolde (2018). pheatmap: Pretty Heatmaps. R package version 1.0.10.
```

Additional analysis

Additional analysis#1: %coefficient of variation of peptide retention time (RT) to reassure the consistency chromatography applied in SWATH acquisition

Hide

```
RT <- read_excel(data_path, sheet = "Observed RT")
RT <- RT %>% filter(Decoy == "FALSE")
RT <- RT[, c(2, 8:length(RT))]
colnames(RT) <- c("Peptides", sample_label)
tRT <- t(RT[, 2:length(RT)])
colnames(tRT) <- RT$Peptides
tRT <- data.frame(group, tRT)
## Group RT average
tmp_RT <- tRT %>%
  gather(Peptides, RT, -group) %>%
  dplyr::group_by(group, Peptides) %>%
  dplyr::summarize(group_mean = mean(RT)) %>%
  spread(Peptides, group_mean)
gr_RT_avr <- as.data.frame(tmp_RT[ , 2:length(tmp_RT)])
rownames(gr_RT_avr) <- tmp_RT$group
## Group RT SD
tmp_RT <-  tRT %>%
  gather(Peptides, RT, -group) %>%
  dplyr::group_by(group, Peptides) %>%
  dplyr::summarize(group_sd = sd(RT)) %>%
  spread(Peptides, group_sd)
gr_RT_sd <- as.data.frame(tmp_RT[ , 2:length(tmp_RT)])
rownames(gr_RT_sd) <- tmp_RT$group
## Coefficient of variation
cv_RT <- 100 *gr_RT_sd/gr_RT_avr
cv_RT <- data.frame(group = tmp_RT$group, cv_RT)
cv_RT$group <- factor(cv_RT$group, ordered = TRUE,
                levels = c("Q741x+inh", "WT+inh", "Q741x", "WT"))
CV_RT <- cv_RT %>% gather(Peptides, CV, -group)
# Calculate median-CV of each group
medianCV_RT <- CV_RT %>% dplyr::group_by(group) %>% summarise(CV = round(median(CV
), 1))
```

Hide

Hide

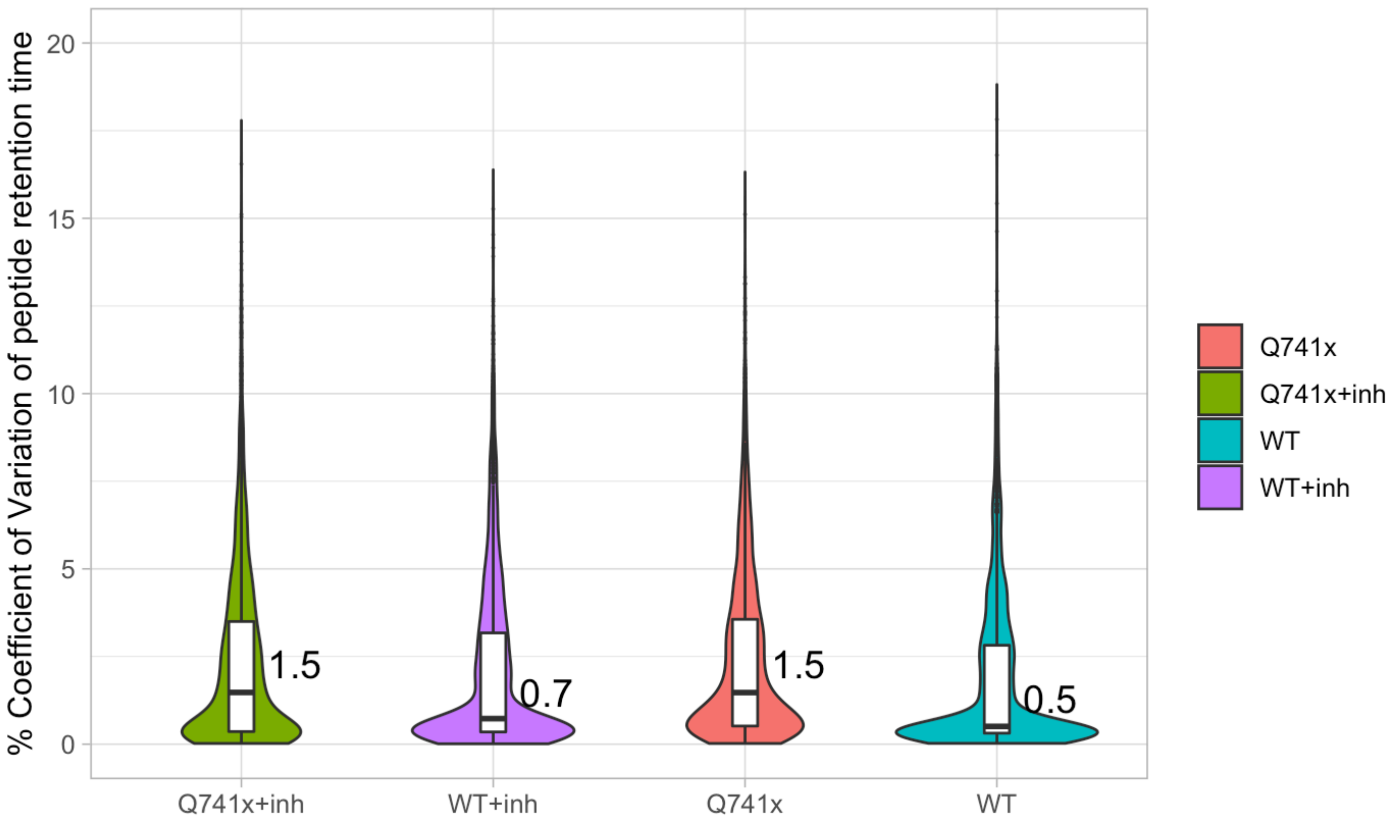
```
print(paste0("Median-CV of peptide RT: Q741x+inh, ", medianCV_RT[1,2], "%; WT+inh,
", medianCV_RT[2,2], "%; Q741x, ", medianCV_RT[3,2], "%; WT, ", medianCV_RT[4,2],
"%"))
```

```
[1] "Median-CV of peptide RT: Q741x+inh, 1.5%; WT+inh, 0.7%; Q741x, 1.5%; WT, 0.5%
"
```

And here is the plot;

Hide

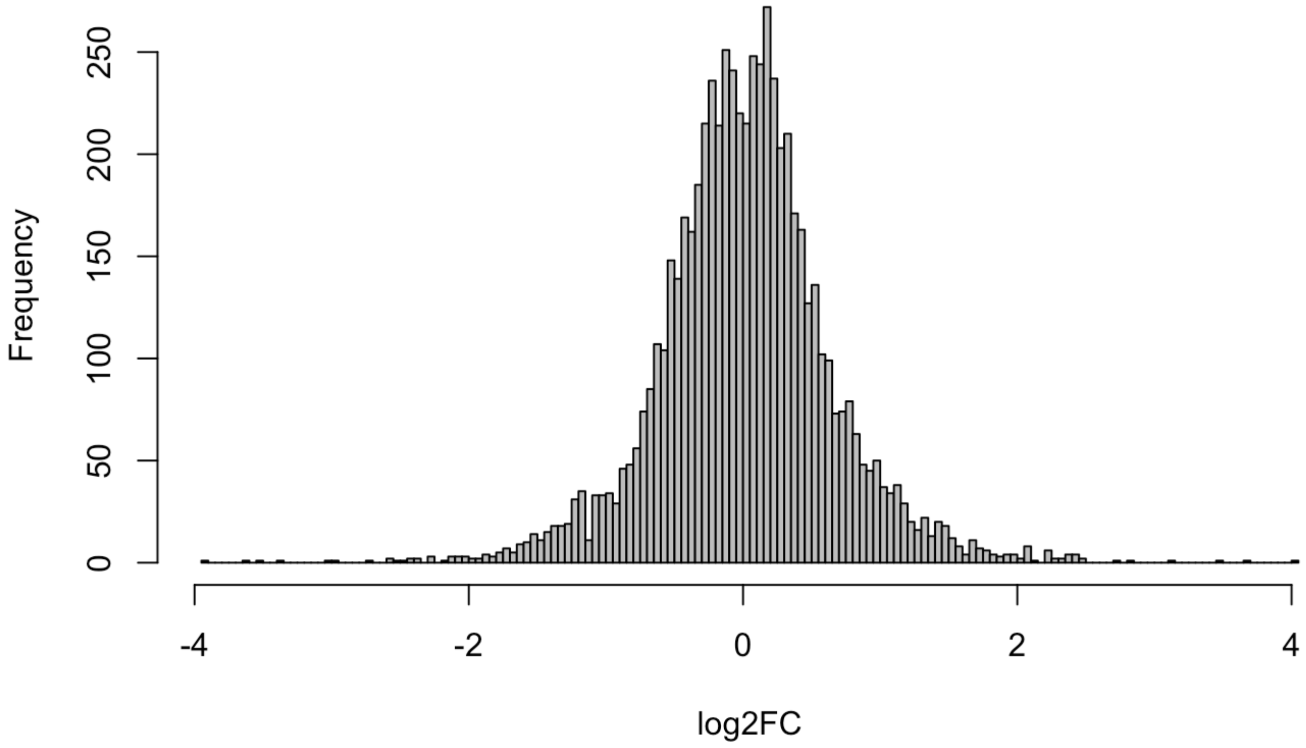
```
# Violin plot of inter-group CV
plot.cv_RT <- ggplot(CV_RT, aes(x=group, y=CV)) +
              geom_violin(aes(fill = as.character(group)), trim=FALSE, width = 0.8
, #aes(fill = group),
                          na.rm = TRUE, position = "dodge")+
              geom_boxplot(width=0.1, fill = 'white', outlier.size = 0,
                          na.rm = TRUE, position = "dodge")+
              #geom_boxplot(width=0.3, outlier.size = 0.1, na.rm = TRUE, position
= "dodge", aes(fill = as.character(group)))+
              geom_text(data = medianCV_RT, aes(label = CV), position = position_d
odge(width = 1),
                          hjust = -0.5, vjust = -0.5, size = 5) +
              ylim(0, 20)+
              labs(fill = "")+
              xlab("") + ylab("% Coefficient of Variation of peptide retention tim
e") +
              theme_light(base_size = 12)

plot.cv_RT
```



Additional analysis#2: Visualizing the overall shape of comparative data by a histogram of distribution of log2FC;

Hide

```
hist(long_ano.fc$log2FC, breaks = 120, col = "grey", xlab = "log2FC", main = "")
```


Frequency
log2FC