```r
#set working directory
setwd("~/04_Champion_Lab/02_N-terminal_Acetylation/OnePot/NAT5")

#import libraries
library(ggplot2)
library(dplyr)
library(stringr)
library(tidyr)
library(ggrepel)
library(QFeatures)
library(corrplot)
library(limma)
library(tibble)

#set colors
red <- hue_pal()(3)[1]
green <- hue_pal()(3)[2]
blue <- hue_pal()(3)[3]

#import protein quant data and filter contaminants
proteins <- read.csv("lfq.proteins.csv")
proteins <- proteins[!grepl("CONTAM", proteins$Accession),]

#parse accession and gene info, save as dictionary
proteins$Accession2 <- str_remove(str_extract(proteins$Accession, "^[^\\|]*\\|"), "\\|")
proteins$gene <- str_remove(str_remove(str_extract(proteins$Accession,
"^[^\\|]*\\|[^\\|]*\\|"),
                                                   "^[^\\|]*\\|"), "\\|")
geneDict <- as.list(proteins$gene)
names(geneDict) <- proteins$Accession2

#clean column names
names(proteins)[which(grepl("tr[0-9].Area", names(proteins)))] <- str_remove(str_extract(
names(proteins)[which(grepl("tr[0-9].Area", names(proteins)))],

"_[^_]+_br[0-9]_tr[0-9]"), "^_")
#set missing values to NA
proteins[proteins == 0] <- NA

#import to Qfeatures and set metadata
Qprot <- readQFeatures(table = proteins,
                       ecol = which(grepl("tr[0-9]", names(proteins))),
                       fnames = "Accession2",
                       name = "raw_proteins")
colnames(Qprot[[1]])
Qprot$bioRep <- str_extract(colnames(Qprot[[1]]), "br[0-9]+")
Qprot$techRep <- str_extract(colnames(Qprot[[1]]), "tr[0-9]+$")
Qprot$condition <- str_extract(colnames(Qprot[[1]]), "^[^_]*")
Qprot$Inj <- colnames(Qprot[[1]])
colData(Qprot)


#log transform, base 2 so that later FC is in base 2
Qprot <- addAssay(Qprot,
                  logTransform((Qprot[["raw_proteins"]]),
                               base = 2),
                  name = "log_proteins")
#normalize
Qprot <- addAssay(Qprot,
                  normalize(Qprot[["log_proteins"]],
                            method = "center.median"),
                  name = "norm_proteins")
#save figure of log and normaliztions
png("LogNormalization.png", units = "px", width = 1500, height = 1000)
par(mfrow = c(1,3))
limma::plotDensities(assay(Qprot[[1]]), legend = F)
limma::plotDensities(assay(Qprot[[2]]), legend = F)
limma::plotDensities(assay(Qprot[[3]]), legend = F)
dev.off()
```

```r

#add in old column data to normalized data
colData(Qprot[["norm_proteins"]]) <- colData(Qprot)

#save as a summary dataframe
brSummaryWide <- Qprot[["norm_proteins"]] %>% assay() %>% data.frame() %>%
rownames_to_column("Accession")
proteinsBR <- Qprot[["norm_proteins"]]
#set variables as factor, and clarify the reference as WT
proteinsBR$condition <- factor(Qprot$condition)
proteinsBR$condition <- relevel(proteinsBR$condition, ref = "WT")


#create model based on values in condition
model_design <- model.matrix(~proteinsBR$condition)

#fit proteins to linear model with lmFit and the matrix we just created
fitted_lm <- proteinsBR %>%
  assay() %>%
  lmFit(design = model_design) %>%
  eBayes()

#pull WT vs deletion
DE_DEL <- topTable(fit = fitted_lm,
                   adjust.method = "BH",
                   number = Inf,
                   coef = "proteinsBR$conditiond1839",
                   sort.by = "p",
                   confint = T)

DE_DEL$Accession <- rownames(DE_DEL)



#calculate corrected z score based on B-H adujsted p-value. Altman,
https://doi.org/10.1136/bmj.d2090
DE_DEL$corrected_z <- sqrt(0.743 - (2.404*log(DE_DEL$adj.P.Val))) - 0.862
#calculate the one directional 95% confidence interval
DE_DEL$error <-  abs(DE_DEL$logFC /DE_DEL$corrected_z) * 1.96


#pull only values with non-missing expression data
DE_DEL <- DE_DEL[!is.na(DE_DEL$AveExpr),]
DE_DEL$CountWT <- NA
DE_DEL$CountExp <- NA

#count number of valid measurments for each strain
for (i in 1:nrow(DE_DEL)) {
  CountDF <- brSummaryWide[brSummaryWide$Accession == DE_DEL$Accession[i],]
  DE_DEL$CountWT[i] <- sum(!is.na(CountDF[,grepl("WT", names(CountDF))][1,]))
  DE_DEL$CountExp[i] <- sum(!is.na(CountDF[,grepl("d1839", names(CountDF))][1,]))
}

#set appropriate negative and positive infinite changes for totally missing values
DE_DEL$logFC[DE_DEL$CountWT == 0] <- Inf
DE_DEL$logFC[DE_DEL$CountExp == 0] <- -Inf

#set filters
adjSigCutoff <- 0.05
NonAdjSigCutoff <- 0.05
LFCcutoff <- 1


#option to plot either adj or non-adj p values
adj <- T

if (adj) {
  proteins_sig <- na.omit(unique(DE_DEL$Accession[DE_DEL$adj.P.Val < adjSigCutoff]))
```

```r
  proteins_sig2 <- na.omit(unique(DE_DEL$Accession[DE_DEL$adj.P.Val < adjSigCutoff &
                                                   abs(DE_DEL$logFC) > LFCcutoff]))
  proteins_label_pos <- na.omit(unique(DE_DEL$Accession[DE_DEL$adj.P.Val < adjSigCutoff &
                                                        DE_DEL$logFC > LFCcutoff]))
  proteins_label_neg <- na.omit(unique(DE_DEL$Accession[DE_DEL$adj.P.Val < adjSigCutoff &
                                                        DE_DEL$logFC < -LFCcutoff]))
} else {
  proteins_sig <- na.omit(unique(DE_DEL$Accession[DE_DEL$P.Value < adjSigCutoff]))
  proteins_sig2 <- na.omit(unique(DE_DEL$Accession[DE_DEL$P.Value < adjSigCutoff &
                                                   abs(DE_DEL$logFC) > LFCcutoff]))
  proteins_label_pos <- na.omit(unique(DE_DEL$Accession[DE_DEL$P.Value < adjSigCutoff &
                                                        DE_DEL$logFC > LFCcutoff]))
  proteins_label_neg <- na.omit(unique(DE_DEL$Accession[DE_DEL$P.Value < adjSigCutoff &
                                                        DE_DEL$logFC < -LFCcutoff]))
}




#add in gene and protein name info from saved dictionaries
DE_DEL$gene <- sapply(DE_DEL$Accession, function(Accession) geneDict[[Accession]])

#set colors
colorsUPDOWN <- c(blue, red)
names(colorsUPDOWN) <- c(T, F)

#add extra area above and below plot for error bars
plotBufferFactor <- 1.5
maxY <- max(DE_DEL$logFC[is.finite(DE_DEL$logFC)]) * plotBufferFactor
minY <- min(DE_DEL$logFC[is.finite(DE_DEL$logFC)]) * plotBufferFactor

#plot MA
ggplot() +
  theme_bw(base_size = 25) +
  theme(panel.grid = element_blank(),
        legend.position = "none") +
  labs(y = "Log2 Fold Change", x = "Normalized Average Expression (Log)",
       title = expression("WT vs " * Delta * italic(emp1))) +
  geom_point(data = DE_DEL[!DE_DEL$Accession %in% proteins_sig &
                             is.finite(DE_DEL$logFC),],
             aes(x = AveExpr, y = logFC), color = "grey50",
             alpha = 0.8, size = 0.8) +
  geom_hline(yintercept = 0, linetype = 1) + #add line at LFC = 0
  geom_hline(yintercept = c(-LFCcutoff,LFCcutoff), linetype = 2) +
  geom_point(data = DE_DEL[DE_DEL$Accession %in% proteins_sig,],
             aes(x = AveExpr, y = logFC, color = logFC > 0), size = 0.8) +
  scale_color_manual(values = colorsUPDOWN) +
  scale_fill_manual(values = colorsUPDOWN) +
  geom_segment(data = DE_DEL[DE_DEL$Accession %in% proteins_sig2,],
               aes(x = AveExpr, y = logFC - error, yend = logFC + error), lwd = 1) +
               #add 95% CI lines
  geom_segment(data = DE_DEL[DE_DEL$Accession %in% proteins_sig2,],
               aes(x = AveExpr, y = logFC - error, yend = logFC + error,
                   color = logFC > 0), lwd = 0.5) +
  geom_point(data = DE_DEL[DE_DEL$Accession %in% proteins_sig2 &
                             is.finite(DE_DEL$logFC),],
             aes(x = AveExpr, y = logFC, fill = logFC > 0), size = 3,
             shape = 21, stroke = 1.5) +
  geom_point(data = DE_DEL[DE_DEL$logFC > 0 & !is.finite(DE_DEL$logFC),],
             shape = 23, aes(x = AveExpr, size = CountExp, fill = logFC > 0), y = maxY,
             stroke = 1, alpha = 0.6) +
  geom_point(data = DE_DEL[DE_DEL$logFC < 0 & !is.finite(DE_DEL$logFC),],
             shape = 23, aes(x = AveExpr, size = CountWT, fill = logFC > 0), y = minY,
             stroke = 1, alpha = 0.6) +
  coord_cartesian(ylim = c(minY, maxY)) +
  scale_size(range = c(1,3)) +
  geom_label_repel(data = DE_DEL[DE_DEL$Accession %in% proteins_label_pos,],
                   aes(x = AveExpr, y = logFC, label = gene), nudge_y = 1, nudge_x = -1.2
```

```
                    ) +
  geom_label_repel(data = DE_DEL[DE_DEL$Accession %in% proteins_label_neg,],
                   aes(x = AveExpr, y = logFC, label = gene), nudge_y = -1, nudge_x = -
                   1.2) +
  geom_label_repel(data = DE_DEL[DE_DEL$logFC > 0 & !is.finite(DE_DEL$logFC) & DE_DEL$
  CountExp > 1,],
                   aes(x = AveExpr, label = gene), y = maxY, nudge_y = -1, nudge_x = -1)
                   +
  geom_label_repel(data = DE_DEL[DE_DEL$logFC < 0 & !is.finite(DE_DEL$logFC) & DE_DEL$
  CountWT > 1,],
                   aes(x = AveExpr, label = gene),y = minY, nudge_y = 1, nudge_x = 1)



ggsave("WTvsDEL_MAplot.png", width = 15, height = 12)

write.table(DE_DEL, "WTvsDELprotein.tsv", sep = "\t", row.names = F)



#add in Acetylation changers
acetylation <- read.csv("1839_acetylation_reported.tsv", sep = "\t")

proteinList <- acetylation$Accession2


ggplot() +
  theme_bw(base_size = 25) +
  theme(panel.grid = element_blank(),
        legend.position = "none") +
  labs(y = "Log2 Fold Change", x = "Normalized Average Expression (Log)",
       title = expression("WT vs " * Delta * italic(emp1))) +
  geom_point(data = DE_DEL,
             aes(x = AveExpr, y = logFC), color = "grey50",
             alpha = 0.8, size = 0.8) +
  geom_hline(yintercept = 0, linetype = 1) + #add line at LFC = 0
  geom_hline(yintercept = c(-LFCcutoff,LFCcutoff), linetype = 2) +
  geom_segment(data = DE_DEL[DE_DEL$Accession %in% proteinList,],
               aes(x = AveExpr, y = logFC - error, yend = logFC + error), lwd = 1) +
               #add 95% CI lines
  geom_segment(data = DE_DEL[DE_DEL$Accession %in% proteinList,],
               aes(x = AveExpr, y = logFC - error, yend = logFC + error), lwd = 0.5,
               color = green) +
  geom_point(data = DE_DEL[DE_DEL$Accession %in% proteinList &
                             is.finite(DE_DEL$logFC),],
             aes(x = AveExpr, y = logFC), size = 3,
             fill = green,
             shape = 21, stroke = 1.5) +
  geom_point(data = DE_DEL[DE_DEL$logFC > 0 & !is.finite(DE_DEL$logFC),],
             shape = 23, aes(x = AveExpr), y = maxY, stroke = 1, alpha = 0.6) +
  geom_point(data = DE_DEL[DE_DEL$logFC < 0 & !is.finite(DE_DEL$logFC),],
             shape = 23, aes(x = AveExpr), y = minY, stroke = 1, alpha = 0.6) +
  coord_cartesian(ylim = c(minY, maxY)) +
  scale_size(range = c(1,3)) +
  geom_label_repel(data = DE_DEL[DE_DEL$Accession %in% proteinList,],
                   aes(x = AveExpr, y = logFC, label = gene), nudge_y = 3, nudge_x = -1.2
                   ,
                   max.overlaps = 40)


ggsave("WTvsDEL_MAplot_acetylationChangers.png", width = 15, height = 12)


#log transform, base 10 for galaxy plots
Qprot <- addAssay(Qprot,
                  logTransform((Qprot[["raw_proteins"]]),
                               base = 10),
                  name = "log_proteins_10")
Qprot <- addAssay(Qprot,
```

```
                    normalize(Qprot[["log_proteins_10"]],
                              method = "center.median"),
                    name = "norm_proteins_10")



#add in old column data to normalized data
colData(Qprot[["norm_proteins_10"]]) <- colData(Qprot)

#pull log 10 data
loglog <- Qprot[["norm_proteins_10"]] %>% assay() %>% data.frame() %>% rownames_to_column
("Accession")

#pivot longer
loglogLong <- loglog %>% pivot_longer(2:ncol(loglog))
loglogLong$strain <- str_extract(loglogLong$name, "^[^_]+")

#average area and count of non na values for each
loglogGrouped <- loglogLong %>% group_by(Accession, strain) %>%
  summarise(Area = mean(value, na.rm = T),
            count = sum(!is.na(value)))

#pivot wider for plotting
loglogWide <- loglogGrouped %>% pivot_wider(id_cols = Accession,
                                            values_from = Area,
                                            names_from = strain)
#plot
ggplot(loglogWide) +
  geom_point(aes(x = WT, y = d1839), color = "grey50",
             size = 0.8, alpha = 0.8) +
  theme_bw(base_size = 25) +
  theme(panel.grid = element_blank(),
        legend.position = "none") +
  geom_abline(slope = 1, intercept = 0, linetype = 2, size = 1) +
  geom_point(data = loglogWide[loglogWide$Accession %in% proteins_sig2,],
             aes(x = WT, y = d1839, fill = WT < d1839), size = 3,
             shape = 21, stroke = 1.5) +
  scale_fill_manual(values = colorsUPDOWN) +
  labs(title = expression("WT vs " * Delta * italic(emp1)),
       x = "Normalized WT expression (log)",
       y = expression("Normalized " * Delta * italic(emp1) * " expression (log)"))

ggsave("galaxyWTvsDelProteinDE.png", width = 10, height = 10)


ggplot(loglogWide) +
  geom_point(aes(x = WT, y = d1839), color = "grey50",
             size = 0.8, alpha = 0.8) +
  theme_bw(base_size = 25) +
  theme(panel.grid = element_blank(),
        legend.position = "none") +
  geom_abline(slope = 1, intercept = 0, linetype = 2, size = 1) +
  geom_point(data = loglogWide[loglogWide$Accession %in% proteinList,],
             aes(x = WT, y = d1839), fill = green, size = 3,
             shape = 21, stroke = 1.5) +
  scale_fill_manual(values = colorsUPDOWN) +
  labs(title = expression("WT vs " * Delta * italic(emp1)),
       x = "Normalized WT expression (log)",
       y = expression("Normalized " * Delta * italic(emp1) * " expression (log)")) +
  geom_label_repel(data = loglogWide[loglogWide$Accession %in% proteinList,],
                   aes(x = WT, y = d1839, label = Accession), nudge_y = 1, nudge_x = -1,
                   max.overlaps = 40)


ggsave("galaxyWTvsDelAcetylationChangers_labs.png", width = 10, height = 10)

ggplot(loglogWide) +
  geom_point(aes(x = WT, y = d1839), color = "grey50",
             size = 0.8, alpha = 0.8) +
```

```
  theme_bw(base_size = 25) +
  theme(panel.grid = element_blank(),
        legend.position = "none") +
  geom_abline(slope = 1, intercept = 0, linetype = 2, size = 1) +
  geom_point(data = loglogWide[loglogWide$Accession %in% proteinList,],
             aes(x = WT, y = d1839), fill = green, size = 3,
             shape = 21, stroke = 1.5) +
  geom_point(data = loglogWide[loglogWide$Accession %in% proteins_sig2,],
             aes(x = WT, y = d1839, fill = WT < d1839), size = 3,
             shape = 21, stroke = 1.5) +
  scale_fill_manual(values = colorsUPDOWN) +
  labs(title = expression("WT vs " * Delta * italic(emp1)),
       x = "Normalized WT expression (log)",
       y = expression("Normalized " * Delta * italic(emp1) * " expression (log)"))

ggsave("galaxyWTvsDelAcetylationChangers_andDE.png", width = 10, height = 10)


#SAME as above but for complement instea of Del
#fit proteins to linear model with lmFit and the matrix we just created
fitted_lm <- proteinsBR %>%
  assay() %>%
  lmFit(design = model_design) %>%
  eBayes()

DE_DEL <- topTable(fit = fitted_lm,
                   adjust.method = "BH",
                   number = Inf,
                   coef = "proteinsBR$conditionc1839",
                   sort.by = "p",
                   confint = T)


DE_DEL$Accession <- rownames(DE_DEL)



#calculate corrected z score based on B-H adujsted p-value. Altman, https://doi.org/10.1136/bmj.d2090
DE_DEL$corrected_z <- sqrt(0.743 - (2.404*log(DE_DEL$adj.P.Val))) - 0.862
#calculate the one directional 95% confidence interval
DE_DEL$error <-  abs(DE_DEL$logFC /DE_DEL$corrected_z) * 1.96



DE_DEL <- DE_DEL[!is.na(DE_DEL$AveExpr),]
DE_DEL$CountWT <- NA
DE_DEL$CountExp <- NA

for (i in 1:nrow(DE_DEL)) {
  CountDF <- brSummaryWide[brSummaryWide$Accession == DE_DEL$Accession[i],]
  DE_DEL$CountWT[i] <- sum(!is.na(CountDF[,grepl("WT", names(CountDF))][1,]))
  DE_DEL$CountExp[i] <- sum(!is.na(CountDF[,grepl("c1839", names(CountDF))][1,]))
}

DE_DEL$logFC[DE_DEL$CountWT == 0] <- Inf
DE_DEL$logFC[DE_DEL$CountExp == 0] <- -Inf


adjSigCutoff <- 0.05
NonAdjSigCutoff <- 0.05
LFCcutoff <- 1



adj <- T

if (adj) {
  proteins_sig <- na.omit(unique(DE_DEL$Accession[DE_DEL$adj.P.Val < adjSigCutoff]))
```

```
  proteins_sig2 <- na.omit(unique(DE_DEL$Accession[DE_DEL$adj.P.Val < adjSigCutoff &
                                                   abs(DE_DEL$logFC) > LFCcutoff]))
  proteins_label_pos <- na.omit(unique(DE_DEL$Accession[DE_DEL$adj.P.Val < adjSigCutoff &
                                                        DE_DEL$logFC > LFCcutoff]))
  proteins_label_neg <- na.omit(unique(DE_DEL$Accession[DE_DEL$adj.P.Val < adjSigCutoff &
                                                        DE_DEL$logFC < -LFCcutoff]))
} else {
  proteins_sig <- na.omit(unique(DE_DEL$Accession[DE_DEL$P.Value < adjSigCutoff]))
  proteins_sig2 <- na.omit(unique(DE_DEL$Accession[DE_DEL$P.Value < adjSigCutoff &
                                                   abs(DE_DEL$logFC) > LFCcutoff]))
  proteins_label_pos <- na.omit(unique(DE_DEL$Accession[DE_DEL$P.Value < adjSigCutoff &
                                                        DE_DEL$logFC > LFCcutoff]))
  proteins_label_neg <- na.omit(unique(DE_DEL$Accession[DE_DEL$P.Value < adjSigCutoff &
                                                        DE_DEL$logFC < -LFCcutoff]))
}




#add in gene and protein name info from saved dictionaries
DE_DEL$gene <- sapply(DE_DEL$Accession, function(Accession) geneDict[[Accession]])



plotBufferFactor <- 1.5
maxY <- max(DE_DEL$logFC[is.finite(DE_DEL$logFC)]) * plotBufferFactor
minY <- min(DE_DEL$logFC[is.finite(DE_DEL$logFC)]) * plotBufferFactor


ggplot() +
  theme_bw(base_size = 25) +
  theme(panel.grid = element_blank(),
        legend.position = "none") +
  labs(y = "Log2 Fold Change", x = "Normalized Average Expression (Log)",
       title = expression("WT vs " * Delta * italic(emp1) * "/comp")) +
  geom_point(data = DE_DEL[!DE_DEL$Accession %in% proteins_sig &
                             is.finite(DE_DEL$logFC),],
             aes(x = AveExpr, y = logFC), color = "grey50",
             alpha = 0.8, size = 0.8) +
  geom_hline(yintercept = 0, linetype = 1) + #add line at LFC = 0
  geom_hline(yintercept = c(-LFCcutoff,LFCcutoff), linetype = 2) +
  geom_point(data = DE_DEL[DE_DEL$Accession %in% proteins_sig,],
             aes(x = AveExpr, y = logFC, color = logFC > 0), size = 0.8) +
  scale_color_manual(values = colorsUPDOWN) +
  scale_fill_manual(values = colorsUPDOWN) +
  geom_segment(data = DE_DEL[DE_DEL$Accession %in% proteins_sig2,],
               aes(x = AveExpr, y = logFC - error, yend = logFC + error), lwd = 1) +
               #add 95% CI lines
  geom_segment(data = DE_DEL[DE_DEL$Accession %in% proteins_sig2,],
               aes(x = AveExpr, y = logFC - error, yend = logFC + error,
                   color = logFC > 0), lwd = 0.5) +
  geom_point(data = DE_DEL[DE_DEL$Accession %in% proteins_sig2 &
                             is.finite(DE_DEL$logFC),],
             aes(x = AveExpr, y = logFC, fill = logFC > 0), size = 3,
             shape = 21, stroke = 1.5) +
  geom_point(data = DE_DEL[DE_DEL$logFC > 0 & !is.finite(DE_DEL$logFC),],
             shape = 23, aes(x = AveExpr, size = CountExp, fill = logFC > 0), y = maxY,
             stroke = 1, alpha = 0.6) +
  geom_point(data = DE_DEL[DE_DEL$logFC < 0 & !is.finite(DE_DEL$logFC),],
             shape = 23, aes(x = AveExpr, size = CountWT, fill = logFC > 0), y = minY,
             stroke = 1, alpha = 0.6) +
  coord_cartesian(ylim = c(minY, maxY)) +
  scale_size(range = c(1,3)) +
  geom_label_repel(data = DE_DEL[DE_DEL$Accession %in% proteins_label_pos,],
                   aes(x = AveExpr, y = logFC, label = gene), nudge_y = 1, nudge_x = -1.2
                   ) +
  geom_label_repel(data = DE_DEL[DE_DEL$Accession %in% proteins_label_neg,],
                   aes(x = AveExpr, y = logFC, label = gene), nudge_y = -1, nudge_x = -
```

```
                    1.2) +
463   geom_label_repel(data = DE_DEL[DE_DEL$logFC > 0 & !is.finite(DE_DEL$logFC) & DE_DEL$
      CountExp > 1,],
464                    aes(x = AveExpr, label = gene), y = maxY, nudge_y = -1, nudge_x = -1)
                       +
465   geom_label_repel(data = DE_DEL[DE_DEL$logFC < 0 & !is.finite(DE_DEL$logFC) & DE_DEL$
      CountWT > 1,],
466                    aes(x = AveExpr, label = gene),y = minY, nudge_y = 1, nudge_x = 1)
467
468
469
470 ggsave("WTvsComp_MAplot.png", width = 15, height = 12)
471 write.table(DE_DEL, "WTvsCOMPprotein.tsv", sep = "\t", row.names = F)
472
473
474
475 proteinList <- acetylation$Accession2
476
477
478 ggplot() +
479   theme_bw(base_size = 25) +
480   theme(panel.grid = element_blank(),
481         legend.position = "none") +
482   labs(y = "Log2 Fold Change", x = "Normalized Average Expression (Log)",
483        title = expression("WT vs " * Delta * italic(emp1) * "/comp")) +
484   geom_point(data = DE_DEL,
485              aes(x = AveExpr, y = logFC), color = "grey50",
486              alpha = 0.8, size = 0.8) +
487   geom_hline(yintercept = 0, linetype = 1) + #add line at LFC = 0
488   geom_hline(yintercept = c(-LFCcutoff,LFCcutoff), linetype = 2) +
489   geom_segment(data = DE_DEL[DE_DEL$Accession %in% proteinList,],
490                aes(x = AveExpr, y = logFC - error, yend = logFC + error), lwd = 1) +
                   #add 95% CI lines
491   geom_segment(data = DE_DEL[DE_DEL$Accession %in% proteinList,],
492                aes(x = AveExpr, y = logFC - error, yend = logFC + error), lwd = 0.5,
493                color = green) +
494   geom_point(data = DE_DEL[DE_DEL$Accession %in% proteinList &
495                              is.finite(DE_DEL$logFC),],
496              aes(x = AveExpr, y = logFC), size = 3,
497              fill = green,
498              shape = 21, stroke = 1.5) +
499   geom_point(data = DE_DEL[DE_DEL$logFC > 0 & !is.finite(DE_DEL$logFC),],
500              shape = 23, aes(x = AveExpr), y = maxY, stroke = 1, alpha = 0.6) +
501   geom_point(data = DE_DEL[DE_DEL$logFC < 0 & !is.finite(DE_DEL$logFC),],
502              shape = 23, aes(x = AveExpr), y = minY, stroke = 1, alpha = 0.6) +
503   coord_cartesian(ylim = c(minY, maxY)) +
504   scale_size(range = c(1,3)) +
505   geom_label_repel(data = DE_DEL[DE_DEL$Accession %in% proteinList,],
506                    aes(x = AveExpr, y = logFC, label = gene), nudge_y = 3, nudge_x = -1.2
                       ,
507                    max.overlaps = 40)
508
509
510 ggsave("WTvsCOMP_MAplot_acetylationChangers.png", width = 15, height = 12)
511
512
513
514 Qprot <- addAssay(Qprot,
515                   logTransform((Qprot[["raw_proteins"]]),
516                                base = 10),
517                   name = "log_proteins_10")
518 Qprot <- addAssay(Qprot,
519                   normalize(Qprot[["log_proteins_10"]],
520                             method = "center.median"),
521                   name = "norm_proteins_10")
522
523
524
525 #add in old column data to normalized data
```

```r
colData(Qprot[["norm_proteins_10"]]) <- colData(Qprot)


loglog <- Qprot[["norm_proteins_10"]] %>% assay() %>% data.frame() %>% rownames_to_column("Accession")

loglogLong <- loglog %>% pivot_longer(2:ncol(loglog))
loglogLong$strain <- str_extract(loglogLong$name, "^[^_]+")

loglogGrouped <- loglogLong %>% group_by(Accession, strain) %>%
  summarise(Area = mean(value, na.rm = T),
            count = sum(!is.na(value)))


loglogWide <- loglogGrouped %>% pivot_wider(id_cols = Accession,
                                            values_from = Area,
                                            names_from = strain)

ggplot(loglogWide) +
  geom_point(aes(x = WT, y = c1839), color = "grey50",
             size = 0.8, alpha = 0.8) +
  theme_bw(base_size = 25) +
  theme(panel.grid = element_blank(),
        legend.position = "none") +
  geom_abline(slope = 1, intercept = 0, linetype = 2, size = 1) +
  geom_point(data = loglogWide[loglogWide$Accession %in% proteins_sig2,],
             aes(x = WT, y = c1839, fill = WT < c1839), size = 3,
             shape = 21, stroke = 1.5) +
  scale_fill_manual(values = colorsUPDOWN) +
  labs(title = expression("WT vs " * Delta * italic(emp1) * "/comp"),
       x = "Normalized WT expression (log)",
       y = expression("Normalized " * Delta * italic(emp1) * "/comp expression (log)"))

ggsave("galaxyWTvsCOMPProteinDE.png", width = 10, height = 10)


ggplot(loglogWide) +
  geom_point(aes(x = WT, y = c1839), color = "grey50",
             size = 0.8, alpha = 0.8) +
  theme_bw(base_size = 25) +
  theme(panel.grid = element_blank(),
        legend.position = "none") +
  geom_abline(slope = 1, intercept = 0, linetype = 2, size = 1) +
  geom_point(data = loglogWide[loglogWide$Accession %in% proteinList,],
             aes(x = WT, y = c1839), fill = green, size = 3,
             shape = 21, stroke = 1.5) +
  scale_fill_manual(values = colorsUPDOWN) +
  labs(title = expression("WT vs " * Delta * italic(emp1) * "/comp"),
       x = "Normalized WT expression (log)",
       y = expression("Normalized " * Delta * italic(emp1) * "/comp expression (log)")) +
  geom_label_repel(data = loglogWide[loglogWide$Accession %in% proteinList,],
                   aes(x = WT, y = c1839, label = Accession), nudge_y = 1, nudge_x = -1,
                   max.overlaps = 40)

ggsave("galaxyWTvsCOMPAcetylationChangers_labs.png", width = 10, height = 10)

ggplot(loglogWide) +
  geom_point(aes(x = WT, y = c1839), color = "grey50",
             size = 0.8, alpha = 0.8) +
  theme_bw(base_size = 25) +
  theme(panel.grid = element_blank(),
        legend.position = "none") +
  geom_abline(slope = 1, intercept = 0, linetype = 2, size = 1) +
  geom_point(data = loglogWide[loglogWide$Accession %in% proteinList,],
             aes(x = WT, y = c1839), fill = green, size = 3,
             shape = 21, stroke = 1.5) +
  geom_point(data = loglogWide[loglogWide$Accession %in% proteins_sig2,],
             aes(x = WT, y = c1839, fill = WT < c1839), size = 3,
             shape = 21, stroke = 1.5) +
```

```
    scale_fill_manual(values = colorsUPDOWN) +
    labs(title = expression("WT vs " * Delta * italic(emp1) * "/comp"),
         x = "Normalized WT expression (log)",
         y = expression("Normalized " * Delta * italic(emp1) * "/comp expression (log)"))

ggsave("galaxyWTvsCOMPAcetylationChangers_andDE.png", width = 10, height = 10)
```