rPhos.Rmd

```
library(plyr)
library(readxl)
library(ggpubr)
library(ggplot2)
library(ggrepel)
library(rstatix)
library(SummarizedExperiment)
library(tidyr)
library(StTools)



library(DEP)

library(curl)
library(stringr)
```

Phosphoproteomics in the hippocampus in mice stress through restrainment

```
data <- read.csv("/mnt/bohacek/Sebastian/05_rPhos/Report_Phospho_Lukasv_All5amples (Normal).xls", sep = "\t")
meta_data <- data.frame(read_xlsx("/mnt/bohacek/Sebastian/05_rPhos/2022-01_DNCFF2_schedulex.xlsx", sheet = "Metadata")[-(4
2,)]
```

```
#Sample 42 error in MS-measurement

meta_data$File <- sort(unique(data$R.FileName))
rownames(meta_data) <- meta_data$File

#better names
meta_data$Restraint <- meta_data$Group
#meta_data$Group= NULL

meta_data$Restraint <- gsub("Control", "Ctrl", meta_data$Restraint)
meta_data$Restraint <- gsub("6 min", "6min", meta_data$Restraint)
meta_data$Restraint <- gsub("45 min", "45min", meta_data$Restraint)
meta_data$Restraint <- gsub("90 min", "90min", meta_data$Restraint)
meta_data$Restraint <- gsub("96 min", "96min", meta_data$Restraint)
meta_data$Restraint <- gsub("135 min", "135min", meta_data$Restraint)


meta_data$Restraint <- as.factor(meta_data$Restraint)
meta_data$Restraint <- factor(meta_data$Restraint, levels = c("Ctrl", "6min", "45min", "90min", "96min", "135min"))

meta_data$Sex <- gsub("M", "male", meta_data$Sex)
meta_data$Sex <- gsub("F", "female", meta_data$Sex)
meta_data$Sex <- as.factor(meta_data$Sex)
meta_data$Sex <- relevel(meta_data$Sex, ref = "male")

#combining data & meta_data

data$Restraint <- "null"

for(i in sort(unique(data$R.FileName))){
  data[data$R.FileName == i,]$Restraint <- as.character(meta_data[i,]$Restraint)
  data$Sex[data$R.FileName == i] <- as.character(meta_data[i,]$Sex)
}

res <- list()
```

```
s2c <- meta_data

s2c$File <- sort(unique(data$R.FileName))
rownames(s2c) <- s2c$FileName
s2c$Replicate <- 0
#data$FG.Quantity <- log2(data$FG.Quantity)

for(i in unique(s2c$Group)){
  s2c$Replicate[s2c$Group == i]  <- 1:length(s2c$Replicate[s2c$Group == i])
}


s2c$Condition2 <- paste(s2c$Restraint, s2c$Sex, sep = "_")


matr <- data[,c("R.FileName","EG.ModifiedPeptide","FG.Quantity")]
matr <- aggregate(FG.Quantity ~ R.FileName + EG.ModifiedPeptide, data = matr, sum)


matr <- spread(matr, key = R.FileName, value = FG.Quantity)

rownames(matr) <- matr[,1]
matr <- matr[,-1]
```

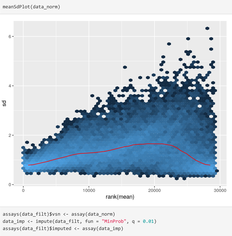
DEP

```
s2csub <- s2c
mat <- matr[,names(matr) %in% s2csub$FileName]
mat$Protein.names <- row.names(mat)
mat$Protein.IDs <- paste("ID_",1:(1:nrow(mat)),sep = "")
data_unique <- make_unique(mat, "Protein.names", "Protein.IDs", delim = "_")

experimental_design <- data.frame(label = as.character(s2csub[,"FileName"]),
                                  condition = as.character(paste(s2csub[,"Restraint"], sep = "_")),
                                  replicate = as.integer(s2csub[,"Replicate"]))
experimental_design[,c("label","condition")] <- lapply(experimental_design[,c("label","condition")], as.character)

#Create DEP data
LFQ_columns <- na.omit(match(s2csub$FileName,colnames(data_unique)))
data_se <- make_se(data_unique, LFQ_columns, experimental_design)
data_filt <- filter_missval(data_se, thr = 2)
data_norm <- normalize_vsn(data_filt)

meanSdPlot(data_norm)
```




```
assays(data_filt)$vsn <- assay(data_norm)
data_imp <- impute(data_filt, fun = "MinProb", q = 0.01)
assays(data_filt)$knnvted <- assay(data_imp)

data_diff_control <- test_diff(data_imp, type = "control", control = "Ctrl")
```

```
## Tested contrasts: X6min_vs_Ctrl, X45min_vs_Ctrl, X135min_vs_Ctrl, X45min_vs_Ctrl, X90min_vs_Ctrl
```

```
data_diff_manual <- test_diff(data_imp, type = "manual", test = c("Ctrl_vs_X6min",
                                                                  "X6min_vs_X45min",
                                                                  "X45min_vs_X90min",
                                                                  "X90min_vs_X96min",
                                                                  "X96min_vs_X135min"))
```

```
## Tested contrasts: Ctrl_vs_X6min, X6min_vs_X45min, X45min_vs_X90min, X90min_vs_X96min, X96min_vs_X135min
```

```
#ADDING DIFF CTRL
data_diff_manual <- data_diff_control

#dep_ctrl <- add_rejections(data_diff_control, alpha = 0.05, lfc = log2(0.5))
dep_man <- add_rejections(data_diff_manual, alpha = 0.05, lfc = log2(0.5))

#Export and build Results Data_Frame
data_results <- get_results(dep_man)
assign <- match(data_results$name,data$EG.ModifiedPeptide)
data_results$Protein <- data[assign,"PG.ProteinAccessions"]
data_results$ModifiedSequence <- data[assign,"EG.PTMLocalizationProbabilities"]
data_results$StrippedSequence <- data[assign,"EG.StrippedSequence"]
data_results$Positions <- data[assign,"EG.PTMPositions..Phospho_STY_NL_98..STY..")]
data_results$Probabilities <- data[assign,"EG.PTMProbabilities..Phospho_STY_NL_98..STY..")]
data_results$ModifiedPeptide <- data[assign,"EG.ModifiedPeptide"]

#data_results_01 <- data_results
```

```
selected <- data_results$name[data_results$X6min_vs_Ctrl_p.val <= 0.05 | data_results$X45min_vs_Ctrl_p.val <= 0.05 | data_re
sults$X6min_vs_Ctrl_p.val >= 0.05 | data_results$X90min_vs_Ctrl_p.val <= 0.05 | data_results$X135min_vs_Ctrl_p.adj <= 0.05]
selected_hard <- data_results$name[data_results$X6min_vs_Ctrl_p.val <= 0.05 & data_results$X45min_vs_Ctrl_p.val <= 0.05 & da
ta_results$X45min_vs_Ctrl_p.val <= 0.05 & data_results$X90min_vs_Ctrl_p.val <= 0.05 & data_results$X135min_vs_Ctrl_p.adj <=
0.05]
```

```
#Produce PDF with DEP plots
setwd("/mnt/bohacek/Sebastian/05_rPhos/")
pdf("rPhos_DEPPlots_AllComparisons.pdf", width = 20, height = 15)
plot_frequency(data_se)
plot_numbers(data_filt)
plot_coverage(data_filt)
plot_normalization(data_filt, data_norm)
#plot_missval(data_filt)
plot_imputation(data_norm, data_imp)
plot_cor(dep_man, significant = TRUE, lower = 0, upper = 1, pal = "Reds")
plot_heatmap(dep_man, type = "centered", kmeans = TRUE,
             k = 5, col_limit = 4, show_row_names = FALSE,
             indicate = c("condition"))
plot_heatmap(dep_man, type = "contrast", kmeans = TRUE,
             k = 5, col_limit = 10, show_row_names = FALSE)
plot_pca(dep_man, x = 1, y = 2, n = 1000, point_size = 4, )

dev.off()
```

```
## png
##   2
```

#Data preparation for SE ##############################

```
data_ModPep <-  aggregate(FG.Quantity ~ R.FileName * EG.ModifiedPeptide * Restraint, data, sum)
data_ModPep$log2 <- log2(data_ModPep$FG.Quantity)


data_iRT <- aggregate(EG.iRTEmpirical ~ R.FileName * EG.ModifiedPeptide * Restraint, data, mean)
data_ApexRT <- aggregate(EG.ApexRT ~ R.FileName * EG.ModifiedPeptide * Restraint, data, mean)
data_Qvalue <- aggregate(EG.Qvalue ~ R.FileName * EG.ModifiedPeptide * Restraint, data, mean)


assay_Quantity <- spread(data_ModPep[,c("R.FileName", "EG.ModifiedPeptide", "FG.Quantity")], key = R.FileName , value = FG.Q
uantity)
rownames(assay_Quantity) <- assay_Quantity[,1]
assay_Quantity[,1] <- NULL


assay_log2 <- spread(data_ModPep[,c("R.FileName", "EG.ModifiedPeptide", "log2")], key = R.FileName , value = log2)
rownames(assay_log2) <- assay_log2[,1]
assay_log2[,1] <- NULL
#boxplot(assay_log2, 2, na.rm=T)


temp_data <- data[!duplicated(data$EG.ModifiedPeptide),][,!colnames(data) %in% c("R.FileName", "Restraint", "Sex", "Stress",
"log2", "EG.ApexRT", "EG.iRTEmpirical", "EG.Qvalue")]
temp_data <- temp_data[,c("EG.ModifiedPeptide",colnames(temp_data)[!colnames(temp_data) == "EG.ModifiedPeptide"])]


temp_data <- temp_data[match(rownames(assay_log2), temp_data$EG.ModifiedPeptide),]
assay_ApexRT <- as.data.frame(data_ApexRT[,c("R.FileName", "EG.ModifiedPeptide", "EG.ApexRT")], key = R.FileName , value = EG.ApexR
T)
rownames(assay_ApexRT) <- assay_ApexRT[,1]
assay_ApexRT[,1] <- NULL
assay_iRTEmpirical <- spread(data_iRT[,c("R.FileName", "EG.ModifiedPeptide", "EG.iRTEmpirical")], key = R.FileName , value =
EG.iRTEmpirical)
rownames(assay_iRTEmpirical) <- assay_iRTEmpirical[,1]
assay_iRTEmpirical[,1] <- NULL
assay_Qvalue <- spread(data_Qvalue[,c("R.FileName", "EG.ModifiedPeptide", "EG.Qvalue")], key = R.FileName , value = EG.Qvalu
e)
rownames(assay_Qvalue) <- assay_Qvalue[,1]
assay_Qvalue[,1] <- NULL

SE.rPhos <- SummarizedExperiment(assays = list(log2 = assay_log2, Quantity = assay_Quantity, ApexRT = assay_ApexRT, iRTEmpir
ical = assay_iRTEmpirical, Qvalue = assay_Qvalue) , colData = meta_data, rowData = temp_data) #, ApexRT = assay_ApexRT, iRT
= assay_iRTEmpirical, Qvalue = assay_Qvalue
)
```

```
se <- SE.rPhos
se$Group <- factor(se$Group,levels = c("Control", "6 min", "45 min", "90 min", "96 min", "135 min"))
se <- se[,order(se$Group, se$Sex)]

library(sechm)
```

```
##
## Attaching package: 'sechm'
```

```
## The following object is masked from 'package:StTools':
##
##     meltSE
```

```
p1 <- sechm(se, features =  selected, assayName = "log2", do.scale = T, top_annotation  = c("Sex","Group"), cluster_rows =
T, cluster_cols = F, show_rownames = F, show_colnames = F)

## `use_raster` is automatically set to TRUE for a matrix with more than
## 2000 rows. You can control `use_raster` argument by explicitly setting
## TRUE/FALSE to it.
##
## Set `ht_opt$message = FALSE` to turn off this message.
```

```
print("Genes with sig. adj. p-value at any time point")
```

```
## [1] "Genes with sig. adj. p-value at any time point"
```

p1


```
pdf(file = "/mnt/bohacek/Sebastian/rPhos_HM_all_clustered.pdf", width = 6, height = 6)
print(p1)
dev.off()
```

```
## png
##   2
```

```
library(sechm)
p2 <- sechm(se, features =  selected_hard, assayName = "log2", do.scale = T, top_annotation  = c("Sex","Group"), cluster_row
s = T, cluster_cols = F, show_rownames = F, show_colnames = F)
print("Genes with sig adj. p-value in all time points")
```

```
## [1] "Genes with sig adj. p-value in all time points"
```

p2