

# Unraveling Protein-Signaling Networks from Multiparameter Single-Cell Data

Elena Tortosa

2/1/2019

## Libraries and work directory

I loaded all required libraries and set the work directory.

```
#install.packages("broom")
library(broom)
library(tidyverse)
library(Hmisc)
library(gridExtra)
library(tidyr)
library(corrplot)
library(dplyr)
library(ggplot2)
library(gridExtra)
library(GGally)
library(d3heatmap)
library(gplots)
library(reshape2)
library(plotly)
library(dunn.test)
library(psych)
library(fBasics)
library(purrr)
setwd("/Users/Tortosae/Desktop/Data science course/Capstone_project")
require(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60), tidy=TRUE)
```

## Data tables

All data was provided in 14 different excel sheets. I saved all as .csv and loaded them in R. They were named with a number (df#) followed by the name of the perturbation + treatment each table contains.

```
df1_cd3cd28 <- read.table(file = "1. cd3cd28.csv", sep = ",",
  header = TRUE)
df2_cd3cd28icam2 <- read.table(file = "2. cd3cd28icam2.csv",
  sep = ",", header = TRUE)
df3_cd3cd28aktinhib <- read.table(file = "3. cd3cd28aktinhib.csv",
  sep = ",", header = TRUE)
df4_cd3cd28g0076 <- read.table(file = "4. cd3cd28g0076.csv",
  sep = ",", header = TRUE)
df5_cd3cd28psitect <- read.table(file = "5. cd3cd28psitect.csv",
  sep = ",", header = TRUE)
df6_cd3cd28u0126 <- read.table(file = "6. cd3cd28u0126.csv",
  sep = ",", header = TRUE)
```

```

df7_cd3cd28ly <- read.table(file = "7. cd3cd28ly.csv", sep = ",",
  header = TRUE)
df8_pma <- read.table(file = "8. pma.csv", sep = ",", header = TRUE)
df9_b2camp <- read.table(file = "9. b2camp.csv", sep = ",", header = TRUE)
df10_cd3cd28icam2aktinhb <- read.table(file = "10. cd3cd28icam2aktinhb.csv",
  sep = ",", header = TRUE)
df11_cd3cd28icam2g0076 <- read.table(file = "11. cd3cd28icam2g0076.csv",
  sep = ",", header = TRUE)
df12_cd3cd28icam2psit <- read.table(file = "12. cd3cd28icam2psit.csv",
  sep = ",", header = TRUE)
df13_cd3cd28icam2u0126 <- read.table(file = "13. cd3cd28icam2u0126.csv",
  sep = ",", header = TRUE)
df14_cd3cd28icam2ly <- read.table(file = "14. cd3cd28icam2ly.csv",
  sep = ",", header = TRUE)

```

## Column names

Only df8 contained two columns with different names (in lower case). I unfied column names.

```
df8_pma <- df8_pma %>% rename(PIP2 = pip2, PIP3 = pip3)
```

## New column for perturbations

Measurements are obtained from two different perturbations (“general perturbation” : GP1 and GP2). At the same time, these perturbations are combined with different treatments (or treatment\_nums) (see below: treatment\_num column). I added a new column called GP to each table to classify the data depending on the general perturbation is applied : GP = 1 for GP1 and GP = 2 for GP2.

```

df1_cd3cd28 <- df1_cd3cd28 %>% mutate(treatment = "cd3cd28") %>%
  mutate(GP = 1)
df2_cd3cd28icam2 <- df2_cd3cd28icam2 %>% mutate(treatment = "cd3cd28icam2") %>%
  mutate(GP = 2)
df3_cd3cd28aktinhb <- df3_cd3cd28aktinhb %>% mutate(treatment = "cd3cd28aktinhb") %>%
  mutate(GP = 1)
df4_cd3cd28g0076 <- df4_cd3cd28g0076 %>% mutate(treatment = "cd3cd28g0076") %>%
  mutate(GP = 1)
df5_cd3cd28psitec <- df5_cd3cd28psitec %>% mutate(treatment = "cd3cd28psitec") %>%
  mutate(GP = 1)
df6_cd3cd28u0126 <- df6_cd3cd28u0126 %>% mutate(treatment = "cd3cd28u0126") %>%
  mutate(GP = 1)
df7_cd3cd28ly <- df7_cd3cd28ly %>% mutate(treatment = "cd3cd28ly") %>%
  mutate(GP = 1)
df8_pma <- df8_pma %>% mutate(treatment = "pma") %>% mutate(GP = 1)
df9_b2camp <- df9_b2camp %>% mutate(treatment = "b2camp") %>%
  mutate(GP = 1)
df10_cd3cd28icam2aktinhb <- df10_cd3cd28icam2aktinhb %>% mutate(treatment = "cd3cd28icam2aktinhb") %>%
  mutate(GP = 2)
df11_cd3cd28icam2g0076 <- df11_cd3cd28icam2g0076 %>% mutate(treatment = "cd3cd28icam2g0076") %>%
  mutate(GP = 2)
df12_cd3cd28icam2psit <- df12_cd3cd28icam2psit %>% mutate(treatment = "cd3cd28icam2psit") %>%
  mutate(GP = 2)
df13_cd3cd28icam2u0126 <- df13_cd3cd28icam2u0126 %>% mutate(treatment = "cd3cd28icam2u0126") %>%

```

```

    mutate(GP = 2)
df14_cd3cd28icam2ly <- df14_cd3cd28icam2ly %>% mutate(treatment = "cd3cd28icam2ly") %>%
    mutate(GP = 2)

```

## New column for treatments (treatment\_num)

As mentioned before, measurements are obtained from two different perturbations (GP1 and GP2). At the same time, these perturbations are combined with different treatments. I added a new column called “treatment\_num” to each table to classify the data depending on the treatment applied : 0 <- no treatment, 1 <- Akt\_inh1, 2 <- PKC\_inh, 3 <- PIP2\_inh, 4 <- MEK\_inh, 5 <- Akt\_inh2, 6 <- PKC\_act and 7 <- PKA\_act.

```

# Add new columns with treatment_nums names
df1_cd3cd28 <- df1_cd3cd28 %>% mutate(treatment = "cd3cd28") %>%
    mutate(treatment_num = 0)
df2_cd3cd28icam2 <- df2_cd3cd28icam2 %>% mutate(treatment = "cd3cd28icam2") %>%
    mutate(treatment_num = 0)
df3_cd3cd28aktinhib <- df3_cd3cd28aktinhib %>% mutate(treatment = "cd3cd28aktinhib") %>%
    mutate(treatment_num = 1)
df4_cd3cd28g0076 <- df4_cd3cd28g0076 %>% mutate(treatment = "cd3cd28g0076") %>%
    mutate(treatment_num = 2)
df5_cd3cd28psitec <- df5_cd3cd28psitec %>% mutate(treatment = "cd3cd28psitec") %>%
    mutate(treatment_num = 3)
df6_cd3cd28u0126 <- df6_cd3cd28u0126 %>% mutate(treatment = "cd3cd28u0126") %>%
    mutate(treatment_num = 4)
df7_cd3cd28ly <- df7_cd3cd28ly %>% mutate(treatment = "cd3cd28ly") %>%
    mutate(treatment_num = 5)
df8_pma <- df8_pma %>% mutate(treatment = "pma") %>% mutate(treatment_num = 6)
df9_b2camp <- df9_b2camp %>% mutate(treatment = "b2camp") %>%
    mutate(treatment_num = 7)
df10_cd3cd28icam2aktinhib <- df10_cd3cd28icam2aktinhib %>% mutate(treatment = "cd3cd28icam2aktinhib") %>%
    mutate(treatment_num = 1)
df11_cd3cd28icam2g0076 <- df11_cd3cd28icam2g0076 %>% mutate(treatment = "cd3cd28icam2g0076") %>%
    mutate(treatment_num = 2)
df12_cd3cd28icam2psit <- df12_cd3cd28icam2psit %>% mutate(treatment = "cd3cd28icam2psit") %>%
    mutate(treatment_num = 3)
df13_cd3cd28icam2u0126 <- df13_cd3cd28icam2u0126 %>% mutate(treatment = "cd3cd28icam2u0126") %>%
    mutate(treatment_num = 4)
df14_cd3cd28icam2ly <- df14_cd3cd28icam2ly %>% mutate(treatment = "cd3cd28icam2ly") %>%
    mutate(treatment_num = 5)

```

## Unique table

I created a unique table for all the perturbations/treatments.

```

alldf <- bind_rows(df1_cd3cd28, df2_cd3cd28icam2, df3_cd3cd28aktinhib,
                    df4_cd3cd28g0076, df5_cd3cd28psitec, df6_cd3cd28u0126, df7_cd3cd28ly,
                    df8_pma, df9_b2camp, df10_cd3cd28icam2aktinhib, df11_cd3cd28icam2g0076,
                    df12_cd3cd28icam2psit, df13_cd3cd28icam2u0126, df14_cd3cd28icam2ly)

```

## Reorder columns

I reorder the columns to have the treatment names and dummy variables first, and after that all the measurements done.

```
alldf <- alldf %>% select(treatment, GP, treatment_num, everything())
```

## Data subsetting

I grouped the data to help in the data visualization

```
GP1 <- subset(alldf, GP == "1", select = c("treatment", "treatment_num",
  "praf", "pmeek", "plcg", "PIP2", "PIP3", "p44.42", "pakts473",
  "PKA", "PKC", "P38", "pjnk"))
GP2 <- subset(alldf, GP == "2", select = c("treatment", "treatment_num",
  "praf", "pmeek", "plcg", "PIP2", "PIP3", "p44.42", "pakts473",
  "PKA", "PKC", "P38", "pjnk"))
```

## Describe the data

I look at some raw data from my dataset.

```
head(alldf)
```

```
##   treatment GP treatment_num praf pmeek plcg PIP2 PIP3 p44.42 pakts473
## 1    cd3cd28  1              0 26.4 13.20  8.82 18.30 58.80  6.61    17.0
## 2    cd3cd28  1              0 35.9 16.50 12.30 16.80  8.13  18.60    32.5
## 3    cd3cd28  1              0 59.4 44.10 14.60 10.20 13.00 14.90    32.5
## 4    cd3cd28  1              0 73.0 82.80 23.10 13.50  1.29  5.83    11.8
## 5    cd3cd28  1              0 33.7 19.80  5.19  9.73 24.80 21.10    46.1
## 6    cd3cd28  1              0 18.8  3.75 17.60 22.10 10.90 11.90    25.7
##   PKA   PKC   P38 pjnk
## 1 414 17.00 44.9 40.0
## 2 352  3.37 16.5 61.5
## 3 403 11.40 31.9 19.5
## 4 528 13.70 28.6 23.1
## 5 305  4.66 25.7 81.3
## 6 610 13.70 49.1 57.8
# class(alldf)
```

I also check the dimensions of my dataset

```
dim(alldf)
```

```
## [1] 11672     14
by_treat <- GP1 %>% group_by(treatment_num)
n_groups(by_treat)

## [1] 8
group_size(by_treat)

## [1] 853 911 723 810 799 848 913 707
```

I also listed the data type of each attribute in my dataset.

```

sapply(alldf, class)

##      treatment          GP treatment_num       praf        pmek
## "character"    "numeric"    "numeric"    "numeric"    "numeric"
##      plcg         PIP2        PIP3     p44.42    pakts473
## "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
##      PKA          PKC        P38      pjnk
## "numeric"    "numeric"    "numeric"    "numeric"

```

I calculated median, IQR and skewness.

```

GP1 %>% select(-treatment) %>% group_by(treatment_num) %>% summarise_all(funs(median,
skewness, IQR))

```

```

## # A tibble: 8 x 34
##   treatment_num praf_median plcg_median PIP2_median PIP3_median
##   <dbl>        <dbl>        <dbl>        <dbl>        <dbl>
## 1 0            52.3        24.4        17          48.3       23.9
## 2 1            49.6        24.1       11.5        142         35.5
## 3 2            269         422        267        466         10.6
## 4 3            57.3        25.6         5          2.10        13.7
## 5 4            231         346        14.7        40.3         20
## 6 5            50          24.4        10.8        38.4        19.2
## 7 6            22.1        17          21.1        32.8        11.9
## 8 7            35.9        1.38        20.4        55.7        22.7
## # ... with 28 more variables: p44.42_median <dbl>, pakts473_median <dbl>,
## #   PKA_median <dbl>, PKC_median <dbl>, P38_median <dbl>,
## #   pjnk_median <dbl>, praf_skewness <dbl>, pmeek_skewness <dbl>,
## #   plcg_skewness <dbl>, PIP2_skewness <dbl>, PIP3_skewness <dbl>,
## #   p44.42_skewness <dbl>, pakts473_skewness <dbl>, PKA_skewness <dbl>,
## #   PKC_skewness <dbl>, P38_skewness <dbl>, pjnk_skewness <dbl>,
## #   praf_IQR <dbl>, pmeek_IQR <dbl>, plcg_IQR <dbl>, PIP2_IQR <dbl>,
## #   PIP3_IQR <dbl>, p44.42_IQR <dbl>, pakts473_IQR <dbl>, PKA_IQR <dbl>,
## #   PKC_IQR <dbl>, P38_IQR <dbl>, pjnk_IQR <dbl>
# install.packages('fBasics') library(fBasics) GP1 %>% select
# (-treatment) %>% group_by(treatment_num) %>% basicStats()

library(psych)
# describe.by(GP1, treatment_num)

```

The further the distribution of the skew value from zero, the larger the skew to the left (negative skew value) or right (positive skew value).

I also looked at correlation in the different treatments.

```

# GP1 %>% select(-treatment) %>% group_by(treatment_num) %>%
# cor()

CorrMatrix <- GP1 %>% select(-treatment) %>% split(.\$treatment_num) %>%
  map(cor)
CorrMatrix

## $`0`
##           treatment_num       praf        pmeek        plcg
## treatment_num 1          NA          NA          NA
## praf          NA 1.000000000 0.793231156 0.025009986

```

```

## pmek          NA  0.793231156  1.000000000  0.006667732
## plcg          NA  0.025009986  0.006667732  1.000000000
## PIP2          NA  0.063294584  0.084906992  0.092750140
## PIP3          NA  0.038396084  0.055070106  0.134639155
## p44.42         NA -0.019248324 -0.014472246  0.002466600
## pakts473      NA -0.027189963 -0.021807150  0.002326521
## PKA           NA -0.004999139 -0.013749669 -0.006452253
## PKC           NA -0.049328028 -0.022648470 -0.006072817
## P38            NA -0.062352098 -0.025565981 -0.002977666
## pjnk          NA  0.003929185 -0.044995201  0.071344230
##               PIP2      PIP3      p44.42      pakts473
## treatment_num   NA        NA        NA        NA
## praf            0.063294584  0.03839608 -0.019248324 -0.027189963
## pmek            0.084906992  0.05507011 -0.014472246 -0.021807150
## plcg            0.092750140  0.13463915  0.002466600  0.002326521
## PIP2            1.000000000  0.27366682 -0.008802046 -0.012385789
## PIP3            0.273666824  1.000000000 -0.030977406 -0.034055372
## p44.42          -0.008802046 -0.03097741  1.000000000  0.991666091
## pakts473        -0.012385789 -0.03405537  0.991666091  1.000000000
## PKA             -0.040767368 -0.04961792  0.388972044  0.435382593
## PKC             -0.014362097  0.03119660  0.001462238  0.005221493
## P38             -0.010723485  0.04921346 -0.003807397 -0.002092585
## pjnk            0.012157192  0.06969519 -0.023074973 -0.022125521
##               PKA      PKC      P38      pjnk
## treatment_num   NA        NA        NA        NA
## praf            -0.004999139 -0.049328028 -0.062352098  0.003929185
## pmek            -0.013749669 -0.022648470 -0.025565981 -0.044995201
## plcg            -0.006452253 -0.006072817 -0.002977666  0.071344230
## PIP2            -0.040767368 -0.014362097 -0.010723485  0.012157192
## PIP3            -0.049617917  0.031196600  0.049213458  0.069695186
## p44.42          0.388972044  0.001462238 -0.003807397 -0.023074973
## pakts473        0.435382593  0.005221493 -0.002092585 -0.022125521
## PKA             1.000000000  0.022792417  0.029805169 -0.057364328
## PKC             0.022792417  1.000000000  0.737275586 -0.206929695
## P38             0.029805169  0.737275586  1.000000000 -0.023436227
## pjnk            -0.057364328 -0.206929695 -0.023436227  1.000000000
##
## $`1`
##               treatment_num      praf      pmek      plcg
## treatment_num          1          NA          NA          NA
## praf                  NA  1.000000000  0.6543711744  0.0148166302
## pmek                  NA  0.654371174  1.0000000000 -0.0009843791
## plcg                  NA  0.014816630 -0.0009843791  1.0000000000
## PIP2                  NA  0.002806440 -0.0320116968  0.1265157208
## PIP3                  NA  0.022382613 -0.0050261591  0.2292799520
## p44.42                 NA  0.017844251 -0.0170879911  0.0300843488
## pakts473                NA  0.017358250 -0.0387185302  0.0498014512
## PKA                   NA  0.079563072  0.0145494647  0.0490266452
## PKC                   NA  0.014247879 -0.0153776811 -0.0001743505
## P38                   NA -0.006444505 -0.0126619284 -0.0077105435
## pjnk                  NA -0.047745025 -0.0160294148 -0.0304544132
##               PIP2      PIP3      p44.42      pakts473
## treatment_num          NA        NA        NA        NA
## praf                  0.002806440  0.0223826126  0.0178442510  0.017358250

```

```

## pmek      -0.032011697 -0.0050261591 -0.0170879911 -0.038718530
## plcg      0.126515721  0.2292799520  0.0300843488  0.049801451
## PIP2      1.0000000000  0.2539154568 -0.0117177269 -0.007056090
## PIP3      0.253915457   1.0000000000 -0.0236047055 -0.022308966
## p44.42    -0.011717727 -0.0236047055  1.0000000000  0.894367989
## pakts473  -0.007056090 -0.0223089659  0.8943679885  1.0000000000
## PKA       0.008841901  0.0009832727  0.2915044346  0.468798610
## PKC       0.026843951  -0.0088344671  0.0207591679  0.051267795
## P38       0.001001583  -0.0220869313  0.0154755972  0.035188572
## pjnk     -0.020607296 -0.0029312442  0.0002449586 -0.005350306
##          PKA          PKC          P38          pjnk
## treatment_num NA          NA          NA          NA
## praf       0.0795630716 0.0142478792 -0.006444505 -0.0477450251
## pmek      0.0145494647 -0.0153776811 -0.012661928 -0.0160294148
## plcg      0.0490266452 -0.0001743505 -0.007710544 -0.0304544132
## PIP2      0.0088419011 0.0268439513  0.001001583 -0.0206072959
## PIP3      0.0009832727 -0.0088344671 -0.022086931 -0.0029312442
## p44.42    0.2915044346 0.0207591679  0.015475597  0.0002449586
## pakts473  0.4687986105 0.0512677951  0.035188572 -0.0053503058
## PKA       1.0000000000 -0.0368043378  0.003912602  0.0001360518
## PKC      -0.0368043378 1.0000000000  0.827303530  0.0611528601
## P38       0.0039126015 0.8273035296  1.0000000000  0.3048573631
## pjnk     0.0001360518 0.0611528601  0.304857363  1.0000000000
##
## $^2^
##          treatment_num      praf      pmek      plcg
## treatment_num      1          NA          NA          NA
## praf            NA 1.0000000000 0.996053803 0.001744788
## pmek            NA 0.996053803 1.0000000000 -0.001467183
## plcg            NA 0.001744788 -0.001467183 1.0000000000
## PIP2            NA 0.001547498 -0.001229328 0.988083700
## PIP3            NA 0.063586888 0.060818011 0.630576783
## p44.42          NA -0.007055850 -0.008192023 0.006384835
## pakts473        NA -0.023567870 -0.025499788 0.002045725
## PKA             NA -0.017937469 -0.019303059 0.008289527
## PKC             NA 0.007849796 0.007222071 0.026384996
## P38             NA 0.012192648 0.010936177 0.026266495
## pjnk            NA -0.066380551 -0.069740362 0.041516107
##          PIP2      PIP3      p44.42      pakts473
## treatment_num      NA          NA          NA          NA
## praf            0.001547498 0.063586888 -0.007055850 -0.023567870
## pmek            -0.001229328 0.060818011 -0.008192023 -0.025499788
## plcg            0.988083700 0.630576783 0.006384835 0.002045725
## PIP2            1.0000000000 0.639912872 0.013115126 0.008008151
## PIP3            0.639912872 1.0000000000 0.027494797 0.020028093
## p44.42          0.013115126 0.027494797 1.0000000000 0.973180921
## pakts473        0.008008151 0.020028093 0.973180921 1.0000000000
## PKA             0.013559033 0.009135856 0.135007843 0.131877825
## PKC             0.036933303 0.025189102 0.007628279 0.016316021
## P38             0.036127282 0.022064093 0.007550670 0.016136848
## pjnk            0.049893443 0.042647104 0.038762963 0.039013349
##          PKA          PKC          P38          pjnk
## treatment_num      NA          NA          NA          NA
## praf            -0.017937469 0.007849796 0.01219265 -0.06638055

```

```

## pmek      -0.019303059 0.007222071 0.01093618 -0.06974036
## plcg      0.008289527 0.026384996 0.02626649  0.04151611
## PIP2      0.013559033 0.036933303 0.03612728  0.04989344
## PIP3      0.009135856 0.025189102 0.02206409  0.04264710
## p44.42    0.135007843 0.007628279 0.00755067  0.03876296
## pakts473  0.131877825 0.016316021 0.01613685  0.03901335
## PKA       1.000000000 0.049814954 0.04644548  0.04641090
## PKC       0.049814954 1.000000000 0.99631134  0.74799506
## P38       0.046445483 0.996311335 1.00000000  0.74402695
## pjnk     0.046410902 0.747995062 0.74402695  1.00000000
##
## $`3`
##          treatment_num      praf      pmek      plcg
## treatment_num      1         NA        NA        NA
## praf            NA 1.000000000 0.68342323 0.0062805687
## pmek            NA 0.683423226 1.00000000 0.0974724083
## plcg            NA 0.006280569 0.09747241 1.0000000000
## PIP2            NA 0.023438956 0.02178010 0.0791545215
## PIP3            NA 0.018776476 0.06957581 0.2021456685
## p44.42          NA -0.045463377 -0.00783438 -0.0236798485
## pakts473        NA -0.066837182 -0.03005500 -0.0281818062
## PKA             NA -0.046146604 -0.01121344 -0.0504306329
## PKC             NA 0.061382395 0.10879203 0.0004964979
## P38             NA 0.102109521 0.18505580 -0.0020814561
## pjnk            NA -0.001006634 -0.03521489 -0.0167193475
##          PIP2      PIP3      p44.42      pakts473
## treatment_num      NA        NA        NA        NA
## praf            0.02343896 0.018776476 -0.0454633768 -0.066837182
## pmek            0.02178010 0.069575812 -0.0078343796 -0.030055000
## plcg            0.07915452 0.202145669 -0.0236798485 -0.028181806
## PIP2            1.00000000 0.415312755 -0.0105962232 -0.020363052
## PIP3            0.41531276 1.000000000 -0.0276601701 -0.005627625
## p44.42          -0.01059622 -0.027660170 1.0000000000 0.898977088
## pakts473        -0.02036305 -0.005627625 0.8989770879 1.000000000
## PKA             0.01014770 0.076183741 0.3734649016 0.492763093
## PKC             -0.03443579 -0.026968222 -0.0245571378 -0.019650108
## P38             -0.03216247 -0.049652974 -0.0178934329 -0.021685650
## pjnk            0.01760287 0.023003405 -0.0006389141 0.008300476
##          PKA      PKC      P38      pjnk
## treatment_num      NA        NA        NA        NA
## praf            -0.046146604 0.0613823954 0.102109521 -0.0010066335
## pmek            -0.011213444 0.1087920272 0.185055805 -0.0352148901
## plcg            -0.050430633 0.0004964979 -0.002081456 -0.0167193475
## PIP2            0.010147701 -0.0344357929 -0.032162470 0.0176028654
## PIP3            0.076183741 -0.0269682223 -0.049652974 0.0230034052
## p44.42          0.373464902 -0.0245571378 -0.017893433 -0.0006389141
## pakts473        0.492763093 -0.0196501080 -0.021685650 0.0083004758
## PKA             1.000000000 0.0196970885 -0.015939810 0.0054131888
## PKC             0.019697088 1.0000000000 0.548152133 -0.1447806840
## P38             -0.015939810 0.5481521332 1.000000000 0.0527610839
## pjnk            0.005413189 -0.1447806840 0.052761084 1.00000000000
##
## $`4`
##          treatment_num      praf      pmek      plcg

```

```

## treatment_num      1          NA          NA          NA
## praf              NA 1.000000000 0.997547034 0.014393293
## pmek              NA 0.997547034 1.000000000 0.013478849
## plcg              NA 0.014393293 0.013478849 1.000000000
## PIP2              NA -0.017149956 -0.019756756 0.203661536
## PIP3              NA -0.015392451 -0.017615355 0.300386246
## p44.42             NA -0.056953550 -0.058183070 -0.004159904
## pakts473           NA -0.054032189 -0.057935350 -0.013144011
## PKA               NA -0.018306015 -0.017793065 0.005267276
## PKC               NA -0.025567449 -0.025061681 0.060182300
## P38                NA -0.030558598 -0.028428755 0.012959449
## pjnk               NA -0.006245316 -0.008910779 0.044810583
##                  PIP2      PIP3      p44.42      pakts473
## treatment_num      NA          NA          NA          NA
## praf              -0.017149956 -0.015392451 -0.056953550 -0.05403219
## pmek              -0.019756756 -0.017615355 -0.058183070 -0.05793535
## plcg              0.203661536 0.300386246 -0.004159904 -0.01314401
## PIP2              1.000000000 0.267245798 -0.030896919 0.01903670
## PIP3              0.267245798 1.000000000 -0.001519511 0.04846579
## p44.42             -0.030896919 -0.001519511 1.000000000 0.53658306
## pakts473           0.019036696 0.048465792 0.536583058 1.000000000
## PKA               -0.007575453 -0.014239755 0.717494637 0.50573189
## PKC               0.092887480 0.009458253 0.029079160 -0.01249728
## P38                0.043596677 0.015495289 0.020975311 -0.02572772
## pjnk               0.040645257 -0.016093663 0.025606817 -0.02000489
##                  PKA      PKC      P38      pjnk
## treatment_num      NA          NA          NA          NA
## praf              -0.018306015 -0.025567449 -0.030558598 -0.006245316
## pmek              -0.017793065 -0.025061681 -0.028428755 -0.008910779
## plcg              0.005267276 0.060182300 0.012959449 0.044810583
## PIP2              -0.007575453 0.092887480 0.043596677 0.040645257
## PIP3              -0.014239755 0.009458253 0.015495289 -0.016093663
## p44.42             0.717494637 0.029079160 0.020975311 0.025606817
## pakts473           0.505731889 -0.012497279 -0.025727720 -0.020004885
## PKA               1.000000000 0.023337635 0.009322763 0.026377314
## PKC               0.023337635 1.000000000 0.736788499 0.263522254
## P38                0.009322763 0.736788499 1.000000000 0.363592209
## pjnk               0.026377314 0.263522254 0.363592209 1.000000000
##
## $^5`
##      treatment_num      praf      pmek      plcg
## treatment_num      1          NA          NA          NA
## praf              NA 1.000000000 0.6544521493 0.0078715909
## pmek              NA 0.654452149 1.0000000000 0.0006919196
## plcg              NA 0.007871591 0.0006919196 1.0000000000
## PIP2              NA 0.055155917 -0.0027997465 0.0331567361
## PIP3              NA 0.080085443 -0.0124483828 0.1091416857
## p44.42             NA 0.014558565 -0.0133585524 -0.0116323775
## pakts473           NA 0.016030859 -0.0178555295 -0.0116967785
## PKA               NA -0.022201213 -0.0148499174 0.0192988822
## PKC               NA -0.032392886 -0.0206089966 -0.0252578984
## P38                NA -0.010565148 -0.0134184009 -0.0427765983
## pjnk               NA -0.014946533 0.0251646746 -0.0115280142
##                  PIP2      PIP3      p44.42      pakts473

```

```

## treatment_num      NA          NA          NA          NA
## praf            0.0551559166  0.080085443  0.014558565  0.016030859
## pmeek           -0.0027997465 -0.012448383 -0.013358552 -0.017855529
## plcg            0.0331567361  0.109141686 -0.011632378 -0.011696778
## PIP2             1.0000000000  0.294416407  0.004328324 -0.004316861
## PIP3             0.2944164072  1.000000000 -0.008365691 -0.018304524
## p44.42           0.0043283236 -0.008365691  1.000000000  0.934408666
## pakts473        -0.0043168610 -0.018304524  0.934408666  1.000000000
## PKA              0.0164266396 -0.021936279  0.401364472  0.538780147
## PKC              -0.0877118385 -0.038575646  0.007489546 -0.006515127
## P38              -0.0007203019  0.021846386 -0.016712724 -0.017188613
## pjnk             -0.0002148966  0.009724781 -0.020830471  0.004281129
##                               PKA          PKC          P38          pjnk
## treatment_num      NA          NA          NA          NA
## praf            -0.0222012126 -0.032392886 -0.0105651483 -0.0149465330
## pmeek           -0.0148499174 -0.020608997 -0.0134184009  0.0251646746
## plcg            0.0192988822 -0.025257898 -0.0427765983 -0.0115280142
## PIP2             0.0164266396 -0.087711839 -0.0007203019 -0.0002148966
## PIP3             -0.0219362791 -0.038575646  0.0218463865  0.0097247810
## p44.42           0.4013644722  0.007489546 -0.0167127240 -0.0208304714
## pakts473        0.5387801470 -0.006515127 -0.0171886133  0.0042811290
## PKA              1.0000000000 -0.012263008  0.0009012257 -0.0057473945
## PKC              -0.0122630077  1.000000000  0.6491534903 -0.1714842170
## P38              0.0009012257  0.649153490  1.0000000000  0.1145096672
## pjnk             -0.0057473945 -0.171484217  0.1145096672  1.0000000000
##
## $`6`
##                               treatment_num      praf        pmeek        plcg
## treatment_num      1          NA          NA          NA
## praf            NA 1.0000000000  0.770180591  0.058893208
## pmeek           NA 0.770180591  1.000000000  0.016714674
## plcg            NA 0.058893208  0.016714674  1.000000000
## PIP2             NA 0.046363271  0.033979220  0.640962095
## PIP3             NA 0.024294813  0.010623531  0.579146803
## p44.42           NA 0.038297711 -0.001788745 -0.018854313
## pakts473         NA 0.032561535 -0.011897555 -0.018666352
## PKA              NA 0.021576411 -0.006198656 -0.002610244
## PKC              NA 0.014930136 -0.004700243  0.044795289
## P38              NA -0.005689966 -0.016895024  0.045857645
## pjnk             NA 0.012870552  0.010247601  0.046665445
##                               PIP2        PIP3        p44.42      pakts473
## treatment_num      NA          NA          NA          NA
## praf            0.046363271  0.02429481  0.038297711  0.03256154
## pmeek           0.033979220  0.01062353 -0.001788745 -0.01189756
## plcg            0.640962095  0.57914680 -0.018854313 -0.01866635
## PIP2             1.0000000000  0.63258107 -0.032368839 -0.04669902
## PIP3             0.632581067  1.000000000 -0.024138939 -0.03140632
## p44.42           -0.032368839 -0.02413894  1.000000000  0.95303688
## pakts473        -0.046699017 -0.03140632  0.953036883  1.000000000
## PKA              -0.061290941 -0.04829442  0.576619572  0.72406236
## PKC              0.044026079  0.01877856 -0.048357736 -0.03467088
## P38              0.040829546  0.01404588 -0.042776643 -0.03243680
## pjnk             0.007384739  0.01377373 -0.009195197 -0.00132233
##                               PKA          PKC          P38          pjnk

```

```

## treatment_num      NA          NA          NA          NA
## praf            0.021576411  0.014930136 -0.005689966  0.012870552
## pmek           -0.006198656 -0.004700243 -0.016895024  0.010247601
## plcg           -0.002610244  0.044795289  0.045857645  0.046665445
## PIP2            -0.061290941  0.044026079  0.040829546  0.007384739
## PIP3            -0.048294418  0.018778564  0.014045882  0.013773734
## p44.42          0.576619572 -0.048357736 -0.042776643 -0.009195197
## pakts473        0.724062360 -0.034670880 -0.032436801 -0.001322330
## PKA             1.000000000  0.023971386  0.007052865  0.025193405
## PKC             0.023971386  1.000000000  0.907072118  0.496259635
## P38              0.007052865  0.907072118  1.000000000  0.638345543
## pjnk            0.025193405  0.496259635  0.638345543  1.000000000
##
## $`7`
## treatment_num      NA          praf        pmek        plcg
## treatment_num      1           NA          NA          NA
## praf              NA  1.000000000  0.157983725 -0.010073292
## pmek              NA  0.157983725  1.000000000 -0.029549853
## plcg              NA -0.010073292 -0.0295498527 1.000000000
## PIP2              NA -0.016992248 -0.0226886286 0.418875769
## PIP3              NA -0.044106953 -0.0528510562 0.380625334
## p44.42            NA -0.012533866 -0.0004946783 0.005068723
## pakts473          NA -0.008960408  0.0051848203 0.006742111
## PKA               NA  0.003906706  0.0465472826 0.013987113
## PKC               NA -0.005248568  0.0262935017 -0.021544286
## P38               NA  0.006458172  0.0231619348 0.042314903
## pjnk              NA  0.023090647 -0.0072806701 -0.018309215
## treatment_num      NA          PIP2        PIP3        p44.42      pakts473
## treatment_num      NA          NA          NA          NA          NA
## praf              -0.01699225 -0.0441069533 -0.0125338664 -0.008960408
## pmek              -0.02268863 -0.0528510562 -0.0004946783 0.005184820
## plcg              0.41887577  0.3806253340  0.0050687227 0.006742111
## PIP2              1.000000000  0.4783927406 -0.0170953978 -0.010885351
## PIP3              0.47839274  1.0000000000 -0.0207879493 -0.018824229
## p44.42            -0.01709540 -0.0207879493  1.0000000000 0.988051041
## pakts473          -0.01088535 -0.0188242294  0.9880510413 1.0000000000
## PKA               0.01555733 -0.0050821508  0.3263554220 0.420457021
## PKC               -0.00255694 -0.0114612752  0.0020829479 -0.003949889
## P38               0.03320940  0.0228570969 -0.0132497857 -0.022564373
## pjnk              0.04481001 -0.0004181645  0.0208459198 0.021436349
## treatment_num      NA          PKA         PKC         P38          pjnk
## treatment_num      NA          NA          NA          NA          NA
## praf              0.003906706 -0.005248568  0.006458172  0.0230906472
## pmek              0.046547283  0.026293502  0.023161935 -0.0072806701
## plcg              0.013987113 -0.021544286  0.042314903 -0.0183092147
## PIP2              0.015557331 -0.002556940  0.033209405  0.0448100078
## PIP3              -0.005082151 -0.011461275  0.022857097 -0.0004181645
## p44.42            0.326355422  0.002082948 -0.013249786 0.0208459198
## pakts473          0.420457021 -0.003949889 -0.022564373 0.0214363488
## PKA               1.000000000 -0.004179191 -0.050281979 0.0302693037
## PKC               -0.004179191  1.0000000000 0.396901710 0.3494718116
## P38               -0.050281979  0.396901710  1.0000000000 0.4473306962
## pjnk              0.030269304  0.349471812  0.447330696  1.0000000000

```

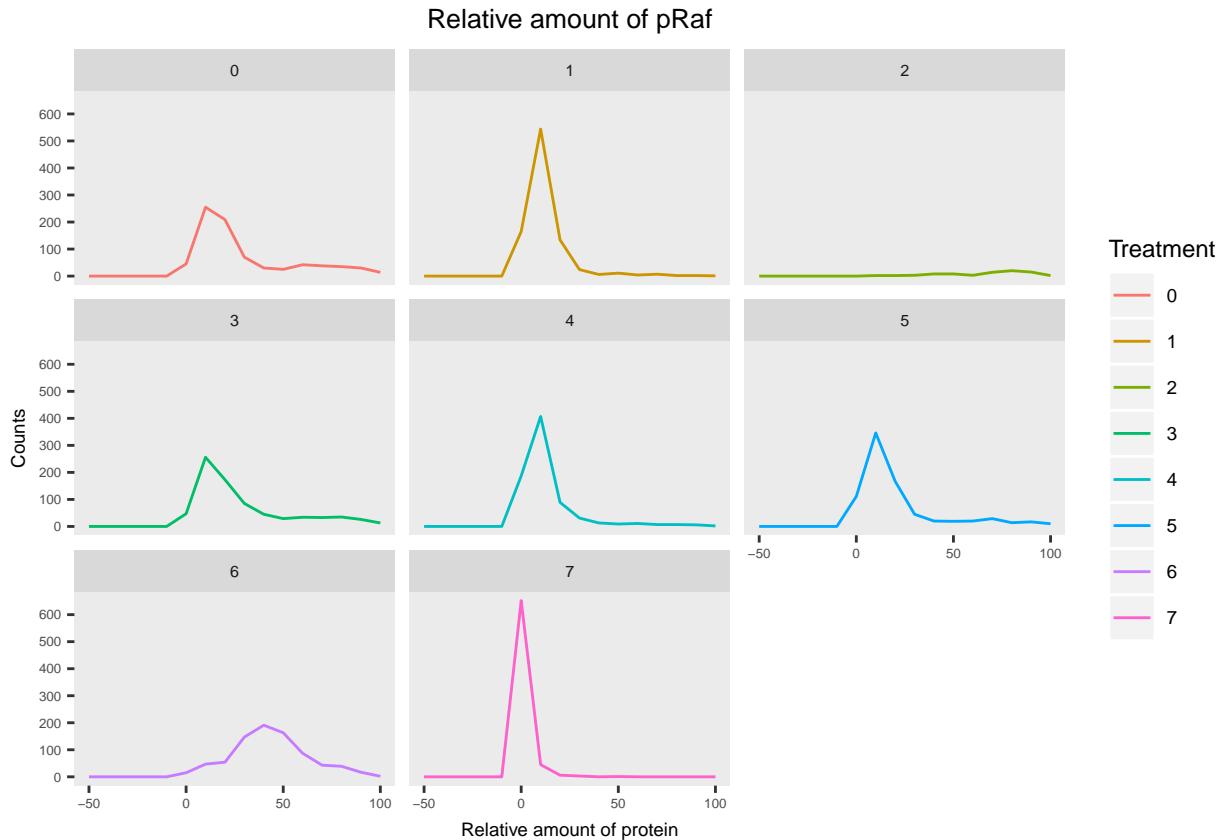
```
# CorrMatrix[[1]][1:3, 1:3]
```

## Data visualization

I made some histograms for some of the proteins in all conditions in order to visualize the distribution of the data.

```
ggplot(GP1, aes(x = pjnk, color = factor(treatment_num))) + geom_freqpoly(binwidth = 10) +
  labs(title = "Relative amount of pRaf", x = "Relative amount of protein",
       y = "Counts", color = "Treatment") + theme(axis.text.x = element_text(size = 5),
                                                 axis.text.y = element_text(size = 5), axis.title.x = element_text(size = 7),
                                                 axis.title.y = element_text(size = 7), plot.title = element_text(hjust = 0.5,
                                                                                         size = 10), strip.text.x = element_text(size = 6), legend.title = element_text(size = 9),
                                                 legend.text = element_text(size = 7), panel.grid.major = element_blank(),
                                                 panel.grid.minor = element_blank()) + xlim(-50, 100) + ylim(0,
                                                                                         600) + scale_y_continuous(breaks = seq(0, 600, by = 100)) +
  facet_wrap(~treatment_num)

## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```



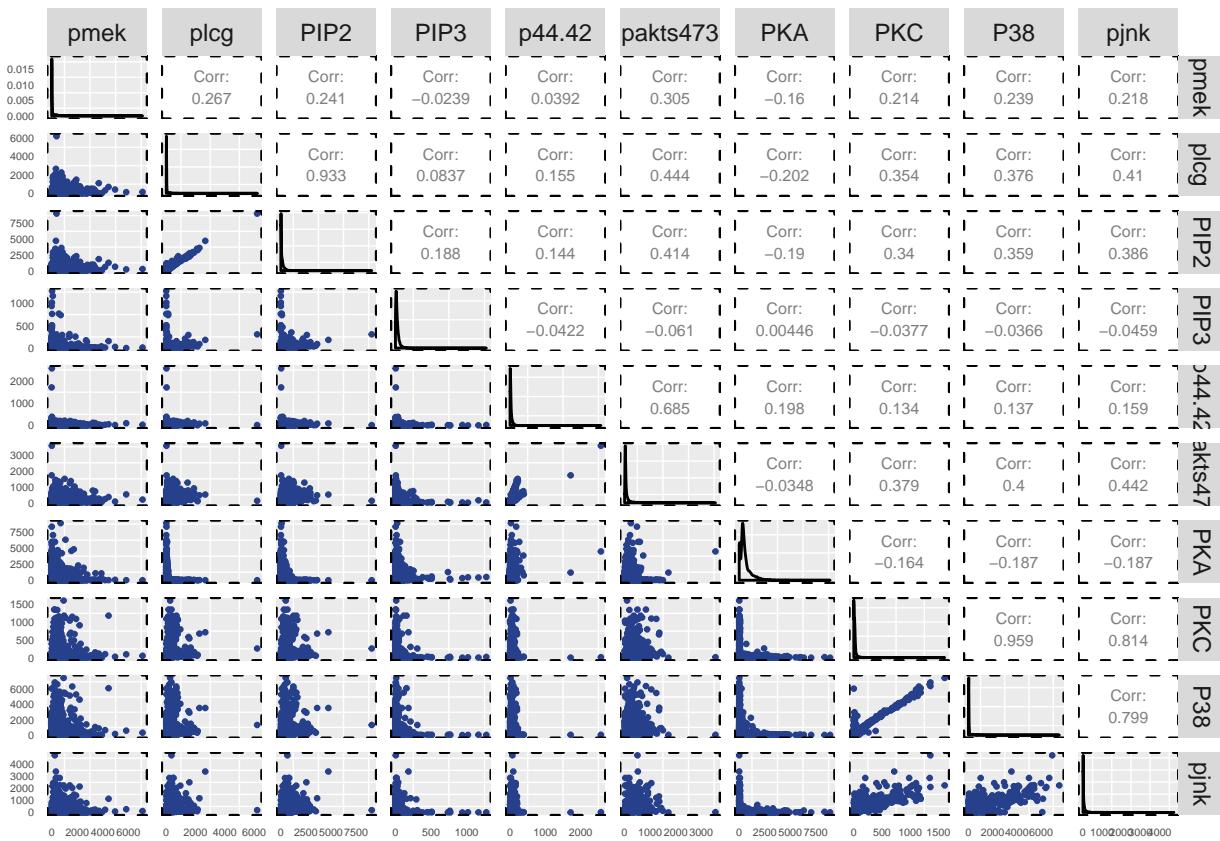
A different way of visualizing the data is with a matrix of plots.

```
CorrPlot <- GP1 %>% select(-treatment, -treatment)
gpairs(CorrPlot, columns = 3:ncol(CorrPlot), upper = list(continuous = wrap("cor",
  size = 2)), lower = list(continuous = wrap("points", size = 0.5,
  color = "royalblue4"))) + theme(legend.position = "none",
```

```

panel.grid.major = element_blank(), axis.text = element_text(size = 4),
axis.ticks = element_blank(), panel.border = element_rect(linetype = "dashed",
colour = "black", fill = NA))

```



## Statistical test to check normality in the data set

I used the Shapiro test to check if the data follows a normal distribution. I grouped the data by treatment (treatment\_num) and protein (variable\_name) and applied the test to each single group.

```
GP1.num <- GP1 %>% select(-treatment)
```

```
normalityGP1 <- GP1.num %>% group_by(treatment_num) %>% summarise_all(.funs = funs(statistic = shapiro.test(.)$statistic,
p.value = shapiro.test(.)$p.value))
normalityGP1
```

```

## # A tibble: 8 x 23
##   treatment_num praf_statistic pmek_statistic plcg_statistic PIP2_statistic
##   <dbl>          <dbl>          <dbl>          <dbl>          <dbl>
## 1 0            0.738          0.512          0.730          0.712
## 2 1            0.922          0.717          0.789          0.843
## 3 2            0.640          0.657          0.622          0.681
## 4 3            0.831          0.554          0.880          0.194
## 5 4            0.726          0.743          0.700          0.621
## 6 5            0.921          0.711          0.689          0.664
## 7 6            0.744          0.708          0.763          0.707
## 8 7            0.748          0.420          0.801          0.773
## # ... with 18 more variables: PIP3_statistic <dbl>,

```

```

## #  p44.42_statistic <dbl>, pakts473_statistic <dbl>, PKA_statistic <dbl>,
## #  PKC_statistic <dbl>, P38_statistic <dbl>, pjnk_statistic <dbl>,
## #  praf_p.value <dbl>, pmek_p.value <dbl>, plcg_p.value <dbl>,
## #  PIP2_p.value <dbl>, PIP3_p.value <dbl>, p44.42_p.value <dbl>,
## #  pakts473_p.value <dbl>, PKA_p.value <dbl>, PKC_p.value <dbl>,
## #  P38_p.value <dbl>, pjnk_p.value <dbl>

library("broom") # for function `tidy`
shapiro_test_results <- GP1.num %>% gather(key = "variable_name",
  value = "value", praf:pjnk) %>% group_by(variable_name, treatment_num) %>%
  do(tidy(shapiro.test(.value))) %>% ungroup() %>% select(-method)

shapiro_test_results

## # A tibble: 88 x 4
##   variable_name treatment_num statistic  p.value
##   <chr>           <dbl>     <dbl>      <dbl>
## 1 P38                  0    0.851  1.26e-27
## 2 P38                  1    0.913  2.29e-22
## 3 P38                  2    0.665  2.28e-35
## 4 P38                  3    0.564  8.83e-41
## 5 P38                  4    0.851  9.43e-27
## 6 P38                  5    0.671  1.50e-37
## 7 P38                  6    0.593  9.63e-42
## 8 P38                  7    0.287  1.06e-45
## 9 p44.42                0    0.0826 4.31e-53
## 10 p44.42               1    0.644  8.77e-40
## # ... with 78 more rows

```

The null hypothesis here is that the sample being tested is normally distributed. Since the p Values are less than the significance level of 0.05, we reject the null hypothesis. Therefore, the tested sample do not follow a normal distribution.

## Statistical test to check significance (Kruskal-Wallis test)

Since my data doesn't follow a normal distribution, I applied the Kruskal-Wallis test as a non-parametric method for testing whether samples originate from the same distribution. I tested if, for each single protein, there are statistical significant differences between the different treatments.

```

df_KruskWall <- GP1 %>% gather(key, value, -treatment, -treatment_num)
lapply(split(df_KruskWall, df_KruskWall$key), function(d) {
  kruskal.test(value ~ treatment_num, data = d)
})

## $P38
##
##  Kruskal-Wallis rank sum test
##
## data:  value by treatment_num
## Kruskal-Wallis chi-squared = 2171.5, df = 7, p-value < 2.2e-16
##
## 
## $p44.42
##
##  Kruskal-Wallis rank sum test

```

```

##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 2303.7, df = 7, p-value < 2.2e-16
##
##
## $pakts473
##
## Kruskal-Wallis rank sum test
##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 2501.4, df = 7, p-value < 2.2e-16
##
##
## $PIP2
##
## Kruskal-Wallis rank sum test
##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 3289.6, df = 7, p-value < 2.2e-16
##
##
## $PIP3
##
## Kruskal-Wallis rank sum test
##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 1110.1, df = 7, p-value < 2.2e-16
##
##
## $pjnk
##
## Kruskal-Wallis rank sum test
##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 3935.4, df = 7, p-value < 2.2e-16
##
##
## $PKA
##
## Kruskal-Wallis rank sum test
##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 2412.7, df = 7, p-value < 2.2e-16
##
##
## $PKC
##
## Kruskal-Wallis rank sum test
##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 3419.5, df = 7, p-value < 2.2e-16
##
##
## $plcg

```

```

## 
## Kruskal-Wallis rank sum test
##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 3288.9, df = 7, p-value < 2.2e-16
##
##
## $pmek
##
## Kruskal-Wallis rank sum test
##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 4485.2, df = 7, p-value < 2.2e-16
##
##
## $praf
##
## Kruskal-Wallis rank sum test
##
## data: value by treatment_num
## Kruskal-Wallis chi-squared = 3191.8, df = 7, p-value < 2.2e-16

```

As the p-values are less than the significance level 0.05, we can conclude that there are significant differences between the treatment groups for all the proteins.

## Pairwise Test For Multiple Comparisons Of Mean Rank Sums (Dunn's-Test)

I calculated pairwise multiple comparisons between different treatments, for each single protein, according to the Dunn's-Test.

```

lapply(split(df_KruskWall, df_KruskWall$key), function(x) {
  dunn.test(df_KruskWall$value, df_KruskWall$treatment_num,
            kw = TRUE, method = "bonferroni", table = TRUE)
})

# df3 <- df_KruskWall %>% group_by(key) %>%
# do(tidy(dunn.test(x= .value, g = .$treatment_num))) df3

```

It didn't work!

## Mean for the different proteins for all treatments

In order to simplify the data, I calculate the median for each protein and for each treatment.

```

stats_GP1 <- GP1 %>% group_by(treatment_num) %>% summarise_at(vars(praf:pjnk),
  median, na.rm = TRUE)
head(stats_GP1)

## # A tibble: 6 x 12
##   treatment_num praf  pmek  plcg    PIP2   PIP3 p44.42 pakts473    PKA    PKC
##   <dbl>     <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl>     <dbl> <dbl> <dbl>
## 1          0    52.3  24.4   17    48.3  23.9  14.9     29.2  437  13.6
## 2          1    49.6  24.1  11.5  142    35.5  17.9     34.3  519  18.3
## 3          2    269   422   267   466    10.6  53.3     325    10  86.6

```

```

## 4      3 57.3 25.6   5     2.10 13.7 20.5     41.8 478 16.7
## 5      4 231 346   14.7 40.3 20     2.71 56.2 211 2.35
## 6      5 50 24.4 10.8 38.4 19.2 12.2     25.3 494. 15.1
## # ... with 2 more variables: P38 <dbl>, pjnk <dbl>
stats_GP2 <- GP2 %>% group_by(treatment_num) %>% summarise_at(vars(praf:pjnk),
  median, na.rm = TRUE)
head(stats_GP2)

## # A tibble: 6 x 12
##   treatment_num praf pmek plcg PIP2 PIP3 p44.42 pakts473 PKA PKC
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1          0 43.7 29.7 21.1 67.6 12.5 17.5 34.3 655 10.6
## 2          1 44.1 31.1 8.13 81.3 20    29.7 50.9 500 7.91
## 3          2 231 343 202 372 19.8 62.6 346 519 35.5
## 4          3 41.8 31.1 7.7 65.8 11.5 15.6 42.6 533 15.7
## 5          4 36.2 30.8 9.82 63.8 18.9 21.9 50.5 813 14.1
## 6          5 35.2 26.4 8.51 32.2 13.6 24.8 50.9 828 13
## # ... with 2 more variables: P38 <dbl>, pjnk <dbl>

```

## Data normalization

Using scale(), I performed a row-wise normalisation of the proteins, so that we normalise the proteins across each experiment and we can compare experiments.

```

GP1.n <- stats_GP1 %>% select(-treatment_num)
GP1.n2 <- scale(GP1.n)
GP1.n2t <- t(GP1.n2)
GP2.n <- stats_GP2 %>% select(-treatment_num)
GP2.n2 <- scale(GP2.n)

```

I check median and sd of normalized data to verify that normalization is correct.

```
round(colMeans(GP1.n2), 1)
```

```

##   praf    pmek    plcg    PIP2    PIP3 p44.42 pakts473 PKA
##   0       0       0       0       0       0       0       0       0
##   PKC    P38    pjnk
##   0       0       0

```

```
apply(GP1.n2, 2, sd)
```

```

##   praf    pmek    plcg    PIP2    PIP3 p44.42 pakts473 PKA
##   1       1       1       1       1       1       1       1       1
##   PKC    P38    pjnk
##   1       1       1

```

```
round(colMeans(GP2.n2), 1)
```

```

##   praf    pmek    plcg    PIP2    PIP3 p44.42 pakts473 PKA
##   0       0       0       0       0       0       0       0       0
##   PKC    P38    pjnk
##   0       0       0

```

```
apply(GP2.n2, 2, sd)
```

```

##   praf    pmek    plcg    PIP2    PIP3 p44.42 pakts473 PKA
##   1       1       1       1       1       1       1       1       1

```

```
##      PKC      P38      pjnk
##      1         1         1
```

## Means visualization

### Lineplots

I plotted the medians for each protein in each condition using a line plot. X axis represent all treatments (from 0 to 7) and Y axis represent the relative amount of protein. Each line represent the median of each single protein for each treatment.

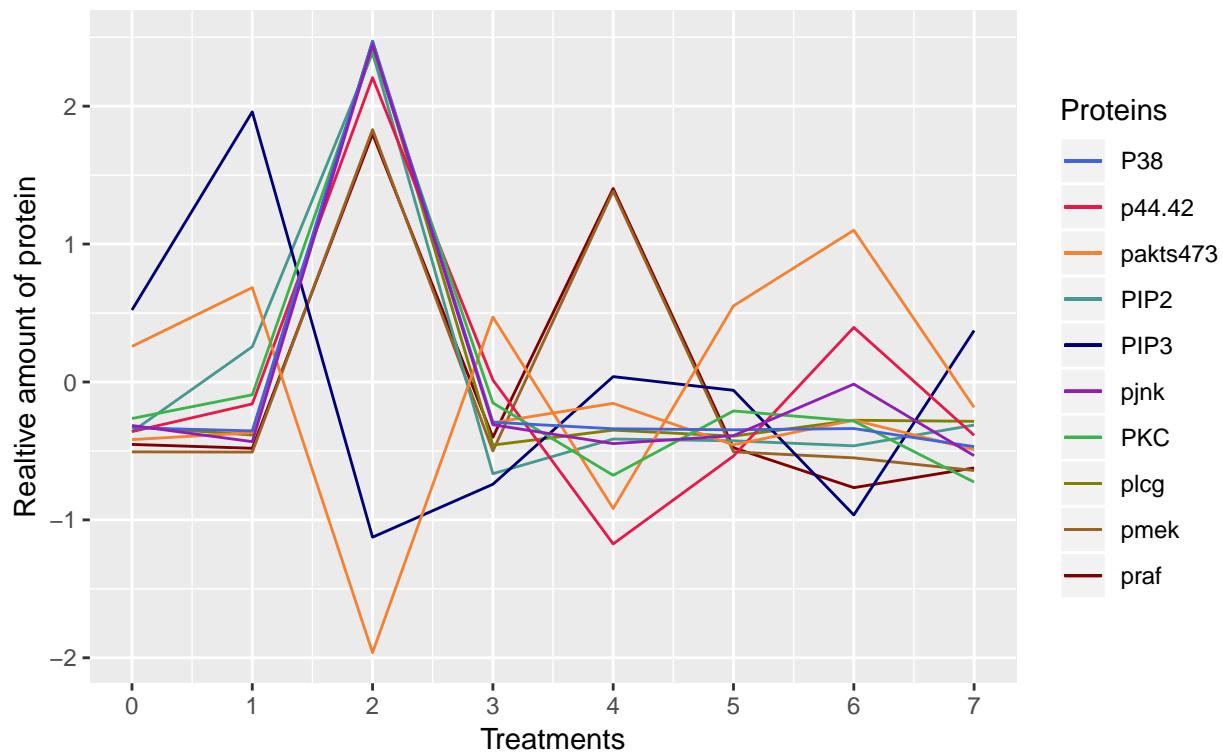
Option#1

```
stats_GP1.n <- as.data.frame(scale(stats_GP1))
head(stats_GP1.n)

##   treatment_num      praf      pmek      plcg      PIP2      PIP3
## 1    -1.4288690 -0.4527949 -0.5066905 -0.3233945 -0.3611310  0.52176936
## 2    -1.0206207 -0.4808349 -0.5084537 -0.3848604  0.2552957  1.95857044
## 3    -0.6123724  1.7976788  1.8301631  2.4705103  2.3868032 -1.12559740
## 4    -0.2041241 -0.4008689 -0.4996376 -0.4575019 -0.6650682 -0.74162470
## 5     0.2041241  1.4030410  1.3834808 -0.3490984 -0.4137608  0.03870693
## 6     0.6123724 -0.4766809 -0.5066905 -0.3926833 -0.4265893 -0.06038280
##      p44.42      pakts473      PKA      PKC      P38      pjnk
## 1 -0.35979198 -0.4179128  0.2577529 -0.2650414 -0.3319723 -0.3156085
## 2 -0.15925767 -0.3682887  0.6841990 -0.0930732 -0.3549154 -0.4329174
## 3  2.20704724  2.4602853 -1.9628874  2.4059535  2.4716744  2.4452990
## 4  0.01453874 -0.2953121  0.4709760 -0.1516155 -0.2918219 -0.3102762
## 5 -1.17462975 -0.1551970 -0.9175743 -0.6766673 -0.3394288 -0.4471367
## 6 -0.53693063 -0.4558607  0.5515847 -0.2101579 -0.3468853 -0.3890747

lab <- labs(title = "Average amount of protein", subtitle = "Medians for each protein and treatment",
            x = "Treatments", y = "Realtive amount of protein", color = "Treatment")
ggplot(stats_GP1.n) + geom_line(aes(x = c(0:7), y = praf, colour = "praf")) +
  geom_line(aes(c(0:7), y = pmek, colour = "pmek")) + geom_line(aes(c(0:7),
  y = plcg, colour = "plcg")) + geom_line(aes(c(0:7), y = PIP2,
  colour = "PIP2")) + geom_line(aes(c(0:7), y = PIP3, colour = "PIP3")) +
  geom_line(aes(c(0:7), y = p44.42, colour = "p44.42")) + geom_line(aes(c(0:7),
  y = pakts473, colour = "pakts473")) + geom_line(aes(c(0:7), y = PKA,
  colour = "pakts473")) + geom_line(aes(c(0:7), y = PKC,
  colour = "PKC")) + geom_line(aes(c(0:7), y = P38, colour = "P38")) +
  geom_line(aes(c(0:7), y = pjnk, colour = "pjnk")) + scale_x_continuous(name = "Treatments",
  breaks = c(0:7), labels = c(0:7)) + lab + scale_color_manual(name = "Proteins",
  values = c(praf = "#800000", pmek = "#9A6324", plcg = "#808000",
  PIP2 = "#469990", PIP3 = "#000075", p44.42 = "#e6194B",
  pakts473 = "#f58231", PKA = "#ffe119", PKC = "#3cb44b",
  P38 = "#4363d8", pjnk = "#911eb4"))
```

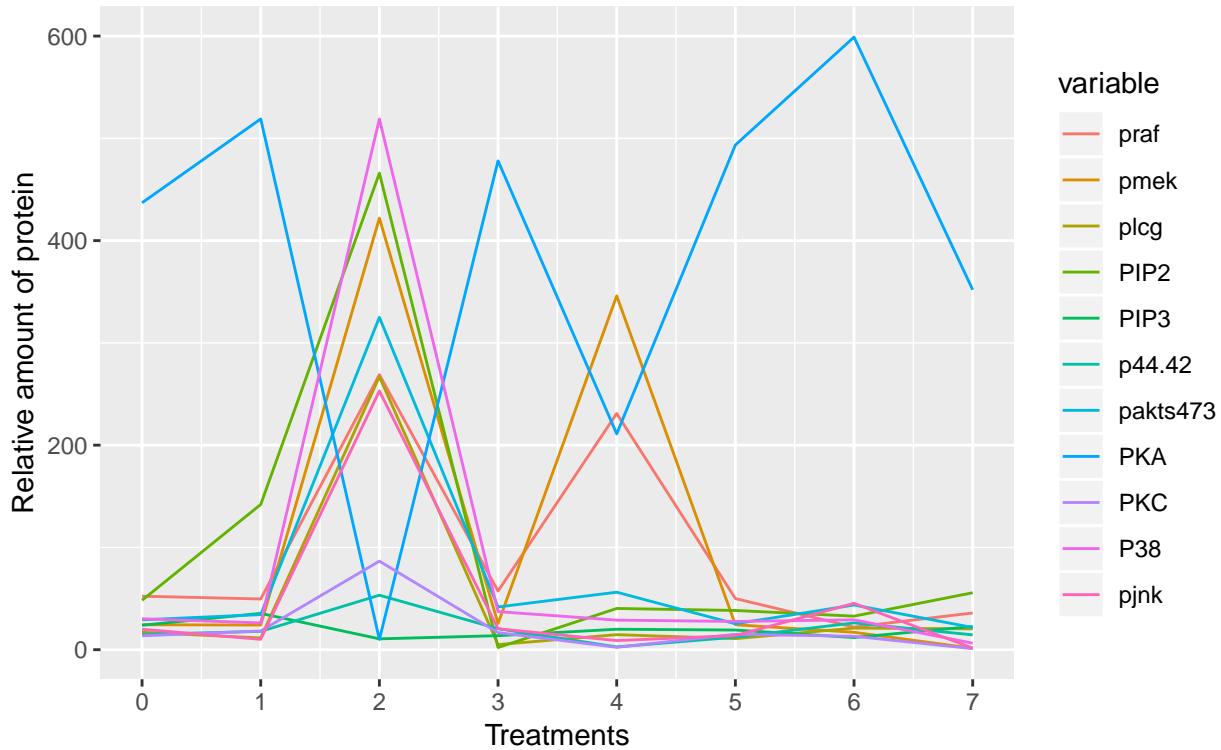
Average amount of protein  
 Medians for each protein and treatment



Option#2

```
test_data_long <- melt(stats_GP1, id = "treatment_num") # convert to long format
ggplot(data = test_data_long, aes(x = treatment_num, y = value,
  colour = variable)) + geom_line() + scale_x_continuous(name = "Treatments",
  breaks = c(0:7), labels = c(0:7)) + scale_y_continuous(name = "Relative amount of protein") +
  labs(title = "Average amount of protein", subtitle = "Medians for each protein and treatment")
```

## Average amount of protein Medians for each protein and treatment



## Protein and treatment clustering

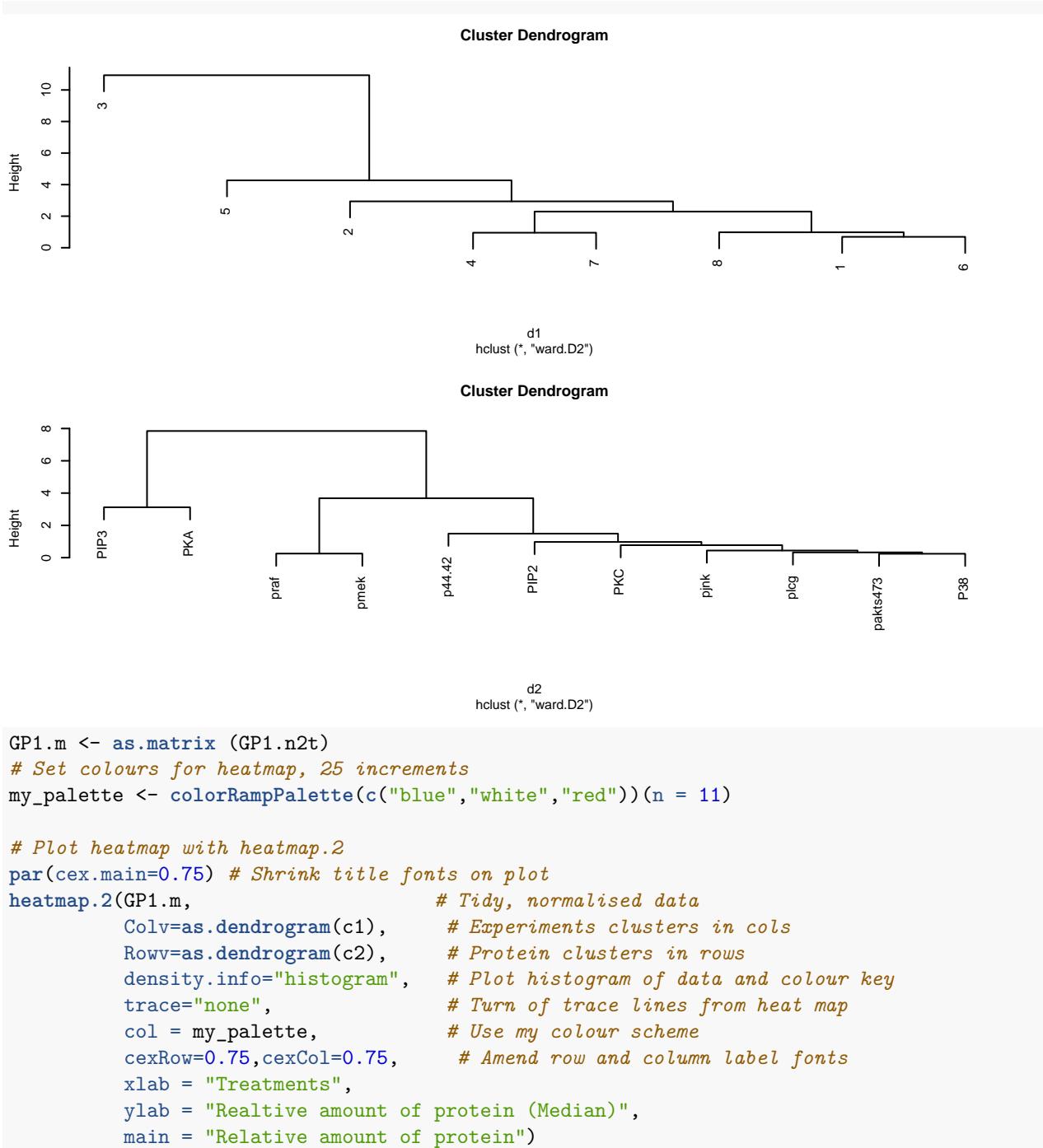
I clustered the proteins and treatments based on their medians.

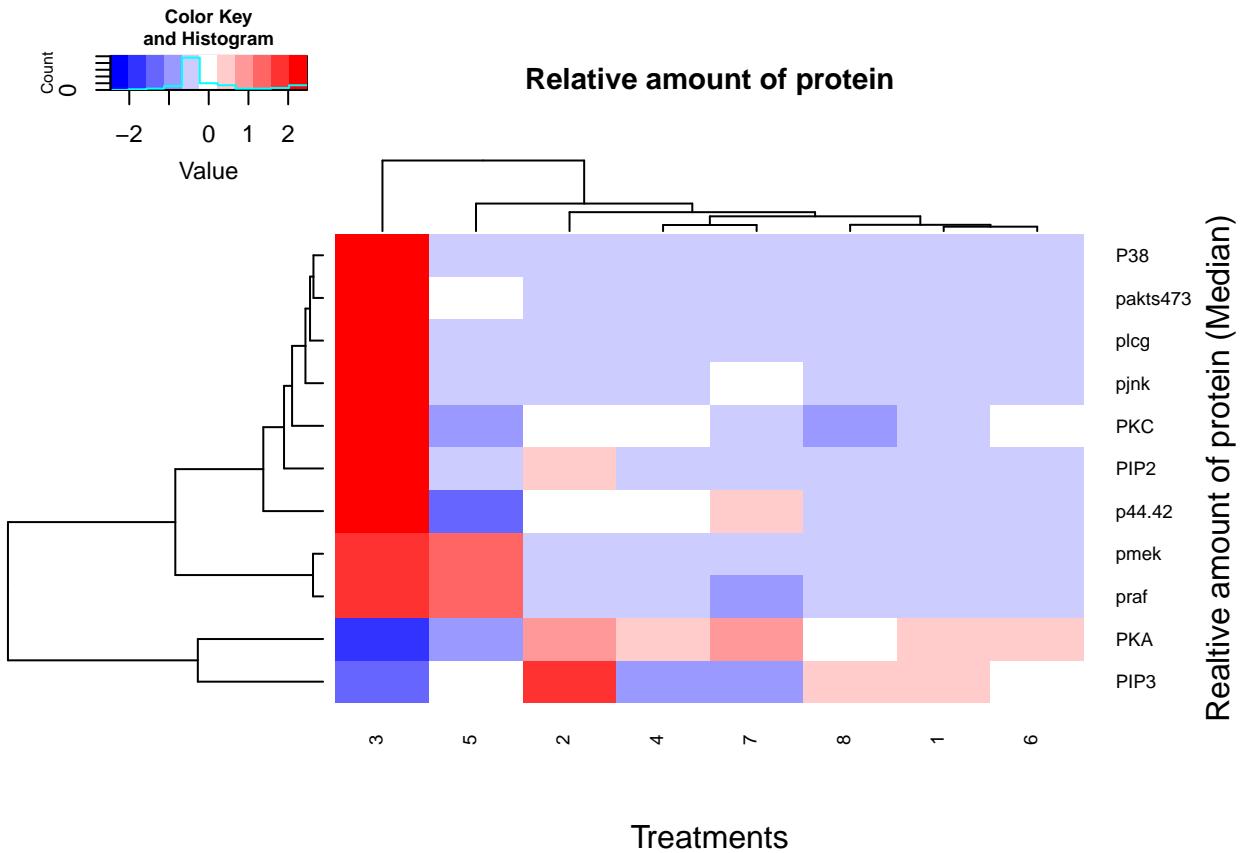
First, I calculated distance between experiments and protein in rows and cluster the data based on these distances. Afterwards, I checked the clustering by plotting dendograms. Finally, I plotted a heatmap together with the dendograms.

```
d1 <- dist(GP1.n2, method = "euclidean", diag = FALSE, upper = FALSE)
round(d1, 3)

##      1     2     3     4     5     6     7
## 2 1.648
## 3 8.427 8.718
## 4 1.387 2.873 8.270
## 5 3.091 3.905 7.915 3.338
## 6 0.690 2.174 8.547 0.938 3.161
## 7 1.927 3.139 8.323 0.952 4.044 1.505
## 8 0.744 2.032 8.662 1.583 3.104 1.058 2.152

d2 <- dist(GP1.n2t, method = "euclidean", diag = FALSE, upper = TRUE)
# Clustering distance between experiments using Ward linkage
c1 <- hclust(d1, method = "ward.D2", members = NULL)
# Clustering distance between proteins using Ward linkage
c2 <- hclust(d2, method = "ward.D2", members = NULL)
# Check clustering by plotting dendograms
par(mfrow=c(2,1), cex=0.5) # Make 2 rows, 1 col plot frame and shrink labels
plot(c1); plot(c2) # Plot both cluster dendograms
```





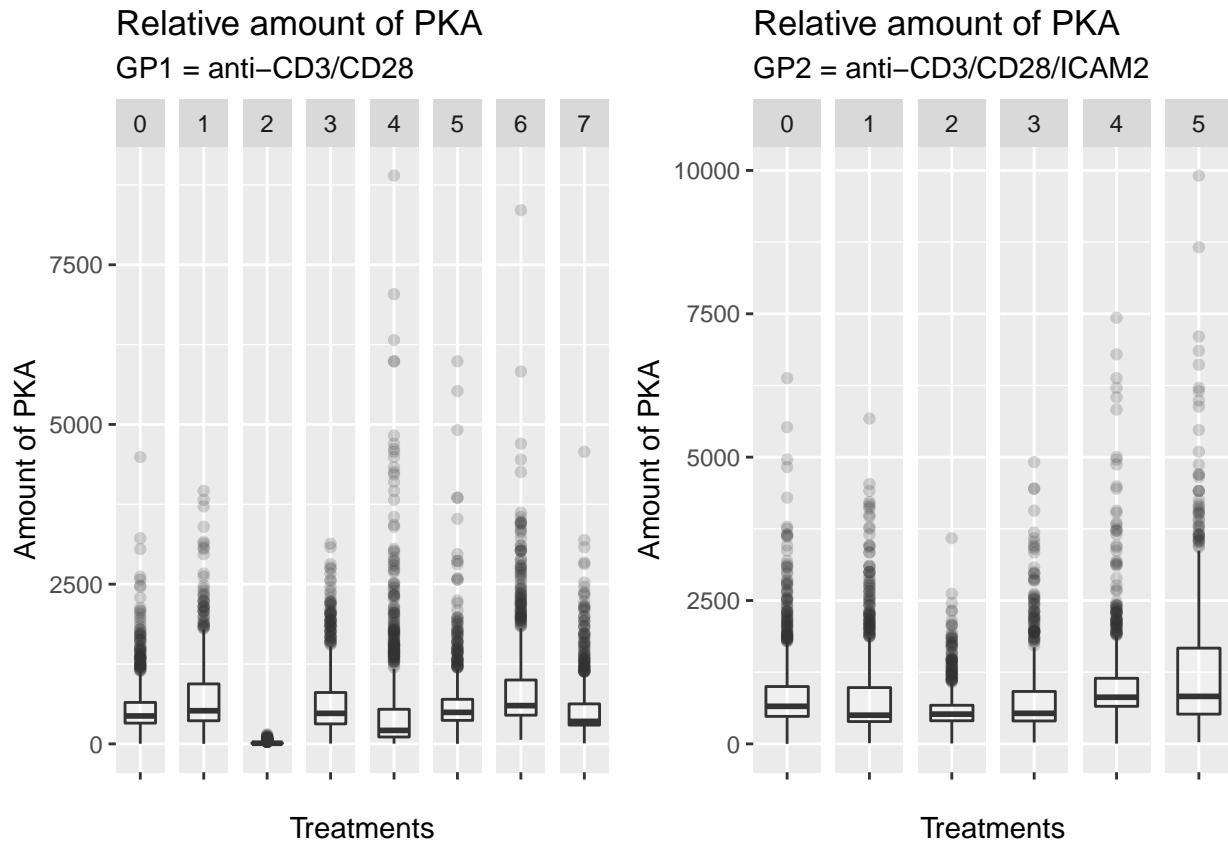
### Box-plots per protein (PKA, PKC, p38, JNK,...)

I represented a graph representing the amount of protein (in this case only for PKA), for each perturbation (GP1 and GP2) and for each single treatment (from 0 to 7). The idea would be to do similar graphs for the different variables.

```
PKA_GP1 <- ggplot(GP1, aes(x = "", y = PKA)) + geom_boxplot(aes(),
  alpha = 0.2) + facet_grid(. ~ treatment_num) + labs(title = "Relative amount of PKA",
  subtitle = "GP1 = anti-CD3/CD28", x = "Treatments", y = "Amount of PKA")

PKA_GP2 <- ggplot(GP2, aes(x = "", y = PKA)) + geom_boxplot(aes(),
  alpha = 0.2) + facet_grid(. ~ treatment_num) + labs(title = "Relative amount of PKA",
  subtitle = "GP2 = anti-CD3/CD28/ICAM2", x = "Treatments",
  y = "Amount of PKA")

grid.arrange(PKA_GP1, PKA_GP2, nrow = 1)
```



### Comparing perturbations/treatment\_nums for PKC

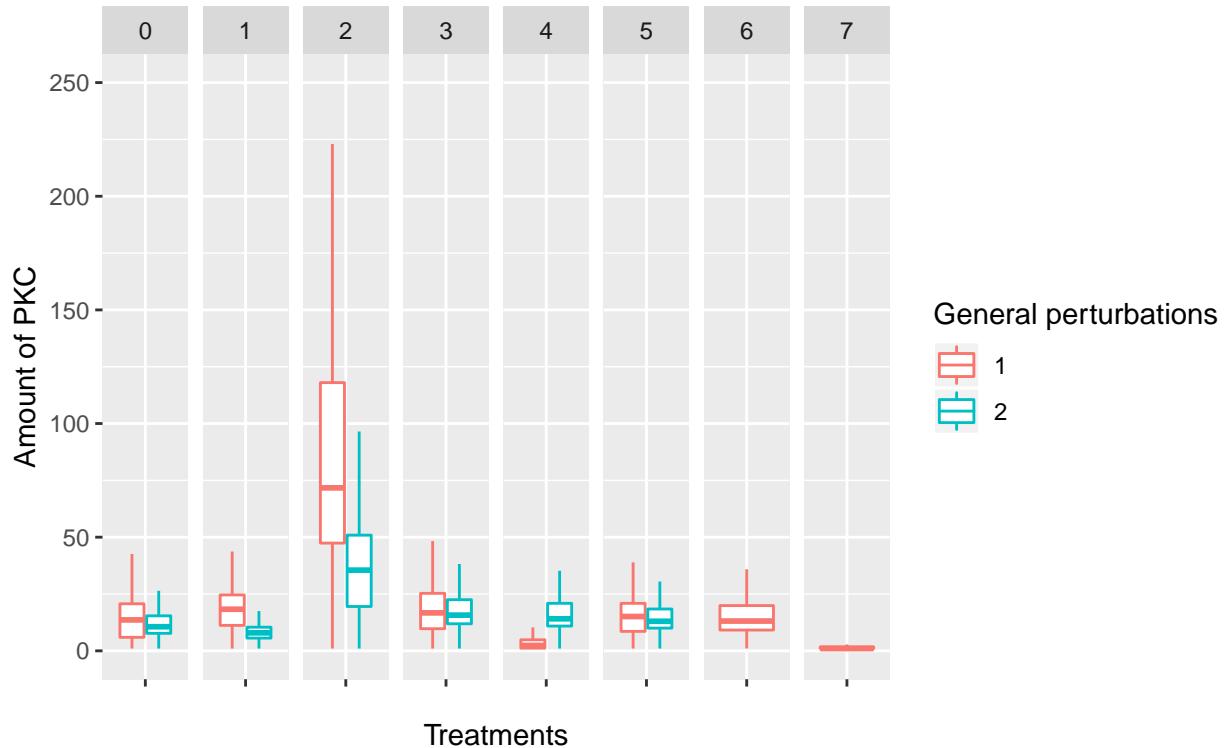
Why are values in X different for depending where I put PKC (X or y)?

The idea is the same that before, but both perturbations in just one graph. I represented a graph representing the amount of protein (in this case only for PKC), for each perturbation (GP1 and GP2) and for each single treatment (from 0 to 7). The idea would be to do similar graphs for the different variables.

```
ggplot(allpdf, aes(x = "", y = PKC, color = factor(GP))) + geom_boxplot(outlier.shape = NA) +
  ylim(0, 250) + facet_grid(. ~ treatment_num) + labs(title = "Relative amount PKC",
  subtitle = "Comparing both general perturbations", x = "Treatments",
  y = "Amount of PKC", color = "General perturbations")
```

## Relative amount PKC

### Comparing both general perturbations



We can see that both perturbations follow similar trends, although there are few differences for treatments 1, 2 and 4.

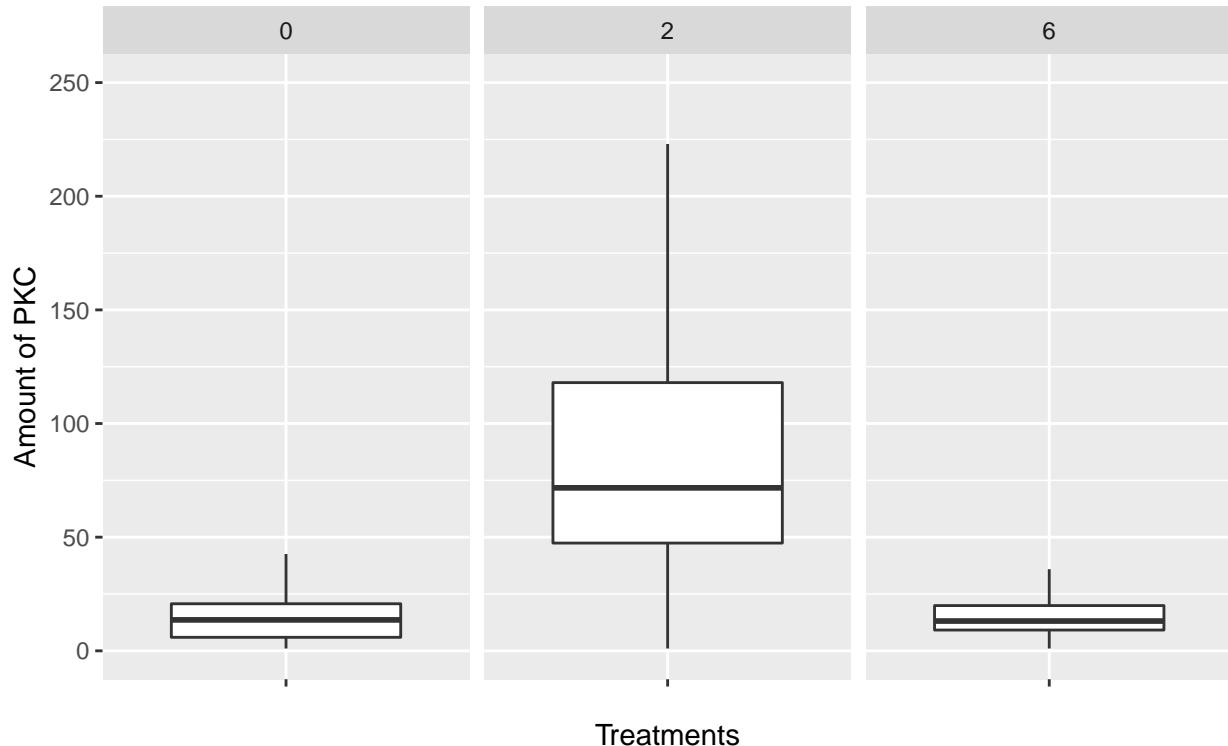
### Checking activation vs inhibition

I check how amount of PKC changes when we activate or inhibit PKC. I represented the amount of protein for PKC in control treatments (cond = 0), when PKC is activated (cond = 6) and when PKC is inhibited (cond = 2)

```
PKC_actvsinh <- GP1 %>% dplyr::filter(treatment_num %in% c("0",
  "2", "6"))
ggplot(PKC_actvsinh, aes(x = "", y = PKC)) + geom_boxplot(outlier.shape = NA) +
  ylim(0, 250) + facet_grid(. ~ treatment_num) + labs(title = "Relative amount of PKC",
  subtitle = "GP1 = anti-CD3/CD28", x = "Treatments", y = "Amount of PKC")
```

## Relative amount of PKC

GP1 = anti-CD3/CD28



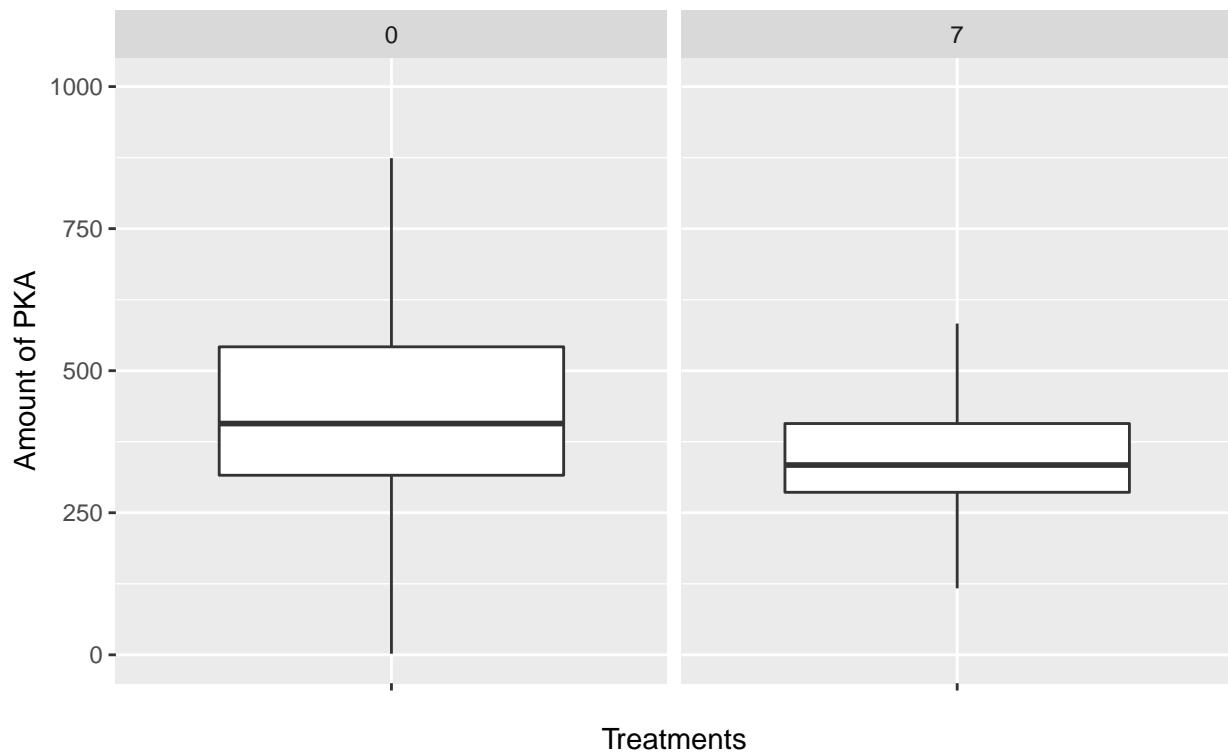
We can see that levels of pPKC increases when we use an activator for this protein.

I also check how the amount of PKA changes when we activate PKA. I represented the amount of protein for PKA in control treatments (cond = 0) and when PKA is activated (cond = 6).

```
PKA_act <- GP1 %>% dplyr::filter(treatment_num %in% c("0", "7"))
ggplot(PKA_act, aes(x = "", y = PKA)) + geom_boxplot(outlier.shape = NA) +
  ylim(0, 1000) + facet_grid(. ~ treatment_num) + labs(title = "Relative amount PKA",
  subtitle = "GP = anti-CD3/CD28", x = "Treatments", y = "Amount of PKA")
```

## Relative amount PKA

GP = anti-CD3/CD28

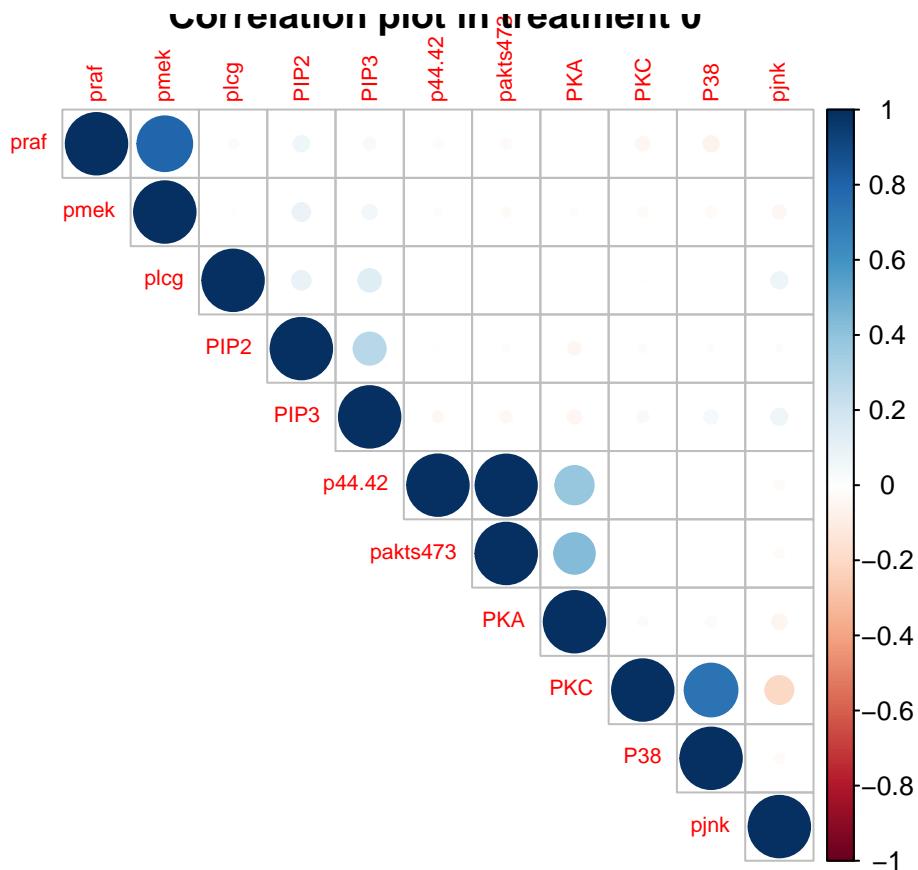


In this case, we don't see big changes in the amount of protein when it's activated.

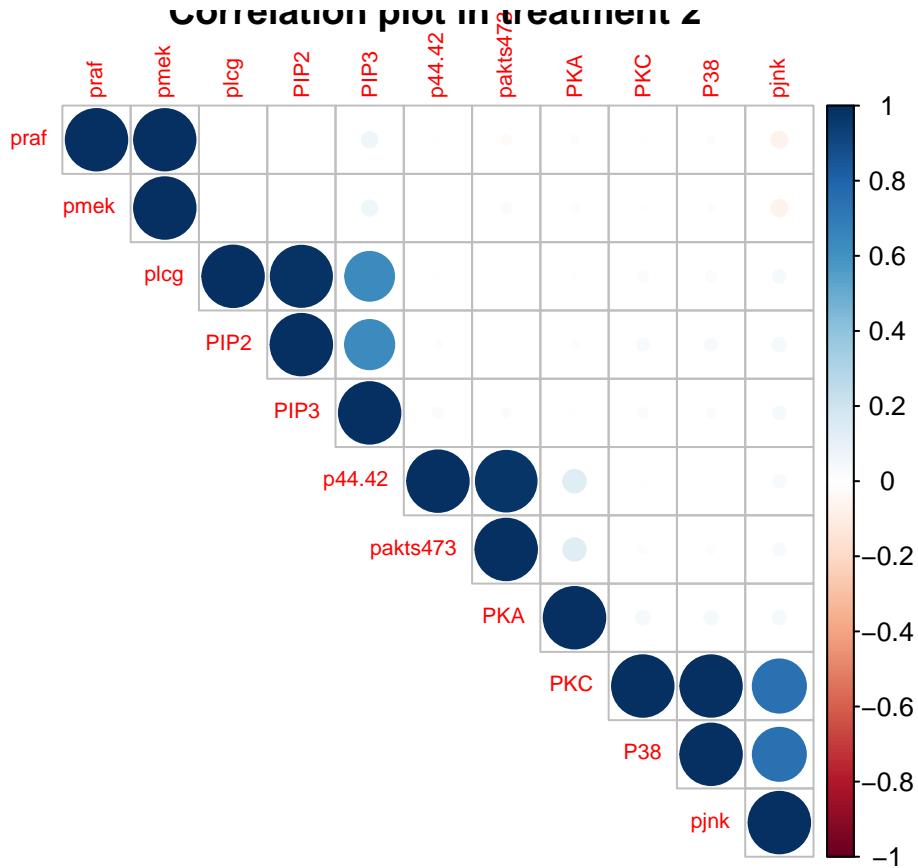
## Looking at some correlations

We check correlations between different proteins change among different treatments (in this case, treatment 0 and 2).

```
CM_0 <- GP1 %>% dplyr::filter(treatment_num == "0") %>% select(-treatment,
  -treatment_num) %>% cor()
corrplot(CM_0, type = "upper", title = "Correlation plot in treatment 0",
  tl.cex = 0.7)
```



```
CM_2 <- GP1 %>% dplyr::filter(treatment_num == "2") %>% select(-treatment,
  -treatment_num) %>% cor()
corrplot(CM_2, type = "upper", title = "Correlation plot in treatment 2",
  tl.cex = 0.7)
```



We see how correlations change among treatments. For example, in treatment 0 there is almost no correlation between plcg and PIP2, and this correlation increases in treatment 2.

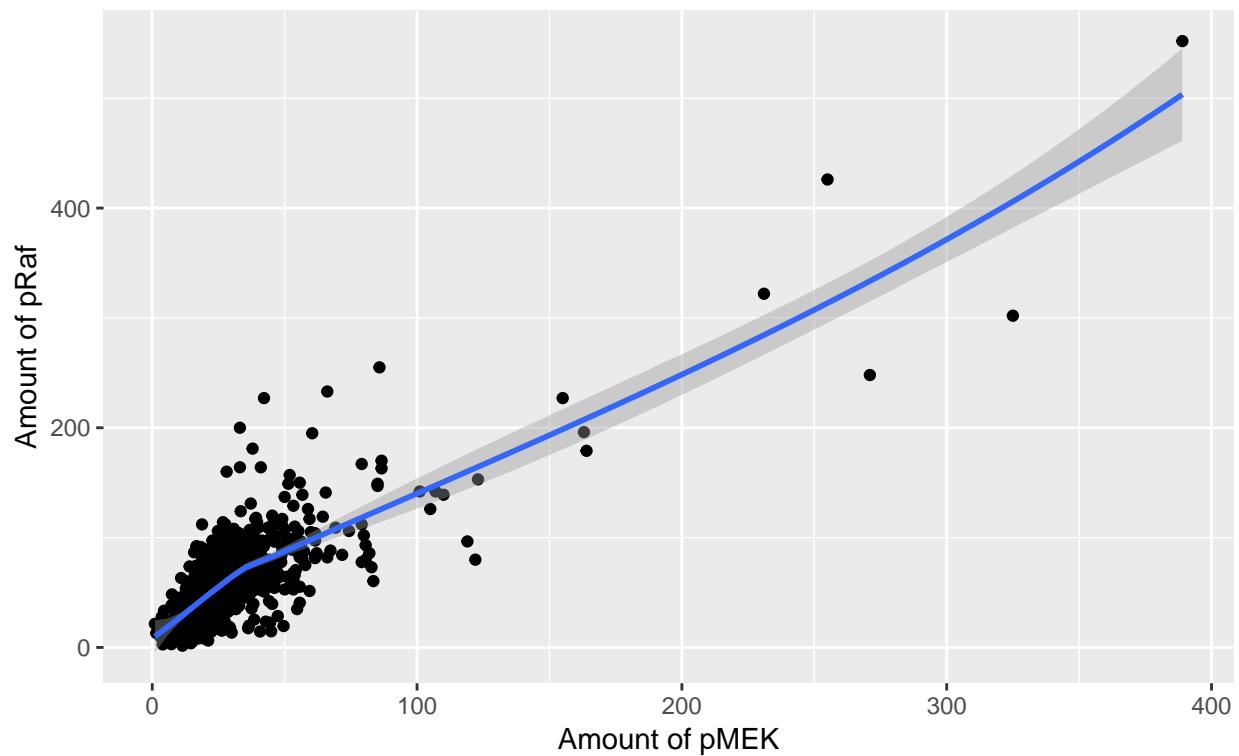
### Some examples of good and bad correlation

How do correlations changes among proteins in the same condition? I represent here the correlation between some proteins, Mek vs Raf and PKA vs PIP2 in control situations (treatment = 0).

```
pmekvspraf <- GP1 %>% dplyr::filter(treatment_num == "0")
ggplot(pmekvspraf, aes(x = pmek, y = praf)) + geom_point() +
  geom_smooth(method = "loess") + labs(subtitle = "pMEK vs pRaf",
  y = "Amount of pRaf", x = "Amount of pMEK", title = "Correlation plot")
```

## Correlation plot

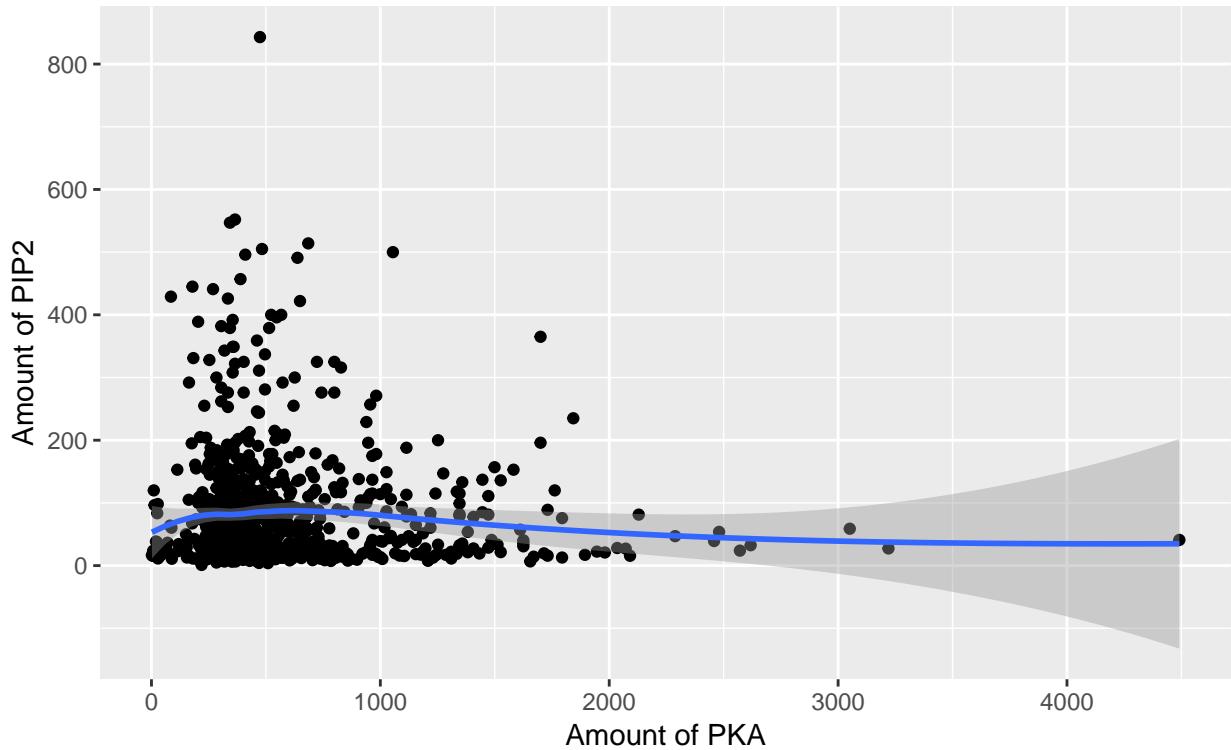
pMEK vs pRaf



```
PKAvsPIP2 <- GP1 %>% dplyr::filter(treatment_num == "0")
ggplot(PKAvsPIP2, aes(x = PKA, y = PIP2)) + geom_point() + geom_smooth(method = "loess") +
  labs(subtitle = "PKA vs PIP2", y = "Amount of PIP2", x = "Amount of PKA",
       title = "Correlation plot")
```

## Correlation plot

PKA vs PIP2



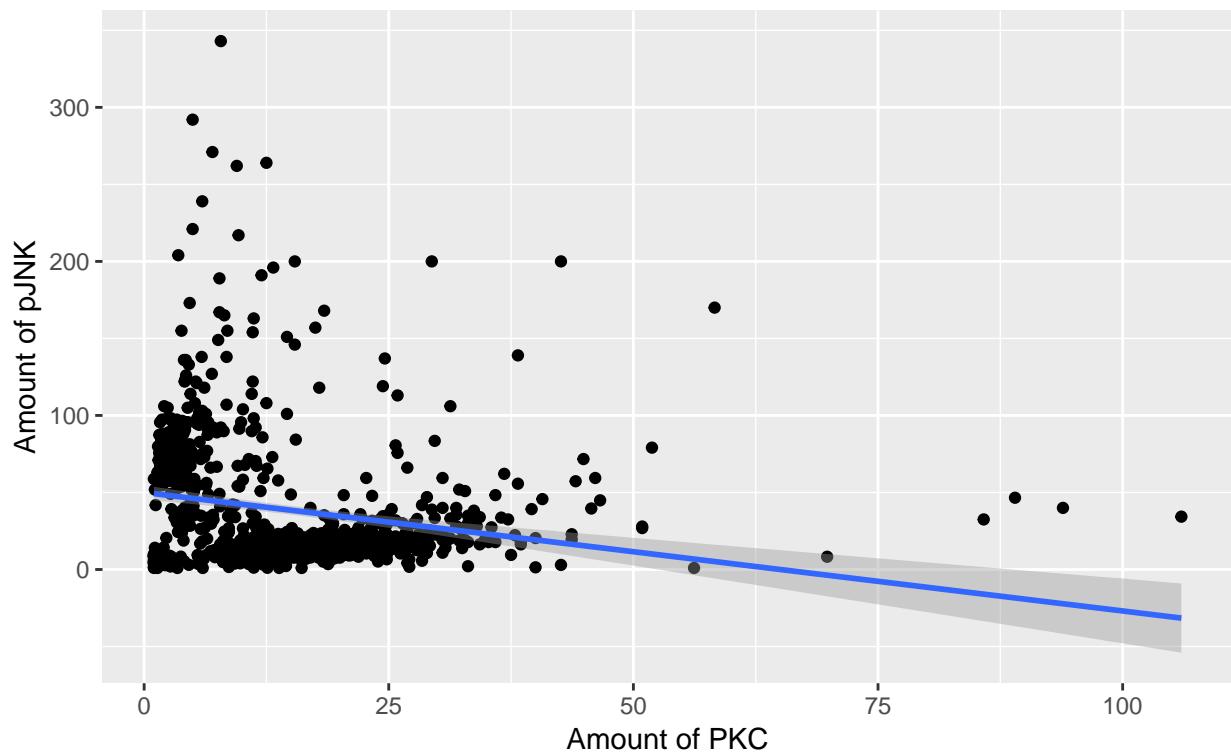
```
# tg1 <- tableGrob(a) tg2 <- tableGrob(b)
# grid.arrange(tg1,tg2, nrow=2, ncol=1)
```

We can see that pMEK and pRaf correlate quite well, whereas there is no correlation between PKA and PIP2. How do correlations changes among treatments? Here, I represent the correlation between PKC vs pJNK in control situations (treatment = 0) and when PKC is inhibited (treatment =2).

```
PKCvspJNK0 <- GP1 %>% dplyr::filter(treatment_num == "0")
ggplot(PKCvspJNK0, aes(x = PKC, y = pjnk)) + geom_point() + geom_smooth(method = "lm") +
  labs(subtitle = "PKC vs pJNK", y = "Amount of pJNK", x = "Amount of PKC",
       title = "Correlation of PKC vs pJNK, Treatment = 0")
```

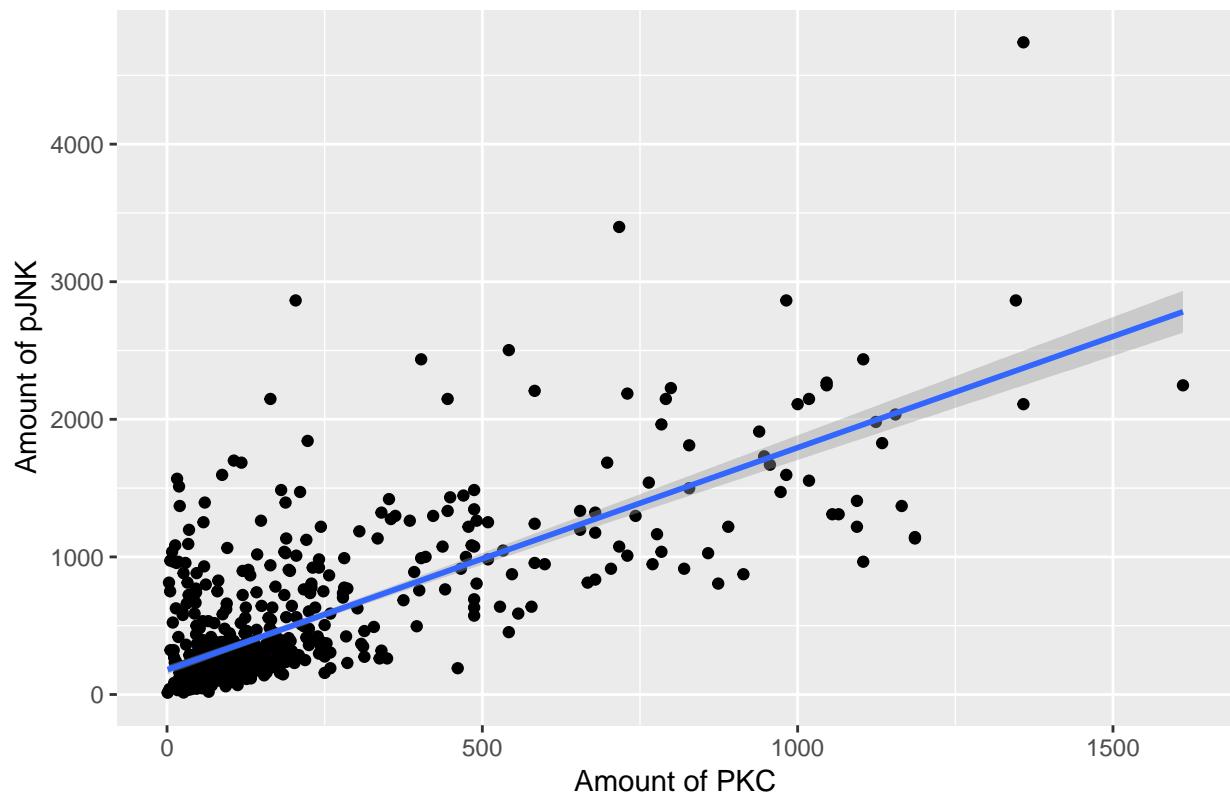
## Correlation of PKC vs pJNK, Treatment = 0

### PKC vs pJNK



```
PKCvspJNK_2 <- GP1 %>% dplyr::filter(treatment_num == "2")
ggplot(PKCvspJNK_2, aes(x = PKC, y = pJNK)) + geom_point() +
  geom_smooth(method = "lm") + labs(y = "Amount of pJNK", x = "Amount of PKC",
  title = "Correlation of PKC vs pJNK, Treatment = 2")
```

## Correlation of PKC vs pJNK, Treatment = 2



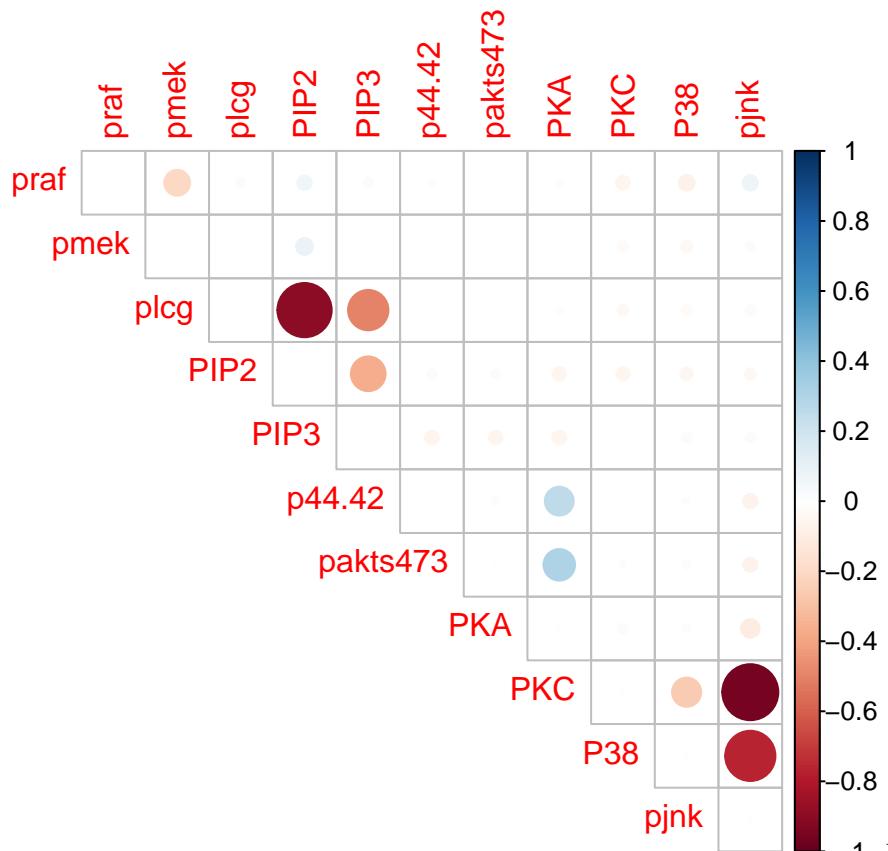
```
# grid.arrange(cond0, cond2, nrow = 1)
```

We observed that when we inhibit PKC, it starts to correlate with pJNK.

### Checking how correlations change among treatments

I think it would be interesting to study how correlations change between different treatments? A way of visualizing this could be subtracting correlation coefficient for different treatments. Here, I represent the subtraction of correlation coefficient in treatment 2 from treatment 0.

```
CM2m0 <- CM_0 - CM_2  
corrplot(CM2m0, type = "upper")
```



Here, we can see how correlation decreases for pairs of proteins like PKC and pjnk or plcg and PIP2 between treatment 0 and treatment 2.