Proteomics analysis of Abeta-expressing flies

Yizhou Yu

updated: Jan-16-2023 ¶

Data curation

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(boot)
library(ggplot2)
library(pdp)
```

```
##
## Attaching package: 'pdp'
##
## The following object is masked from 'package:purrr':
##
##     partial
```

```
library("FactoMineR")
library("factoextra")
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
```

```
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
##
##     compact
```

Load data Note: the protein quantification values have been log2 transformed

```
dt = read.csv("data/P239 Ivana Celardo TMT experiment3.csv")
dt <- subset(dt, select = -c(X))
dt$gene_name = sub("\\ OS=.*", "", dt$Description)
#delete all before GN
dt$symbol = gsub(".*GN=","",dt$Description)
#delete all before PE
dt$symbol = gsub(" PE=.*","",dt$symbol)
head(dt$symbol)
```

```
## [1] "bt"    "bt"    "sls"   "Mhc"   "Rfabg" "Mhc"
```

Descriptive statistics of the number of detected genes

```
print(paste0("Total detected: ",nrow(dt)))
```

```
## [1] "Total detected: 4822"
```

```
print(paste0("Significant: ",nrow(subset(dt, adj.P.Val <= 0.05))))
```

```
## [1] "Significant: 1625"
```

```
print(paste0("Significantly up: ",nrow(subset(dt, adj.P.Val <= 0.05 & logFC > 0))))
```

```
## [1] "Significantly up: 723"
```

```
print(paste0("Significantly down: ",nrow(subset(dt, adj.P.Val <= 0.05 & logFC < 0))))
```

```
## [1] "Significantly down: 902"
```

Add FC up vs down

```
#colnames(dt)

#no change
dt$cutoff1 = "no_change"

#Increased
dt$cutoff1[dt$logFC >= 1] <- "up"

#Decreased
dt$cutoff1[dt$logFC <= -1] <- "down"

c1_up = subset(dt, cutoff1 =="up")
nrow(c1_up)
```

```
## [1] 106
```

```
c1_down = subset(dt, cutoff1 =="down")
nrow(c1_down)
```

```
## [1] 28
```

```
write.csv(c1_up,"data_out/cutoff1_up.csv")
write.csv(c1_down,"data_out/cutoff1_down.csv")
write.csv(dt,"data_out/annotated_dt.csv")
```

Add another FC up vs down `log(1.5,2) = 0.5849625`

```
#colnames(dt)

#no change
dt$cutoff.6 = "no_change"

#Increased
dt$cutoff.6[dt$logFC >= 0.5849625] <- "up"

#Decreased
dt$cutoff.6[dt$logFC <= -0.5849625] <- "down"

c.6_up = subset(dt, cutoff.6 =="up")
nrow(c.6_up)
```

```
## [1] 349
```

```
c.6_down = subset(dt, cutoff.6 =="down")
nrow(c.6_down)
```

```
## [1] 189
```

Add another FC up vs down Add cutoff of 0

```
#colnames(dt)

#no change
dt$cutoff0 = "no_change"

#Increased
dt$cutoff0[dt$logFC >0] <- "up"

#Decreased
dt$cutoff0[dt$logFC <0] <- "down"

nrow(subset(dt, cutoff0 =="up"))
```

```
## [1] 2272
```

```
nrow(subset(dt, cutoff0 =="down"))
```

```
## [1] 2546
```

PCA analysis

Prepare data for PCA

Only select significant variables

```
pca_dt = subset(dt,adj.P.Val <= 0.05, select = c(symbol, daGAL4_plus1,daGAL4_plus2,daGAL4_plus3,daGAL4_

pca_dt_dedup = unique(pca_dt)
pca_dt_dedup_t = as.data.frame(t(pca_dt_dedup))
#here, the variables are made into characters...

colnames(pca_dt_dedup_t) <- pca_dt_dedup_t[1,]
pca_dt_dedup_t = pca_dt_dedup_t[-1,]

#de duplicate
pca_dt_dedup_t <- pca_dt_dedup_t[, !duplicated(colnames(pca_dt_dedup_t))]

#fix the structure here
pca_dt_dedup_t <- mutate_all(pca_dt_dedup_t, function(x) as.numeric(as.character(x)))

labels = row.names(pca_dt_dedup_t)
pca_dt_dedup_t$genotype = row.names(pca_dt_dedup_t)
#delete last character
pca_dt_dedup_t$genotype = substr(pca_dt_dedup_t$genotype,1,nchar(pca_dt_dedup_t$genotype)-1)

ncol(pca_dt_dedup_t)
```

```
## [1] 1607
```

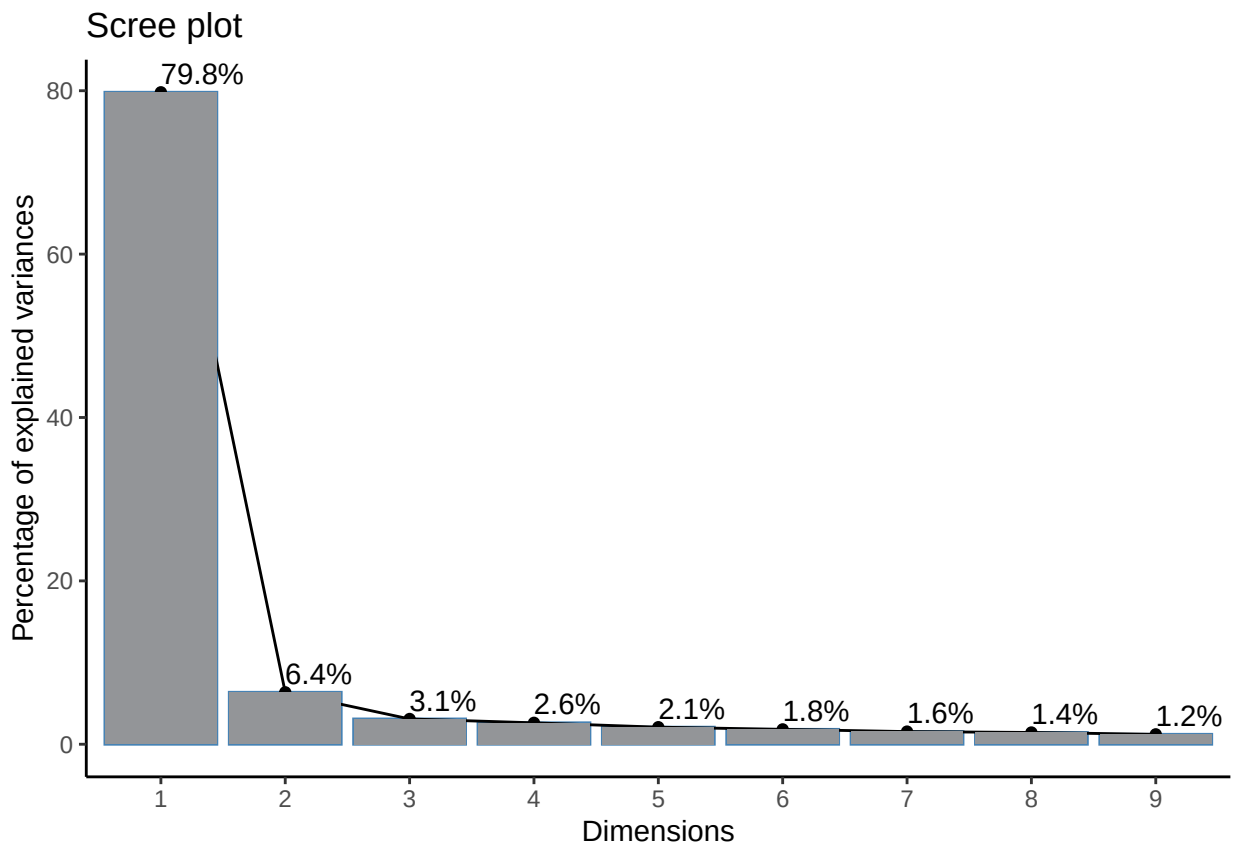
```
#colnames(pca_dt_dedup_t) <- make.names(colnames(pca_dt_dedup_t))
```

1607 columns, the last one are the groups

Run PCA

Visualise

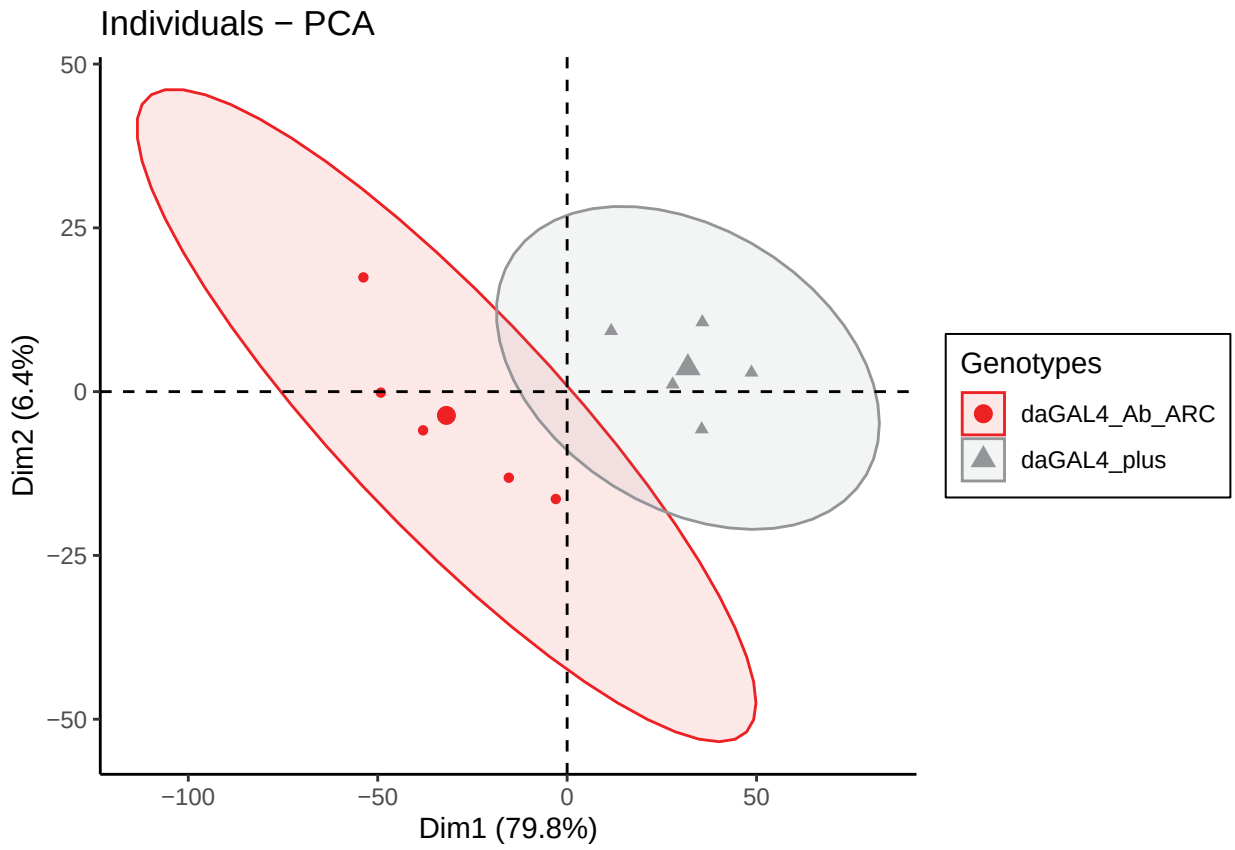
```
pca.10 <- PCA(pca_dt_dedup_t[,1:1606], scale.unit = TRUE, ncp = 10, graph = FALSE)
fviz_eig(pca.10, addlabels = TRUE) + theme_classic()+ geom_bar(stat = "identity",fill="#939598")+
  theme(
    panel.background = element_rect(fill = "transparent"), # bg of the panel
    plot.background = element_rect(fill = "transparent", color = NA), # bg of the plot
    panel.grid.major = element_blank(), # get rid of major grid
    panel.grid.minor = element_blank(), # get rid of minor grid
    legend.background = element_rect(fill = "transparent"), # get rid of legend bg
    legend.box.background = element_rect(fill = "transparent") # get rid of legend panel bg
  )
```




```
ggsave("fig/PCA_varianceExplained_10PC.pdf", width = 6, height = 4,  bg = "transparent")
```

PC1 and 2 together explain 59% of the variance; adding PC3 increases this to 67.5%

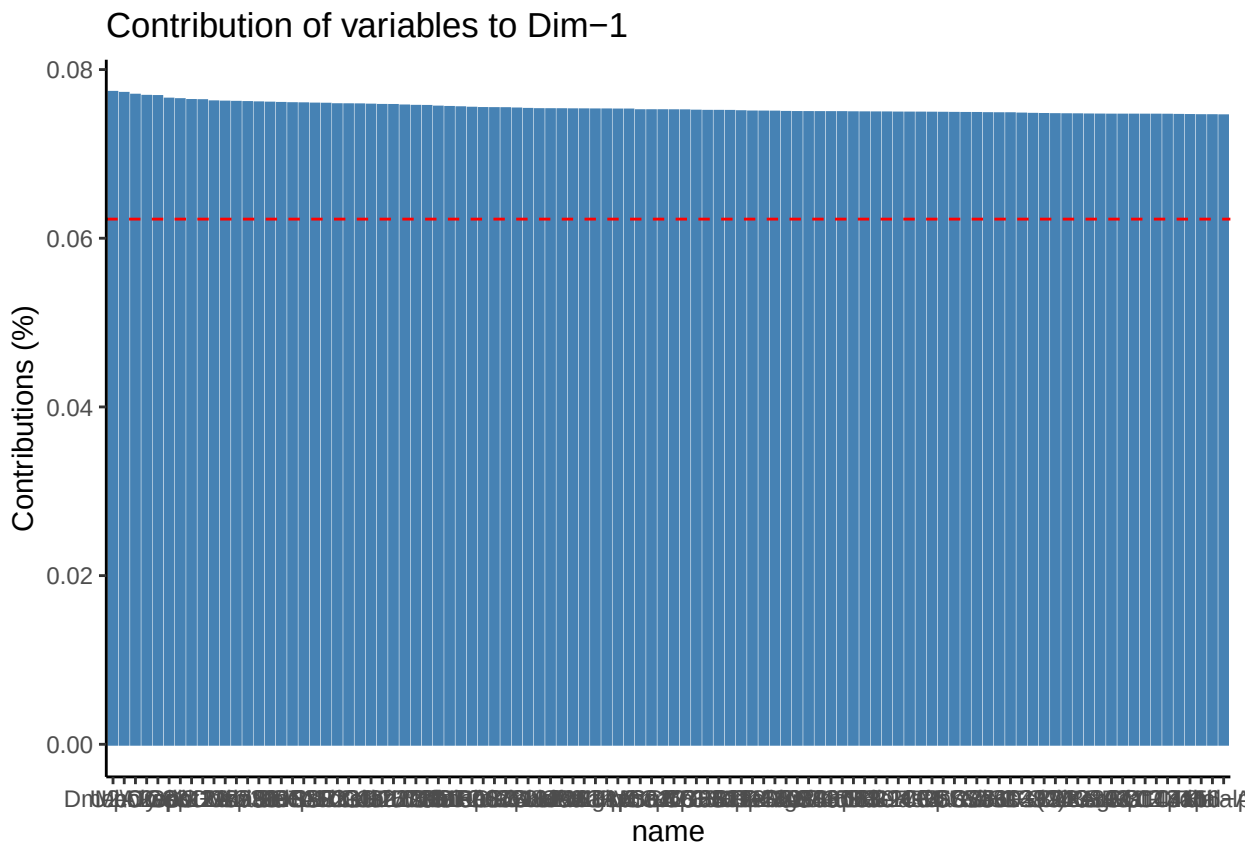
```
fviz_pca_ind(pca.10,
             geom.ind = "point", # show points only (nbut not "text")
             col.ind = pca_dt_dedup_t$genotype, # color by groups
             palette = c("#ED2024","#939598"),
             addEllipses = TRUE, # Concentration ellipses
             legend.title = "Genotypes"
             ) + theme_classic()+
  theme(
    panel.background = element_rect(fill = "transparent"), # bg of the panel
    plot.background = element_rect(fill = "transparent", color = NA), # bg of the plot
    panel.grid.major = element_blank(), # get rid of major grid
    panel.grid.minor = element_blank(), # get rid of minor grid
    legend.background = element_rect(fill = "transparent"), # get rid of legend bg
    legend.box.background = element_rect(fill = "transparent") # get rid of legend panel bg
  )
```




```
ggsave("fig/PCA_general_graph.pdf", width = 6, height = 4,  bg = "transparent")
```

The red dashed line on the graph above indicates the expected average contribution.

```
pca_vars = get_pca_var(pca.10)
fviz_contrib(pca.10, choice = "var", axes = 1, top = 100) + theme_classic()
```



Get genes that are the highest contributors

```
pca_contrib = as.data.frame(pca_vars$contrib)
pca_contrib$names = row.names(pca_contrib)
pca_contrib_sorted <- arrange(pca_contrib,desc(Dim.1))
head(pca_contrib_sorted,10)
```

```
##         Dim.1        Dim.2        Dim.3        Dim.4        Dim.5        Dim.6
## 1  0.07729930 0.0001529367 1.095999e-03 1.478828e-02 1.964051e-04 1.736510e-05
## 2  0.07718393 0.0055918012 1.915789e-04 5.776716e-04 9.629066e-03 1.199209e-07
## 3  0.07696605 0.0003320131 6.336122e-04 1.123539e-03 1.239769e-02 1.492796e-02
## 4  0.07683366 0.0128746346 2.623166e-04 1.691855e-05 2.149203e-03 2.094973e-03
## 5  0.07680808 0.0004171575 3.173216e-03 1.253574e-02 7.872362e-05 7.478522e-03
## 6  0.07650160 0.0132261176 7.102179e-03 1.192094e-03 1.916691e-03 4.762470e-04
## 7  0.07643669 0.0004107493 2.294763e-03 3.386982e-05 1.105877e-02 2.310667e-06
## 8  0.07633457 0.0008850327 8.631833e-07 1.840898e-02 6.276574e-03 3.542810e-02
## 9  0.07631553 0.0093362823 4.749736e-03 4.996830e-03 1.147708e-03 3.814634e-03
## 10 0.07617795 0.0064891883 1.520360e-03 1.737154e-03 4.163377e-02 1.784647e-03
##           Dim.7        Dim.8        Dim.9         names
## 1  7.598675e-03 8.702707e-04 7.666136e-04           U2A
## 2  1.797399e-03 9.564815e-05 5.015507e-03          Mec2
## 3  1.014606e-02 1.259357e-03 5.370564e-03          poly
## 4  4.537094e-05 2.280229e-03 3.023186e-04 Dmel\\CG33129
## 5  4.595537e-04 7.770500e-07 3.042807e-02           Pep
## 6  1.255770e-03 1.182396e-04 4.165751e-03           wdp
## 7  1.742533e-04 1.942300e-02 5.379704e-02         Gprk2
```

```
## 8  8.706426e-04 1.048910e-03 9.683291e-05        obst-A
## 9  1.613880e-02 7.704406e-03 2.844955e-03  Dmel\\CG7781
## 10 7.269491e-05 1.573031e-03 3.244491e-03       CG11899
```

Get genes that are higher than the average contributor

```
mean(pca_contrib$Dim.1)
```

```
## [1] 0.0622665
```

Determined using mean(pca_contrib$Dim.1) -> 0.0622665

```
pca_contrib_high = subset(pca_contrib, Dim.1 >= 0.0622665)
pca_contrib_high_names = subset(pca_contrib_high, select = c(names))
dt_simplified = subset(dt, select = c(symbol,logFC, cutoff.6))
pca_contrib_high_names = merge(pca_contrib_high_names, dt_simplified,
                               by.x = "names", by.y = "symbol")
nrow(pca_contrib_high_names)
```

```
## [1] 1079
```

```
pca_contrib_high_names_list = unique(subset(pca_contrib_high_names, select = c(names, cutoff.6)))
pca_contrib_high_names_list_higher_cutoff = subset(pca_contrib_high_names_list, select = names, cutoff.

write.csv(pca_contrib_high_names, "data_out/pca_high_contribution.csv", row.names = F)
write.csv(pca_contrib_high_names_list, "data_out/pca_high_contribution_geneNamesOnly.csv", row.names = 
write.csv(pca_contrib_high_names_list_higher_cutoff, "data_out/pca_contrib_high_names_list_higher_0.6cu
```

String plot of the STRING results

```
string_dt = read.csv("data_out/PCA_STRING/enrichment.Keyword_PCR_STRING_up_FC1.5.tsv", sep = "\t")
go_target = c("Ubiquinone","NAD","Respiratory chain","One-carbon metabolism")
string_dt_splot = string_dt[string_dt$term.description %in% go_target, ]
string_dt_splot$cat = ifelse(string_dt_splot$term.description == "One-carbon metabolism", yes = "One-ca
string_dt_splot_subset = subset(string_dt_splot, select = c(term.description, cat,
```

remake the df

```
# U10

string_dt_splot_subset_u10_proteins = string_dt_splot_subset[string_dt_splot_subset$term.description=="
string_dt_splot_subset_u10_proteins = as.list(strsplit(string_dt_splot_subset_u10_proteins, ","))


string_dt_splot_subset_u10 = data.frame(from = rep("Ubiquinone",length(string_dt_splot_subset_u10_prote
                                        to = string_dt_splot_subset_u10_proteins)
colnames(string_dt_splot_subset_u10)<-c("from","to")
```

```
# NAD
string_dt_splot_subset_proteins_nad =  as.list(strsplit(string_dt_splot_subset[string_dt_splot_subset$te


string_dt_splot_subset_nad = data.frame(from = rep("NAD",length(string_dt_splot_subset_proteins_nad)),
                                        to = string_dt_splot_subset_proteins_nad)
colnames(string_dt_splot_subset_nad)<-c("from","to")

# resp
string_dt_splot_subset_proteins_resp =  as.list(strsplit(string_dt_splot_subset[string_dt_splot_subset$t


string_dt_splot_subset_resp = data.frame(from = rep("Respiratory chain",length(string_dt_splot_subset_pr
                                        to = string_dt_splot_subset_proteins_resp)
colnames(string_dt_splot_subset_resp)<-c("from","to")

# 1c
string_dt_splot_subset_proteins_1c =  as.list(strsplit(string_dt_splot_subset[string_dt_splot_subset$te


string_dt_splot_subset_1c = data.frame(from = rep("One-carbon metabolism",length(string_dt_splot_subset_
                                        to = string_dt_splot_subset_proteins_1c)
colnames(string_dt_splot_subset_1c)<-c("from","to")
```

```
string_dt_splot_bind = rbind(string_dt_splot_subset_u10,
                             string_dt_splot_subset_nad)
string_dt_splot_bind = rbind(string_dt_splot_bind,
                             string_dt_splot_subset_resp)
string_dt_splot_bind = rbind(string_dt_splot_bind,
                             string_dt_splot_subset_1c)
```

Thickness based on strength

```
string_dt_splot_bind = merge(string_dt_splot_bind,
                             subset(string_dt, select =
                                      c(term.description,strength)),
                             by.x = "from",
                             by.y = "term.description")
string_dt_splot_bind$colour = ifelse(string_dt_splot_bind$from == "One-carbon metabolism", yes = "One-ca
```

Note: since the plot is turned -90 degrees, the labels' order need to be reversed.

```
library(circlize)
```

```
## ========================================
## circlize version 0.4.15
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
```

```
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================
```

```
gene_list = sort(labels(summary(as.factor(string_dt_splot_bind$to))))

pdf("fig/proteomics_STRING_PCA_FC1.5_chordDiag.pdf")
circos.par(start.degree = -90)
chordDiagram(string_dt_splot_bind, annotationTrack = c("name", "grid"), scale = TRUE, big.gap = 20, ord
circos.clear()
```