#stringApp\_01\_GO\_KEGG\_volcanoplot.R

#stringApp1.5.1\_Cytoscape3.8.0→stringApp1.6.0\_Cytoscape3.8.1\_Java11.0.6

#stringApp出力データから

#1\_P,C,IntにおけるGO\_KEGG\_top50ラベル

#2\_P,C,Int全てについてGO\_KEGG\_top50抽出

#3\_GO\_KEGG termのplot描写

#4\_P,C,Intにおけるvolcano plot描写(top50赤色)

#5\_GO\_KEGG\_top50に含まれるDEP list作成 →別script

#############################################################

rm(list = ls(all.names = TRUE))

#setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/AMY/stringApp") #作業ディレクトリ設定

#setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/HIP/stringApp") #作業ディレクトリ設定

#setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/NAc/stringApp") #作業ディレクトリ設定

#setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/PFC/stringApp") #作業ディレクトリ設定

#setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/STR/stringApp") #作業ディレクトリ設定

setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/PFC/stringApp") #作業ディレクトリ設定

getwd()#作業ディレクトリ確認

dir() #作業ディレクトリ内のファイル表示

options(digits=2) #change digit2桁表示指定

##############################################################

library(EnhancedVolcano)

library(magrittr)

library(tidyverse) #ライブラリtidyverse(ggplot2,dplyr),gcookbook読み込み

library(dplyr)

library(scales) #muted()関数使用のため

library(gridExtra) #svg出力のため

library(rJava)

library(readxl) #エクセル読み込み

library(openxlsx) #JAVA不使用で大きなデータも読み込める

library(cowplot)

#############################################################

#CSV入力

dat1 <- read.csv("String\_twANOVA\_Pq005\_filt.csv", stringsAsFactors = T, fileEncoding = "UTF-8-BOM") #CSVの読み込み

dat2 <- read.csv("String\_twANOVA\_Cq005\_filt.csv", stringsAsFactors = T, fileEncoding = "UTF-8-BOM") #CSVの読み込み

dat3 <- read.csv("String\_twANOVA\_PxCq005\_filt.csv", stringsAsFactors = T, fileEncoding = "UTF-8-BOM") #CSVの読み込み

#############################################################

#行番号挿入,重複削除,列名置換

dat1 <- dat1 %>% mutate(No = row\_number()) %>% #行番号挿入

mutate(group = "PCP") %>% #シート番号挿入

distinct(description,.keep\_all=TRUE) %>% #重複削除

mutate(log10p = -log10(`FDR.value`)) #p値→-log10p値変換

names(dat1)[which(names(dat1)=="X..genes" ) ] <- "Overlap" #列名置換

dat2 <- dat2 %>% mutate(No = row\_number()) %>% #行番号挿入

mutate(group = "CLZ") %>% #シート番号挿入

distinct(description,.keep\_all=TRUE) %>% #重複削除

mutate(log10p = -log10(`FDR.value`)) #p値→-log10p値変換

names(dat2)[which(names(dat2)=="X..genes" ) ] <- "Overlap" #列名置換

dat3 <- dat3 %>% mutate(No = row\_number()) %>% #行番号挿入

mutate(group = "Interaction") %>% #シート番号挿入

distinct(description,.keep\_all=TRUE) %>% #重複削除

mutate(log10p = -log10(`FDR.value`)) #p値→-log10p値変換

names(dat3)[which(names(dat3)=="X..genes" ) ] <- "Overlap" #列名置換

#############################################################

#列の作成と結合

dat1 <- dat1 %>% as.data.frame() %>%

mutate(DBsourse = case\_when( #DBsourse列作成

`category` == "GO Process" ~ "BP", #categoryのGO ProcessをBP

`category` == "GO Component" ~ "CC", #categoryのGO ComponentをCC

`category` == "GO Function" ~ "MF", #categoryのGO FunctionをMF

`category` == "KEGG Pathways" ~ "KEGG", #categoryのKEGG PathwaysをKEGG

TRUE ~ as.character(`category`)

)) %>%

mutate(term = paste( #term列作成

!!!rlang::syms(c("description", "DBsourse")), #シンボル化：map(as.symbol)と同じ

sep="\_")) #description\_DBsourseとして結合

dat2 <- dat2 %>% as.data.frame() %>%

mutate(DBsourse = case\_when( #DBsourse列作成

`category` == "GO Process" ~ "BP", #categoryのGO ProcessをBP

`category` == "GO Component" ~ "CC", #categoryのGO ComponentをCC

`category` == "GO Function" ~ "MF", #categoryのGO FunctionをMF

`category` == "KEGG Pathways" ~ "KEGG", #categoryのKEGG PathwaysをKEGG

TRUE ~ as.character(`category`)

)) %>%

mutate(term = paste( #term列作成

!!!rlang::syms(c("description", "DBsourse")), #シンボル化：map(as.symbol)と同じ

sep="\_")) #description\_DBsourseとして結合

dat3 <- dat3 %>% as.data.frame() %>%

mutate(DBsourse = case\_when( #DBsourse列作成

`category` == "GO Process" ~ "BP", #categoryのGO ProcessをBP

`category` == "GO Component" ~ "CC", #categoryのGO ComponentをCC

`category` == "GO Function" ~ "MF", #categoryのGO FunctionをMF

`category` == "KEGG Pathways" ~ "KEGG", #categoryのKEGG PathwaysをKEGG

TRUE ~ as.character(`category`)

)) %>%

mutate(term = paste( #term列作成

!!!rlang::syms(c("description", "DBsourse")), #シンボル化：map(as.symbol)と同じ

sep="\_")) #description\_DBsourseとして結合

#############################################################

#GO抽出

dat1\_GO <- dat1 %>% #dat1データの

dplyr::filter(category=="GO Component" | #category列がGO Componentまたは

category=="GO Process" | #category列がGO Processまたは

category=="GO Function") #category列がGO Function

dat1\_CC <- dat1 %>% dplyr::filter(category=="GO Component")

dat1\_BP <- dat1 %>% dplyr::filter(category=="GO Process")

dat1\_MF <- dat1 %>% dplyr::filter(category=="GO Function")

dat2\_GO <- dat2 %>% #dat2データの

dplyr::filter(category=="GO Component" | #category列がGO Componentまたは

category=="GO Process" | #category列がGO Processまたは

category=="GO Function") #category列がGO Function

dat2\_CC <- dat2 %>% dplyr::filter(category=="GO Component")

dat2\_BP <- dat2 %>% dplyr::filter(category=="GO Process")

dat2\_MF <- dat2 %>% dplyr::filter(category=="GO Function")

dat3\_GO <- dat3 %>% #dat3データの

dplyr::filter(category=="GO Component" | #category列がGO Componentまたは

category=="GO Process" | #category列がGO Processまたは

category=="GO Function") #category列がGO Function

dat3\_CC <- dat3 %>% dplyr::filter(category=="GO Component")

dat3\_BP <- dat3 %>% dplyr::filter(category=="GO Process")

dat3\_MF <- dat3 %>% dplyr::filter(category=="GO Function")

#KEGG抽出

dat4\_KEGG <- dat1 %>% dplyr::filter(category=="KEGG Pathways")#dat1データのcategory列がKEGG Pathways

dat5\_KEGG <- dat2 %>% dplyr::filter(category=="KEGG Pathways")#dat2データのcategory列がKEGG Pathways

dat6\_KEGG <- dat3 %>% dplyr::filter(category=="KEGG Pathways")#dat3データのcategory列がKEGG Pathways

#############################################################

#top10データの抽出

dat1\_GO10 <- dat1\_GO %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p) #log10pで降順ソートし上位10のデータを保存

dat1\_BP10 <- dat1\_BP %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat1\_CC10 <- dat1\_CC %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat1\_MF10 <- dat1\_MF %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat2\_GO10 <- dat2\_GO %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat2\_BP10 <- dat2\_BP %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat2\_CC10 <- dat2\_CC %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat2\_MF10 <- dat2\_MF %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat3\_GO10 <- dat3\_GO %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat3\_BP10 <- dat3\_BP %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat3\_CC10 <- dat3\_CC %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat3\_MF10 <- dat3\_MF %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat4\_KEGG10 <- dat4\_KEGG %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat5\_KEGG10 <- dat5\_KEGG %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

dat6\_KEGG10 <- dat6\_KEGG %>% arrange(desc(log10p)) %>% top\_n(n = 10, wt = log10p)

#top20データの抽出

dat1\_GO20 <- dat1\_GO %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p) #log10pで降順ソートし上位20のデータを保存

dat1\_BP20 <- dat1\_BP %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat1\_CC20 <- dat1\_CC %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat1\_MF20 <- dat1\_MF %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat2\_GO20 <- dat2\_GO %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat2\_BP20 <- dat2\_BP %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat2\_CC20 <- dat2\_CC %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat2\_MF20 <- dat2\_MF %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat3\_GO20 <- dat3\_GO %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat3\_BP20 <- dat3\_BP %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat3\_CC20 <- dat3\_CC %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat3\_MF20 <- dat3\_MF %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat4\_KEGG20 <- dat4\_KEGG %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat5\_KEGG20 <- dat5\_KEGG %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

dat6\_KEGG20 <- dat6\_KEGG %>% arrange(desc(log10p)) %>% top\_n(n = 20, wt = log10p)

#############################################################

#top10データをラベル

dat1\_GO10 <- dat1\_GO10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat1\_BP10 <- dat1\_BP10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat1\_CC10 <- dat1\_CC10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat1\_MF10 <- dat1\_MF10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat2\_GO10 <- dat2\_GO10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat2\_BP10 <- dat2\_BP10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat2\_CC10 <- dat2\_CC10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat2\_MF10 <- dat2\_MF10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat3\_GO10 <- dat3\_GO10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat3\_BP10 <- dat3\_BP10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat3\_CC10 <- dat3\_CC10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat3\_MF10 <- dat3\_MF10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat4\_KEGG10 <- dat4\_KEGG10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat5\_KEGG10 <- dat5\_KEGG10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat6\_KEGG10 <- dat6\_KEGG10 %>% mutate(top10 = if\_else(log10p > 0, true = TRUE, false = FALSE))

#top20データをラベル

dat1\_GO20 <- dat1\_GO20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat1\_BP20 <- dat1\_BP20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat1\_CC20 <- dat1\_CC20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat1\_MF20 <- dat1\_MF20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat2\_GO20 <- dat2\_GO20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat2\_BP20 <- dat2\_BP20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat2\_CC20 <- dat2\_CC20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat2\_MF20 <- dat2\_MF20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat3\_GO20 <- dat3\_GO20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat3\_BP20 <- dat3\_BP20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat3\_CC20 <- dat3\_CC20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat3\_MF20 <- dat3\_MF20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat4\_KEGG20 <- dat4\_KEGG20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat5\_KEGG20 <- dat5\_KEGG20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat6\_KEGG20 <- dat6\_KEGG20 %>% mutate(top20 = if\_else(log10p > 0, true = TRUE, false = FALSE))

#############################################################

#dat1,2,3(GO)とdat4,5,6(KEGG)を1つのデータフレーム(GO)に結合

GO10 <- rbind(dat1\_GO10, dat2\_GO10, dat3\_GO10) #data1,2,3結合

KEGG10 <- rbind(dat4\_KEGG10, dat5\_KEGG10, dat6\_KEGG10) #data4,5,6結合

#Top20

GO20 <- rbind(dat1\_GO20, dat2\_GO20, dat3\_GO20) #data1,2,3結合

KEGG20 <- rbind(dat4\_KEGG20, dat5\_KEGG20, dat6\_KEGG20) #data4,5,6結合

#############################################################

#Top10

#GO <- GO10

#KEGG <- KEGG10

#Top20

GO <- GO20

KEGG <- KEGG20

#############################################################

#重複削除,top10データの抽出

GO\_t <- GO %>%

arrange(desc(log10p)) %>% #log10pで降順ソート

distinct(description,.keep\_all=TRUE) %>% #重複削除

top\_n(n = 10, wt = log10p) #log10p上位

KEGG\_t <- KEGG %>%

arrange(desc(log10p)) %>% #log10pで降順ソート

distinct(description,.keep\_all=TRUE) %>% #重複削除

top\_n(n = 10, wt = log10p) #log10p上位

#############################################################

#重複削除,top20データの抽出

#GO\_t <- GO %>%

# arrange(desc(log10p)) %>% #log10pで降順ソート

# distinct(description,.keep\_all=TRUE) %>% #重複削除

# top\_n(n = 20, wt = log10p) #log10p上位

#KEGG\_t <- KEGG %>%

# arrange(desc(log10p)) %>% #log10pで降順ソート

# distinct(description,.keep\_all=TRUE) %>% #重複削除

# top\_n(n = 20, wt = log10p) #log10p上位

#############################################################

#重複削除,top50データの抽出

#GO\_t <- GO %>%

# arrange(desc(log10p)) %>% #log10pで降順ソート

# distinct(description,.keep\_all=TRUE) %>% #重複削除

# top\_n(n = 50, wt = log10p) #log10p上位

#KEGG\_t <- KEGG %>%

# arrange(desc(log10p)) %>% #log10pで降順ソート

# distinct(description,.keep\_all=TRUE) %>% #重複削除

# top\_n(n = 50, wt = log10p) #log10p上位

#############################################################

#topの絞り込み

t(colnames(GO\_t))

GO\_term <- data.frame(GO\_t[,c(3,5,12)])

t(colnames(KEGG\_t))

KEGG\_term <- data.frame(KEGG\_t[,c(3,5,12)])

#Annotation作成

t(colnames(dat1\_GO))

#代入

#dat1\_t\_GO <- dat1\_GO10

#dat2\_t\_GO <- dat2\_GO10

#dat3\_t\_GO <- dat3\_GO10

#dat4\_t\_KEGG <- dat4\_KEGG10

#dat5\_t\_KEGG <- dat5\_KEGG10

#dat6\_t\_KEGG <- dat6\_KEGG10

#データ処理

dat1\_GO\_rm <- dat1\_GO[,c(-3,-5)]

dat2\_GO\_rm <- dat2\_GO[,c(-3,-5)]

dat3\_GO\_rm <- dat3\_GO[,c(-3,-5)]

dat4\_KEGG\_rm <- dat4\_KEGG[,c(-3,-5)]

dat5\_KEGG\_rm <- dat5\_KEGG[,c(-3,-5)]

dat6\_KEGG\_rm <- dat6\_KEGG[,c(-3,-5)]

#データにAnnotationを結合

#欠損値

dat1\_t\_GO <- left\_join(GO\_term, dat1\_GO\_rm, by = "term.name") #term.name識別子としてmerge

dat2\_t\_GO <- left\_join(GO\_term, dat2\_GO\_rm, by = "term.name") #term.name識別子としてmerge

dat3\_t\_GO <- left\_join(GO\_term, dat3\_GO\_rm, by = "term.name") #term.name識別子としてmerge

dat4\_t\_KEGG <- left\_join(KEGG\_term, dat4\_KEGG\_rm, by = "term.name") #term.name識別子としてmerge

dat5\_t\_KEGG <- left\_join(KEGG\_term, dat5\_KEGG\_rm, by = "term.name") #term.name識別子としてmerge

dat6\_t\_KEGG <- left\_join(KEGG\_term, dat6\_KEGG\_rm, by = "term.name") #term.name識別子としてmerge

#############################################################

#グループ挿入

dat1\_t\_GO$group <- "PCP"

dat2\_t\_GO$group <- "CLZ"

dat3\_t\_GO$group <- "Interaction"

dat4\_t\_KEGG$group <- "PCP"

dat5\_t\_KEGG$group <- "CLZ"

dat6\_t\_KEGG$group <- "Interaction"

#############################################################

#dat1,2,3(GO)とdat4,5,6(KEGG)を1つのデータフレーム(GO)に再結合

GO\_t <- rbind(dat1\_t\_GO, dat2\_t\_GO, dat3\_t\_GO) #dat1,2,3結合

KEGG\_t <- rbind(dat4\_t\_KEGG, dat5\_t\_KEGG, dat6\_t\_KEGG) #dat4,5,6結合

#############################################################

#NA(欠損値)の置換

#ifelse(is.na(GO\_t$log10p), 0, GO\_t$log10p)

#replace(GO\_t$log10p, which(is.na(GO\_t$log10p)), 0)

GO\_t$log10p[is.na(GO\_t$log10p)] <- 0

#ifelse(is.na(KEGG\_t$log10p), 0, KEGG\_t$log10p)

#replace(KEGG\_t$log10p, which(is.na(KEGG\_t$log10p)), 0)

KEGG\_t$log10p[is.na(KEGG\_t$log10p)] <- 0

#############################################################

#NA(欠損値)の置換

#ifelse(is.na(GO\_t$`Overlap`), 0, GO\_t$`Overlap`)

#replace(GO\_t$`Overlap`, which(is.na(GO\_t$`Overlap`)), 0)

GO\_t$`Overlap`[is.na(GO\_t$`Overlap`)] <- 0

#ifelse(is.na(KEGG\_t$`Overlap`), 0, KEGG\_t$`Overlap`)

#replace(KEGG\_t$`Overlap`, which(is.na(KEGG\_t$`Overlap`)), 0)

KEGG\_t$`Overlap`[is.na(KEGG\_t$`Overlap`)] <- 0

#############################################################

#descriptionの順序の入れ替えのために

#データをファクタに変換する

#library(dplyr)

dat1\_GO10 <- dat1\_GO10 %>% mutate(description = as.factor(description)) #descriptionをファクタに変換

dat2\_GO10 <- dat2\_GO10 %>% mutate(description = as.factor(description)) #descriptionをファクタに変換

dat3\_GO10 <- dat3\_GO10 %>% mutate(description = as.factor(description)) #descriptionをファクタに変換

dat4\_KEGG10 <- dat4\_KEGG10 %>% mutate(description = as.factor(description)) #descriptionをファクタに変換

dat5\_KEGG10 <- dat5\_KEGG10 %>% mutate(description = as.factor(description)) #descriptionをファクタに変換

dat6\_KEGG10 <- dat6\_KEGG10 %>% mutate(description = as.factor(description)) #descriptionをファクタに変換

GO\_t <- GO\_t %>% mutate(description = as.factor(description)) #descriptionをファクタに変換

KEGG\_t <- KEGG\_t %>% mutate(description = as.factor(description)) #descriptionをファクタに変換

#############################################################

#列の作成と結合

GO\_t <- GO\_t %>% as.data.frame() %>%

mutate(DBsourse = case\_when( #DBsourse列作成

`category` == "GO Process" ~ "BP", #categoryのGO ProcessをBP

`category` == "GO Component" ~ "CC", #categoryのGO ComponentをCC

`category` == "GO Function" ~ "MF", #categoryのGO FunctionをMF

`category` == "KEGG Pathways" ~ "KEGG", #categoryのKEGG PathwaysをKEGG

TRUE ~ as.character(`category`)

)) %>%

mutate(term = paste( #term列作成

!!!rlang::syms(c("description", "DBsourse")), #シンボル化：map(as.symbol)と同じ

sep="\_")) #description\_DBsourseとして結合

KEGG\_t <- KEGG\_t %>% as.data.frame() %>%

mutate(DBsourse = case\_when( #DBsourse列作成

`category` == "GO Process" ~ "BP", #categoryのGO ProcessをBP

`category` == "GO Component" ~ "CC", #categoryのGO ComponentをCC

`category` == "GO Function" ~ "MF", #categoryのGO FunctionをMF

`category` == "KEGG Pathways" ~ "KEGG", #categoryのKEGG PathwaysをKEGG

TRUE ~ as.character(`category`)

)) %>%

mutate(term = paste( #term列作成

!!!rlang::syms(c("description", "DBsourse")), #シンボル化：map(as.symbol)と同じ

sep="\_")) #description\_DBsourseとして結合

#############################################################

#ggplot2,GO

plotGO <- ggplot()+

theme\_gray()+ #フォント指定base\_family = "Arial", "HiraKakuPro-W3"など

geom\_point(data = GO\_t, aes(x = log10p,

y = reorder(x = term, X = log10p),

#position = "jitter", #geom\_jitter

color = group,

alpha = 0.9,

size = `Overlap`))

plotGO1 <- ggplot() + theme\_gray() + geom\_point(data = dat1\_GO10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotGO2 <- ggplot() + theme\_gray() + geom\_point(data = dat2\_GO10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotGO3 <- ggplot() + theme\_gray() + geom\_point(data = dat3\_GO10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotBP1 <- ggplot() + theme\_gray() + geom\_point(data = dat1\_BP10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotBP2 <- ggplot() + theme\_gray() + geom\_point(data = dat2\_BP10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotBP3 <- ggplot() + theme\_gray() + geom\_point(data = dat3\_BP10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotCC1 <- ggplot() + theme\_gray() + geom\_point(data = dat1\_CC10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotCC2 <- ggplot() + theme\_gray() + geom\_point(data = dat2\_CC10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotCC3 <- ggplot() + theme\_gray() + geom\_point(data = dat3\_CC10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotMF1 <- ggplot() + theme\_gray() + geom\_point(data = dat1\_MF10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotMF2 <- ggplot() + theme\_gray() + geom\_point(data = dat2\_MF10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotMF3 <- ggplot() + theme\_gray() + geom\_point(data = dat3\_MF10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

#############################################################

#ggplot2,KEGG

plotKEGG <- ggplot() + theme\_gray() + geom\_point(data = KEGG\_t, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotKEGG4 <- ggplot() + theme\_gray() + geom\_point(data = dat4\_KEGG10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotKEGG5 <- ggplot() + theme\_gray() + geom\_point(data = dat5\_KEGG10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

plotKEGG6 <- ggplot() + theme\_gray() + geom\_point(data = dat6\_KEGG10, aes(x = log10p, y = reorder(x = term, X = log10p), color = group, alpha = 0.9, size = `Overlap`))

############################################################################################

#svg出力

#GO

#svg(file="stringApp\_GO\_top10\_all.svg") #ファイル名指定

svg(file="stringApp\_GO\_top10.svg") #ファイル名指定

print(plotGO) #プロット作成

dev.off() #svg出力

#svg(file="stringApp\_GO\_top10\_P.svg") #ファイル名指定

#print(plotGO1) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_GO\_top10\_C.svg") #ファイル名指定

#print(plotGO2) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_GO\_top10\_PxC.svg") #ファイル名指定

#print(plotGO3) #プロット作成

#dev.off() #svg出力

#BP

#svg(file="stringApp\_BP\_top10\_P.svg") #ファイル名指定

#print(plotBP1) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_BP\_top10\_C.svg") #ファイル名指定

#print(plotBP2) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_BP\_top10\_PxC.svg") #ファイル名指定

#print(plotBP3) #プロット作成

#dev.off() #svg出力

#CC

#svg(file="stringApp\_CC\_top10\_P.svg") #ファイル名指定

#print(plotCC1) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_CC\_top10\_C.svg") #ファイル名指定

#print(plotCC2) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_CC\_top10\_PxC.svg") #ファイル名指定

#print(plotCC3) #プロット作成

#dev.off() #svg出力

#MF

#svg(file="stringApp\_MF\_top10\_P.svg") #ファイル名指定

#print(plotMF1) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_MF\_top10\_C.svg") #ファイル名指定

#print(plotMF2) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_MF\_top10\_PxC.svg") #ファイル名指定

#print(plotMF3) #プロット作成

#dev.off() #svg出力

#KEGG

#svg(file="stringApp\_KEGG\_top10\_all.svg") #ファイル名指定

svg(file="stringApp\_KEGG\_top10.svg") #ファイル名指定

print(plotKEGG) #プロット作成

dev.off() #svg出力

#svg(file="stringApp\_KEGG\_top10\_P.svg") #ファイル名指定

#print(plotKEGG4) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_KEGG\_top10\_C.svg") #ファイル名指定

#print(plotKEGG5) #プロット作成

#dev.off() #svg出力

#svg(file="stringApp\_KEGG\_top10\_PxC.svg") #ファイル名指定

#print(plotKEGG6) #プロット作成

#dev.off() #svg出力

#P359################################################################

#レシピ14.1 PDFベクタファイルへの出力

#GO

pdf(file="stringApp\_GO\_top10\_all.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

#pdf(file="stringApp\_GO\_top10.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotGO) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_GO\_top10\_P.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotGO1) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_GO\_top10\_C.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotGO2) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_GO\_top10\_PxC.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotGO3) #プロット作成

#dev.off() #pdf出力

#BP

#pdf(file="stringApp\_BP\_top10\_P.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotBP1) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_BP\_top10\_C.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotBP2) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_BP\_top10\_PxC.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotBP3) #プロット作成

#dev.off() #pdf出力

#CC

#pdf(file="stringApp\_CC\_top10\_P.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotCC1) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_CC\_top10\_C.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotCC2) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_CC\_top10\_PxC.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotCC3) #プロット作成

#dev.off() #pdf出力

#MF

#pdf(file="stringApp\_MF\_top10\_P.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotMF1) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_MF\_top10\_C.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotMF2) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_MF\_top10\_PxC.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotMF3) #プロット作成

dev.off() #pdf出力

#KEGG

pdf(file="stringApp\_KEGG\_top10\_all.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

#pdf(file="stringApp\_KEGG\_top10.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotKEGG) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_KEGG\_top10\_P.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotKEGG4) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_KEGG\_top10\_C.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotKEGG5) #プロット作成

#dev.off() #pdf出力

#pdf(file="stringApp\_KEGG\_top10\_PxC.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plotKEGG6) #プロット作成

dev.off() #pdf出力

#############################################################

#ggplot2

#volcano plot

volGO1 <- ggplot(dat1\_GO10, aes(x = `Overlap`, y = log10p, #xy値設定

shape = top10, colour = top10, fill = `Overlap` #カラー設定

)) + geom\_point(size = dat1\_GO10$`Overlap` \* 0.05, #プロットサイズ設定

alpha = 0.8) + #プロット透明度設定

theme\_bw()+ scale\_shape\_manual(values = c(21,21)) + #プロット形状設定

scale\_colour\_manual(values = c("black","red")) + scale\_fill\_gradient2(low = "dark grey", mid = "black", high = "dark red", midpoint = 1) +

scale\_x\_continuous(limits = c(0, 150)) + #x軸の範囲設定

scale\_y\_continuous(limits = c(0, 35)) #y軸の範囲設定

volGO1 #グラフをみて範囲を調整

volGO2 <- ggplot(dat2\_GO10, aes(x = `Overlap`, y = log10p, shape = top10, colour = top10, fill = `Overlap`)) + geom\_point(size = dat1\_GO10$`Overlap` \* 0.05, alpha = 0.8) + theme\_bw()+ scale\_shape\_manual(values = c(21,21)) + scale\_colour\_manual(values = c("black","red")) + scale\_fill\_gradient2(low = "dark grey", mid = "black", high = "dark red", midpoint = 1) +

scale\_x\_continuous(limits = c(0, 150)) +

scale\_y\_continuous(limits = c(0, 35))

volGO2

volGO3 <- ggplot(dat3\_GO10, aes(x = `Overlap`, y = log10p, shape = top10, colour = top10, fill = `Overlap`)) + geom\_point(size = dat1\_GO10$`Overlap` \* 0.05, alpha = 0.8) + theme\_bw()+ scale\_shape\_manual(values = c(21,21)) + scale\_colour\_manual(values = c("black","red")) + scale\_fill\_gradient2(low = "dark grey", mid = "black", high = "dark red", midpoint = 1) +

scale\_x\_continuous(limits = c(0, 150)) +

scale\_y\_continuous(limits = c(0, 35))

volGO3

volKEGG4 <- ggplot(dat4\_KEGG10, aes(x = `Overlap`, y = log10p, shape = top10, colour = top10, fill = `Overlap`)) + geom\_point(size = dat1\_GO10$`Overlap` \* 0.05, alpha = 0.8) + theme\_bw()+ scale\_shape\_manual(values = c(21,21)) + scale\_colour\_manual(values = c("black","red")) + scale\_fill\_gradient2(low = "dark grey", mid = "black", high = "dark red", midpoint = 1) +

scale\_x\_continuous(limits = c(0, 150)) +

scale\_y\_continuous(limits = c(0, 35))

volKEGG4

volKEGG5 <- ggplot(dat5\_KEGG10, aes(x = `Overlap`, y = log10p, shape = top10, colour = top10, fill = `Overlap`)) + geom\_point(size = dat1\_GO10$`Overlap` \* 0.05, alpha = 0.8) + theme\_bw()+ scale\_shape\_manual(values = c(21,21)) + scale\_colour\_manual(values = c("black","red")) + scale\_fill\_gradient2(low = "dark grey", mid = "black", high = "dark red", midpoint = 1) +

scale\_x\_continuous(limits = c(0, 150)) +

scale\_y\_continuous(limits = c(0, 35))

volKEGG5

volKEGG6 <- ggplot(dat6\_KEGG10, aes(x = `Overlap`, y = log10p, shape = top10, colour = top10, fill = `Overlap`)) + geom\_point(size = dat1\_GO10$`Overlap` \* 0.05, alpha = 0.8) + theme\_bw()+ scale\_shape\_manual(values = c(21,21)) + scale\_colour\_manual(values = c("black","red")) + scale\_fill\_gradient2(low = "dark grey", mid = "black", high = "dark red", midpoint = 1) +

scale\_x\_continuous(limits = c(0, 150)) +

scale\_y\_continuous(limits = c(0, 35))

volKEGG6

#############################################################

#P359################################################################

#レシピ14.1 PDFベクタファイルへの出力

#GO

pdf("stringApp\_volGO\_top10\_all.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

#pdf("stringApp\_volGO\_top10\_P.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(volGO1) #プロット作成

#dev.off() #PDF出力

#pdf("stringApp\_volGO\_top10\_C.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(volGO2) #プロット作成

#dev.off() #PDF出力

#pdf("stringApp\_volGO\_top10\_PxC.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(volGO3) #プロット作成

dev.off() #PDF出力

#KEGG

pdf("stringApp\_volKEGG\_top10\_all.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

#pdf("stringApp\_volKEGG\_top10\_P.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(volKEGG4) #プロット作成

#dev.off() #PDF出力

#pdf("stringApp\_volKEGG\_top10\_C.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(volKEGG5) #プロット作成

#dev.off() #PDF出力

#pdf("stringApp\_volKEGG\_top10\_PxC.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(volKEGG6) #プロット作成

dev.off() #PDF出力

#############################################################

#xslx出力

smp <- list("GO"=GO\_t, "GO\_P"=dat1\_GO10, "GO\_C"=dat2\_GO10, "GO\_PxC"=dat3\_GO10, "KEGG"=KEGG\_t, "KEGG\_P"=dat4\_KEGG10, "KEGG\_C"=dat5\_KEGG10, "KEGG\_PxC"=dat6\_KEGG10) #GO,KEGGのtop10リスト作成

write.xlsx(smp, "GO\_KEGG\_top10.xlsx") #GOシート,KEGGシート出力

#############################################################

#エクセルでRの特殊記号"|"を","に置換し、GO\_KEGGr.xlsxとして保存！！！