#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/SWATH") #作業ディレクトリ設定

#setwd("~/GoogleDrive/マイドライブ/0\_Work/R/SWATH") #作業ディレクトリ設定

#setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/AMY/2\_Cytoscape") #作業ディレクトリ設定

#setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/HIP/2\_Cytoscape") #作業ディレクトリ設定

#setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/NAc/2\_Cytoscape") #作業ディレクトリ設定

#setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/PFC/2\_Cytoscape") #作業ディレクトリ設定

setwd("/Users/akira\_yoshimi/Dropbox/0\_Work/R/Perseus\_Like\_Analysis/STR/2\_Cytoscape") #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

str(dat1)

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

#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" ~ "MP", #categoryのGO FunctionをMP

`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" ~ "MP", #categoryのGO FunctionをMP

`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" ~ "MP", #categoryのGO FunctionをMP

`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

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

#Inf(無限大)の置換:無いので処理しない

#ifelse((is.infinite(dat1$log10p) & (dat1$log10p > 0)), 100, dat1$log10p) #Infの場合に100に置換したい

#replace(dat1$log10p, which(is.infinite(dat1$log10p) & (dat1$log10p > 0)), 100) #Infの場合に100に置換

#dat1$log10p[is.infinite(dat1$log10p)& (dat1$log10p > 0)] <- 100 #置換の実行

#dat1$log10p #置換結果の表示

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

#i列目の条件で抽出

#i <- 3 #3列目の条件

#varname <- colnames(dat1)[i] #条件抽出する列名varname

#x <- dat1 %>% dplyr::filter\_(paste(varname, "==", 1)) #数字のみ抽出可能

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

#GO抽出

dat1\_f <- 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

dat2\_f <- 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

dat3\_f <- 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

#KEGG抽出

dat4\_f <- dat1 %>% #dat1データの

dplyr::filter(category=="KEGG Pathways") #category列がKEGG Pathways

dat5\_f <- dat2 %>% #dat2データの

dplyr::filter(category=="KEGG Pathways") #category列がKEGG Pathways

dat6\_f <- dat3 %>% #dat3データの

dplyr::filter(category=="KEGG Pathways") #category列がKEGG Pathways

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

#top50データの抽出

dat1\_ft <- dat1\_f %>%

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

top\_n(n = 50, wt = log10p) #log10p上位50のデータを保存

dat2\_ft <- dat2\_f %>% #P\_GOのデータから

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

top\_n(n = 50, wt = log10p) #log10p上位50のデータを保存

dat3\_ft <- dat3\_f %>%

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

top\_n(n = 50, wt = log10p) #log10p上位50のデータを保存

dat4\_ft <- dat4\_f %>% #P\_KEGGのデータから

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

top\_n(n = 50, wt = log10p) #log10p上位50のデータを保存

dat5\_ft <- dat5\_f %>% #C\_KEGGのデータから

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

top\_n(n = 50, wt = log10p) #log10p上位50のデータを保存

dat6\_ft <- dat6\_f %>% #PxC\_KEGGのデータから

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

top\_n(n = 50, wt = log10p) #log10p上位50のデータを保存

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

#NA(欠損値)の置換

#ifelse(is.na(dat1\_ft$log10p), 0, dat1\_ft$log10p)

#replace(dat1\_ft$log10p, which(is.na(dat1\_ft$log10p)), 0)

#dat1\_ft$log10p[is.na(dat1\_ft$log10p)] <- 0

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

#top50データをラベル

dat1\_f <- mutate(dat1\_f, top50 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat2\_f <- mutate(dat2\_f, top50 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat3\_f <- mutate(dat3\_f, top50 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat4\_f <- mutate(dat4\_f, top50 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat5\_f <- mutate(dat5\_f, top50 = if\_else(log10p > 0, true = TRUE, false = FALSE))

dat6\_f <- mutate(dat6\_f, top50 = 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

GO <- rbind(dat1\_f, dat2\_f, dat3\_f) #data1,2,3結合

KEGG <- rbind(dat4\_f, dat5\_f, dat6\_f) #data4,5,6結合

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

#重複削除,top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\_f))

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

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

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

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

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

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

#データにAnnotationを結合

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

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

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

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

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

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

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

#グループ挿入

dat1\_t$group <- "PCP" #グループ挿入

dat2\_t$group <- "CLZ" #グループ挿入

dat3\_t$group <- "Interaction" #グループ挿入

dat4\_t$group <- "PCP" #グループ挿入

dat5\_t$group <- "CLZ" #グループ挿入

dat6\_t$group <- "Interaction" #グループ挿入

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

#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, dat2\_t, dat3\_t) #dat1,2,3結合

KEGG\_t <- rbind(dat4\_t, dat5\_t, dat6\_t) #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\_ft <- dat1\_ft %>% mutate(description = as.factor(description)) #descriptionをファクタに変換

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

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

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

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

dat6\_ft <- dat6\_ft %>% 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" ~ "MP", #categoryのGO FunctionをMP

`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" ~ "MP", #categoryのGO FunctionをMP

`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

plot1 <- 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`))

plot2 <- ggplot()+

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

geom\_point(data = dat1\_ft, aes(x = log10p,

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

#position = "jitter", #geom\_jitter

color = group,

alpha = 0.9,

size = `Overlap`))

plot3 <- ggplot()+

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

geom\_point(data = dat2\_ft, aes(x = log10p,

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

#position = "jitter", #geom\_jitter

color = group,

alpha = 0.9,

size = `Overlap`))

plot4 <- ggplot()+

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

geom\_point(data = dat3\_ft, aes(x = log10p,

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

#position = "jitter", #geom\_jitter

color = group,

alpha = 0.9, #プロット透明度設定

size = `Overlap`))

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

#ggplot2,KEGG

plot5 <- ggplot()+

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

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

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

#position = "jitter", #geom\_jitter

color = group,

alpha = 0.9,

size = `Overlap`))

plot6 <- ggplot()+

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

geom\_point(data = dat4\_ft, aes(x = log10p,

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

#position = "jitter", #geom\_jitter

color = group,

alpha = 0.9,

size = `Overlap`))

plot7 <- ggplot()+

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

geom\_point(data = dat5\_ft, aes(x = log10p,

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

#position = "jitter", #geom\_jitter

color = group,

alpha = 0.9,

size = `Overlap`))

plot8 <- ggplot()+

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

geom\_point(data = dat6\_ft, aes(x = log10p,

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

#position = "jitter", #geom\_jitter

color = group,

alpha = 0.9,

size = `Overlap`))

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

#svg出力

svg(file="stringAppGO.svg") #ファイル名指定

print(plot1) #プロット作成

dev.off() #svg出力

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

print(plot2) #プロット作成

dev.off() #svg出力

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

print(plot3) #プロット作成

dev.off() #svg出力

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

print(plot4) #プロット作成

dev.off() #svg出力

svg(file="stringAppKEGG.svg") #ファイル名指定

print(plot5) #プロット作成

dev.off() #svg出力

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

print(plot6) #プロット作成

dev.off() #svg出力

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

print(plot7) #プロット作成

dev.off() #svg出力

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

print(plot8) #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

pdf("stringAppGO.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plot1) #プロット作成

dev.off() #PDF出力

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

print(plot2) #プロット作成

dev.off() #PDF出力

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

print(plot3) #プロット作成

dev.off() #PDF出力

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

print(plot4) #プロット作成

dev.off() #PDF出力

pdf("stringAppKEGG.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(plot5) #プロット作成

dev.off() #PDF出力

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

print(plot6) #プロット作成

dev.off() #PDF出力

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

print(plot7) #プロット作成

dev.off() #PDF出力

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

print(plot8) #プロット作成

dev.off() #PDF出力

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

#ggplot2

#volcano plot

plot1 <- ggplot(dat1\_f, aes(x = `Overlap`, y = log10p, #xy値設定

shape = top50,

colour = top50,

fill = `Overlap` #カラー設定

)) +

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

alpha = 0.8) #プロット透明度設定

vol1 <- plot1 +

theme\_bw()+

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

scale\_colour\_manual(values = c("black","red"))+ #プロット形状設定

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

vol1 #グラフをみて範囲を調整

plot2 <- ggplot(dat2\_f, aes(x = `Overlap`, y = log10p, #xy値設定

shape = top50,

colour = top50,

fill = `Overlap` #カラー設定

)) +

geom\_point(size = dat2\_f$`Overlap` \* 0.02, #プロットサイズ設定

alpha = 0.8) #プロット透明度設定

vol2 <- plot2 +

theme\_bw()+

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

scale\_colour\_manual(values = c("black","red"))+ #プロット形状設定

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

vol2 #グラフをみて範囲を調整

plot3 <- ggplot(dat3\_f, aes(x = `Overlap`, y = log10p, #xy値設定

shape = top50,

colour = top50,

fill = `Overlap` #カラー設定

)) +

geom\_point(size = dat3\_f$`Overlap` \* 0.01, #プロットサイズ設定

alpha = 0.8) #プロット透明度設定

vol3 <- plot3 +

theme\_bw()+

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

scale\_colour\_manual(values = c("black","red"))+ #プロット形状設定

scale\_fill\_gradient2(

low = "dark grey",

mid = "black",

high = "dark red",

midpoint = 1

)+

scale\_x\_continuous(limits = c(0, 350)) + #x軸の範囲設定

scale\_y\_continuous(limits = c(0, 70)) #y軸の範囲設定

vol3 #グラフをみて範囲を調整

plot4 <- ggplot(dat4\_f, aes(x = `Overlap`, y = log10p, #xy値設定

shape = top50,

colour = top50,

fill = `Overlap` #カラー設定

)) +

geom\_point(size = dat4\_f$`Overlap` \* 0.2, #プロットサイズ設定

alpha = 0.8) #プロット透明度設定

vol4 <- plot4 +

theme\_bw()+

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

scale\_colour\_manual(values = c("black","red"))+ #プロット形状設定

scale\_fill\_gradient2(

low = "dark grey",

mid = "black",

high = "dark red",

midpoint = 1

)+

scale\_x\_continuous(limits = c(0, 20)) + #x軸の範囲設定

scale\_y\_continuous(limits = c(0, 10)) #y軸の範囲設定

vol4 #グラフをみて範囲を調整

plot5 <- ggplot(dat5\_f, aes(x = `Overlap`, y = log10p, #xy値設定

shape = top50,

colour = top50,

fill = `Overlap` #カラー設定

)) +

geom\_point(size = dat5\_f$`Overlap` \* 0.2, #プロットサイズ設定

alpha = 0.8) #プロット透明度設定

vol5 <- plot5 +

theme\_bw()+

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

scale\_colour\_manual(values = c("black","red"))+ #プロット形状設定

scale\_fill\_gradient2(

low = "dark grey",

mid = "black",

high = "dark red",

midpoint = 1

)+

scale\_x\_continuous(limits = c(0, 30)) + #x軸の範囲設定

scale\_y\_continuous(limits = c(0, 15)) #y軸の範囲設定

vol5 #グラフをみて範囲を調整

plot6 <- ggplot(dat6\_f, aes(x = `Overlap`, y = log10p, #xy値設定

shape = top50,

colour = top50,

fill = `Overlap` #カラー設定

)) +

geom\_point(size = dat6\_f$`Overlap` \* 0.07, #プロットサイズ設定

alpha = 0.8) #プロット透明度設定

vol6 <- plot6 +

theme\_bw()+

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

scale\_colour\_manual(values = c("black","red"))+ #プロット形状設定

scale\_fill\_gradient2(

low = "dark grey",

mid = "black",

high = "dark red",

midpoint = 1

)+

scale\_x\_continuous(limits = c(0, 60)) + #x軸の範囲設定

scale\_y\_continuous(limits = c(0, 40)) #y軸の範囲設定

vol6 #グラフをみて範囲を調整

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

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

#レシピ14.1 PDFベクタファイルへの出力

pdf("stringApp\_vol1.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(vol1) #プロット作成

dev.off() #PDF出力

pdf("stringApp\_vol2.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(vol2) #プロット作成

dev.off() #PDF出力

pdf("stringApp\_vol3.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(vol3) #プロット作成

dev.off() #PDF出力

pdf("stringApp\_vol4.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(vol4) #プロット作成

dev.off() #PDF出力

pdf("stringApp\_vol5.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(vol5) #プロット作成

dev.off() #PDF出力

pdf("stringApp\_vol6.pdf", width = 20/2.54, height = 20/2.54, useDingbats = FALSE) #幅,高さcm(inch/2.54)

print(vol6) #プロット作成

dev.off() #PDF出力

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

#xslx出力

smp <- list("GO"=GO\_t, "GO\_P"=dat1\_ft, "GO\_C"=dat2\_ft, "GO\_PxC"=dat3\_ft, "KEGG"=KEGG\_t, "KEGG\_P"=dat4\_ft, "KEGG\_C"=dat5\_ft, "KEGG\_PxC"=dat6\_ft) #GO,KEGGのtop50リスト作成

write.xlsx(smp, "GO\_KEGG.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！！！