**Supplementary File 2. The R language codes for GSEA pathway enrichment analysis.**

library(DOSE)

library(GOSemSim)

library(clusterProfiler)

library(org.Hs.eg.db)

library(org.Mm.eg.db)

library(org.Rn.eg.db)

library(dplyr)

library(GO.db)

#

get\_GO\_data <- function(OrgDb, ont, keytype) {

GO\_Env <- get\_GO\_Env()

use\_cached <- FALSE

if (exists("organism", envir=GO\_Env, inherits=FALSE) &&

exists("keytype", envir=GO\_Env, inherits=FALSE)) {

org <- get("organism", envir=GO\_Env)

kt <- get("keytype", envir=GO\_Env)

if (org == DOSE:::get\_organism(OrgDb) &&

keytype == kt &&

exists("goAnno", envir=GO\_Env, inherits=FALSE)) {

## https://github.com/GuangchuangYu/clusterProfiler/issues/182

## && exists("GO2TERM", envir=GO\_Env, inherits=FALSE)){

use\_cached <- TRUE

}

}

if (use\_cached) {

goAnno <- get("goAnno", envir=GO\_Env)

} else {

OrgDb <- GOSemSim:::load\_OrgDb(OrgDb)

kt <- keytypes(OrgDb)

if (! keytype %in% kt) {

stop("keytype is not supported...")

}

kk <- keys(OrgDb, keytype=keytype)

goAnno <- suppressMessages(

AnnotationDbi::select(OrgDb, keys=kk, keytype=keytype,

columns=c("GOALL", "ONTOLOGYALL")))

goAnno <- unique(goAnno[!is.na(goAnno$GOALL), ])

assign("goAnno", goAnno, envir=GO\_Env)

assign("keytype", keytype, envir=GO\_Env)

assign("organism", DOSE:::get\_organism(OrgDb), envir=GO\_Env)

}

if (ont == "ALL") {

GO2GENE <- unique(goAnno[, c(2,1)])

} else {

GO2GENE <- unique(goAnno[goAnno$ONTOLOGYALL == ont, c(2,1)])

}

GO\_DATA <- DOSE:::build\_Anno(GO2GENE, get\_GO2TERM\_table())

goOnt.df <- goAnno[, c("GOALL", "ONTOLOGYALL")] %>% unique

goOnt <- goOnt.df[,2]

names(goOnt) <- goOnt.df[,1]

assign("GO2ONT", goOnt, envir=GO\_DATA)

return(GO\_DATA)

}

get\_GO\_Env <- function () {

if (!exists(".GO\_clusterProfiler\_Env", envir = .GlobalEnv)) {

pos <- 1

envir <- as.environment(pos)

assign(".GO\_clusterProfiler\_Env", new.env(), envir=envir)

}

get(".GO\_clusterProfiler\_Env", envir = .GlobalEnv)

}

get\_GO2TERM\_table <- function() {

GOTERM.df <- get\_GOTERM()

GOTERM.df[, c("go\_id", "Term")] %>% unique

}

get\_GOTERM <- function() {

pos <- 1

envir <- as.environment(pos)

if (!exists(".GOTERM\_Env", envir=envir)) {

assign(".GOTERM\_Env", new.env(), envir)

}

GOTERM\_Env <- get(".GOTERM\_Env", envir = envir)

if (exists("GOTERM.df", envir = GOTERM\_Env)) {

GOTERM.df <- get("GOTERM.df", envir=GOTERM\_Env)

} else {

GOTERM.df <- toTable(GOTERM)

assign("GOTERM.df", GOTERM.df, envir = GOTERM\_Env)

}

return(GOTERM.df)

}