NetBAS for tumor genes

HBG

11/4/2018

This script perform GO enrichment using NetBAS

for 51 PDAC cell high BF genes

```
panc.file <- read.csv("../rnf43.csv",header=TRUE,stringsAsFactors=F)
panc.gene <- panc.file$gene
panc.panc <- panc.file$panc.mean
panc.np <- panc.file$nonpanc.mean
quant <- quantile(panc.panc, probs = seq(0,1,1/20))
quant.np <- quantile(panc.np, probs=seq(0,1,1/20))

# top 10% BF genes (1718) in tumor cells
panc.top10 <- panc.gene[which(panc.panc > quant[19])]
length(panc.top10)
```

```
## [1] 1718
```

```
# genes (8458) of 0% to 50% BF factors in normal cells
np.bottom50 <- panc.gene[which(panc.np < quant.np[11])]
length(np.bottom50)
```

```
## [1] 8458
```

```
# the overlap genes between top 10% tumor and (0-50%) normal cells
gene.list <- panc.gene[which(panc.top10 %in% np.bottom50)]
length(gene.list)
```

```
## [1] 51
```

```
gene.list
```

```
##  [1] "FZD5"     "WLS"      "HNF1A"    "RBM15"    "PPCS"     "PORCN"
##  [7] "TPK1"     "ADRBK1"   "NANS"     "PPARG"    "SLC2A1"   "STK40"
## [13] "MPI"      "ALG3"     "WNT3"     "TRIB1"    "STX4"     "SLC25A1"
## [19] "WDR26"    "VPS4B"    "MMACHC"   "PPP5C"    "KATNA1"   "ARHGEF39"
## [25] "GALE"     "MOGS"     "PCBD1"    "PGM3"     "PRR12"    "AP2M1"
## [31] "PTCH2"    "RIF1"     "TFB1M"    "DHX29"    "MTR"      "NPC1"
## [37] "PPP2R4"   "ALG9"     "PPCDC"    "GOLGA7"   "KLK3"     "EDC3"
## [43] "LAMA3"    "ACO2"     "MTRR"     "RANBP17"  "NDE1"     "FAM221B"
## [49] "FDFT1"    "DPF2"     "ELMSAN1"
```

```
## read the original network
network <- read.csv("../Data/human.pin.csv", header=T, stringsAsFactors=F)
geneA <- network$geneA
geneB <- network$geneB

GOcategory.file <- read.csv("../Data/human.cc.term.csv",header=TRUE, stringsAsFactors=F)
cc.go.cat <- GOcategory.file$GO.term
```

```
cc.dim <- length(cc.go.cat)

GOterm.file <- read.csv("../Data/human.cc.gene.term.csv", header=T, stringsAsFactors=F)
cc.GO.gene <- GOterm.file$gene  #it should be changed to System for yeast pin
cc.GO.term <- GOterm.file$GO.term

vec <- numeric(length=cc.dim)

for (i in 1:length(gene.list)) {
    orf <- as.character(gene.list[i])
    intA <- geneB[which(geneA %in% orf)]
    for (j in 1:length(intA)) {
        ccA <- cc.GO.term[which(cc.GO.gene %in% intA[j])]
        for (k in 1:length(ccA)) {
           na <- which(cc.go.cat %in% ccA[k])
           vec[na] <- vec[na] + 1
        }
    }

    intB <- geneA[which(geneB %in% orf)]
    for (s in 1:length(intB)) {
        ccB <- cc.GO.term[which(cc.GO.gene %in% intB[s])]
        for (t in 1:length(ccB)) {
           nb <- which(cc.go.cat %in% ccB[t])
           vec[nb] <- vec[nb] + 1
        }
    }
}

write.table(vec, file="hs.rnf43.list.cc.txt", col.names=F, row.names=F, quote=F)
```

```
# Now the ms02star permutations
for (p in 1:100) {
permutation.file <- paste("../ms02star/human/", "ms02.", p, ".csv", sep="")
permutation <- read.csv(permutation.file, header=T, stringsAsFactors = F)
geneA <- permutation$id1
geneB <- permutation$id2

vecp <-  numeric(length = cc.dim)
for (i in 1:length(gene.list)) {
    orf <- as.character(gene.list[i])
    intA <- geneB[which(geneA %in% orf)]
    for (j in 1:length(intA)) {
        ccA <- cc.GO.term[which(cc.GO.gene %in% intA[j])]
        for (k in 1:length(ccA)) {
           na <- which(cc.go.cat %in% ccA[k])
           vecp[na] <- vecp[na] + 1
        }
    }

    intB <- geneA[which(geneB %in% orf)]
    for (s in 1:length(intB)) {
        ccB <- cc.GO.term[which(cc.GO.gene %in% intB[s])]
        for (t in 1:length(ccB)) {
```

```
            nb <- which(cc.go.cat %in% ccB[t])
            vecp[nb] <- vecp[nb] + 1
        }
    }
}

output <- paste("ms02.human", "/", "rnf43.list", "/", "ms02.", p, ".cc.matrix.csv", sep="")

write.table(vecp, file = output, col.names=F, row.names=F, quote=F)
}
```

```
library("microbenchmark")
library("matrixStats")

conn.dim <- 1

hspin <- matrix(as.numeric(unlist(read.table("hs.rnf43.list.cc.txt", header=F, sep=","))), nrow=cc.dim,
obs <- c(hspin)

perm <- c()
for (i in 1:100) {
    name <- paste("ms02.human", "/", "rnf43.list", "/", "ms02.", i, ".cc.matrix.csv", sep="")
    mat <- matrix(as.numeric(unlist(read.table(name, header=F, sep=","))), nrow=cc.dim, ncol=conn.dim)
    perm <- rbind(perm, c(mat))
}

mean <- colMeans(perm)
std <- colSds(perm)

zscore <- round((obs - mean)/std, 3)

z <- matrix(zscore, nrow=cc.dim, ncol=conn.dim)

write.table(z, file="hs.rnf43.list.cc.z.csv", sep=",", row.names=F, col.names=F, quote=F)
```

```
library('gplots')
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library('GO.db')
```

```
## Loading required package: AnnotationDbi
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
```

```
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind,
##     colMeans, colnames, colSums, dirname, do.call, duplicated,
##     eval, evalq, Filter, Find, get, grep, grepl, intersect,
##     is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##     paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##     Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which, which.max,
##     which.min

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:gplots':
##
##     space

## The following object is masked from 'package:base':
##
##     expand.grid

##
order <- order(z)
cc.go.cat <- cc.go.cat[order]
z <- z[order]

z <- t(z)
colnames(z) <- cc.go.cat

enriched.list <- cc.go.cat[which(z >= 3)]
```

```
enriched <- c("GO.ID", "GO.Term", "Z-score")
for (i in 1:length(enriched.list)) {
  term <- Term(GOID(as.character(enriched.list[i])))
  enriched <- rbind(enriched, c(as.character(enriched.list[i]),term,
                                z[which(cc.go.cat %in% enriched.list[i])]))
}

print(enriched)
```

```
##
## enriched "GO.ID"
##          "GO:0005783"
##          "GO:0005774"
##          "GO:0030176"
##          "GO:0043202"
##          "GO:0071556"
##          "GO:0098793"
##          "GO:0043005"
##          "GO:1904115"
##          "GO:0005874"
##          "GO:0005770"
##          "GO:0062023"
##          "GO:0070382"
##          "GO:0005604"
##          "GO:0008021"
##          "GO:0005769"
##          "GO:0005905"
##          "GO:0005814"
##          "GO:0008305"
##          "GO:0005771"
##          "GO:0032588"
##          "GO:0005829"
##          "GO:0016342"
##          "GO:0005615"
##          "GO:0072562"
##          "GO:0048471"
##          "GO:0005794"
##          "GO:0031901"
##          "GO:0005576"
##          "GO:0005871"
##          "GO:0030136"
##          "GO:0030424"
##          "GO:0070062"
##          "GO:0030672"
##          "GO:0030658"
##          "GO:0030669"
##          "GO:0005764"
##          "GO:0000777"
##          "GO:0030665"
##          "GO:0005765"
##          "GO:0031201"
##          "GO:0030666"
##          "GO:0031012"
##          "GO:0016592"
```

```
##          "GO:0031902"
##          "GO:0005796"
##          "GO:0005788"
##          "GO:0043231"
##          "GO:0009986"
##          GO:0005783
## enriched "GO.Term"
##          "endoplasmic reticulum"
##          "vacuolar membrane"
##          "integral component of endoplasmic reticulum membrane"
##          "lysosomal lumen"
##          "integral component of lumenal side of endoplasmic reticulum membrane"
##          "presynapse"
##          "neuron projection"
##          "axon cytoplasm"
##          "microtubule"
##          "late endosome"
##          "collagen-containing extracellular matrix"
##          "exocytic vesicle"
##          "basement membrane"
##          "synaptic vesicle"
##          "early endosome"
##          "clathrin-coated pit"
##          "centriole"
##          "integrin complex"
##          "multivesicular body"
##          "trans-Golgi network membrane"
##          "cytosol"
##          "catenin complex"
##          "extracellular space"
##          "blood microparticle"
##          "perinuclear region of cytoplasm"
##          "Golgi apparatus"
##          "early endosome membrane"
##          "extracellular region"
##          "kinesin complex"
##          "clathrin-coated vesicle"
##          "axon"
##          "extracellular exosome"
##          "synaptic vesicle membrane"
##          "transport vesicle membrane"
##          "clathrin-coated endocytic vesicle membrane"
##          "lysosome"
##          "condensed chromosome kinetochore"
##          "clathrin-coated vesicle membrane"
##          "lysosomal membrane"
##          "SNARE complex"
##          "endocytic vesicle membrane"
##          "extracellular matrix"
##          "mediator complex"
##          "late endosome membrane"
##          "Golgi lumen"
##          "endoplasmic reticulum lumen"
##          "intracellular membrane-bounded organelle"
```

```
##          "cell surface"
##
## enriched "Z-score"
##          "3.056"
##          "3.064"
##          "3.102"
##          "3.165"
##          "3.22"
##          "3.282"
##          "3.344"
##          "3.372"
##          "3.402"
##          "3.445"
##          "3.457"
##          "3.503"
##          "3.562"
##          "3.65"
##          "3.66"
##          "3.699"
##          "3.701"
##          "3.736"
##          "3.814"
##          "3.866"
##          "3.907"
##          "3.911"
##          "3.966"
##          "4.049"
##          "4.171"
##          "4.176"
##          "4.251"
##          "4.257"
##          "4.306"
##          "4.336"
##          "4.393"
##          "4.506"
##          "4.722"
##          "4.752"
##          "5.303"
##          "5.318"
##          "5.347"
##          "5.466"
##          "5.482"
##          "5.621"
##          "5.729"
##          "6.184"
##          "6.349"
##          "6.58"
##          "6.611"
##          "6.702"
##          "6.987"
##          "7.158"
```

```
write.table(enriched, file="hs.rnf43.list.cc.enriched.csv", row.names=F, col.names=F, quote=F, sep="\t"
```

```
###No suppressed terms have been found
#sup.list <- cc.go.cat[which(z <= -3)]
#sup <- c("GO.ID", "GO.Term", "Z-score")
#for (i in 1:length(sup.list)) {
#  term <- Term(GOID(as.character(sup.list[i])))
#  sup <- rbind(sup, c(as.character(sup.list[i]),term,
#                z[which(cc.go.cat %in% sup.list[i])]))
#}

#print(sup)

#write.table(sup, file="human.pdcd1.cc.suppressed.csv", row.names=F, col.names=F, quote=F, sep="\t")
###

#Note that there may be "inf" Z-scores owing to lack of sampling (i.e., zero in standard deviations)
#We can also extract the GO-terms for the gene for comparison
```