

# 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"   "ARHGFEF39"
## [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.bp.term.csv",header=TRUE, stringsAsFactors=F)
bp.go.cat <- GOcategory.file$GO.term
```

```

bp.dim <- length(bp.go.cat)

G0term.file <- read.csv("../Data/human.bp.gene.term.csv", header=T, stringsAsFactors=F)
bp.GO.gene <- G0term.file$gene #it should be changed to System for yeast pin
bp.GO.term <- G0term.file$G0.term

vec <- numeric(length=bp.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)) {
    bpA <- bp.GO.term[which(bp.GO.gene %in% intA[j])]
    for (k in 1:length(bpA)) {
      na <- which(bp.go.cat %in% bpA[k])
      vec[na] <- vec[na] + 1
    }
  }

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

write.table(vec, file="hs.rnf43.list.bp.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 = bp.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)) {
      bpA <- bp.GO.term[which(bp.GO.gene %in% intA[j])]
      for (k in 1:length(bpA)) {
        na <- which(bp.go.cat %in% bpA[k])
        vecp[na] <- vecp[na] + 1
      }
    }

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

```

```

        nb <- which(bp.go.cat %in% bpB[t])
        vecp[nb] <- vecp[nb] + 1
    }
}

output <- paste("ms02.human", "/", "rnf43.list", "/", "ms02.", p, ".bp.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.bp.txt", header=F, sep=","))), nrow=bp.dim,
obs <- c(hspin)

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

write.table(z, file="hs.rnf43.list.bp.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)
bp.go.cat <- bp.go.cat[order]
z <- z[order]

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

enriched.list <- bp.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(bp.go.cat %in% enriched.list[i])]))
}

print(enriched)

```

```

##
## enriched "GO.ID"
## "GO:0044331"
## "GO:0000186"
## "GO:0034614"
## "GO:0035116"
## "GO:0060348"
## "GO:0007059"
## "GO:0030301"
## "GO:0017156"
## "GO:0050807"
## "GO:0007420"
## "GO:0071277"
## "GO:0060349"
## "GO:0045892"
## "GO:0042472"
## "GO:0006338"
## "GO:0007623"
## "GO:0030522"
## "GO:0001658"
## "GO:0006898"
## "GO:0006986"
## "GO:0045599"
## "GO:0050873"
## "GO:0034446"
## "GO:0030324"
## "GO:0042026"
## "GO:0007155"
## "GO:0010389"
## "GO:0043401"
## "GO:0042730"
## "GO:0060384"
## "GO:0051591"
## "GO:0006418"
## "GO:0007052"
## "GO:0050767"
## "GO:0030521"
## "GO:0034605"
## "GO:0006953"
## "GO:0007018"
## "GO:0048146"
## "GO:0002576"
## "GO:0022008"
## "GO:1901673"
## "GO:0001934"

```

##	"G0:0030514"
##	"G0:0007223"
##	"G0:0006812"
##	"G0:0001843"
##	"G0:0030879"
##	"G0:0043547"
##	"G0:0016192"
##	"G0:0008203"
##	"G0:0003341"
##	"G0:0042632"
##	"G0:0051085"
##	"G0:0048706"
##	"G0:0001523"
##	"G0:1900034"
##	"G0:0046718"
##	"G0:0033627"
##	"G0:0006508"
##	"G0:0031122"
##	"G0:0007224"
##	"G0:0045669"
##	"G0:0034620"
##	"G0:0006997"
##	"G0:0030198"
##	"G0:0034613"
##	"G0:0016079"
##	"G0:0019216"
##	"G0:0030574"
##	"G0:0045600"
##	"G0:0030216"
##	"G0:0090263"
##	"G0:0070268"
##	"G0:0006629"
##	"G0:0006303"
##	"G0:0036258"
##	"G0:0016042"
##	"G0:0071300"
##	"G0:0015721"
##	"G0:0097711"
##	"G0:0008202"
##	"G0:0061024"
##	"G0:0022617"
##	"G0:0090103"
##	"G0:0060071"
##	"G0:0045893"
##	"G0:0016055"
##	"G0:0090090"
##	"G0:0039702"
##	"G0:0048843"
##	"G0:0009235"
##	"G0:1904886"
##	"G0:0030182"
##	"G0:0045165"
##	"G0:0035567"
##	"G0:0060070"

```

##          GO:0044331
## enriched "GO.Term"
##          "cell-cell adhesion mediated by cadherin"
##          "activation of MAPKK activity"
##          "cellular response to reactive oxygen species"
##          "embryonic hindlimb morphogenesis"
##          "bone development"
##          "chromosome segregation"
##          "cholesterol transport"
##          "calcium ion regulated exocytosis"
##          "regulation of synapse organization"
##          "brain development"
##          "cellular response to calcium ion"
##          "bone morphogenesis"
##          "negative regulation of transcription, DNA-templated"
##          "inner ear morphogenesis"
##          "chromatin remodeling"
##          "circadian rhythm"
##          "intracellular receptor signaling pathway"
##          "branching involved in ureteric bud morphogenesis"
##          "receptor-mediated endocytosis"
##          "response to unfolded protein"
##          "negative regulation of fat cell differentiation"
##          "brown fat cell differentiation"
##          "substrate adhesion-dependent cell spreading"
##          "lung development"
##          "protein refolding"
##          "cell adhesion"
##          "regulation of G2/M transition of mitotic cell cycle"
##          "steroid hormone mediated signaling pathway"
##          "fibrinolysis"
##          "innervation"
##          "response to cAMP"
##          "tRNA aminoacylation for protein translation"
##          "mitotic spindle organization"
##          "regulation of neurogenesis"
##          "androgen receptor signaling pathway"
##          "cellular response to heat"
##          "acute-phase response"
##          "microtubule-based movement"
##          "positive regulation of fibroblast proliferation"
##          "platelet degranulation"
##          "neurogenesis"
##          "regulation of mitotic spindle assembly"
##          "positive regulation of protein phosphorylation"
##          "negative regulation of BMP signaling pathway"
##          "Wnt signaling pathway, calcium modulating pathway"
##          "cation transport"
##          "neural tube closure"
##          "mammary gland development"
##          "positive regulation of GTPase activity"
##          "vesicle-mediated transport"
##          "cholesterol metabolic process"
##          "cilium movement"

```

```

## "cholesterol homeostasis"
## "chaperone cofactor-dependent protein refolding"
## "embryonic skeletal system development"
## "retinoid metabolic process"
## "regulation of cellular response to heat"
## "viral entry into host cell"
## "cell adhesion mediated by integrin"
## "proteolysis"
## "cytoplasmic microtubule organization"
## "smoothened signaling pathway"
## "positive regulation of osteoblast differentiation"
## "cellular response to unfolded protein"
## "nucleus organization"
## "extracellular matrix organization"
## "cellular protein localization"
## "synaptic vesicle exocytosis"
## "regulation of lipid metabolic process"
## "collagen catabolic process"
## "positive regulation of fat cell differentiation"
## "keratinocyte differentiation"
## "positive regulation of canonical Wnt signaling pathway"
## "cornification"
## "lipid metabolic process"
## "double-strand break repair via nonhomologous end joining"
## "multivesicular body assembly"
## "lipid catabolic process"
## "cellular response to retinoic acid"
## "bile acid and bile salt transport"
## "ciliary basal body-plasma membrane docking"
## "steroid metabolic process"
## "membrane organization"
## "extracellular matrix disassembly"
## "cochlea morphogenesis"
## "Wnt signaling pathway, planar cell polarity pathway"
## "positive regulation of transcription, DNA-templated"
## "Wnt signaling pathway"
## "negative regulation of canonical Wnt signaling pathway"
## "viral budding via host ESCRT complex"
## "negative regulation of axon extension involved in axon guidance"
## "cobalamin metabolic process"
## "beta-catenin destruction complex disassembly"
## "neuron differentiation"
## "cell fate commitment"
## "non-canonical Wnt signaling pathway"
## "canonical Wnt signaling pathway"
##
## enriched "Z-score"
## "3.01"
## "3.014"
## "3.029"
## "3.068"
## "3.069"
## "3.078"
## "3.083"

```



##	"3.103"
##	"3.117"
##	"3.122"
##	"3.128"
##	"3.129"
##	"3.131"
##	"3.16"
##	"3.182"
##	"3.214"
##	"3.285"
##	"3.303"
##	"3.315"
##	"3.37"
##	"3.385"
##	"3.388"
##	"3.396"
##	"3.42"
##	"3.451"
##	"3.458"
##	"3.482"
##	"3.484"
##	"3.494"
##	"3.499"
##	"3.514"
##	"3.553"
##	"3.565"
##	"3.575"
##	"3.598"
##	"3.604"
##	"3.605"
##	"3.613"
##	"3.619"
##	"3.638"
##	"3.669"
##	"3.689"
##	"3.7"
##	"3.733"
##	"3.739"
##	"3.747"
##	"3.75"
##	"3.793"
##	"3.797"
##	"3.803"
##	"3.823"
##	"3.843"
##	"3.843"
##	"3.995"
##	"4.011"
##	"4.038"
##	"4.059"
##	"4.109"
##	"4.187"
##	"4.199"
##	"4.218"

```
##      "4.234"
##      "4.286"
##      "4.292"
##      "4.297"
##      "4.333"
##      "4.536"
##      "4.55"
##      "4.573"
##      "4.585"
##      "4.643"
##      "4.703"
##      "4.769"
##      "4.836"
##      "4.878"
##      "4.953"
##      "4.984"
##      "5.043"
##      "5.086"
##      "5.111"
##      "5.467"
##      "5.555"
##      "6.017"
##      "6.113"
##      "6.224"
##      "6.25"
##      "6.703"
##      "6.911"
##      "7.163"
##      "7.918"
##      "8.259"
##      "9.218"
##      "9.266"
##      "10.133"
##      "10.469"
##      "12.956"
##      "17.879"
```

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

```
###No suppressed terms have been found
```

```
#sup.list <- bp.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(bp.go.cat %in% sup.list[i])]))
#}
```

```
#print(sup)
```

```
#write.table(sup, file="human.pdcd1.bp.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*