yeast PIN Z-score, multi-labeled nodes, foreach

H Qin

11/5/2018

```
rm(list=ls())
debug = 1
largeGOFlag = 1;
#library(igraph)
library(foreach)
library(doMC)
```

```
## Loading required package: iterators
```

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

```
registerDoMC(5)

pairs= read.csv("Data/yeast.pin.csv", colClasses = c("character", "character"))
#if ( debug > 9) { pairs = pairs[1:10000, ]}
names(pairs) = c("name1", "name2")
```

Load yeast GO terms

```
YeastBP = read.csv("Data/yeast.bp.gene.term.csv", colClasses = c("character", "character", "character",
cats = YeastBP;  #cats is a general name
names(cats) = c("id", "gene", "sgd","type", "GO")
```

```
if( largeGOFlag > 0 ){ #pick large GO terms
  tmp = table( cats$GO)
  largeGOterms =as.character( names(tmp[ tmp >= mean(tmp)]) )
  cats$large_flag = ifelse( cats$GO %in% largeGOterms, 1, 0 )
  table(cats$large_flag)
  cats = cats[ cats$large_flag == 1, ]
}
```

function to find out all combinations of two vectors

```
allCombinationsOfTwoVectors = function (els1, els2 ) {
  tagbuffer = c();
  for (e1 in els1) {
    for (e2 in els2) {
         tmp = sort(c(e1, e2));
         current_tag = paste(tmp[1], tmp[2], sep="_")
         tagbuffer = c(tagbuffer, current_tag)
    }
  }
  return( tagbuffer)
}
```

```
allCombinationsOfTwoVectors( c("one", "two"), c("red", "blue", "orange"))
```

```
## [1] "one_red"    "blue_one"   "one_orange" "red_two"    "blue_two"
## [6] "orange_two"
```

```
x = allCombinationsOfTwoVectors( c("1", "2", "red"), c("red", "blue", "orange", "2"))
length(x)
```

```
## [1] 12
```

```
table(x)
```

```
## x
##        1_2     1_blue   1_orange      1_red        2_2     2_blue
##          1          1          1          1          1          1
##   2_orange      2_red   blue_red orange_red    red_red
##          1          2          1          1          1
```

```
start = Sys.time()
alltags = c()
for ( i in 1:length(pairs[,1])){
  sub1 = cats[ cats$id == pairs$name1[i], ]
  sub2 = cats[ cats$id == pairs$name2[i], ]
  els1 = sub1$GO
  els2 = sub2$GO
  if ( is.null(sub1) ) { els1 = c("NA") }
  if ( is.null(sub2) ) { els2 = c("NA") }
  tagbuffer = allCombinationsOfTwoVectors ( els1, els2 )  #all combinations
  alltags = c( alltags, tagbuffer) #combine with dataframe buffer
}
stop1 = Sys.time()
print(paste( "alltags.foreach runtime is ", stop1 - start, sep = " ") )
```

```
## [1] "alltags.foreach runtime is  3.64782598416011"
```

```
F.obs = data.frame(  table(as.character( alltags))  )
names(F.obs) = c("tag", "freq")
F.obs$tag = as.character(F.obs$tag)
F.obs = F.obs[ order(F.obs$tag), ]
stop2 = Sys.time()
print(paste( "Fobs runtime is ", stop2 - stop1, sep = " ") )
```

```
## [1] "Fobs runtime is  0.0504450798034668"
```

```
start = Sys.time()
alltags2 = c()
alltags2 = foreach(i=1:nrow(pairs), .combine = rbind ) %dopar% {
  #print(i)
  sub1 = cats[ cats$id == pairs$name1[i], ]
  sub2 = cats[ cats$id == pairs$name2[i], ]
  els1 = sub1$GO
  els2 = sub2$GO
  if ( is.null(sub1) ) { els1 = c("NA") }
  if ( is.null(sub2) ) { els2 = c("NA") }
  tagbuffer = allCombinationsOfTwoVectors ( els1, els2 )  #all combinations
  data.frame(tagbuffer)
}
```

```
stop1 = Sys.time()
print(paste( "alltags2 foreach runtime is ", stop1 - start, sep = " ") )
```

```
## [1] "alltags2 foreach runtime is  3.46106046438217"
```

```
F.obs.foreach  =  data.frame( table(as.character( alltags2[,1] ) ) )
names(F.obs.foreach) = c("tag", "freq")
F.obs.foreach$tag = as.character(F.obs.foreach$tag)
#F.obs.foreach = F.obs.foreach[ order(F.obs.foreach$tag), ]

stop2 = Sys.time()
print(paste( "F.obs.foreach table() runtime is ", stop2 - stop1, sep = " ") )
```

```
## [1] "F.obs.foreach table() runtime is  0.0435981750488281"
```

compare single core and multiple-core results. Passed.

```
table( F.obs$freq == F.obs.foreach$freq )
```

```
##
## TRUE
## 1423
```

```
table( F.obs$tag == F.obs.foreach$tag )
```

```
##
## TRUE
## 1423
```

```
#cbind( F.obs$tag, F.obs.foreach$tag )
```

Analyze MS02 null networks

```
ms02files = list.files(path='yeastMS02')
if (debug > 0 ) {ms02files = ms02files[1: 10] }
F.ms02 = data.frame(matrix(data=NA, nrow=1, ncol=3)) #null distributions
names(F.ms02) = c('tag', 'freq', 'file')
start = Sys.time()

for (file in ms02files ){
  start.file  = Sys.time()
  ms02_pairs= read.csv(paste("yeastMS02/", file, sep=''),
                       colClasses = c("character", "character"))
  ms02_pairs = ms02_pairs[,1:2]
  if ( debug > 5 ) {     ms02_pairs = ms02_pairs[1:1000, ]  }

  alltagsMS02 = c()
  alltagsMS02 = foreach(i=1:nrow(ms02_pairs), .combine = rbind ) %dopar% {
    sub1 = cats[ cats$id == ms02_pairs$id1[i], ]
    sub2 = cats[ cats$id == ms02_pairs$id2[i], ]
    els1 = sub1$GO
```

```
    els2 = sub2$GO
    if ( is.null(sub1) ) { els1 = c("NA") }
    if ( is.null(sub2) ) { els2 = c("NA") }
    tagbufferMS02 = allCombinationsOfTwoVectors ( els1, els2 )  #all combinations
    data.frame(tagbufferMS02)
  }
  stop.file = Sys.time()
  print(paste( "alltagsMS02 foreach single file runtime is ", stop.file - start.file, sep = " ") )

  F.ms02current = data.frame( table(alltagsMS02[,1]))
  F.ms02current$file = file
  names(F.ms02current) = c('tag', 'freq', 'file')
  F.ms02 =  data.frame( rbind(F.ms02, data.frame(F.ms02current)) )
}
```

```
## [1] "alltagsMS02 foreach single file runtime is  3.17934985160828"
## [1] "alltagsMS02 foreach single file runtime is  3.12119778394699"
## [1] "alltagsMS02 foreach single file runtime is  3.05151321490606"
## [1] "alltagsMS02 foreach single file runtime is  3.02785186767578"
## [1] "alltagsMS02 foreach single file runtime is  3.17542383273443"
## [1] "alltagsMS02 foreach single file runtime is  3.32671111424764"
## [1] "alltagsMS02 foreach single file runtime is  3.87613354921341"
## [1] "alltagsMS02 foreach single file runtime is  3.2539294163386"
## [1] "alltagsMS02 foreach single file runtime is  3.16472880045573"
## [1] "alltagsMS02 foreach single file runtime is  3.10835864941279"
```

```
F.ms02 = F.ms02[ !is.na(F.ms02$tag), ]
summary(F.ms02)
```

```
##      tag                freq            file
##  Length:14303       Min.   :   1.0   Length:14303
##  Class :character   1st Qu.:  62.0   Class :character
##  Mode  :character   Median : 143.0   Mode  :character
##                     Mean   : 252.2
##                     3rd Qu.: 316.0
##                     Max.   :3300.0
```

```
stop = Sys.time()
print(paste( "MS02 tag counts",length(ms02files), " files, runtime", stop - start, sep = " ") )
```

```
## [1] "MS02 tag counts 10  files, runtime 32.288107351462"
```