Application Qian 2001 method on Fournier2015 data (Z-normalized by genes)

s_morimoto

2016/07/06

This source code is built under R ver.3.1.2

You are using R version 3.2.2 (2015-08-14)

SETUP parameters

```
rm()

##--- setting : data  ##
dataDirectry   <- "/Users/mos/Dropbox/Draft_201603Morimoto/Analysis/Data"

dataPath_trc     <- "Gene2011_testResult_ana_20160706.xlsx"
                    #'Fournier_20151018_trc.xlsx'
                    #"Gene2011_testResult_ana.xlsx" #
dataPath_prt     <- "ProtSWATH_testResult_ana_20160706.xlsx"
                    #'Fournier_20151018_prt.xlsx'
                    #"ProtSWATH_testResult_ana.xlsx" #
dataPath_pathway <- "2015_Selevsek_proteins_KEGG.xlsx" #
bioproc <- 'pentose' #'GlySerThrMetabo' #'pentose'# #NULL#

typeBoolean  <- FALSE
sheet    = 1
sheetPathway = 1

header   = TRUE

startRow_trc = 1
startRow_prt = 1
startRow_pw  = 1

endRow_trc = 40027 #2296
endRow_prt = 15535 #2296
endRow_pw  = 54#54

timePoint <- 5
repet     <- 1

discrete_x_label <- c('inter-temporal')
continuous_y_label <- c('0min is 1')


cols <- c('0min','30min','60min','90min','120min')
        #c('min0','min20','min40','min60','min120','min240','min360')
        #c('T1-T1','T2-T1','T3-T1','T4-T1','T5-T1')

## end of setting : data  ---##
```

```
columnName <- c(rep(cols,repet))

## --- setting : data processing before analysis #

z_norm <- FALSE  ## Z-Normalize by genes and datanames ( TRUE/ FALSE)
interTemporal <- FALSE

## end of setting : data processing before analysis ---##

## --- setting : filenames of the OUTPUT FILEs ##

allOutput.prefix <- 'Selevsek_2015'

outputDirectry <-
  "/Users/mos/Dropbox/Draft_201603Morimoto/Analysis/Output"

file_dataGQ.prefix         <- 'dataGQ_Selevsek_20160706'
file_data_idConvert.prefix <- 'id.convert'
file_save.image     <- 'Selevsek_20160706.RData'

## end of setting : filenames of the OUTPUT FILEs ---##
```

```
 # Gene id convert (via bioMart)
#
#bioMartDB      <- "fungi_mart_29" # 2015.10.28 DB update                                     # biomaRt
#bioMartDataSet <- 'scerevisiae_eg_gene'
#                              # biomaRt::listDatasets(db)
#inputName      <- 'wikigene_name'   # 'wikigene_name' or 'ensembl_gene_id'
#                              # biomaRt::listFilters(sceg)
#outputName     <- 'uniprot_swissprot_accession'
#                              # biomaRt::listFilters(sceg)

bioMartHost    <- 'fungi.ensembl.org' # 2015.11.10 updated (HOST='biomart.org' has stopped)
bioMartDB      <- "fungal_mart" # 2015.11.10 DB update
                                  # biomaRt::listMarts(host=bioMartHost)
bioMartDataSet <- 'scerevisiae_eg_gene'
                                  # biomaRt::listDatasets(db)
inputName_1      <-  'ensembl_gene_id' # 'wikigene_name' or 'ensembl_gene_id'
                                  # biomaRt::listFilters(sceg)
outputName     <- 'uniprot_swissprot_accession'
                                  # biomaRt::listFilters(sceg)s
```

```
funcDirectry <- '/Users/mos/Dropbox/Draft_201603Morimoto/Analysis/PG/Functions'

EscoreCalc <- 'func_for_calcEscore_20160123.R'
```

LOAD PACKAGES

```
## Loading required package: plyr
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
```

```
##
##        'package:plyr'       :
##
##      arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
##        'package:stats'       :
##
##      filter, lag
##
##        'package:base'       :
##
##      intersect, setdiff, setequal, union
##
## Loading required package: tidyr
## Loading required package: xlsx
## Loading required package: rJava
## Loading required package: xlsxjars
## Loading required package: readxl
## Loading required package: ggplot2
## Loading required package: gplots
##
## Attaching package: 'gplots'
##
##        'package:stats'       :
##
##      lowess
##
## Loading required package: GMD
## Loading required package: pvclust
## Loading required package: reshape2
## Loading required package: pander
## Loading required package: stringr
## Loading required package: biomaRt
```

definition of functions

```
makeDifData <- function(data,var){
  for (i in 2:length(var)){
    end   <- var[i]
    start <- var[i-1]
    rescol <- paste('d',end,sep='_')
    ddata  <- data[,end] - data[,start]
    data[,rescol] <- ddata
  }
  return(data)
  }
```

Qian,Gerstein (J.Mol.Biol.,2001)

```
source(file = sprintf(fmt = '%s/%s',funcDirectry,EscoreCalc))

##--- Example : function gq_method ##
```

```r
gq_method(c(0,0,0,0,1,1,1,-1,1,1,1,1),c(0,0,0,0,1,1,1,1,1,1,1,-1),4,3,'pos')
```

```
## $mat
##    X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15
## 1   0  0  0  0  0  0  0  0  0   0   0   0   0   0   0
## 2   0  0  0  0  0  0  0  0  0   0   0   0   0   0   0
## 3   0  0  0  0  0  0  0  0  0   0   0   0   0   0   0
## 4   0  0  0  0  0  0  0  0  0   0   0   0   0   0   0
## 5   0  0  0  0  0  0  0  0  0   0   0   0   0   0   0
## 6   0  0  0  0  0  0  0  0  0   0   0   0   0   0   0
## 7   0  0  0  0  0  0  1  1  1   1   0   1   1   1   0
## 8   0  0  0  0  0  0  1  2  2   2   0   1   2   2   0
## 9   0  0  0  0  0  0  1  2  3   3   0   1   2   3   1
## 10  0  0  0  0  0  0  0  0  1   2   0   0   0   1   4
## 11  0  0  0  0  0  0  0  0  0   0   0   0   0   0   0
## 12  0  0  0  0  0  0  1  1  1   1   0   1   1   1   0
## 13  0  0  0  0  0  0  1  2  2   2   0   1   2   2   0
## 14  0  0  0  0  0  0  1  2  3   3   0   1   2   3   1
## 15  0  0  0  0  0  0  1  2  3   4   0   1   2   3   2
## 
## $index
## [1] 10 10
## 
## $score
## [1] 0 4
## 
## $subPosiDisVec
## [1] 0 0 0
## 
## $subMaxVec
## [1] 0 3 3
```

```r
gq_method(c(-1.597670,1.722744,1.699152,-2.219394,4.418399,2.081298,4.085573),
          c(-5.823882,9.529135,-5.602938,2.634425,3.772645,-8.844689,-3.819055),7,1,'neg')
```

```
## $mat
##   X1        X2        X3        X4        X5        X6       X7        X8
## 1  0  0.000000  0.000000  0.000000  0.000000  0.000000  0.00000  0.000000
## 2  0  0.000000 15.224413  0.000000  4.208942  6.027442  0.00000  0.000000
## 3  0 10.033058  0.000000 24.876841  0.000000  0.000000 21.26458  6.579254
## 4  0  9.895661  0.000000  9.520243 20.400552  0.000000 15.02847 27.753732
## 5  0  0.000000 31.044566  0.000000 15.367070 28.773538  0.00000  6.552483
## 6  0 25.732234  0.000000 55.800581  0.000000  0.000000 67.85290 16.874109
## 7  0 12.121234  5.899265 11.661384 50.317558  0.000000 18.40843 75.801495
## 8  0 23.793895  0.000000 28.790477  0.898248 34.904141 36.13562 34.011462
## 
## $index
## [1] 7 8
## 
## $score
## [1] -1.00000 75.80149
## 
```

```
## $subPosiDisVec
## [1] -1
##
## $subMaxVec
## [1] 75.80149
```

```
## end of Example : function gq_method ---##
```

LOAD DATA

```
raw_data.trc <- read_excel(
  sprintf(fmt = '%s/%s',
          dataDirectry,
          dataPath_trc
          ),
  sheet,
  col_names = header,
  col_types = NULL,
  na = "",
  skip = startRow_trc-1
  )
raw_data.prt <- read_excel(
  sprintf( fmt = '%s/%s',
           dataDirectry,
           dataPath_prt),
  sheet,
  col_names = header,
  col_types = NULL,
  na = "",
  skip = startRow_trc-1
  )
```

```
data.trc <- mutate(raw_data.trc,
                   dtname='trc'
                   ) %>%
  dplyr::select(id,var,val,dtname)

data.prt <- mutate(raw_data.prt,
                   dtname='prt'
                   ) %>%
  dplyr::select(id,var,val,dtname)

if (z_norm==TRUE){
  dataLong <- bind_rows(data.trc,data.prt) %>%
    rename(val2=val) %>%
    group_by(id,dtname) %>%  # " dataLong <- as.data.frame(dataLong) "
    mutate(val=scale(val2, center = TRUE, scale = TRUE)) %>%
    dplyr::select(-val2)
}else
  dataLong <- bind_rows(data.trc,data.prt)

attributes(dataLong$val) <- NULL
  # attrs are created by scale function which causes errors when this data treated as data.frame
```

```
dataLong <- as.data.frame(dataLong)
  # ungroup the BY-groups created by  " %>% group_by(id,dtname)) "
```

LOAD gene filtering DATA

```
data_ana <-
  dataLong

#data_ana$var <- factor(data_ana$var,levels=cols)

if(1-is.null(bioproc)){
  pathwayData   <- read.xlsx(
    sprintf(fmt = '%s/%s',
            dataDirectry,
            dataPath_pathway),
    sheetPathway,
    header=header,
    startRow=startRow_pw,
    endRow=endRow_pw,
    colIndex=1:3, dtname='pw') %>%
    filter(pathway==bioproc) %>%
    mutate(id=as.character(id),
           dtname=as.character(dtname),
           protein=as.character(protein)
           )
  data_ana <- inner_join(dataLong,
                         pathwayData %>% dplyr::select(-dtname)
                         ,by='id')
  summ_data_ana <- data_ana %>%
    group_by(dtname,pathway,var) %>%
    summarise(
      n=n(),mean=mean(val),sd=sd(val),
      min=min(val),median=median(val),max=max(val))
  pander(summ_data_ana)
  }else{
    summ_data_ana <- data_ana %>%
      group_by(dtname,var) %>%
      summarise(
        n=n(),mean=mean(val),sd=sd(val),
        min=min(val),median=median(val),max=max(val))
    pander(summ_data_ana)
    }
```

dtname	pathway	var	n	mean	sd	min	median	max
prt	pentose	0min	22	1	0	1	1	1
prt	pentose	120min	22	1.707	1.494	0.7108	1.217	6.907
prt	pentose	15min	22	1.154	0.3104	0.8267	1.058	1.904
prt	pentose	30min	22	1.472	1.074	0.8046	1.095	5.335
prt	pentose	60min	22	1.621	1.48	0.7556	1.173	6.943

dtname	pathway	var	n	mean	sd	min	median	max
prt	pentose	90min	22	1.71	1.505	0.7097	1.151	7.07
trc	pentose	0min	22	1	0	1	1	1
trc	pentose	120min	22	1.889	2.138	0.6643	1.036	8.815
trc	pentose	240min	22	1.624	1.381	0.7423	1.032	5.352
trc	pentose	30min	22	7.911	15.85	0.2606	0.9693	55.72
trc	pentose	60min	22	3.515	5.877	0.6373	1.072	21.41
trc	pentose	90min	22	1.872	1.971	0.6373	1.043	6.964

```r
w.timePoint <- timePoint

if(interTemporal==TRUE){
  difdata_ana <- data_ana
  difdata_ana <- makeDifData(difdata_ana%>%spread(key=var,value=val),cols) %>%
    dplyr::select(id,dtname,starts_with('d_')
           ) %>%
    gather(var,val,starts_with('d_'))
  data_ana <- difdata_ana
  timePoint <- timePoint-1
}
```

```r
if(interTemporal==TRUE){
  dataGQ <- data_ana %>%
    spread(key=var,value=val) %>%
    dplyr::select(id,dtname,starts_with("d_"))
  }else{
  dataGQ <- data_ana %>%
    spread(key=var,value=val) %>%
    dplyr::select(id,dtname,one_of(cols))
  }

write.csv(dataGQ,
          file=sprintf(fmt = '%s/%s_%s_output.csv',
                  outputDirectry,
                  allOutput.prefix,
                  file_dataGQ.prefix
                  )
          )
```

```r
# "XML content does not seem to be XML:"
# means "You are not connected to internet"

db.DL <- useMart(bioMartDB,host=bioMartHost) # listMarts(host=bioMartHost)
sceg <- useDataset(bioMartDataSet, mart = db.DL) # listDatasets(db)

id_convert <- getBM(
  attributes = c(inputName_1,outputName),
  filters =  c(inputName_1), # listFilters(sceg)
  values = dataGQ$id, #
  mart = sceg
  )
id_convert[,'id']    <- id_convert[,inputName_1]
id_convert[,'uniid'] <- id_convert[,outputName]

if(is.null(bioproc)) bioproc <- 'all'
write.csv(id_convert %>%
            dplyr::select(id,uniid),
          file=sprintf(fmt = '%s/%s_%s_%s.csv',
                 outputDirectry,
                 allOutput.prefix,
                 file_data_idConvert.prefix,
                 bioproc)
          )
```

Qian,Gerstein (J.Mol.Biol.,2001)

```
#gq_list_pos <- list()
#for(i in 1:length(unique(dataGQ$id))){
#  gene <- unique(dataGQ$id)[i]
#  data <- dataGQ %>%
#    filter(id==gene)
#    if(nrow(data)>=2){
#      gq_res <-  gq_method(data[1,3:length(data)],
#                           data[2,3:length(data)],
#                           timepoint=timePoint,
#                           rep=repet,'pos')
#      gq_list_i <- list()
#      gq_list_i[[1]] <- data
#      gq_list_i[[2]] <- gq_res
#      gq_list_pos[[i]] <- gq_list_i
#      }
#  }
#pander(gq_list_pos)
```

```
if(is.null(bioproc)) bioproc <- 'all'

save(
  data_ana,
  dataGQ,
  timePoint,
  repet,
  bioproc,
  discrete_x_label,
  continuous_y_label,
  cols,
  gq_method,
  file = sprintf(fmt = '%s/%s_%s_%s',
                 outputDirectry,
                 allOutput.prefix,
                 bioproc,
                 file_save.image
                 )
   )
```