R implementation

Sleiman Bassim, PhD

November 9, 2015

1 Loaded functions:

```
#source("/media/Data/Dropbox/humanR/01funcs.R")
rm(list=ls())
#setwd("/media/Data/Dropbox/humanR/PD/")
#setwd("~/Dropbox/humanR/PD/")
###load("PD.Rdata", .GlobalEnv)
#lsos(pat="")
```

2 Load packages.

[1] MUST install topGO package

```
pkgs <- c('topGO', 'dplyr','ggplot2', 'tidyr', 'clusterProfiler')
lapply(pkgs, require, character.only = TRUE)
```

3 Install **TopGO** from bioconductor here.

```
#source("http://bioconductor.org/biocLite.R")
#biocLite("topGO")
```

4 ## 1 Merge all GO terms by genome

```
aplke <- read.table("./data/merge/Aplke.txt")
dim(aplke)
auran <- read.table("./data/merge/auran.txt")
dim(auran)
aurli <- read.table("./data/merge/Aurli.txt")
dim(aurli)
fracy <- read.table("./data/merge/fracy.txt")
dim(fracy)
phatr <- read.table("./data/merge/Phatr.txt")
dim(phatr)
phyca <- read.table("./data/merge/Phyca.txt")
dim(phyca)
phyci <- read.table("./data/merge/Phyci.txt")
dim(phyci)
physo <- read.table("./data/merge/Physo.txt")
dim(physo)
pramo <- read.table("./data/merge/Pramorumv.txt")
dim(pramo)
psemu <- read.table("./data/merge/Psemu.txt")
dim(psemu)
schag <- read.table("./data/merge/Schag.txt")
dim(schag)
thaps <- read.table("./data/merge/Thaps.txt")
dim(thaps)
```

5 Merge all data.

```
Warning in outer_join_impl(x, y, by$x, by$y):  joining factors with different
```

```
levels, coercing to character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
Warning in outer_join_impl(x, y, by$x, by$y):  joining factor and character vector,
coercing into character vector
```

2 Shell scripts

Get the number of unique GO-terms from 13 out of 14 files. Without Auran. **There is 2965 unique GO terms between all genomes.**

> All shell script are issued in linux. If windows is used excel might work

```
find . -name "*tab" | xargs cut -f 5 | sort - | uniq | wc -l
```

Get the number of GO-terms in all 13 files. Redundancy between terms is included in the count. **There is 368,808 GO terms between genomes**

```
find . -name "*tab" | xargs cut -f 5  | wc -l
```

Extract the *gene names* and all the associated *GO terms* from Auran file. **There is 1834 unique GO terms and 6296 annotated genes in Auran.**

```
cat photo/Auran1_GO.tab | cut -f 1,5 | sed 's/ | /, /g' \
awk 'NR > 1' > photo/auran.geneID2GO.tab
```

Extract *gene names* and *GO terms* from all other files (x13).

```
cat Fracy1_goinfo_FilteredModels1.tab | cut -f 1,5 | awk 'NR > 1' > fracy.txt
```

Get a list of GOs that figure most in a genome.

```
cat Phatr2_bd_unmapped_goinfo_FilteredModels1.tab | cut -f5 | sort - | uniq -c \
 | sort - | awk '{ if ($1 >= 10) print $0}'

     10 GO:0006810
     10 GO:0007242
     11 GO:0006508
     13 GO:0007165
     13 GO:0008270
     16 GO:0008152
     17 GO:0016021
     18 GO:0016020
     22 GO:0005524
     25 GO:0003824
```

Get the list of genes with the previous list of most GOs.

```
cat Phatr2_bd_unmapped_goinfo_FilteredModels1.tab | cut -f5 | sort - | uniq -c \
 | sort - | awk '{ if ($1 >= 10) print $2}' > phatr2.go.txt

cat Phatr2_bd_unmapped_goinfo_FilteredModels1.tab | grep -Fwf phatr2.go.txt - | less

7       3736    ATP binding     molecular_function      GO:0005524
18      9320    membrane        cellular_component      GO:0016020
18      4927    transport       biological_process      GO:0006810
```

```
18      9321    integral to membrane    cellular_component      GO:0016021
29      3736    ATP binding     molecular_function      GO:0005524
36      9320    membrane        cellular_component      GO:0016020
52      9321    integral to membrane    cellular_component      GO:0016021
116     3736    ATP binding     molecular_function      GO:0005524
```

2.1 Create function to automate Fishers test

Create a function.

↰ This function here can be copy/pasted in any R terminal. But first install topGO library

```
doFisher <- function(data, ontology,
                     algorithm="classic",
                     statistic="fisher",
                     table=20,
                     n=3){
# function to find the most interesting GO terms
    df <- readMappings(file = data)
    df <- inverseList(df)
    df <- inverseList(df)
# create matrix
# matrix must be a vector with 2 factors
# that is: interesting and not interesting genes
    geneNames <- names(df)
    geneList <- factor(as.integer(geneNames %in% geneNames[-1]))
    names(geneList) <- geneNames
    GOdata <- new("topGOdata",
              ontology = ontology,
              allGenes = geneList,
              annot = annFUN.gene2GO,
              gene2GO = df)
# Run Fishers test
# show table of significant GO terms
    results <- runTest(GOdata, algorithm=algorithm, statistic=statistic)
    maps <- GenTable(GOdata,
                     classicFisher=results,
                     orderBy="classicFisher",
                     topNodes=table)
    return(maps[, c(1:n)])
}
```

3 Photo: Auran

Read in data. Create list of genes and their corresponding GO-terms.

```
auran <- readMappings(file = "./data/photo/auran.geneID2GO.txt")
geneNames <- names(auran)
length(geneNames)

[1] 6296
```

Create an analysis matrix.

```
geneList <- factor(as.integer(geneNames %in% geneNames[-1]))
names(geneList) <- geneNames
GOdata <- new("topGOdata",
              ontology = "MF",
              allGenes = geneList,
              annot = annFUN.gene2GO,
              gene2GO = auran)


Building most specific GOs ..... ( 1180 GO terms found. )

Build GO DAG topology .......... ( 1562 GO terms and 1917 relations. )

Annotating nodes ............... ( 5650 genes annotated to the GO terms. )
```

Run Fishers Exact test.

```
resultFis <- runTest(GOdata, algorithm="classic", statistic="fisher")


 -- Classic Algorithm --

 the algorithm is scoring 1562 nontrivial nodes
 parameters:
 test statistic:  fisher
```

Show table.

```
Gomaps <- GenTable(GOdata,
                   classicFisher=resultFis,
                   orderBy="classicFisher",
                   topNodes=20)
Gomaps[,c(1:3)]
```

```
        GO.ID                                        Term Annotated
1  GO:0016740                      transferase activity      1293
2  GO:0003676                      nucleic acid binding       800
3  GO:0016491                   oxidoreductase activity       660
4  GO:0043169                             cation binding       656
5  GO:0046872                         metal ion binding       656
6  GO:0016772 transferase activity, transferring phosp...       640
7  GO:0005215                      transporter activity       496
8  GO:0016301                           kinase activity       470
9  GO:0016773 phosphotransferase activity, alcohol gro...       445
10 GO:0003677                               DNA binding       400
11 GO:0016887                           ATPase activity       381
12 GO:0004672                   protein kinase activity       372
13 GO:0022857          transmembrane transporter activity       359
14 GO:0022892     substrate-specific transporter activity       341
15 GO:0005509                      calcium ion binding       330
16 GO:0016788 hydrolase activity, acting on ester bond...       329
17 GO:0004674    protein serine/threonine kinase activity       328
18 GO:0008233                       peptidase activity       328
19 GO:0022891 substrate-specific transmembrane transpo...       322
20 GO:0070011 peptidase activity, acting on L-amino ac...       320
```

4 Diatoms

4.1 FRACY

Cellular components

```
doFisher(data = "./data/diatome/fracy.txt",
```

```
        ontology = "CC", algorithm = "classic",
        statistic = "fisher", table = 20, n = 3)


Building most specific GOs ..... ( 161 GO terms found. )

Build GO DAG topology .......... ( 327 GO terms and 694 relations. )

Annotating nodes ............... ( 3388 genes annotated to the GO terms. )

 -- Classic Algorithm --

 the algorithm is scoring 327 nontrivial nodes
 parameters:
 test statistic:  fisher
        GO.ID                                        Term Annotated
1  GO:0016020                                    membrane      1490
2  GO:0032991                       macromolecular complex       929
3  GO:0005737                                   cytoplasm       793
4  GO:0044425                                membrane part       709
5  GO:0043234                              protein complex       629
6  GO:0031224             intrinsic component of membrane       572
7  GO:0016021              integral component of membrane       566
8  GO:0044444                           cytoplasmic part       539
9  GO:0043228              non-membrane-bounded organelle       406
10 GO:0043232 intracellular non-membrane-bounded organ...       403
11 GO:0044422                             organelle part       397
12 GO:0044446               intracellular organelle part       397
13 GO:0030529                  ribonucleoprotein complex       286
14 GO:1902494                           catalytic complex       222
15 GO:0005840                                   ribosome       214
16 GO:0012505                          endomembrane system       121
17 GO:0031090                           organelle membrane       117
18 GO:0005694                                   chromosome       115
19 GO:0005739                                mitochondrion        88
20 GO:0044427                           chromosomal part        85
```

4.2 Phatr2 unmapped

Cellular component.

```
doFisher(data = "./data/diatome/Phatr2.unmapped.txt",
```

```
        ontology = "CC", algorithm = "classic",
        statistic = "fisher", table = 20, n = 3)


Building most specific GOs ..... ( 20 GO terms found. )

Build GO DAG topology .......... ( 56 GO terms and 101 relations. )

Annotating nodes ............... ( 65 genes annotated to the GO terms. )

 -- Classic Algorithm --

 the algorithm is scoring 56 nontrivial nodes
 parameters:
 test statistic:  fisher
        GO.ID                                        Term Annotated
1  GO:0000015           phosphopyruvate hydratase complex         2
2  GO:0000502                             proteasome complex         3
3  GO:0005575                             cellular_component        65
4  GO:0005622                                  intracellular        37
5  GO:0005623                                           cell        37
6  GO:0005634                                         nucleus         8
7  GO:0005737                                       cytoplasm        17
8  GO:0005739                                     mitochondrion         3
9  GO:0005740                          mitochondrial envelope         3
10 GO:0005743                mitochondrial inner membrane         1
11 GO:0005829                                        cytosol         2
12 GO:0005839                      proteasome core complex         3
13 GO:0005840                                      ribosome         2
14 GO:0005856                                   cytoskeleton         2
15 GO:0005950                anthranilate synthase complex         2
16 GO:0005963 magnesium-dependent protein serine/threo...         1
17 GO:0008287 protein serine/threonine phosphatase com...         1
18 GO:0009317              acetyl-CoA carboxylase complex         1
19 GO:0009331 glycerol-3-phosphate dehydrogenase compl...         1
20 GO:0015629                             actin cytoskeleton         2
```

4.3 Phatr2 chromosomes

Cellular components.

```
doFisher(data = "./data/diatome/Phatr2.chromosomes.txt",
```

```
        ontology = "CC", algorithm = "classic",
        statistic = "fisher", table = 20, n = 3)


Building most specific GOs ..... ( 122 GO terms found. )

Build GO DAG topology .......... ( 270 GO terms and 566 relations. )

Annotating nodes ............... ( 1496 genes annotated to the GO terms. )

 -- Classic Algorithm --

 the algorithm is scoring 270 nontrivial nodes
 parameters:
 test statistic:  fisher
        GO.ID                                Term Annotated
1  GO:0000015    phosphopyruvate hydratase complex         2
2  GO:0000139                       Golgi membrane         9
3  GO:0000148   1,3-beta-D-glucan synthase complex         1
4  GO:0000151             ubiquitin ligase complex        74
5  GO:0000152     nuclear ubiquitin ligase complex         2
6  GO:0000159  protein phosphatase type 2A complex         1
7  GO:0000228                  nuclear chromosome         3
8  GO:0000313                organellar ribosome         1
9  GO:0000323                     lytic vacuole         1
10 GO:0000428  DNA-directed RNA polymerase complex         4
11 GO:0000502                  proteasome complex        24
12 GO:0000781         chromosome, telomeric region         1
13 GO:0000784 nuclear chromosome, telomeric region         1
14 GO:0000785                           chromatin        23
15 GO:0000786                         nucleosome        17
16 GO:0000922                       spindle pole         2
17 GO:0002178         palmitoyltransferase complex         2
18 GO:0005575                  cellular_component      1496
19 GO:0005576              extracellular region         6
20 GO:0005622                     intracellular       951
```

4.4 Thaps

```
doFisher(data = "./data/diatome/thaps3.txt",
```

```
        ontology = "CC", algorithm = "classic",
        statistic = "fisher", table = 20, n = 3)


Building most specific GOs ..... ( 126 GO terms found. )

Build GO DAG topology .......... ( 274 GO terms and 576 relations. )

Annotating nodes ............... ( 1751 genes annotated to the GO terms. )

 -- Classic Algorithm --

 the algorithm is scoring 274 nontrivial nodes
 parameters:
 test statistic:  fisher
        GO.ID                                Term Annotated
1  GO:0000015   phosphopyruvate hydratase complex         2
2  GO:0000139                      Golgi membrane        12
3  GO:0000145                             exocyst         1
4  GO:0000148  1,3-beta-D-glucan synthase complex         1
5  GO:0000151            ubiquitin ligase complex        80
6  GO:0000152    nuclear ubiquitin ligase complex         1
7  GO:0000159 protein phosphatase type 2A complex         1
8  GO:0000228                 nuclear chromosome         1
9  GO:0000313               organellar ribosome         1
10 GO:0000323                     lytic vacuole         1
11 GO:0000428 DNA-directed RNA polymerase complex         5
12 GO:0000502                  proteasome complex        30
13 GO:0000775      chromosome, centromeric region         1
14 GO:0000785                        chromatin        40
15 GO:0000786                      nucleosome        31
16 GO:0000922                    spindle pole         2
17 GO:0002178       palmitoyltransferase complex         3
18 GO:0005575                cellular_component      1751
19 GO:0005576              extracellular region        50
20 GO:0005622                     intracellular      1125
```

5 Clustering of GO terms

Select one GO term from a Cellular Component analysis and use it to group all genes under that term by BP and MF. The script below will do just that.

1. Get genes related to 1 CC GO term
2. Get BP GOs for all these genes
3. Select BP GOs with at least 10 genes
4. Restructure and label data

```
cat Fracy1_goinfo_FilteredModels1.tab | grep "GO:0016020" | \
awk '{print $1}' | grep -Fwf - Fracy1_goinfo_FilteredModels1.tab | \
grep "biological_process" | cut -f 3 | sort - | uniq -c | \
awk '{if ($1 >= 10) print $0}' | sort -n | \
perl -pe 's/([0-9]) /\1\t/g' | \
awk '{print $0,"\t","GO:0016020","\t","membrane","\t","fracy","\t","1"}' | nl -
```

6 System Information

The version number of R and packages loaded for generating the vignette were:

```
###save(list=ls(pattern=".*|.*"),file="PD.Rdata")
```

```
sessionInfo()

R version 3.2.1 (2015-06-18)
Platform: x86_64-unknown-linux-gnu (64-bit)
Running under: elementary OS Luna

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats4    parallel  stats     graphics  grDevices utils
[7] datasets  methods   base

other attached packages:
 [1] clusterProfiler_2.2.7 tidyr_0.2.0           ggplot2_1.0.1
 [4] dplyr_0.4.2           topGO_2.20.0          SparseM_1.6
 [7] GO.db_3.1.2           RSQLite_1.0.0         DBI_0.3.1
[10] AnnotationDbi_1.30.1  GenomeInfoDb_1.4.3    IRanges_2.2.9
[13] S4Vectors_0.6.6       Biobase_2.28.0        graph_1.46.0
[16] BiocGenerics_0.14.0   knitr_1.10.5          RevoUtilsMath_3.2.1

loaded via a namespace (and not attached):
 [1] Rcpp_0.11.6       XVector_0.8.0     formatR_1.2
 [4] highr_0.5         plyr_1.8.3        zlibbioc_1.14.0
 [7] tools_3.2.1       digest_0.6.8      DOSE_2.6.6
[10] evaluate_0.7      gtable_0.1.2      lattice_0.20-31
[13] png_0.1-7         igraph_1.0.1      proto_0.3-10
[16] httr_1.0.0        stringr_1.0.0     Biostrings_2.36.4
[19] grid_3.2.1        qvalue_2.0.0      R6_2.0.1
[22] GOSemSim_1.26.0   DO.db_2.9         reshape2_1.4.1
[25] magrittr_1.5      splines_3.2.1     scales_0.2.5
[28] MASS_7.3-41       KEGGREST_1.8.1    assertthat_0.1
[31] colorspace_1.2-6  stringi_0.5-5     munsell_0.4.2
```