An introduction to FSim

Qiang Hu, Song Liu

November 14, 2013

Contents

1 Introduction

FSim is an R package to search functionally similar (FSim) genes for objective gene, a set of functional keywords or a biological pathway. The function compare the functional relation between genes based on the Gene Ontology annotation. A new algorithm is proposed to analyze the relation between genes based on the GO annotation of genes. Our package is able to search the most functionally similar genes by comparing the GO terms between genes.

```
> library(FSim)
```

```
groupGOTerms:         GOBPTerm, GOMFTerm, GOCCTerm environments built.
```

2 Similar score calculation

2.1 Comparison between two genes

A algorithm is proposed to compare the functionally similarity between two genes. Given two gene IDs, the function `calSim` can calculate the similar scores based on annotated GO terms in different ontology or terms from "ALL" ontologies. For example, the similarity between human gene "9" and "10" can be calculated as follows.

```
> calSim(g1="9", g2="10", ontology="BP", an.go=an.Hs.egGO)

  value     ASE z.value
      1       0     Inf

> calSim(g1="9", g2="10", ontology="ALL", an.go=an.Hs.egGO)

      value         ASE     z.value
 0.81383985  0.06884736 11.82093042
```

2.2 Comparison between gene and GO terms

Given a gene and a group of GO terms, the functional relation can also be calculated.

```
> terms <- names(get("10", org.Hs.egGO))
> calSim(g1="9", tids=terms, an.go=an.Hs.egGO)

      value         ASE     z.value
 0.96832224  0.03097859 31.25779241
```

3 Search example

3.1 Search by gene

The function `SearchGene` can be used to search functionally similar genes for an objective gene. For example, we can use the function to find the most funcionally related genes for an objective gene "NAT2" in the GO annotation database.

```
> SearchGene(symbol="NAT2", an.go=an.Hs.egGO, targets="ALL", n=10)

       Symbol sharedTerms     value        ASE   z.value
9        NAT1           4 0.9683222 0.03097859 31.257792
126     ADH1C           3 0.5145827 0.10940881  4.703303
125     ADH1B           3 0.5021795 0.10945535  4.587985
119391  GSTO2           3 0.4838813 0.10641653  4.547050
130      ADH6           3 0.4850249 0.10940879  4.433144
124     ADH1A           3 0.4167641 0.10793765  3.861156
10380   BPNT1           3 0.3892837 0.10675123  3.646644
127      ADH4           3 0.2792058 0.09823665  2.842175
131      ADH7           3 0.2428469 0.09390942  2.585969
1312     COMT           3 0.2016357 0.08548145  2.358824
```

The option n is set to 10, then the function return 10 functionally related genes ordered by Z values. The search database can be set by the targets option, which can be "ALL" to search in all GO annotated genes and abso can be a set of customized genes to specify search range. For example, the function between gene "9" and a gene set can be compared d by the option "targets".

```
> SearchGene(gene="9", targets=c("10", "100", "124"),
+            an.go=an.Hs.egGO)

    Symbol sharedTerms      value        ASE    z.value
10    NAT2           4 0.79272288 0.07169913 11.0562406
124  ADH1A           3 0.41676414 0.10793765  3.8611564
100    ADA           2 0.04848653 0.05405582  0.8969715
```

3.2 Search by GO terms

A group of GO terms can aslo be used to search functionally related genes. For exmaple,

```
> t1 <- names(get("9", org.Hs.egGO))
> t1

[1] "GO:0006805" "GO:0044281" "GO:0005829" "GO:0004060"

> SearchGene(terms=t1, an.go=an.Hs.egGO, n=5)

       Symbol sharedTerms     value        ASE   z.value
9        NAT1           4 0.9683222 0.03097859 31.257792
10       NAT2           4 0.7927229 0.07169913 11.056241
119391  GSTO2           3 0.4838813 0.10641653  4.547050
124     ADH1A           3 0.4167641 0.10793765  3.861156
10380   BPNT1           3 0.3892837 0.10675123  3.646644
```

3.3 Search by keywords

The function can also be used to analyze the functionally similar genes with a group of biological keywords. For example, we try to search for genes related with function "chromatin remodeling" and "histone binding".

```
> t2 <- SearchTerm(fun=c("chromatin remodeling", "histone binding"))
> t2

        GOID Ontology                                                Term
1 GO:0006338       BP                                chromatin remodeling
2 GO:0031055       BP                chromatin remodeling at centromere
4 GO:0043044       BP                ATP-dependent chromatin remodeling
5 GO:0043156       BP chromatin remodeling in response to cation stress
3 GO:0031011       CC                                        Ino80 complex
6 GO:0031493       MF                        nucleosomal histone binding
7 GO:0042393       MF                                    histone binding
```

Then we select part of the returned GO terms to search function related genes.

```
> SearchGene(terms=t2$GOID[c(1,3,6)], an.go=an.Hs.egGO, n=5)

        Symbol sharedTerms      value        ASE   z.value
6594   SMARCA1           2 0.19106779 0.10368965 1.8426891
8467   SMARCA5           2 0.11672897 0.08034512 1.4528445
56916 SMARCAD1           2 0.11354577 0.09762054 1.1631341
6598   SMARCB1           2 0.10045014 0.09250804 1.0858531
10014    HDAC5           1 0.06268819 0.07465530 0.8397018
```

3.4 Search by gene set

In order to search functionally similar genes for a gene set, we need summary the objective gene set to a group of GO terms first. GO over-represent analysis can be used to discover major biological functions for a gene set. The `ovreGO` integrated functions from *topGO* can be used to find over-represented GO terms for a gene set. For example, we have a gene set from KEGG database.

```
> library(KEGG.db)
> geneset <- get("hsa00232", KEGGPATHID2EXTID)
> geneset

[1] "10"   "1544" "1548" "1549" "1553" "7498" "9"
```

The function `ovreGO` can be used to find major represent terms. All human genes from KEGG database are used as backgroud.

```
> paths <- as.list(KEGGPATHID2EXTID)
> paths <- paths[grep("^hsa", names(paths))]
> allgenes <- unique(unlist(paths))
> BPterms <- ovreGO(genes=geneset, allgenes=allgenes,
+                   ontology="BP", nterm=10)

Building most specific GOs .....        ( 7290 GO terms found. )

Build GO DAG topology ..........        ( 10717 GO terms and 24506 relations. )

Annotating nodes ...............        ( 5341 genes annotated to the GO terms. )

                        -- Parent-Child Algorithm --

                the algorithm is scoring 310 nontrivial nodes
                parameters:
                        test statistic:  fisher : joinFun = union

        Level 13:        1 nodes to be scored.

        Level 12:        5 nodes to be scored.

        Level 11:        7 nodes to be scored.

        Level 10:        14 nodes to be scored.
```

```
        Level 9:        22 nodes to be scored.

        Level 8:        29 nodes to be scored.

        Level 7:        34 nodes to be scored.

        Level 6:        48 nodes to be scored.

        Level 5:        66 nodes to be scored.

        Level 4:        47 nodes to be scored.

        Level 3:        25 nodes to be scored.

        Level 2:        11 nodes to be scored.
```

```
> BPterms
```

```
        GO.ID                                       Term Annotated Significant
1  GO:0006805               xenobiotic metabolic process       128           5
2  GO:0009410            response to xenobiotic stimulus       129           5
3  GO:0071466   cellular response to xenobiotic stimulus       128           5
4  GO:0042737                      drug catabolic process        10           2
5  GO:0009403                 toxin biosynthetic process         1           1
6  GO:0017144                      drug metabolic process        27           2
7  GO:0042738           exogenous drug catabolic process         8           2
8  GO:1900746 regulation of vascular endothelial growt...         1           1
9  GO:0071615                      oxidative deethylation         1           1
10 GO:0019748                secondary metabolic process        28           2
   Expected result1    Score   wScore
1      0.14 2.5e-07 6.602335 5.148161
2      0.14 7.4e-06 5.132325 2.286814
3      0.14 5.0e-05 4.297514 2.872269
4      0.01 0.00023 3.629172 2.484255
5      0.00 0.00047 3.323665 2.141917
6      0.03 0.00073 3.133715 1.593601
7      0.01 0.00084 3.076731 2.637198
8      0.00 0.00098 3.009876 2.736251
9      0.00 0.00254 2.594393 2.594393
10     0.03 0.00264 2.578419 1.211618
```

The results from `ovreGO` show the most over-represented terms ordered by p values. Top 10 GO terms are used to stand for the the major biological functions of the gene set.

```
> SearchGene(terms=BPterms$GO.ID, targets="ALL",
+            an.go=an.Hs.egGO, n=5)
```

```
     Symbol sharedTerms     value        ASE  z.value
1544 CYP1A2           6 0.2763440 0.08658507 3.191590
```

```
1555 CYP2B6           3 0.2892603 0.10908530 2.651689
1548 CYP2A6           3 0.2596739 0.10542761 2.463054
1559 CYP2C9           4 0.2281457 0.10100676 2.258718
1576 CYP3A4           4 0.2039875 0.09571657 2.131162
```

The returned genes are all from "CYP" gene family because most of the interested gene set are also from this family.

4 Evaluation

The functionally related genes should get higer similar scores as our method proposed. Here we use a set of genes from KEGG pathway to simply evaluate our method. First, we use GO over-represented algorithm to find the major functions of the gene set. Then the over-represented GO terms are used to calculate the similar scores with the gene set and other randomly selected genes using our method.

```
> MFterms <- ovreGO(genes=geneset, allgenes=allgenes,
+                   ontology="MF", nterm=10)

Building most specific GOs .....        ( 2798 GO terms found. )

Build GO DAG topology ..........        ( 3291 GO terms and 4043 relations. )

Annotating nodes ...............        ( 5604 genes annotated to the GO terms. )

                        -- Parent-Child Algorithm --

                the algorithm is scoring 54 nontrivial nodes
                parameters:
                        test statistic:  fisher : joinFun = union

          Level 8:        1 nodes to be scored.

          Level 7:        2 nodes to be scored.

          Level 6:        9 nodes to be scored.

          Level 5:        13 nodes to be scored.

          Level 4:        14 nodes to be scored.

          Level 3:        11 nodes to be scored.

          Level 2:        3 nodes to be scored.

> CCterms <- ovreGO(genes=geneset, allgenes=allgenes,
+                   ontology="CC", nterm=10)
```

```
Building most specific GOs .....        ( 927 GO terms found. )

Build GO DAG topology ..........        ( 1129 GO terms and 2200 relations. )

Annotating nodes ...............        ( 5671 genes annotated to the GO terms. )

                        -- Parent-Child Algorithm --

                the algorithm is scoring 34 nontrivial nodes
                parameters:
                        test statistic:  fisher : joinFun = union

        Level 11:        1 nodes to be scored.

        Level 10:        2 nodes to be scored.

        Level 9:        4 nodes to be scored.

        Level 8:        4 nodes to be scored.

        Level 7:        4 nodes to be scored.

        Level 6:        2 nodes to be scored.

        Level 5:        2 nodes to be scored.

        Level 4:        3 nodes to be scored.

        Level 3:        6 nodes to be scored.

        Level 2:        5 nodes to be scored.
```

```
> allterms <- c(BPterms$GO.ID, MFterms$GO.ID, CCterms$GO.ID)
```

The gene set from the pathway is a group of genes related in biological process, molecular function and cellular component, so GO terms from the three ontologies are used. The `allterms` are the over-represented GO terms to do the comparisons.

```
> score1 <- SearchGene(terms=allterms, targets=geneset,
+                      an.go=an.Hs.egGO, ontology="ALL",
+                      term2ancestor=FALSE)
> score1
```

```
      Symbol sharedTerms      value        ASE  z.value
1548  CYP2A6           8 0.48132473 0.09157462 5.256093
1553 CYP2A13           4 0.50315919 0.10053025 5.005053
1544  CYP1A2          12 0.37643401 0.08242933 4.566748
1549  CYP2A7           3 0.46774831 0.10900862 4.290930
9       NAT1           2 0.22369629 0.14357073 1.558091
```

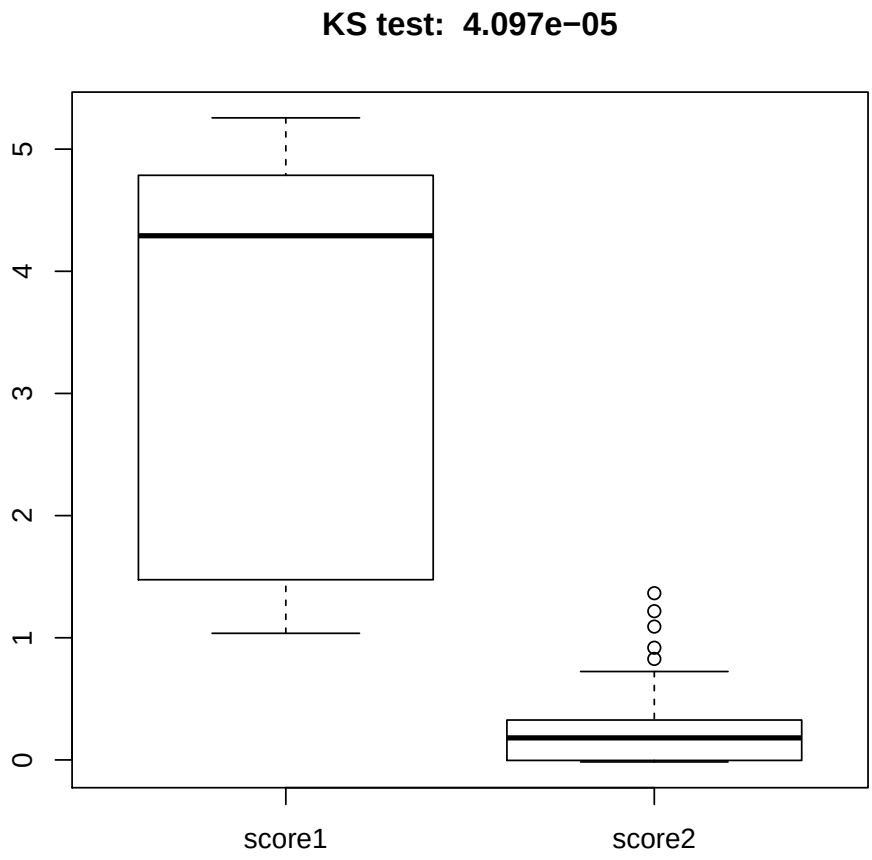
```
10      NAT2           2 0.18714047 0.13443331 1.392069
7498     XDH           2 0.07473017 0.07209355 1.036572
```

Then we randomly select 100 genes as control group to calculate the similar scores.

```
> set.seed(1)
> ctlgene <- sample(setdiff(allgenes, geneset), 50)
> score2 <- SearchGene(terms=allterms, targets=ctlgene,
+                      an.go=an.Hs.egGO, ontology="ALL",
+                      term2ancestor=FALSE)
> head(score2)

       Symbol sharedTerms      value        ASE   z.value
126129  CPT1C           0 0.15454810 0.11321277 1.3651118
23225  NUP210           1 0.13995903 0.11493444 1.2177292
3040     HBA2           1 0.11925483 0.10925182 1.0915592
1080     CFTR           0 0.07690279 0.08371221 0.9186567
8540     AGPS           0 0.10011470 0.12107255 0.8268984
125965 COX6B2           0 0.09894737 0.13672301 0.7237067

> pv <- suppressWarnings(ks.test(score1$z.value, score2$z.value))
> pv

        Two-sample Kolmogorov-Smirnov test

data:  score1$z.value and score2$z.value
D = 0.9388, p-value = 4.097e-05
alternative hypothesis: two-sided
```

The `ks.test` shows the two scores are significantly different. The scores from the `geneset` are significantly higer than the randomly selected genes. A boxplot can be used to show the detailed distribution of the two groups of scores.

```
> boxplot(score1$z.value, score2$z.value,
+         names=c("score1", "score2"),
+         main=paste("KS test: ", format(pv$p.value, digits=4)))
```

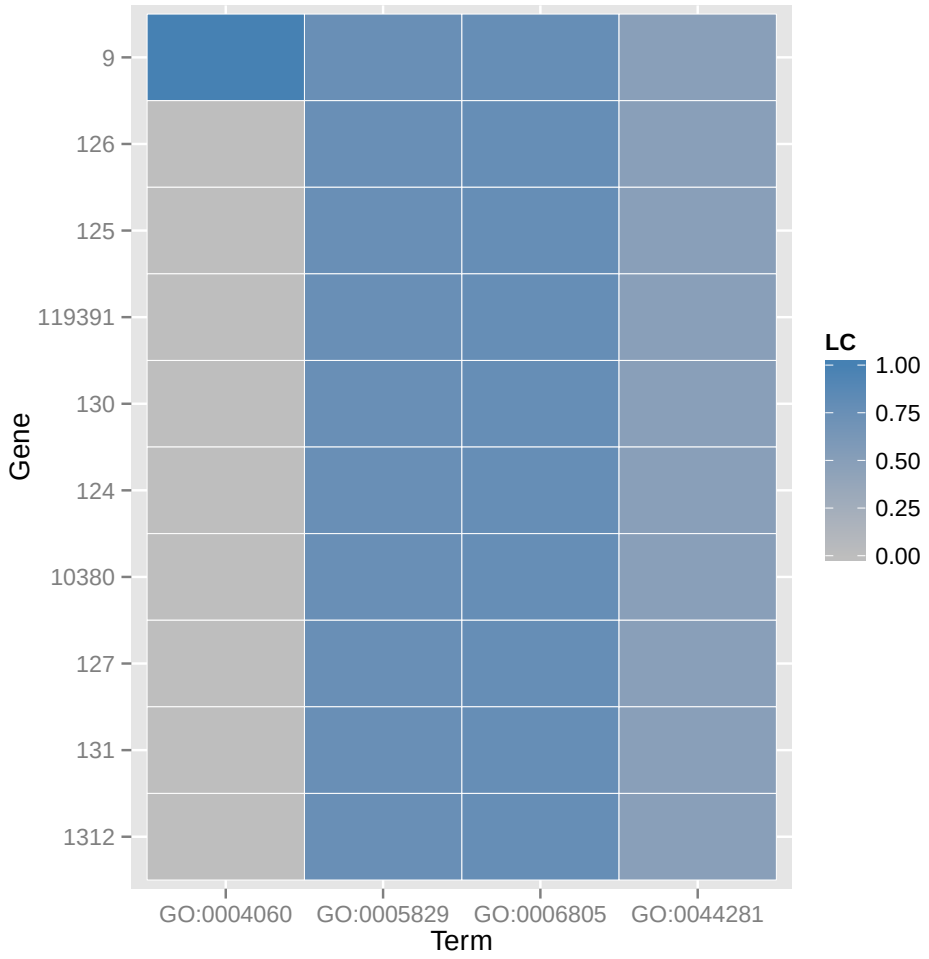


5 Visualization

5.1 Heatmap

Basically, functionally related genes share part of GO terms. The search results only show the number of shared terms. The details can be plotted with heatmap.

```
> res1 <- SearchGene(symbol="NAT2", an.go=an.Hs.egGO,
+                    targets="ALL", n=10, plot=TRUE)
```

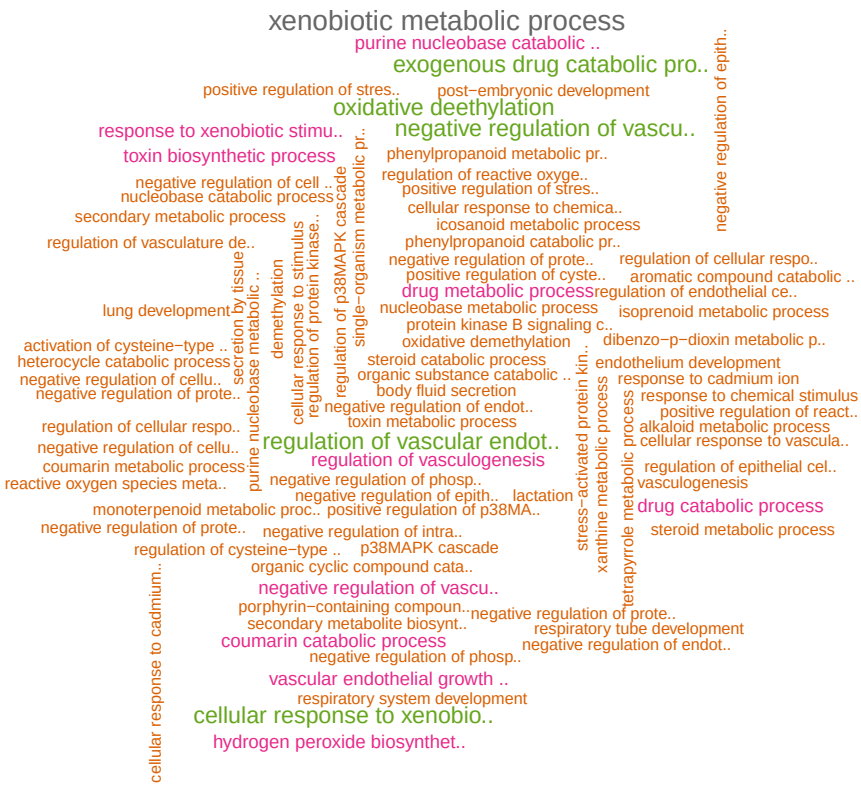


5.2 Wordcloud

The GO over-represent analysis return the major biological functions of a gene set. The results can also be visualized by word cloud plot. The function `ovreGO` can also be used to plot the wordcloud with the option "plot=TRUE".

```
> res2 <- ovreGO(genes=geneset, allgenes=allgenes,
+                ontology="BP", plot=TRUE, scale=c(1,0.5))

Building most specific GOs .....        ( 7290 GO terms found. )

Build GO DAG topology ..........        ( 10717 GO terms and 24506 relations. )

Annotating nodes ...............        ( 5341 genes annotated to the GO terms. )

                        -- Parent-Child Algorithm --

                the algorithm is scoring 310 nontrivial nodes
                parameters:
                        test statistic:  fisher : joinFun = union
```

```
Level 13:        1 nodes to be scored.

Level 12:        5 nodes to be scored.

Level 11:        7 nodes to be scored.

Level 10:        14 nodes to be scored.

Level 9:        22 nodes to be scored.

Level 8:        29 nodes to be scored.

Level 7:        34 nodes to be scored.

Level 6:        48 nodes to be scored.

Level 5:        66 nodes to be scored.

Level 4:        47 nodes to be scored.

Level 3:        25 nodes to be scored.

Level 2:        11 nodes to be scored.
```



6 Session information

```
> sessionInfo()

R version 3.0.2 (2013-09-25)
Platform: x86_64-pc-linux-gnu (64-bit)

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [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] grid      parallel  stats     graphics  grDevices utils     datasets
[8] methods   base
```

```
other attached packages:
 [1] FSim_0.1.6           reshape2_1.2.2       ggplot2_0.9.3.1
 [4] wordcloud_2.4        RColorBrewer_1.0-5   Rcpp_0.10.6
 [7] vcd_1.3-1            topGO_2.14.0         SparseM_1.03
[10] graph_1.40.0         KEGG.db_2.10.1       org.Hs.eg.db_2.10.1
[13] GO.db_2.10.1         RSQLite_0.11.4       DBI_0.2-7
[16] AnnotationDbi_1.24.0 Biobase_2.22.0       BiocGenerics_0.8.0

loaded via a namespace (and not attached):
 [1] IRanges_1.20.5   MASS_7.3-29      colorspace_1.2-4 dichromat_2.0-0
 [5] digest_0.6.3     gtable_0.1.2     labeling_0.2     lattice_0.20-24
 [9] munsell_0.4.2    plyr_1.8         proto_0.3-10     scales_0.2.3
[13] slam_0.1-30      stats4_3.0.2     stringr_0.6.2    tools_3.0.2
```