clusterProfiler: an R package for Statistical Analysis and Visulization of Functional Profiles for Genes and Gene Clusters

Guangchuang Yu

March 25, 2011

1 Introduction

In recently years, high-throughput experimental techniques such as microarray and mass spectrometry can identify many lists of genes and gene products. The most widely used strategy for high-throughput data analysis is to identify different gene clusters based on their expression profiles. Another commonly used approach is to annotate these genes to biological knowledge, such as Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG), and identify the statistically significantly enriched categories. These two different strategies were implemented in many bioconductor packages, such as *Mfuzz* and *BHC* for clustering analysis and *GOstats* for GO enrichment analysis.

After clustering analysis, researchers not only want to determine whether there is a common theme to a particular gene cluster, but also would like to compare the biological themes among gene clusters, which have different expression profiles. There is no existing tools to bridge this gap, and we designed *clusterProfiler*, for comparing functional profiles among gene clusters.

This document presents an introduction to the use of *clusterProfiler*, an R package for the analysis of lists of genes and gene clusters based on their GO annotation distribution or enrichment categories of GO and KEGG, and provides methods for visulization.

2 Quick start

The following lines provide a quick and simple example on the use of *clusterProfiler* to explore gene list and compare gene clusters.

The analysis proceeds as follows:

- First a sample dataset is loaded. This dataset contains 5 gene clusters.

```
> require(clusterProfiler)
> data(gcSample)
> gcSample

$C1
 [1] "23753" "57222" "5036"  "5037"  "10111" "10856" "6228"
 [8] "9361"  "1537"  "3376"  "6124"  "4175"  "2539"

$C2
 [1] "6629"  "10291" "7094"  "3843"  "6611"  "10399" "10576"
 [8] "4705"  "5216"  "6697"  "5868"  "80777" "1973"  "1938"
```

```
> print(plot(xx, type = "dot", caption = "MF Ontology Distribution Comparison"))
```
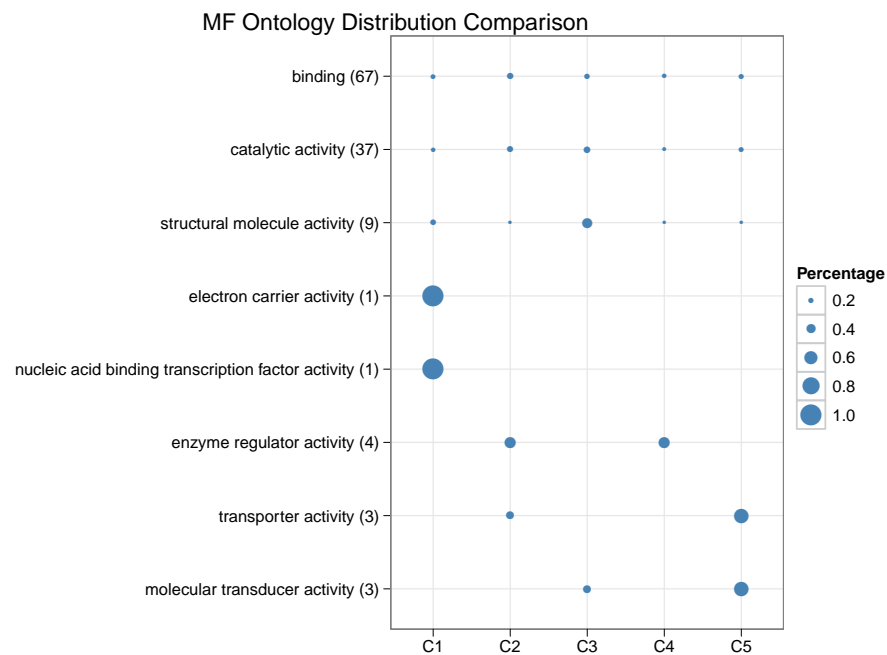
MF Ontology Distribution Comparison

Figure 1: Example of comparing MF ontology distribution using dotplot.

```
[15] "23450" "9343"  "1917"  "9520"


$C3
 [1] "4905"   "10383"  "10953"  "645958" "7280"   "10381"
 [7] "5869"   "5985"   "23197"  "290"    "309"    "10577"
[13] "23071"  "121504" "2495"   "653226" "84617"


$C4
 [1] "51552" "8336"  "302"   "5984"  "50814" "8813"  "871"
 [8] "81"    "23344" "4134"  "10262" "22919" "159"


$C5
 [1] "11171"  "8243"   "112464" "2194"   "9318"   "79026"
 [7] "1654"   "65003"  "6240"   "3476"   "6238"   "3836"
[13] "4176"   "1017"   "249"
```

- GO distribution among a set of gene clusters can be compared by *compareCluster*, and plotted by bar chart or dot chart.

```
> xx <- compareCluster(gcSample, fun = groupGO,
+     organism = "human", ont = "MF", level = 2)
```

By default, only top 5 categories of each cluster was plotted. User can change the parameter *limit* to specify how many categories of each cluster to be plotted, and if *limit* set to NULL, the whole result will

```
> print(plot(xx, type = "bar", limit = NULL, by = "count",
+     caption = "MF Ontology Distribution Comparison"))
```
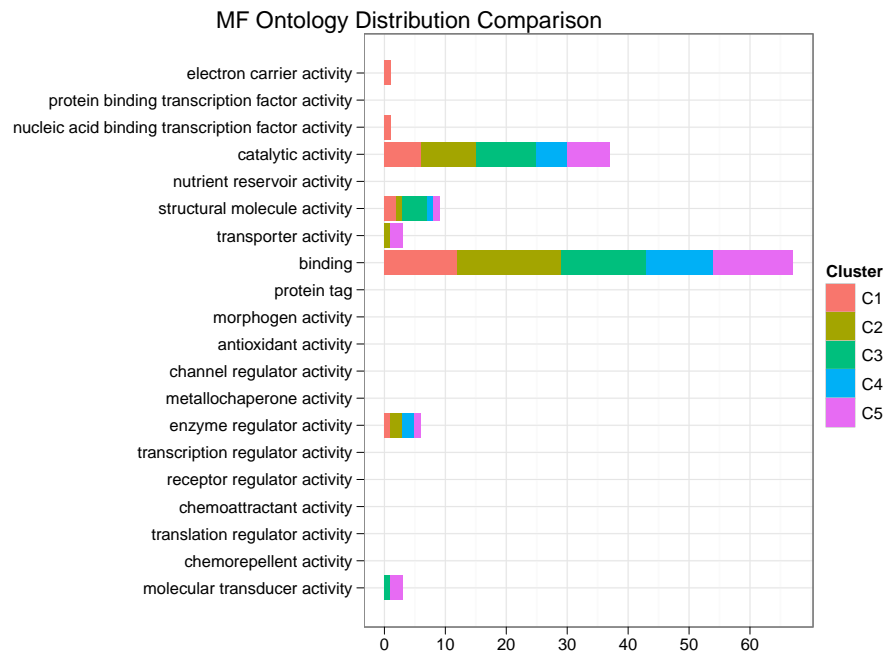

Figure 2: Example of comparing MF ontology distribution using barplot.

be plotted. By default, the dot sizes were based on their corresponding row percentage, and user can set the parameter *by* to "count" to make the comparison based on gene counts.

We chose "percentage" as default parameter to represent the sizes of dots, since some categories may contain a large number of genes, and make the dot sizes of those small categories too small to compare. To provide the full information, we also provide number of identified genes in each category (numbers in parentheses), as shown in Figure 1. If the dot sizes were based on "count", the parentheses will not showed as in Figure 2.

In the bar chart, color is used to differ distinct clusters.

• GO or KEGG enrichment analysis among a set of gene clusters can also be compared by *compareCluster* as shown in the following examples.

```
> xx <- compareCluster(gcSample, fun = enrichGO,
+     organism = "human", ont = "CC", pvalueCutoff = 0.01,
+     testDirection = "over")
> head(summary(xx))

  Cluster      GOID                    Description
1      C1 GO:0044446  intracellular organelle part
2      C1 GO:0044422                 organelle part
3      C1 GO:0030529     ribonucleoprotein complex
4      C1 GO:0070013 intracellular organelle lumen
```

3

```
5       C1 GO:0043233              organelle lumen
6       C1 GO:0031974      membrane-enclosed lumen
        Pvalue Qvalue OddsRatio ExpCount Count GeneSetSize
1 1.149368e-05      0 25.256415 4.193321    12          13
2 1.344710e-05      0 24.753880 4.250527    12          13
3 5.327956e-04      0 13.916396 0.405148     4          13
4 5.441960e-04      0  7.765743 1.702092     7          13
5 6.132116e-04      0  7.595756 1.735005     7          13
6 6.816604e-04      0  7.447439 1.764784     7          13
  Size
1 5351
2 5424
3  517
4 2172
5 2214
6 2252
                                                       GeneID
1 23753/57222/5036/5037/10111/10856/6228/9361/1537/6124/4175/2539
2 23753/57222/5036/5037/10111/10856/6228/9361/1537/6124/4175/2539
3                                       5036/10856/6228/6124
4                      23753/5036/10111/10856/9361/6124/4175
5                      23753/5036/10111/10856/9361/6124/4175
6                      23753/5036/10111/10856/9361/6124/4175
```

The p-values indicate that which categories are more likely to have biological meanings. The dots in the image are color-encoded based on their corresponding p-values. Color gradient ranging from blue to red correspond to in order of increasing p-values. Blue indicate lower p-values, and red indicate higher p-values. P-values were filtered out by the threshold giving by parameter *pvalueCutoff*.

We also provide FDR-corrected q-values, which were calculated by *fdrtool*, to control false positive discovery rate. FDR control is necessary since enrichment analysis carrying out hundreds, if not thousands, of tests.

```
> xx <- compareCluster(gcSample, fun = enrichKEGG,
+     organism = "human", pvalueCutoff = 0.05)
> head(summary(xx))

  Cluster pathwayID
1      C1  hsa03010
2      C1  hsa00290
3      C1  hsa03450
4      C2  hsa03040
5      C2  hsa00790
6      C3  hsa05130
                                 Description annoGeneRatio
1                                   Ribosome          2/13
2 Valine, leucine and isoleucine biosynthesis          1/13
3               Non-homologous end-joining          1/13
4                                 Spliceosome          4/18
5                          Folate biosynthesis          1/18
6      Pathogenic Escherichia coli infection          4/17
  annoBgRatio     pvalue     qvalue
```

```
> print(plot(xx, caption = "CC Ontology Enrichment Comparison"))
```
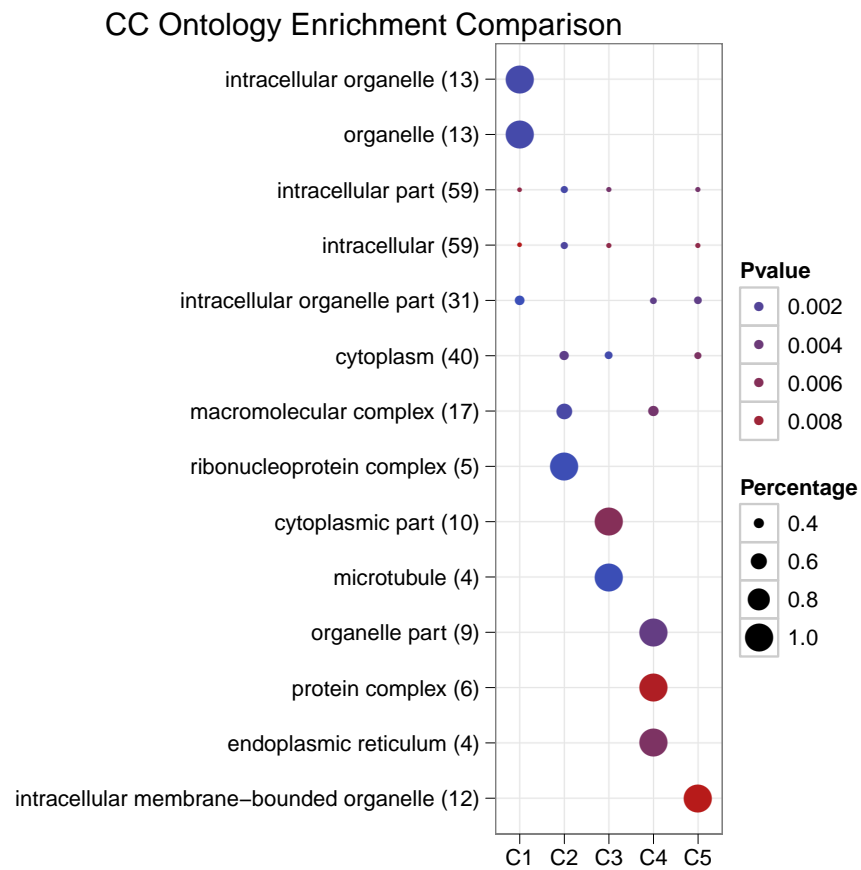

Figure 3: Example of comparing CC ontology enrichment among gene clusters.

```
> print(plot(xx, type = "dot", caption = "KEGG Enrichment Comparison"))
```
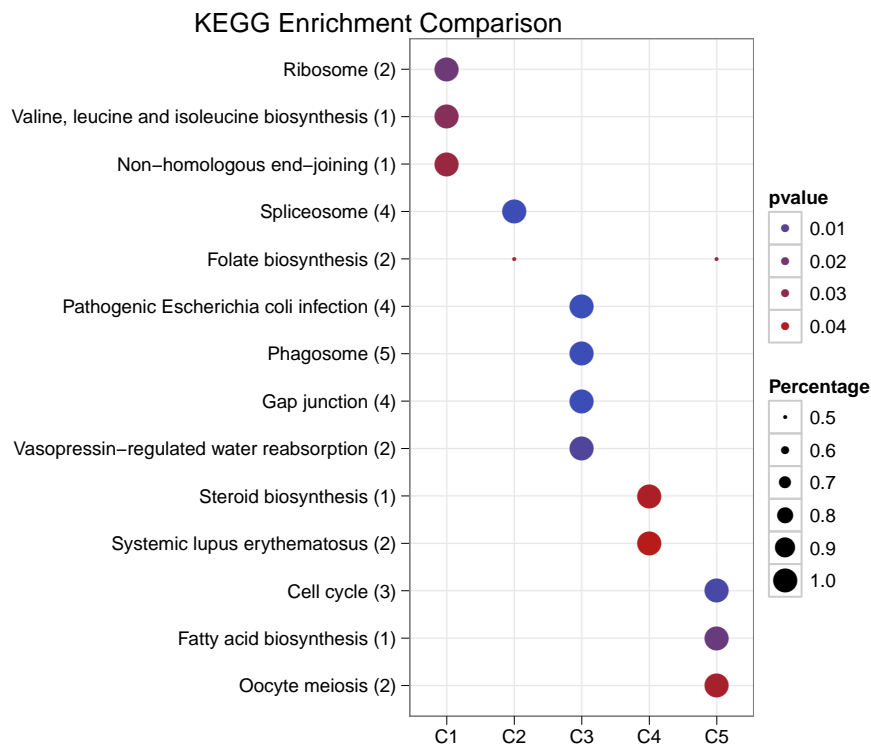

Figure 4: Example of comparing KEGG enrichment among gene clusters.

```
1       88/5504 1.758231e-02 1.000000000
2       11/5504 2.569952e-02 1.000000000
3       14/5504 3.260190e-02 1.000000000
4      128/5504 6.630825e-04 1.000000000
5       11/5504 3.542300e-02 1.000000000
6       59/5504 2.554952e-05 0.005397106
                          geneID Count
1               6228/6124     2
2                    3376     1
3                   10111     1
4   6629/10291/23450/9343     4
5                    6697     1
6 10383/7280/10381/84617     4
```

- The internal functions for annotating gene and enrichment analysis was groupGO, enrichGO and enrichKEGG, which was designed to analyze one particular gene list. Gene list can be projected to GO at a given level by groupGO. GO enrichment analysis were also provided by enrichGO for exploring biological themes of a given gene list. The internal algorithm in enrichGO was hyperGTest provided by *Category*. enrichGO extend *GOstats*(Falcon et al., 2007) by providing corresponding enrichment gene list and the FDR-corrected q-values. KEGG enrichment analysis were also supported by enrichKEGG.

```
> yy <- groupGO(gcSample[[1]], organism = "human",
+     ont = "BP", level = 2)
> yy <- enrichGO(gcSample[[1]], organism = "human",
+     ont = "BP", pvalueCutoff = 0.01, testDirection = "over")

> yy <- enrichKEGG(gcSample[[3]], organism = "human",
+     pvalueCutoff = 0.01)
> head(summary(yy))

      pathwayID                              Description
05130  hsa05130    Pathogenic Escherichia coli infection
04145  hsa04145                                Phagosome
04540  hsa04540                              Gap junction
04962  hsa04962 Vasopressin-regulated water reabsorption
      annoGeneRatio annoBgRatio       pvalue      qvalue
05130          4/17      59/5504 2.554952e-05 0.005397106
04145          5/17     159/5504 8.823916e-05 0.009219728
04540          4/17      90/5504 1.352278e-04 0.010247594
04962          2/17      44/5504 7.871612e-03 0.376048432
                         geneID Count
05130        10383/7280/10381/84617     4
04145 10383/7280/10381/5869/84617     5
04540        10383/7280/10381/84617     4
04962                     4905/5869     2
```

The outputs of groupGO, enrichGO and enrichKEGG can also be visualized by plot.

3 Session Information

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

```
R version 2.12.0 (2010-10-15)
Platform: i686-pc-linux-gnu (32-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=C              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      stats     graphics  grDevices utils
[6] datasets  methods   base

other attached packages:
 [1] GO.db_2.5.0            org.Hs.eg.db_2.5.0
 [3] clusterProfiler_0.99.6 fdrtool_1.2.6
 [5] KEGG.db_2.4.5          RSQLite_0.9-2
 [7] DBI_0.2-5              AnnotationDbi_1.13.17
```

```
> print(plot(yy, caption = "BP ontology categrories at level 2"))
```
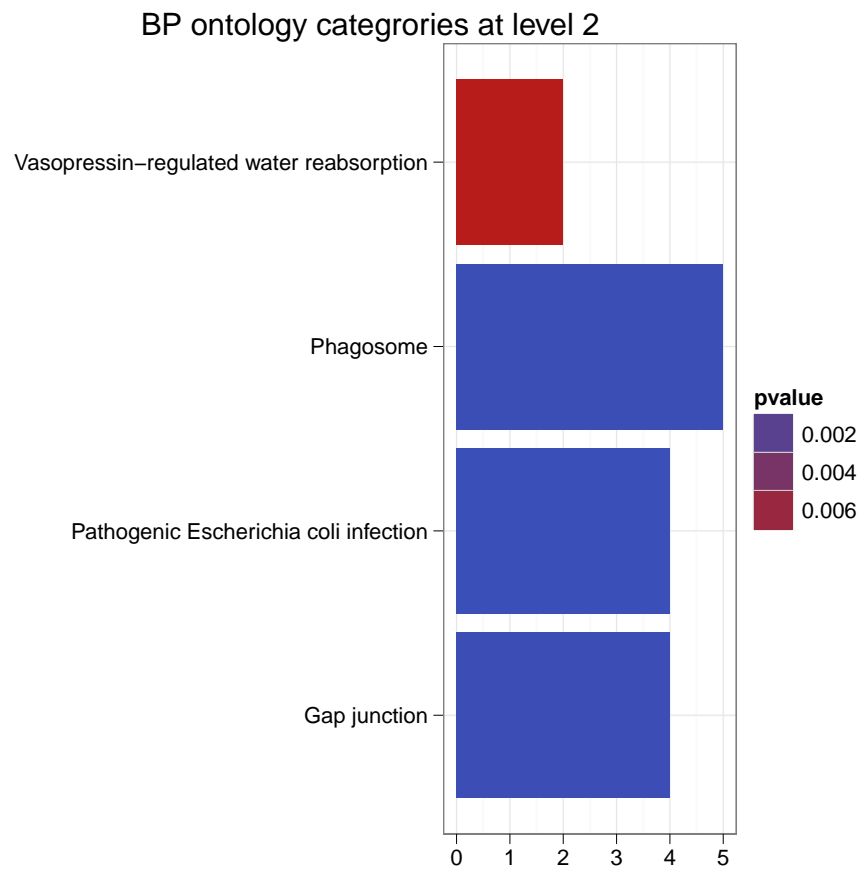

Figure 5: Example of annotating gene list to BP ontology at level 2.

```
 [9] Biobase_2.11.10        ggplot2_0.8.8
[11] proto_0.3-8            reshape_0.8.3
[13] plyr_1.2.1

loaded via a namespace (and not attached):
 [1] Category_2.16.0    GOstats_2.16.0
 [3] GSEABase_1.12.0    RBGL_1.26.0
 [5] XML_3.2-0          annotate_1.28.0
 [7] digest_0.4.2       genefilter_1.32.0
 [9] graph_1.28.0       org.Mm.eg.db_2.5.0
[11] splines_2.12.0     survival_2.35-8
[13] tools_2.12.0       xtable_1.5-6
```

References

S. Falcon, , and R. Gentleman. Using gostats to test gene lists for go term association. *Bioinformatics*, 23: 257–258, 2007.