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

Guangchuang Yu

Jinan University, Guangzhou, China

July 29, 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*(Falcon et al., 2007) for GO enrichment analysis.

After clustering analysis, researchers not only want to determine whether there is a common theme of a particular gene cluster, but also to compare the biological themes among gene clusters, which have different expression profiles. To bridge this gap, 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 a set of genes 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"
[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"
```

- Use `groupGO` for genes classification based on GO distribution at a specific level.

```
> x <- groupGO(gene = gcSample[[1]], organism = "human",
+     ont = "CC", level = 2, readable = TRUE)
> summary(x)

                GOID               Description Count
GO:0005576 GO:0005576      extracellular region     1
GO:0005623 GO:0005623                      cell    13
GO:0019012 GO:0019012                   virion     0
GO:0031974 GO:0031974   membrane-enclosed lumen     7
GO:0032991 GO:0032991    macromolecular complex     6
GO:0043226 GO:0043226                 organelle    13
GO:0044421 GO:0044421 extracellular region part     1
GO:0044422 GO:0044422            organelle part    12
GO:0044423 GO:0044423               virion part     0
GO:0044456 GO:0044456              synapse part     1
GO:0044464 GO:0044464                 cell part    13
GO:0045202 GO:0045202                   synapse     1
GO:0055044 GO:0055044                  symplast     0
                                                                                Ge
GO:0005576                                                                      P
GO:0005623 SDF2L1/ERGIC1/PA2G4/PEBP1/RAD50/RUVBL2/RPS23/LONP1/CYC1/IARS/RPL4/MCM6/
GO:0019012
GO:0031974                                      SDF2L1/PA2G4/RAD50/RUVBL2/LONP1/RPL4/
GO:0032991                                           PA2G4/RAD50/RUVBL2/RPS23/RPL4/
GO:0043226 SDF2L1/ERGIC1/PA2G4/PEBP1/RAD50/RUVBL2/RPS23/LONP1/CYC1/IARS/RPL4/MCM6/
GO:0044421                                                                      P
GO:0044422      SDF2L1/ERGIC1/PA2G4/PEBP1/RAD50/RUVBL2/RPS23/LONP1/CYC1/RPL4/MCM6/
GO:0044423
GO:0044456                                                                      P
GO:0044464 SDF2L1/ERGIC1/PA2G4/PEBP1/RAD50/RUVBL2/RPS23/LONP1/CYC1/IARS/RPL4/MCM6/
GO:0045202                                                                      P
GO:0055044
```

- Use `enrichGO` for GO enrichment analysis.

```
> y <- enrichGO(gene = gcSample[[2]], organism = "human",
+     ont = "MF", pvalueCutoff = 0.01, readable = TRUE)
```

- Use `enrichKEGG` for KEGG pathway enrichment analysis.

```
> z <- enrichKEGG(gene = gcSample[[3]], organism = "human",
+     pvalueCutoff = 0.05, readable = TRUE)
> summary(z)
```

```
      pathwayID                                 Description
05130  hsa05130    Pathogenic Escherichia coli infection
04145  hsa04145                               Phagosome
04540  hsa04540                             Gap junction
04962  hsa04962 Vasopressin-regulated water reabsorption
04614  hsa04614                 Renin-angiotensin system
      GeneRatio  BgRatio       pvalue       qvalue
05130      4/17  58/5894 1.826892e-05 0.0002115348
04145      5/17 156/5894 5.827611e-05 0.0003373880
04540      4/17  90/5894 1.039489e-04 0.0004012064
04962      2/17  44/5894 6.898981e-03 0.0199707355
04614      1/17  17/5894 4.798133e-02 0.1111146614
                              geneID Count
05130       TUBB2C/TUBB2A/TUBB3/TUBB6     4
04145 TUBB2C/TUBB2A/TUBB3/RAB5B/TUBB6     5
04540       TUBB2C/TUBB2A/TUBB3/TUBB6     4
04962                      NSF/RAB5B     2
04614                          ANPEP     1
```

The input parameters of *gene* is a vector of entrez genes or ORF IDs (for yeast), and *organism* must be one of "human", "mouse", and "yeast", according to the gene IDs. For GO analysis, *ont* must be assigned to one of "BP", "MF", and "CC" for biological process, molecular function and cellular component, respectively. In `groupGO`, the *level* specify the GO level for gene projection. In enrichment analysis, the *pvalueCutoff* is to restrict the result based on their pvalues. Consider multiple testing, qvalues are also provided, for estimating FDR. The *readable* is a logical parameter, if TRUE, the gene IDs will map to gene symbols.

In addition, these results can be visualized by our `plot` function. For example:

```
> plot(x, title = "CC Ontology Classification, level 2",
+     font.size = 12)

> plot(z, title = "KEGG Enrichment")
```

- 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)
> plot(xx, title = "MF Ontology Distribution Comparison")

> yy <- compareCluster(gcSample, fun = enrichGO,
+     organism = "human", ont = "CC", pvalueCutoff = 0.01)
> plot(yy, title = "CC Ontology Enrichment Comparison")
```

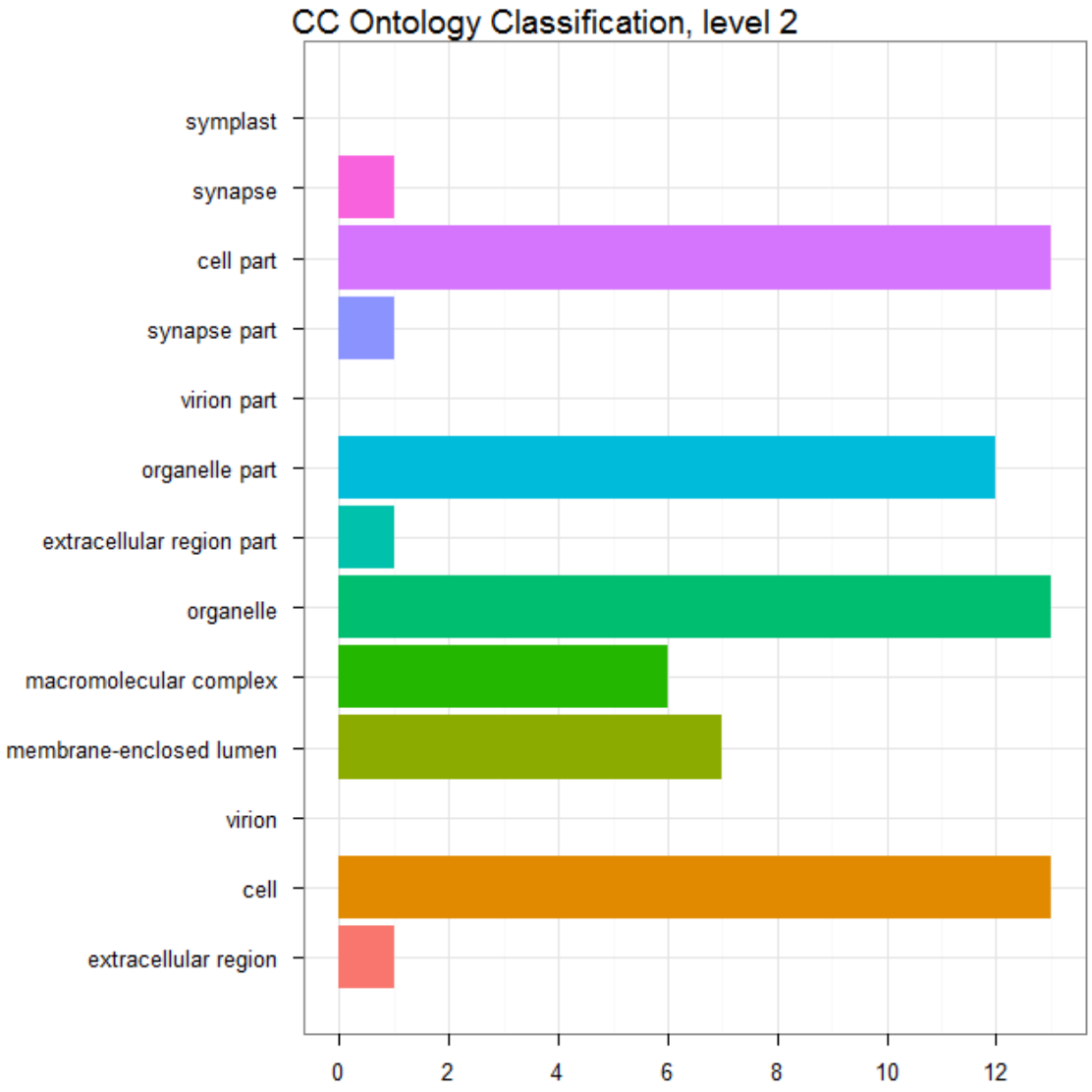

Figure 1: Example of gene classification

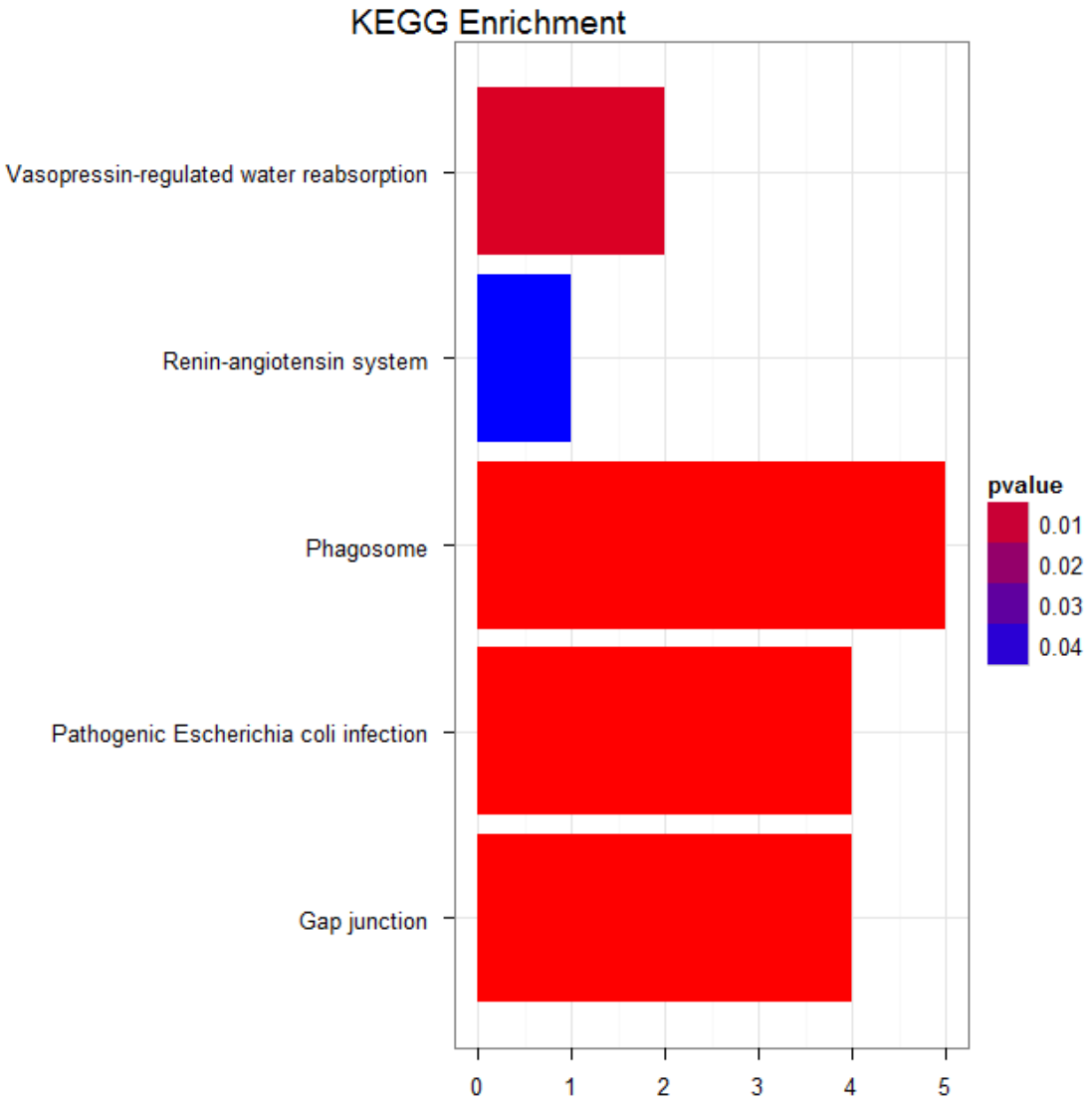

Figure 2: Example of KEGG enrichment analysis

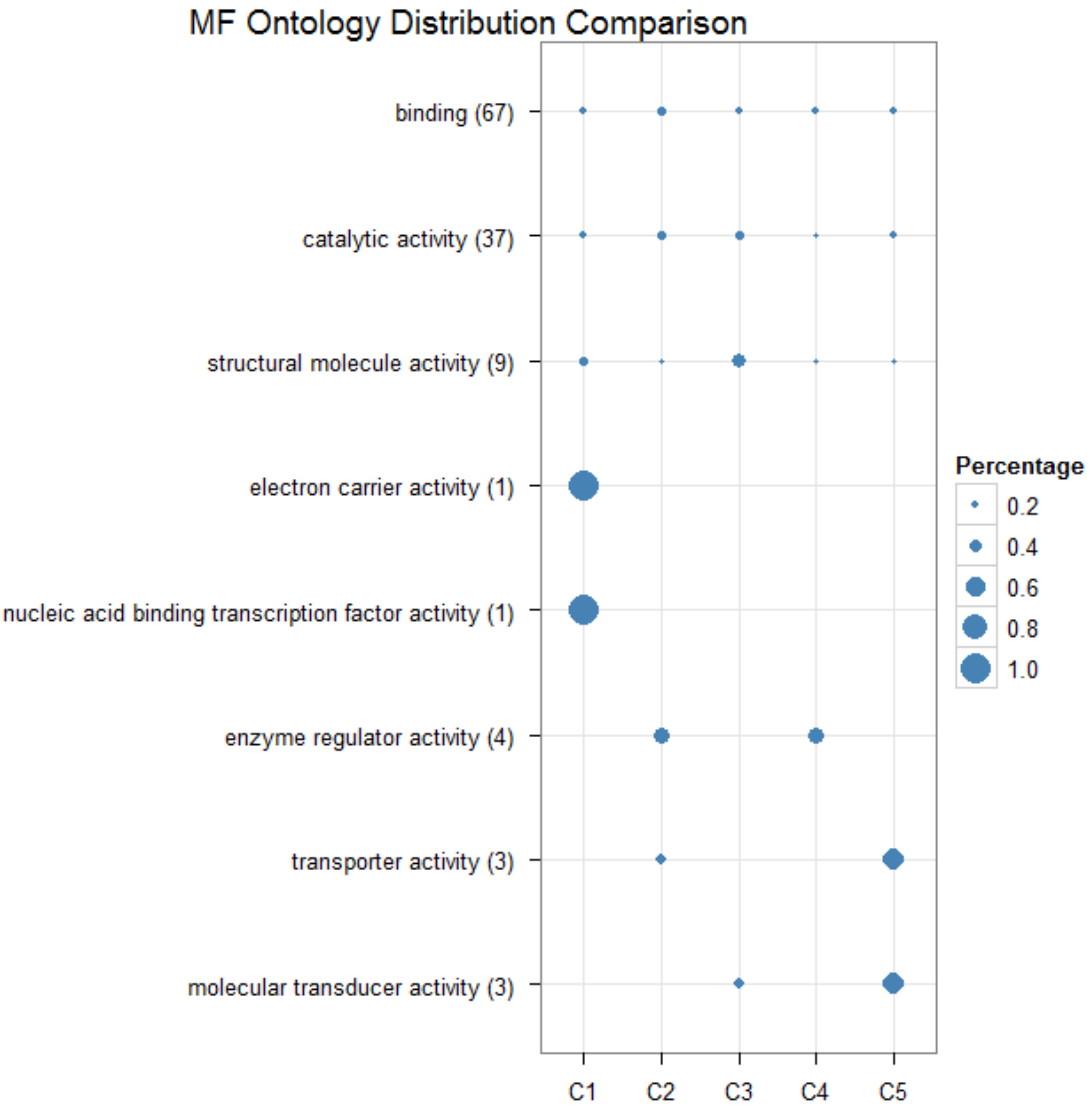

Figure 3: Example of comparing MF ontology distribution using dotplot

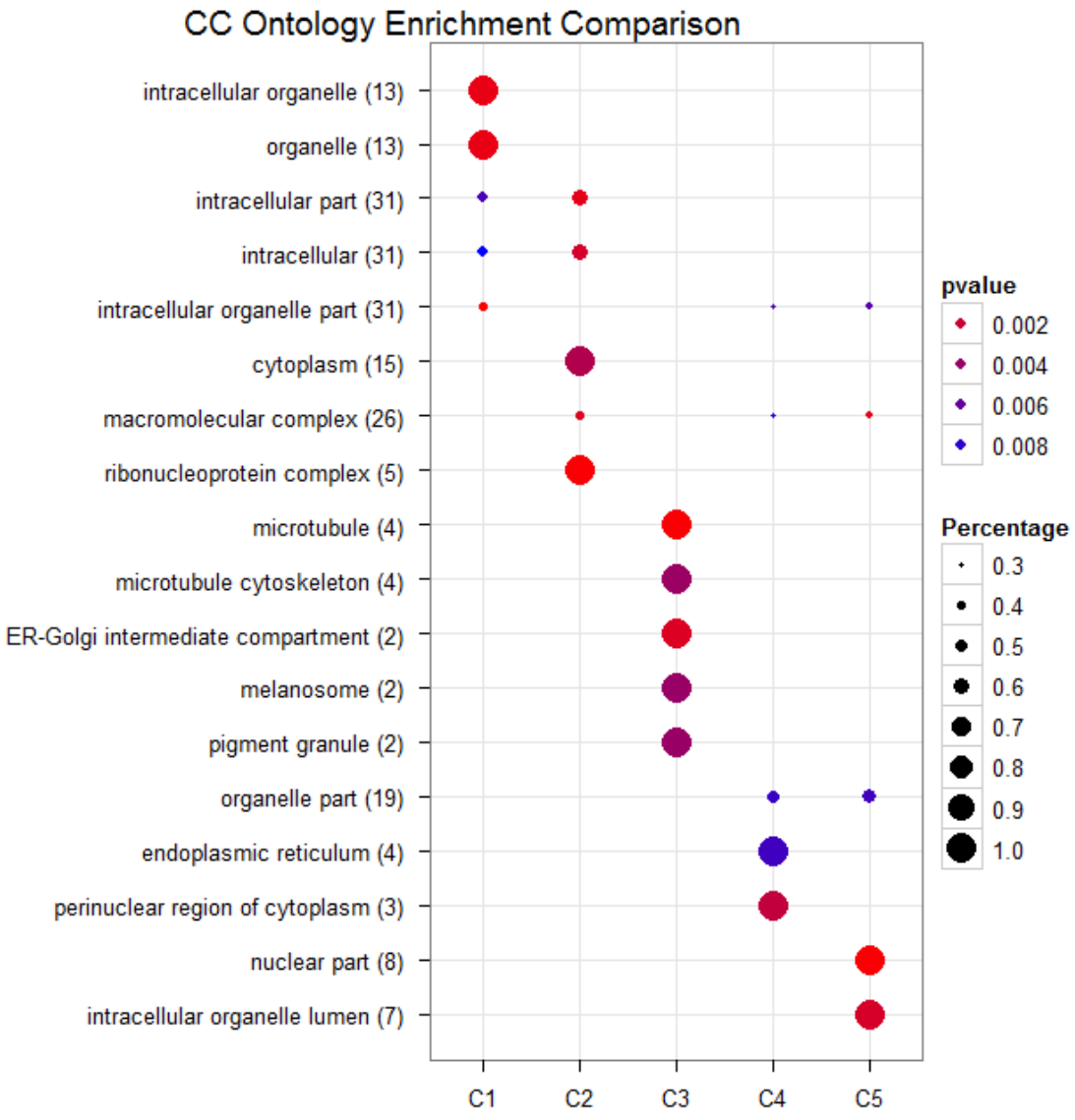

Figure 4: Example of comparing CC ontology enrichment using dot chart

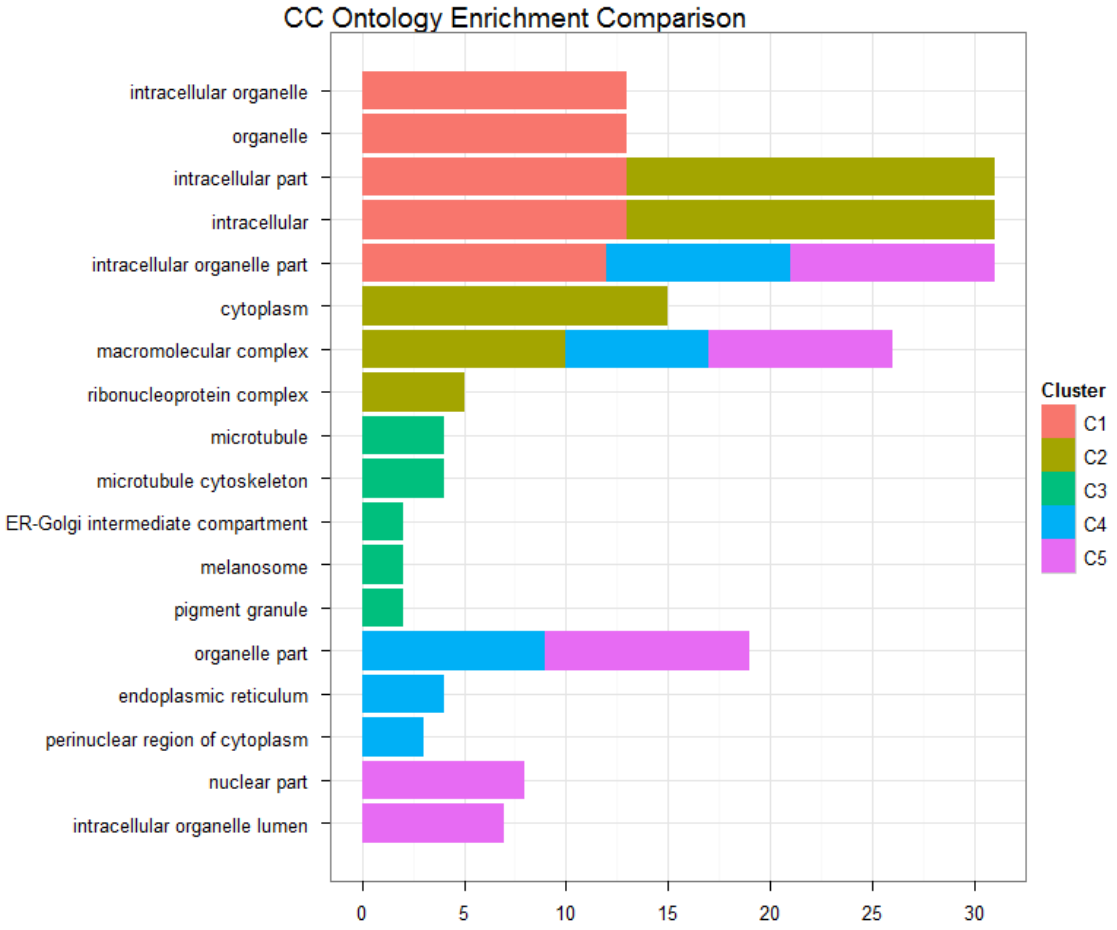

Figure 5: Example of comparing CC ontology enrichment using bar chart

```
> plot(yy, title = "CC Ontology Enrichment Comparison",
+     type = "bar", by = "count")
```

```
> zz <- compareCluster(gcSample, fun = enrichKEGG,
+     organism = "human", pvalueCutoff = 0.05)
> plot(zz, title = "KEGG Pathway Enrichment Comparison")
```

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

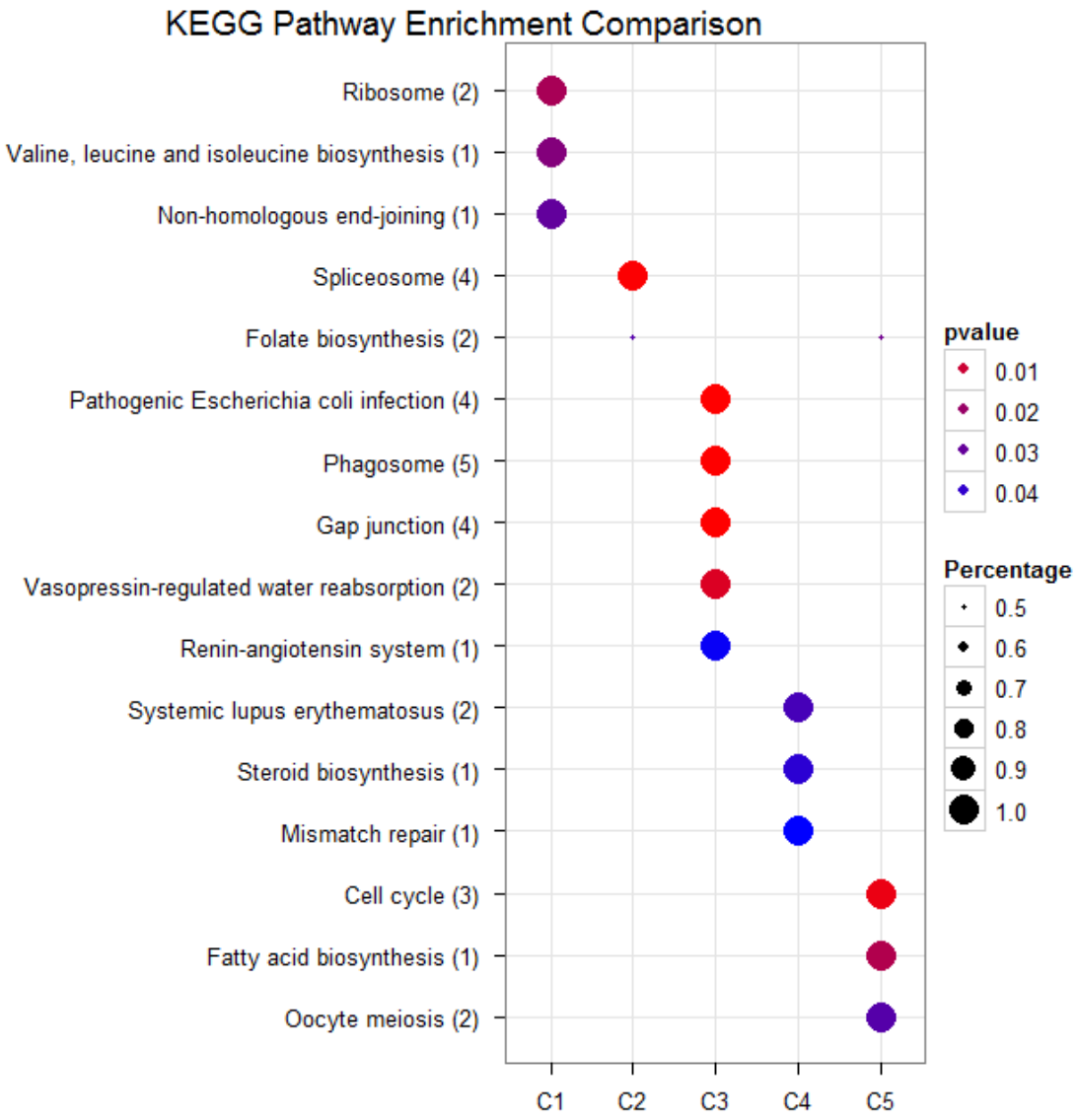

Figure 6: Example of comparing KEGG enrichment among gene clusters

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 red to blue correspond to in order of increasing p-values. red indicate low p-values (high enrichment), and blue indicate high p-values (low enrichment). P-values were filtered out by the threshold giving by parameter *pvalueCutoff*.

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

3 Session Information

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

```
R version 2.13.0 (2011-04-13)
Platform: i386-pc-mingw32/i386 (32-bit)

locale:
[1] LC_COLLATE=C
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] grid      stats     graphics  grDevices utils
[6] datasets  methods   base

other attached packages:
[1] clusterProfiler_1.0.6 RSQLite_0.9-4
[3] DBI_0.2-5             ggplot2_0.8.9
[5] proto_0.3-9.2         reshape_0.8.4
[7] plyr_1.5.2

loaded via a namespace (and not attached):
 [1] AnnotationDbi_1.14.1 Biobase_2.12.2
 [3] GO.db_2.5.0          KEGG.db_2.5.0
 [5] org.Hs.eg.db_2.5.0   org.Mm.eg.db_2.5.0
 [7] org.Sc.sgd.db_2.5.0  qvalue_1.26.0
 [9] tcltk_2.13.0         tools_2.13.0
```

References

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