Statistical analysis and visualization of functional profiles for genes and gene clusters

Guangchuang Yu (guangchuangyu@gmail.com)
School of Public Health, The University of Hong Kong

2017-03-08

Contents

1 Abstract
- 1.1 Supported Analysis
- 1.2 Supported ontologies/pathways
- 1.3 Visualization

2 Citation
3 Introduction
4 `bitr`: Biological Id TranslatoR
- 4.1 `bitr_kegg`: converting biological IDs using KEGG API

5 GO Analysis
- 5.1 Supported organisms
- 5.2 GO classification
- 5.3 GO over-representation test
  - 5.3.1 drop specific GO terms or level
  - 5.3.2 test GO at sepcific level
  - 5.3.3 reduce redundancy of enriched GO terms
- 5.4 GO Gene Set Enrichment Analysis
- 5.5 GO Semantic Similarity Analysis

6 KEGG analysis
- 6.1 KEGG over-representation test
- 6.2 KEGG Gene Set Enrichment Analysis
- 6.3 KEGG Module over-representation test
- 6.4 KEGG Module Gene Set Enrichment Analysis

7 Disease analysis
8 Reactome pathway analysis
9 DAVID functional analysis
10 Universal enrichment analysis
- 10.1 Using MSigDB gene set collections

11 Functional analysis of NGS data
12 Visualization
- 12.1 barplot
- 12.2 dotplot
- 12.3 enrichMap
- 12.4 cnetplot
- 12.5 plotGOgraph
- 12.6 gseaplot
- 12.7 browseKEGG
- 12.8 pathview from pathview package

13 Biological theme comparison
- 13.1 Formula interface of compareCluster
- 13.2 Visualization of profile comparison

14 Homepage
15 Session Information
References

1 Abstract

clusterProfiler implements methods to analyze and visualize functional profiles of genomic coordinates (supported by *ChIPseeker*), gene and gene clusters.

1.1 Supported Analysis

- Over-Representation Analysis
- Gene Set Enrichment Analysis
- Biological theme comparison

1.2 Supported ontologies/pathways

- Disease Ontology (via DOSE)
- Network of Cancer Gene (via DOSE)
- DisGeNET (via DOSE)
- Gene Ontology (supports many species with GO annotation query online via AnnotationHub)
- KEGG Pathway and Module with latest online data (supports more than 4000 species listed in http://www.genome.jp/kegg/catalog/org_list.html)
- Reactome Pathway (via ReactomePA)
- DAVID (via RDAVIDWebService)
- Molecular Signatures Database
    - hallmark gene sets
    - positional gene sets
    - curated gene sets
    - motif gene sets
    - computational gene sets
    - GO gene sets
    - oncogenic signatures
    - immunologic signatures
- Other Annotations
    - from other sources (e.g. DisGeNET as an example)
    - user's annotation
    - customized ontology
    - and many others

1.3 Visualization

- barplot
- cnetplot
- dotplot
- enrichMap
- gseaplot
- plotGOgraph (via topGO package)
- upsetplot

2 Citation

If you use *clusterProfiler* in published research, please cite:

G Yu, LG Wang, Y Han, QY He. clusterProfiler: an R package for comparing biological themes among gene clusters. ***OMICS: A Journal of Integrative Biology*** 2012, 16(5):284-287. doi:[10.1089/omi.2011.0118](http://dx.doi.org/10.1089/omi.2011.0118)

3 Introduction

In recently years, high-throughput experimental techniques such as microarray, RNA-Seq and mass spectrometry can detect cellular molecules at systems-level. These kinds of analyses generate huge quantitaties of data, which need to be given a biological interpretation. A commonly used approach is via clustering in the gene dimension for grouping different genes based on their similarities[1].

To search for shared functions among genes, a common way is to incorporate the biological knowledge, such as Gene Ontology (GO) and Kyoto Encyclopedia of Genes and Genomes (KEGG), for identifying predominant biological themes of a collection of genes.

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. The manual step to choose interesting clusters followed by enrichment analysis on each selected cluster is slow and tedious. To bridge this gap, we designed *clusterProfiler*[2], for comparing and visualizing functional profiles among gene clusters.

4 `bitr`: Biological Id TranslatoR

clusterProfiler provides `bitr` and `bitr_kegg` for converting ID types. Both `bitr` and `bitr_kegg` support many species including model and many non-model organisms.

```
x <- c("GPX3",  "GLRX",   "LBP",   "CRYAB", "DEFB1", "HCLS1",   "SOD2",   "HSPA2",
       "ORM1",  "IGFBP1", "PTHLH", "GPC3",  "IGFBP3","TOB1",    "MITF",   "NDRG1",
       "NR1H4", "FGFR3",  "PVR",   "IL6",   "PTPRM", "ERBB2",   "NID2",   "LAMB1",
       "COMP",  "PLS3",   "MCAM",  "SPP1",  "LAMC1", "COL4A2",  "COL4A1", "MYOC",
       "ANXA4", "TFPI2",  "CST6",  "SLPI",  "TIMP2", "CPM",     "GGT1",   "NNMT",
       "MAL",   "EEF1A2", "HGD",   "TCN2",  "CDA",   "PCCA",    "CRYM",   "PDXK",
       "STC1",  "WARS",  "HMOX1", "FXYD2", "RBP4",   "SLC6A12", "KDELR3", "ITM2B")
eg = bitr(x, fromType="SYMBOL", toType="ENTREZID", OrgDb="org.Hs.eg.db")
head(eg)
```

```
##   SYMBOL ENTREZID
## 1   GPX3     2878
## 2   GLRX     2745
## 3    LBP     3929
## 4  CRYAB     1410
## 5  DEFB1     1672
## 6  HCLS1     3059
```

User should provides an annotation package, both *fromType* and *toType* can accept any types that supported.

User can use *keytypes* to list all supporting types.

```
library(org.Hs.eg.db)
keytypes(org.Hs.eg.db)
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"
##  [5] "ENSEMBLTRANS" "ENTREZID"     "ENZYME"       "EVIDENCE"
##  [9] "EVIDENCEALL"  "GENENAME"     "GO"           "GOALL"
## [13] "IPI"          "MAP"          "OMIM"         "ONTOLOGY"
## [17] "ONTOLOGYALL"  "PATH"         "PFAM"         "PMID"
## [21] "PROSITE"      "REFSEQ"       "SYMBOL"       "UCSCKG"
## [25] "UNIGENE"      "UNIPROT"
```

We can translate from one type to other types.

```
ids <- bitr(x, fromType="SYMBOL", toType=c("UNIPROT", "ENSEMBL"), OrgDb="org.Hs.eg.db")
head(ids)
```

```
##   SYMBOL    UNIPROT         ENSEMBL
## 1   GPX3     P22352 ENSG00000211445
## 2   GLRX A0A024RAM2 ENSG00000173221
## 3   GLRX     P35754 ENSG00000173221
## 4    LBP     P18428 ENSG00000129988
## 5    LBP     Q8TCF0 ENSG00000129988
## 6  CRYAB     P02511 ENSG00000109846
```

For GO analysis, user don't need to convert ID, all ID type provided by `OrgDb` can be used in `groupGO`, `enrichGO` and `gseGO` by specifying `keytype` parameter.

4.1 `bitr_kegg`: converting biological IDs using KEGG API

```
data(gcSample)
hg <- gcSample[[1]]
head(hg)
```

```
## [1] "4597"  "7111"  "5266"  "2175"  "755"   "23046"
```

```
eg2np <- bitr_kegg(hg, fromType='kegg', toType='ncbi-proteinid', organism='hsa')
head(eg2np)
```

```
##     kegg ncbi-proteinid
## 1 146691   NP_001076437
## 2  23148   NP_001139806
## 3  64221      NP_071765
## 4    119   NP_001171983
## 5   4109   NP_001011543
## 6  51314      NP_057700
```

The ID type (both `fromType` & `toType`) should be one of 'kegg', 'ncbi-geneid', 'ncbi-proteinid' or 'uniprot'. The 'kegg' is the primary ID used in KEGG database. The data source of KEGG was from NCBI. A rule of thumb for the 'kegg' ID is `entrezgene` ID for eukaryote species and `Locus` ID for prokaryotes.

Many prokaryote species don't have entrezgene ID available. For example we can check the gene information of `ece:Z5100` in http://www.genome.jp/dbget-bin/www_bget?ece:Z5100, which have `NCBI-ProteinID` and `UnitProt` links in the `Other DBs` Entry, but not `NCBI-GeneID`.

If we try to convert `Z5100` to `ncbi-geneid`, `bitr_kegg` will throw error of `ncbi-geneid is not supported`.

```
bitr_kegg("Z5100", fromType="kegg", toType='ncbi-geneid', organism='ece')
```

```
## Error in KEGG_convert(fromType, toType, organism) :
## ncbi-geneid is not supported for ece ...
```

We can of course convert it to `ncbi-proteinid` and `uniprot`:

```
bitr_kegg("Z5100", fromType="kegg", toType='ncbi-proteinid', organism='ece')
```

```
##    kegg ncbi-proteinid
## 1 Z5100       AAG58814
```

```
bitr_kegg("Z5100", fromType="kegg", toType='uniprot', organism='ece')
```

```
##    kegg uniprot
## 1 Z5100  Q7DB85
```

5 GO Analysis

5.1 Supported organisms

GO analyses (`groupGO()`, `enrichGO()` and `gseGO()`) support organisms that have an `OrgDb` object available.

Bioconductor have already provide `OrgDb` for about 20 species. User can query `OrgDb` online by *AnnotationHub* or build their own by *AnnotationForge*. An example can be found in the vignette of *GOSemSim*.

If user have GO annotation data (in data.frame format with first column of gene ID and second column of GO ID), they can use `enricher()` and `gseGO()` functions to perform over-representation test and gene set enrichment analysis.

If genes are annotated by direction annotation, it should also annotated by its ancestor GO nodes (indirect annation). If user only has direct annotation, they can pass their annotation to `buildGOmap` function, which will infer indirection annotation and generate a `data.frame` that suitable for both `enricher()` and `gseGO()`.

5.2 GO classification

In *clusterProfiler*, `groupGO` is designed for gene classification based on GO distribution at a specific level. Here we use dataset `geneList` provided by *DOSE*. Please refer to vignette of *DOSE* for more details.

```
data(geneList, package="DOSE")
gene <- names(geneList)[abs(geneList) > 2]
gene.df <- bitr(gene, fromType = "ENTREZID",
        toType = c("ENSEMBL", "SYMBOL"),
        OrgDb = org.Hs.eg.db)
head(gene.df)
```

```
##   ENTREZID         ENSEMBL SYMBOL
## 1     4312 ENSG00000196611   MMP1
## 2     8318 ENSG00000093009  CDC45
## 3    10874 ENSG00000109255    NMU
## 4    55143 ENSG00000134690  CDCA8
## 5    55388 ENSG00000065328  MCM10
## 6      991 ENSG00000117399  CDC20
```

```
ggo <- groupGO(gene     = gene,
               OrgDb    = org.Hs.eg.db,
               ont      = "CC",
               level    = 3,
               readable = TRUE)

head(ggo)
```

```
##                    ID                   Description Count GeneRatio
## GO:0005886 GO:0005886               plasma membrane    53    53/207
## GO:0005628 GO:0005628              prospore membrane     0     0/207
## GO:0005789 GO:0005789 endoplasmic reticulum membrane     6     6/207
## GO:0019867 GO:0019867                 outer membrane     3     3/207
## GO:0031090 GO:0031090             organelle membrane    18    18/207
## GO:0034357 GO:0034357        photosynthetic membrane     0     0/207
##
## GO:0005886 S100A9/MELK/S100A8/MARCO/CXCL10/LAMP3/UGT8/SLC7A5/CXCL9/FADS2/ERCC6L/MSLN/IL1R2/KIF18A
## GO:0005628
## GO:0005789
## GO:0019867
## GO:0031090
## GO:0034357
```

The input parameters of *gene* is a vector of gene IDs (can be any ID type that supported by corresponding `OrgDb`).

If *readable* is setting to *TRUE*, the input gene IDs will be converted to gene symbols.

5.3 GO over-representation test

Over-representation test[3] were implemented in *clusterProfiler*. For calculation details and explanation of paramters, please refer to the vignette of *DOSE*.

```
ego <- enrichGO(gene          = gene,
                universe      = names(geneList),
                OrgDb         = org.Hs.eg.db,
                ont           = "CC",
                pAdjustMethod = "BH",
                pvalueCutoff  = 0.01,
                qvalueCutoff  = 0.05,
        readable      = TRUE)
head(ego)
```

```
##                   ID                                Description GeneRatio
## GO:0005819 GO:0005819                                    spindle    24/199
## GO:0000793 GO:0000793                      condensed chromosome    17/199
## GO:0000779 GO:0000779 condensed chromosome, centromeric region    13/199
## GO:0005876 GO:0005876                        spindle microtubule    10/199
## GO:0005875 GO:0005875           microtubule associated complex    14/199
## GO:0000776 GO:0000776                                kinetochore    13/199
##              BgRatio       pvalue     p.adjust       qvalue
## GO:0005819 231/11711 9.909518e-13 2.467470e-10 2.169663e-10
## GO:0000793 156/11711 1.141749e-09 1.200146e-07 1.055296e-07
## GO:0000779  84/11711 1.445959e-09 1.200146e-07 1.055296e-07
## GO:0005876  46/11711 3.833921e-09 2.316468e-07 2.036886e-07
## GO:0005875 110/11711 4.651542e-09 2.316468e-07 2.036886e-07
## GO:0000776 101/11711 1.464436e-08 6.077410e-07 5.343908e-07
##
## GO:0005819 CDCA8/CDC20/KIF23/CENPE/ASPM/DLGAP5/SKA1/NUSAP1/TPX2/NEK2/CDK1/MAD2L1/KIF18A/BIRC5/KI
## GO:0000793                                    CENPE/NDC80/TOP2A/NCAPH/HJURP/SKA1/NEK2/CENPM/
## GO:0000779                                                          CENPE/NDC80/HJURP/SKA
## GO:0005876                                                                        SKA
## GO:0005875                                                  CDCA8/KIF23/CENPE/KIF18A/BIRC5/
## GO:0000776                                                          CENPE/NDC80/HJURP/SKA1/
##            Count
## GO:0005819    24
## GO:0000793    17
## GO:0000779    13
## GO:0005876    10
## GO:0005875    14
## GO:0000776    13
```

As I mentioned before, any gene ID type that supported in `OrgDb` can be directly used in GO analyses. User need to specify the `keytype` parameter to specify the input gene ID type.

```
ego2 <- enrichGO(gene         = gene.df$ENSEMBL,
                 OrgDb         = org.Hs.eg.db,
        keytype       = 'ENSEMBL',
                 ont           = "CC",
                 pAdjustMethod = "BH",
                 pvalueCutoff  = 0.01,
                 qvalueCutoff  = 0.05)
```

Gene ID can be mapped to gene Symbol by using paramter `readable=TRUE` or `setReadable` function.

```
ego2 <- setReadable(ego2, OrgDb = org.Hs.eg.db)
```

5.3.1 drop specific GO terms or level

`enrichGO` test the whole GO corpus and enriched result may contains very general terms. With `dropGO` function, user can remove specific GO terms or GO level from results obtained from both `enrichGO` and `compareCluster`.

5.3.2 test GO at sepcific level

`enrichGO` doesn't contain parameter to restrict the test at specific GO level. Instead, we provide a function `gofilter` to restrict the result at specific GO level. It works with results obtained from both `enrichGO` and `compareCluster`.

5.3.3 reduce redundancy of enriched GO terms

According to issue #28, I implement a `simplify` method to remove redundant GO terms obtained from `enrichGO`. An example can be found in the blog post. It internally call *GOSemSim* to calculate similarities among GO terms and remove those highly similar terms by keeping one representative term. The `simplify` method also works with both outputs from `enrichGO` and `compareCluster`.

5.4 GO Gene Set Enrichment Analysis

A common approach in analyzing gene expression profiles was identifying differential expressed genes that are deemed interesting. The enrichment analysis we demonstrated previous were based on these differential expressed genes. This approach will find genes where the difference is large, but it will not detect a situation where the difference is small, but evidenced in coordinated way in a set of related genes. Gene Set Enrichment Analysis (GSEA)[4] directly addresses this limitation. All genes can be used in GSEA; GSEA aggregates the per gene statistics across genes within a gene set, therefore making it possible to detect situations where all genes in a predefined set change in a small but coordinated way. Since it is likely that many relevant phenotypic differences are manifested by small but consistent changes in a set of genes.

For algorithm details, please refer to the vignette of *DOSE*.

```
ego3 <- gseGO(geneList     = geneList,
              OrgDb        = org.Hs.eg.db,
              ont          = "CC",
              nPerm        = 1000,
              minGSSize    = 100,
              maxGSSize    = 500,
              pvalueCutoff = 0.05,
              verbose      = FALSE)
```

GSEA use permutation test, user can set *nPerm* for number of permutations. Only gene Set size in `[minGSSize, maxGSSize]` will be tested.

5.5 GO Semantic Similarity Analysis

GO semantic similarity can be calculated by *GOSemSim*[1]. We can use it to cluster genes/proteins into different clusters based on their functional similarity and can also use it to measure the similarities among GO terms to reduce the redundancy of GO enrichment results.

6 KEGG analysis

The annotation package, *KEGG.db*, is not updated since 2012. It's now pretty old and in *clusterProfiler*, `enrichKEGG` (for KEGG pathway) and `enrichMKEGG` (for KEGG module) supports downloading latest online version of KEGG data for enrichment analysis. Using *KEGG.db* is also supported by explicitly setting *use_internal_data* parameter to *TRUE*, but it's not recommended.

With this new feature, organism is not restricted to those supported in previous release, it can be any species that have KEGG annotation data available in KEGG database. User should pass abbreviation of academic name to the *organism* parameter. The full list of KEGG supported organisms can be accessed via http://www.genome.jp/kegg/catalog/org_list.html.

clusterProfiler provides `search_kegg_organism()` function to help searching supported organisms.

```
search_kegg_organism('ece', by='kegg_code')
```

```
##     kegg_code                        scientific_name common_name
## 334       ece Escherichia coli O157:H7 EDL933 (EHEC)        <NA>
```

```
ecoli <- search_kegg_organism('Escherichia coli', by='scientific_name')
dim(ecoli)
```

```
## [1] 64  3
```

```
head(ecoli)
```

```
##     kegg_code                        scientific_name common_name
## 329       eco           Escherichia coli K-12 MG1655        <NA>
## 330       ecj            Escherichia coli K-12 W3110        <NA>
## 331       ecd            Escherichia coli K-12 DH10B        <NA>
## 332       ebw                Escherichia coli BW2952        <NA>
## 333      ecok            Escherichia coli K-12 MDS42        <NA>
## 334       ece Escherichia coli O157:H7 EDL933 (EHEC)        <NA>
```

6.1 KEGG over-representation test

```
kk <- enrichKEGG(gene         = gene,
                 organism     = 'hsa',
                 pvalueCutoff = 0.05)
head(kk)
```

```
##                ID                            Description GeneRatio
## hsa04110 hsa04110                             Cell cycle     11/87
## hsa04114 hsa04114                         Oocyte meiosis     10/87
## hsa03320 hsa03320                PPAR signaling pathway      7/87
## hsa04914 hsa04914 Progesterone-mediated oocyte maturation      6/87
## hsa04115 hsa04115                  p53 signaling pathway      5/87
##           BgRatio       pvalue     p.adjust       qvalue
## hsa04110 124/7215 2.295579e-07 4.109087e-05 4.059551e-05
## hsa04114 124/7215 2.043885e-06 1.829277e-04 1.807225e-04
## hsa03320  72/7215 2.268848e-05 1.353746e-03 1.337426e-03
## hsa04914  96/7215 1.005904e-03 4.501419e-02 4.447154e-02
## hsa04115  69/7215 1.392048e-03 4.983532e-02 4.923454e-02
##                                                    geneID Count
## hsa04110 8318/991/9133/890/983/4085/7272/1111/891/4174/9232    11
## hsa04114    991/9133/983/4085/51806/6790/891/9232/3708/5241    10
## hsa03320                 4312/9415/9370/5105/2167/3158/5346     7
## hsa04914                         9133/890/983/4085/891/5241     6
## hsa04115                             9133/6241/983/1111/891     5
```

Input ID type can be `kegg`, `ncbi-geneid`, `ncbi-proteinid` or `uniprot`, an example can be found in the post.

6.2 KEGG Gene Set Enrichment Analysis

```
kk2 <- gseKEGG(geneList     = geneList,
               organism     = 'hsa',
               nPerm        = 1000,
               minGSSize    = 120,
               pvalueCutoff = 0.05,
               verbose      = FALSE)
head(kk2)
```

```
##                ID          Description setSize enrichmentScore       NES
## hsa04510 hsa04510       Focal adhesion     188      -0.4188582 -1.714863
## hsa03013 hsa03013        RNA transport     131       0.4116488  1.740923
## hsa05162 hsa05162              Measles     122       0.3938756  1.646892
## hsa05164 hsa05164          Influenza A     156       0.3651059  1.594990
## hsa05152 hsa05152         Tuberculosis     162       0.3745153  1.648998
## hsa05203 hsa05203 Viral carcinogenesis     164       0.3523856  1.549603
##                pvalue   p.adjust    qvalues rank
## hsa04510 0.001418440 0.01834532 0.01249527 2183
## hsa03013 0.003144654 0.01834532 0.01249527 3383
## hsa05162 0.003144654 0.01834532 0.01249527 2607
## hsa05164 0.003205128 0.01834532 0.01249527 2823
## hsa05152 0.003236246 0.01834532 0.01249527 2823
## hsa05203 0.003257329 0.01834532 0.01249527 3112
##                            leading_edge
## hsa04510 tags=27%, list=17%, signal=23%
## hsa03013 tags=40%, list=27%, signal=29%
## hsa05162 tags=35%, list=21%, signal=28%
## hsa05164 tags=35%, list=23%, signal=27%
## hsa05152 tags=34%, list=23%, signal=27%
## hsa05203 tags=35%, list=25%, signal=26%
##
## hsa04510                                    5228/7424/1499/4636/83660/7059/5295/1288/23396/3910/337
## hsa03013 10460/1978/55110/54913/9688/8894/11260/10799/9631/4116/5042/8761/6396/23165/8662/10248/
## hsa05162                                                                                  898/9134/459
## hsa05164            3627/3576/56649/6352/4599/6772/6347/3838/3126/3112/5645/91543/5646/4938/4940/
## hsa05152          820/51806/6772/64581/3126/3112/8767/3654/1054/1051/3458/1520/11151/1594/50617/
## hsa05203    991/890/983/1111/898/9134/6502/85236/1237/1029/8970/4067/2957/5902/55697/3066/578/10
```

6.3 KEGG Module over-representation test

KEGG Module is a collection of manually defined function units. In some situation, KEGG Modules have a more straightforward interpretation.

```
mkk <- enrichMKEGG(gene = gene,
                   organism = 'hsa')
```

6.4 KEGG Module Gene Set Enrichment Analysis

```
mkk2 <- gseMKEGG(geneList = geneList,
                 species = 'hsa')
```

7 Disease analysis

DOSE[5] supports Disease Ontology (DO) Semantic and Enrichment analysis. The `enrichDO` function is very useful for identifying disease association of interesting genes, and function `gseDO` function is designed for gene set enrichment analysis of *DO*.

In addition, *DOSE* also supports enrichment analysis of Network of Cancer Gene (NCG)[6] and Disease Gene Network[7], please refer to the *DOSE* vignettes.

8 Reactome pathway analysis

ReactomePA[8] uses Reactome as a source of pathway data. The function call of `enrichPathway` and `gsePathway` in *ReactomePA* is consistent with `enrichKEGG` and `gseKEGG`.

9 DAVID functional analysis

clusterProfiler provides enrichment and GSEA analysis with GO, KEGG, DO and Reactome pathway supported internally, some user may prefer GO and KEGG analysis with DAVID[9] and still attracted by the visualization methods provided by *clusterProfiler*[???]. To bridge the gap between DAVID and clusterProfiler, we implemented `enrichDAVID`. This function query enrichment analysis result from DAVID webserver via *RDAVIDWebService*[10] and stored the result as an `enrichResult` instance, so that we can use all the visualization functions in *clusterProfiler* to visualize DAVID results. `enrichDAVID` is fully compatible with `compareCluster` function and comparing enrichment results from different gene clusters is now available with DAVID.

```
david <- enrichDAVID(gene = gene,
                     idType = "ENTREZ_GENE_ID",
                     listType = "Gene",
                     annotation = "KEGG_PATHWAY",
                     david.user = "clusterProfiler@hku.hk")
```

DAVID Web Service has the following limitations:

- A job with more than 3000 genes to generate gene or term cluster report will not be handled by DAVID due to resource limit.
- No more than 200 jobs in a day from one user or computer.
- DAVID Team reserves right to suspend any improper uses of the web service without notice.

For more details, please refer to http://david.abcc.ncifcrf.gov/content.jsp?file=WS.html.

As user has limited usage, please register and use your own user account to run `enrichDAVID`.

10 Universal enrichment analysis

clusterProfiler supports both hypergeometric test and gene set enrichment analyses of many ontology/pathway, but it's still not enough for users may want to analyze their data with unsupported organisms, slim version of GO, novel functional annotation (e.g. GO via BlastGO or KEGG via KAAS), unsupported ontologies/pathways or customized annotations.

clusterProfiler provides `enricher` function for hypergeometric test and `GSEA` function for gene set enrichment analysis that are designed to accept user defined annotation. They accept two additional parameters *TERM2GENE* and *TERM2NAME*. As indicated in the parameter names, *TERM2GENE* is a data.frame with first column of term ID and second column of corresponding mapped gene and *TERM2NAME* is a data.frame with first column of term ID and second column of corresponding term name. *TERM2NAME* is optional.

An example of using `enricher` and `GSEA` to analyze DisGeNet annotation is presented in the post, use clusterProfiler as an universal enrichment analysis tool.

10.1 Using MSigDB gene set collections

The MSigDB is a collection of annotated gene sets, it include 8 major collections:

- H: hallmark gene sets
- C1: positional gene sets
- C2: curated gene sets
- C3: motif gene sets
- C4: computational gene sets
- C5: GO gene sets
- C6: oncogenic signatures
- C7: immunologic signatures

Users can use `enricher` and `GSEA` function to analyze gene set collections downloaded from Molecular Signatures Database (MSigDb). *clusterProfiler* provides a function, `read.gmt`, to parse the gmt file into a *TERM2GENE* `data.frame` that is ready for both `enricher` and `GSEA` functions.

```
gmtfile <- system.file("extdata", "c5.cc.v5.0.entrez.gmt", package="clusterProfiler")
c5 <- read.gmt(gmtfile)

egmt <- enricher(gene, TERM2GENE=c5)
head(egmt)
```

```
##                                           ID              Description
## SPINDLE                                SPINDLE                  SPINDLE
## MICROTUBULE_CYTOSKELETON MICROTUBULE_CYTOSKELETON MICROTUBULE_CYTOSKELETON
## CYTOSKELETAL_PART              CYTOSKELETAL_PART        CYTOSKELETAL_PART
## SPINDLE_MICROTUBULE          SPINDLE_MICROTUBULE      SPINDLE_MICROTUBULE
## MICROTUBULE                        MICROTUBULE              MICROTUBULE
## CYTOSKELETON                       CYTOSKELETON             CYTOSKELETON
##                            GeneRatio  BgRatio       pvalue     p.adjust
## SPINDLE                        11/82  39/5270 7.667674e-12 5.214018e-10
## MICROTUBULE_CYTOSKELETON       16/82 152/5270 8.449298e-10 2.872761e-08
## CYTOSKELETAL_PART              15/82 235/5270 2.414879e-06 5.237096e-05
## SPINDLE_MICROTUBULE             5/82  16/5270 3.080645e-06 5.237096e-05
## MICROTUBULE                     6/82  32/5270 7.740446e-06 1.052701e-04
## CYTOSKELETON                   16/82 367/5270 1.308357e-04 1.482805e-03
##                                 qvalue
## SPINDLE                    4.197043e-10
## MICROTUBULE_CYTOSKELETON   2.312439e-08
## CYTOSKELETAL_PART          4.215619e-05
## SPINDLE_MICROTUBULE        4.215619e-05
## MICROTUBULE                8.473751e-05
## CYTOSKELETON               1.193589e-03
##
## SPINDLE                                                 991/9493/9787/22974/983/332/3832/7272/9055/679
## MICROTUBULE_CYTOSKELETON 991/9493/9133/7153/9787/22974/4751/983/332/3832/7272/9055/6790/24137/41
## CYTOSKELETAL_PART              991/9493/7153/9787/22974/4751/983/332/3832/7272/9055/6790/24137/41
## SPINDLE_MICROTUBULE                                                                  983/332/3832/905
## MICROTUBULE                                                                       983/332/3832/9055/241
## CYTOSKELETON               991/9493/9133/7153/9787/22974/4751/983/332/3832/7272/9055/6790/24137/41
##                            Count
## SPINDLE                       11
## MICROTUBULE_CYTOSKELETON      16
## CYTOSKELETAL_PART             15
## SPINDLE_MICROTUBULE            5
## MICROTUBULE                    6
## CYTOSKELETON                  16
```

```
egmt2 <- GSEA(geneList, TERM2GENE=c5, verbose=FALSE)
head(egmt2)
```

```
##                                                                 ID
## EXTRACELLULAR_REGION                           EXTRACELLULAR_REGION
## EXTRACELLULAR_REGION_PART                 EXTRACELLULAR_REGION_PART
## PROTEINACEOUS_EXTRACELLULAR_MATRIX PROTEINACEOUS_EXTRACELLULAR_MATRIX
## CELL_PROJECTION                                 CELL_PROJECTION
## EXTRACELLULAR_MATRIX                           EXTRACELLULAR_MATRIX
## EXTRACELLULAR_MATRIX_PART                 EXTRACELLULAR_MATRIX_PART
##                                                         Description
## EXTRACELLULAR_REGION                           EXTRACELLULAR_REGION
## EXTRACELLULAR_REGION_PART                 EXTRACELLULAR_REGION_PART
## PROTEINACEOUS_EXTRACELLULAR_MATRIX PROTEINACEOUS_EXTRACELLULAR_MATRIX
## CELL_PROJECTION                                 CELL_PROJECTION
## EXTRACELLULAR_MATRIX                           EXTRACELLULAR_MATRIX
## EXTRACELLULAR_MATRIX_PART                 EXTRACELLULAR_MATRIX_PART
##                                    setSize enrichmentScore       NES
## EXTRACELLULAR_REGION                   401      -0.3860230 -1.704037
## EXTRACELLULAR_REGION_PART              310      -0.4101043 -1.768773
## PROTEINACEOUS_EXTRACELLULAR_MATRIX      93      -0.6355317 -2.341055
## CELL_PROJECTION                         87      -0.4729701 -1.737519
## EXTRACELLULAR_MATRIX                    95      -0.6229461 -2.297479
## EXTRACELLULAR_MATRIX_PART               54      -0.5908035 -1.994617
##                                       pvalue   p.adjust    qvalues rank
## EXTRACELLULAR_REGION               0.001225490 0.03190789 0.02493075 1797
## EXTRACELLULAR_REGION_PART          0.001298701 0.03190789 0.02493075 1897
## PROTEINACEOUS_EXTRACELLULAR_MATRIX 0.001510574 0.03190789 0.02493075 1473
## CELL_PROJECTION                    0.001515152 0.03190789 0.02493075 2280
## EXTRACELLULAR_MATRIX               0.001519757 0.03190789 0.02493075 1473
## EXTRACELLULAR_MATRIX_PART          0.001597444 0.03190789 0.02493075 1794
##                                                        leading_edge
## EXTRACELLULAR_REGION               tags=29%, list=14%, signal=26%
## EXTRACELLULAR_REGION_PART          tags=32%, list=15%, signal=28%
## PROTEINACEOUS_EXTRACELLULAR_MATRIX tags=49%, list=12%, signal=44%
## CELL_PROJECTION                    tags=28%, list=18%, signal=23%
## EXTRACELLULAR_MATRIX               tags=48%, list=12%, signal=43%
## EXTRACELLULAR_MATRIX_PART          tags=59%, list=14%, signal=51%
##
## EXTRACELLULAR_REGION               3910/51162/2878/2717/3373/4153/10406/1301/6750/7474/4925/7450,
## EXTRACELLULAR_REGION_PART
## PROTEINACEOUS_EXTRACELLULAR_MATRIX
## CELL_PROJECTION
## EXTRACELLULAR_MATRIX
## EXTRACELLULAR_MATRIX_PART
```

11 Functional analysis of NGS data

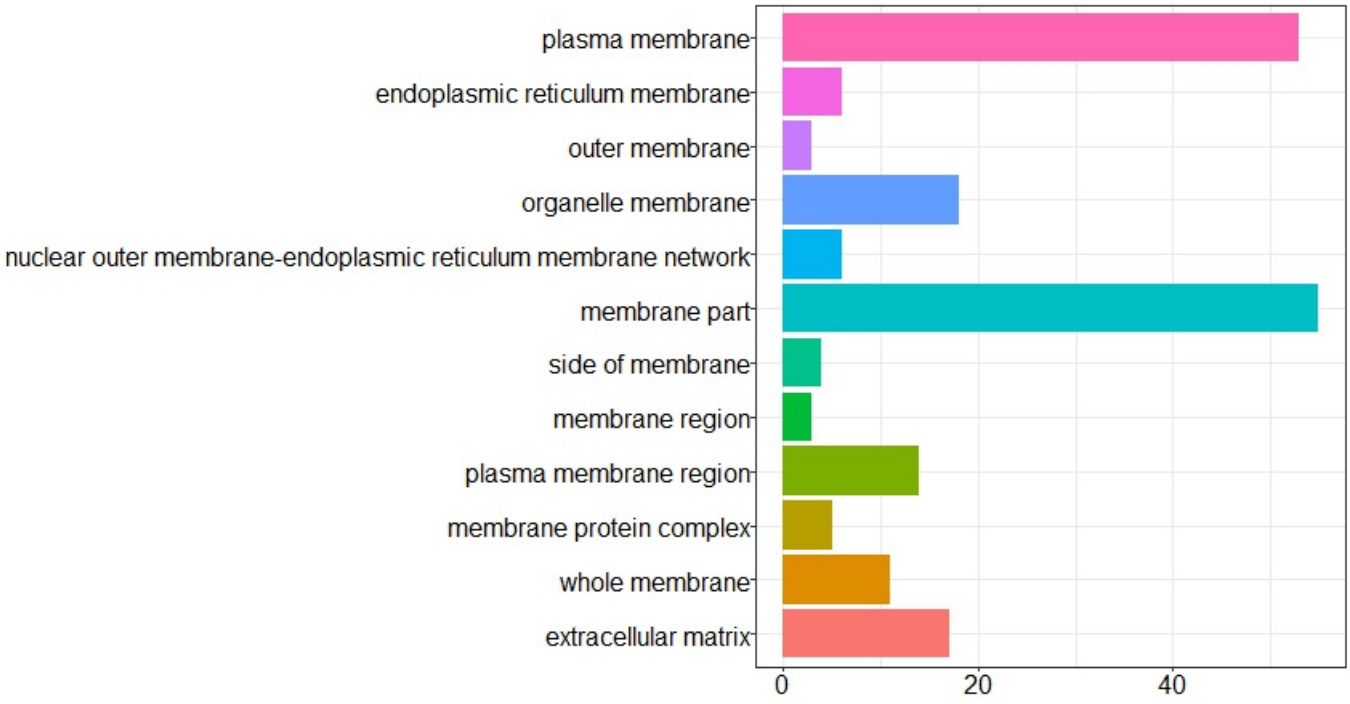
Functional analysis using NGS data (eg, RNA-Seq and ChIP-Seq) can be performed by linking coding and non-coding regions to coding genes via *ChIPseeker*[11] package, which can annotates genomic regions to their nearest genes, host genes, and flanking genes respectivly. In addtion, it provides a function, `seq2gene`, that simultaneously considering host genes, promoter region and flanking gene from intergenic region that may under control via cis-regulation. This function maps genomic regions to genes in a many-to-many manner and facilitate functional analysis. For more details, please refer to *ChIPseeker*.

12 Visualization

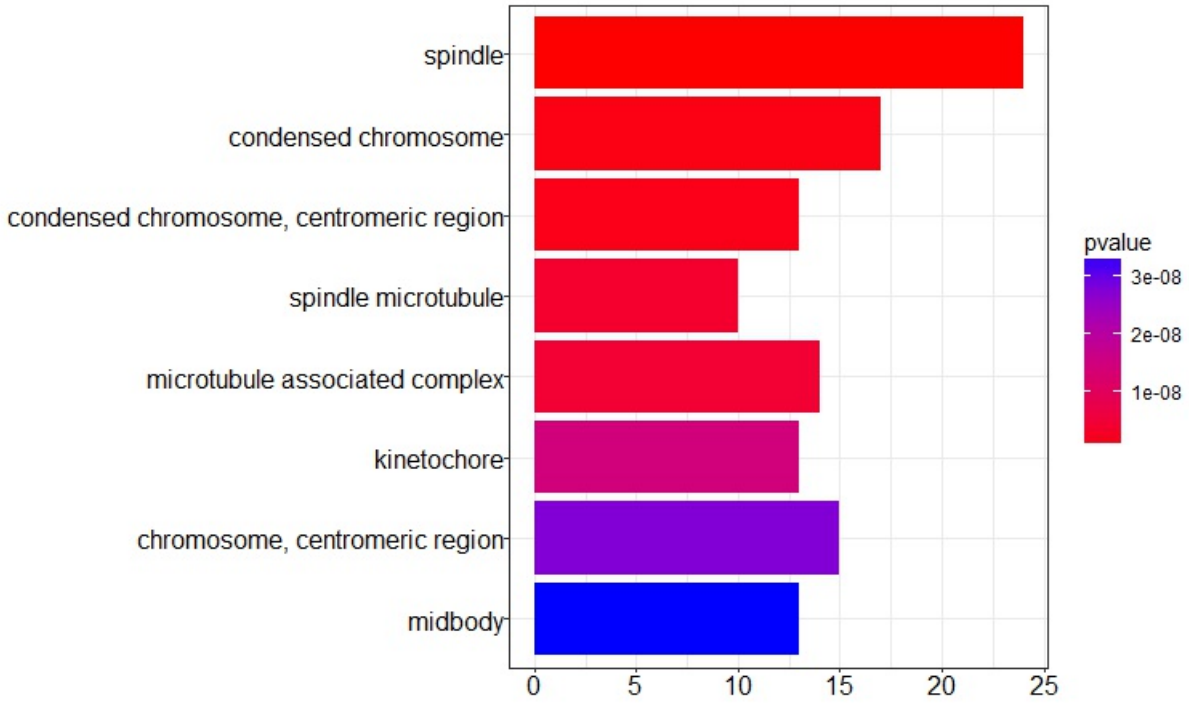
The function calls of `groupGO`, `enrichGO`, `enrichKEGG`, `enrichDO`, `enrichPathway` and `enricher` are consistent and all the output can be visualized by bar plot, enrichment map and category-gene-network plot. It is very common to visualize the enrichment result in bar or pie chart. We believe the pie chart is misleading and only provide bar chart.

12.1 barplot

```
barplot(ggo, drop=TRUE, showCategory=12)
```



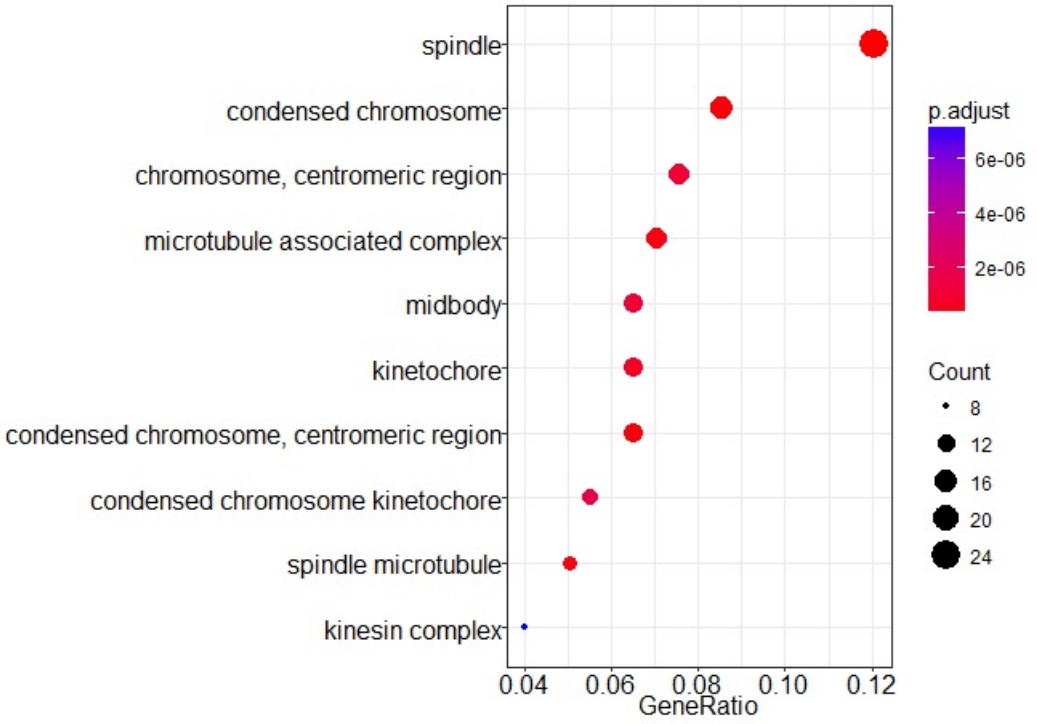

```
barplot(ego, showCategory=8)
```



12.2 dotplot

dotplot is a good alternative to `barplot`.

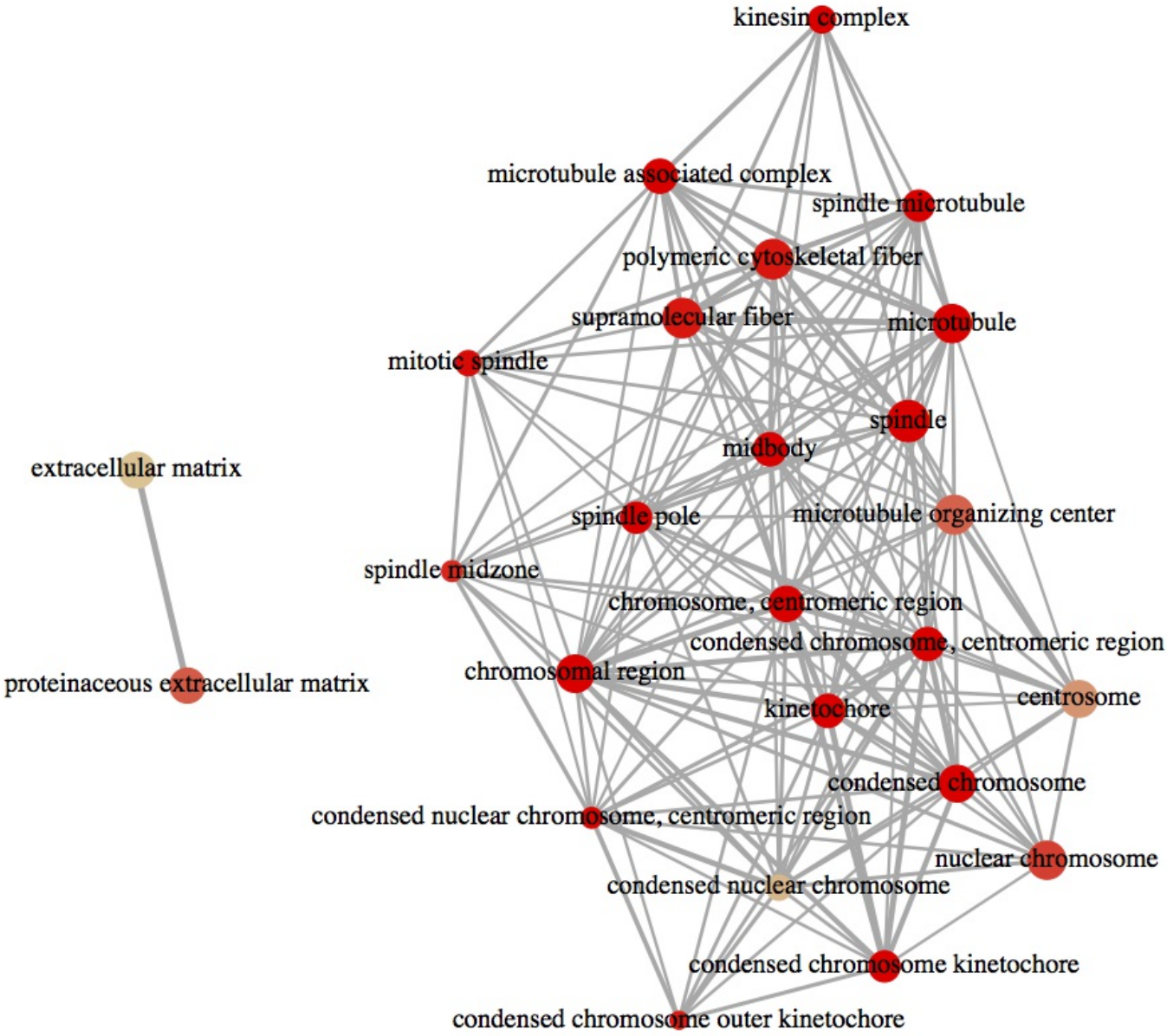
```
dotplot(ego)
```



12.3 enrichMap

Enrichment map can be viusalized by `enrichMap`, which also support results obtained from hypergeometric test and gene set enrichment analysis.

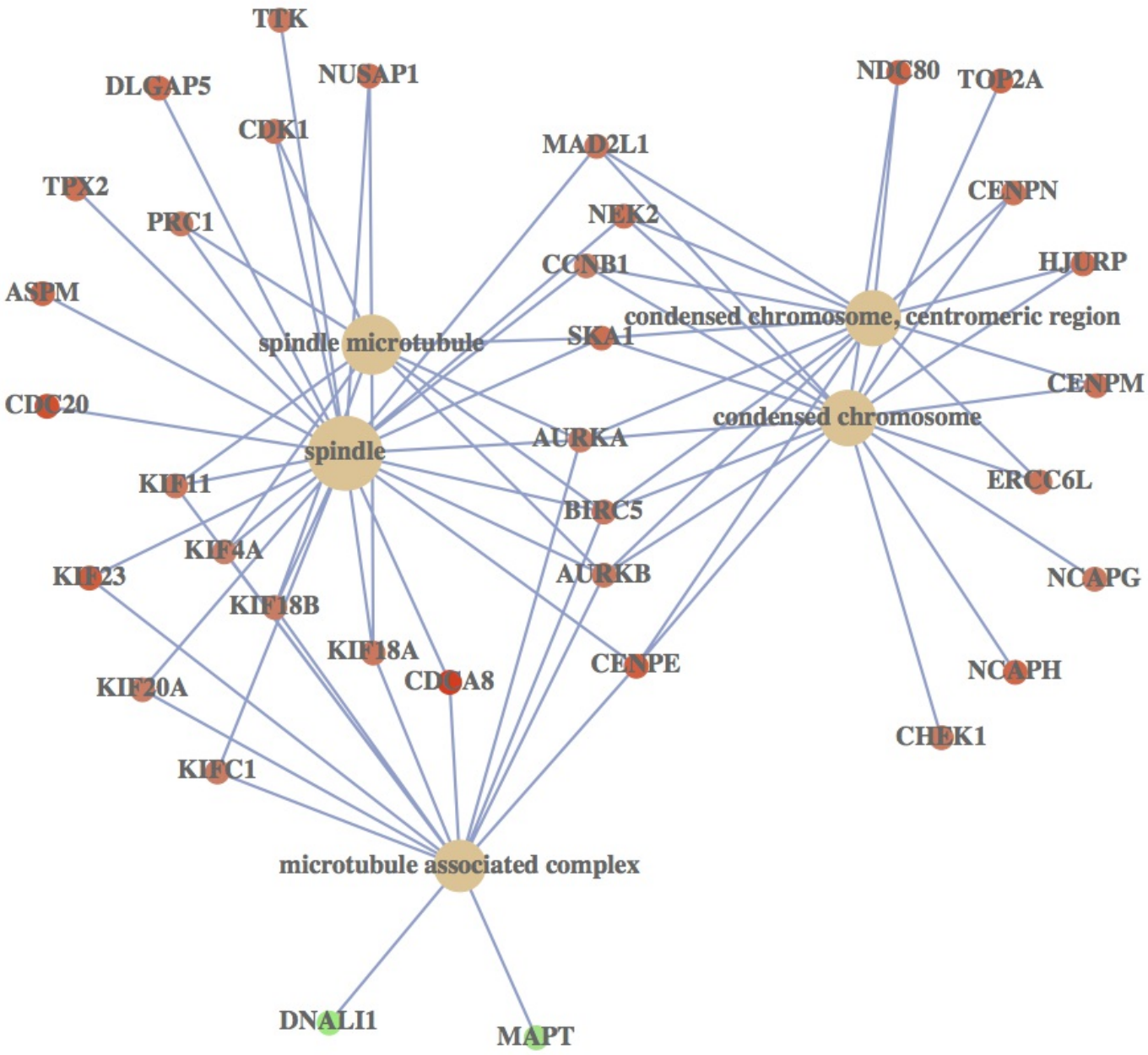
```
enrichMap(ego)
```



12.4 cnetplot

In order to consider the potentially biological complexities in which a gene may belong to multiple annotation categories and provide information of numeric changes if available, we developed `cnetplot` function to extract the complex association.

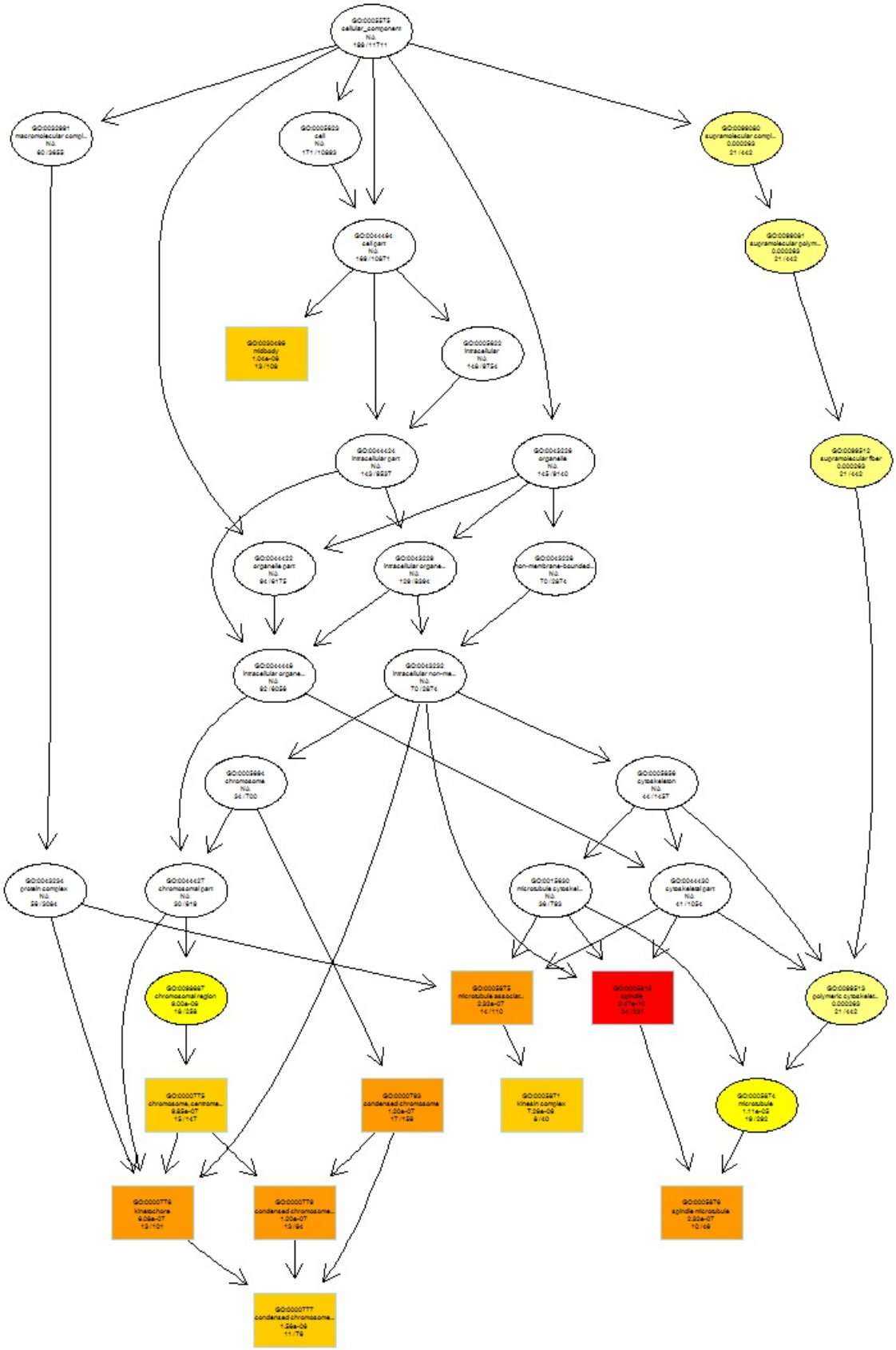
```
## categorySize can be scaled by 'pvalue' or 'geneNum'
cnetplot(ego, categorySize="pvalue", foldChange=geneList)
```



12.5 plotGOgraph

`plotGOgraph`, which is based on *topGO*, can accept output of `enrichGO` and visualized the enriched GO induced graph.

```
plotGOgraph(ego)
```

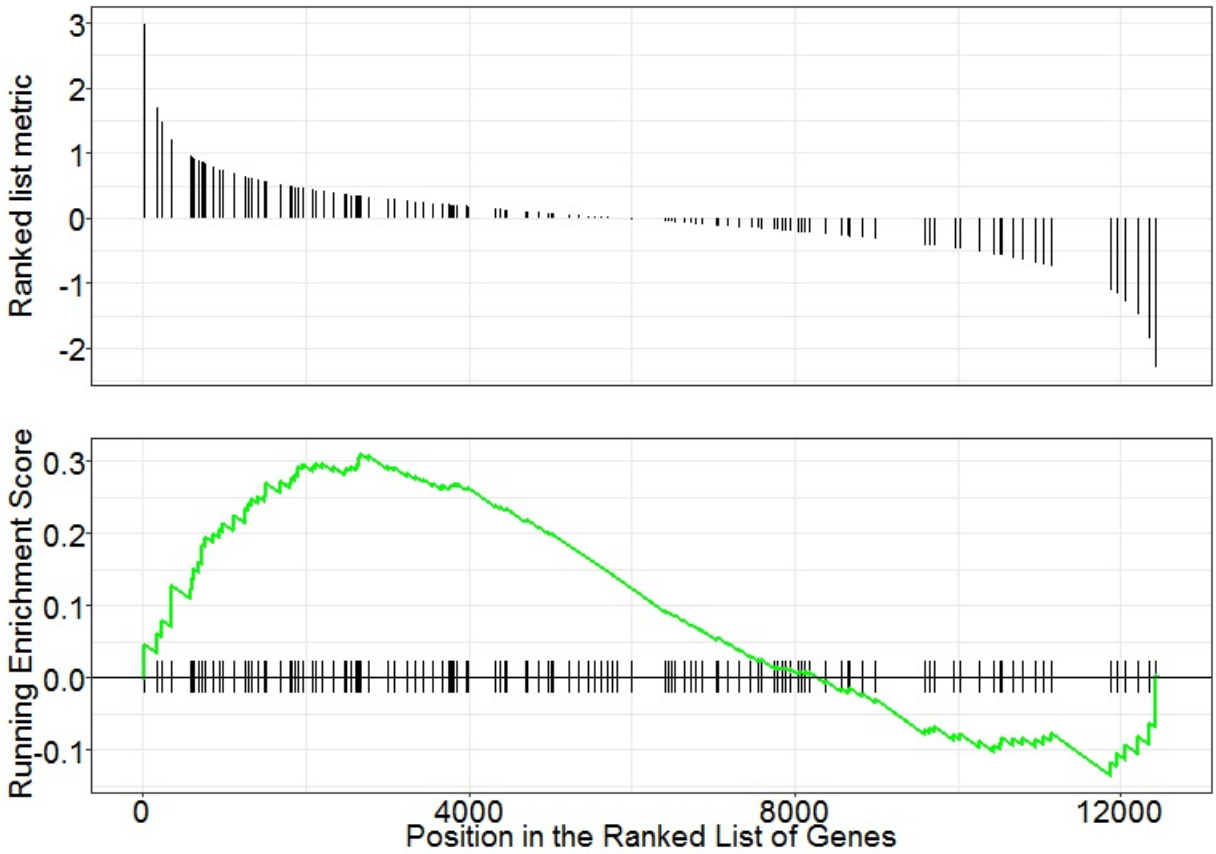


```
## $dag
## A graphNEL graph with directed edges
## Number of Nodes = 34
## Number of Edges = 56
##
## $complete.dag
## [1] "A graph with 34 nodes."
```

12.6 gseaplot

Running score of gene set enrichment analysis and its association of phenotype can be visualized by `gseaplot`.

```
gseaplot(kk2, geneSetID = "hsa04145")
```

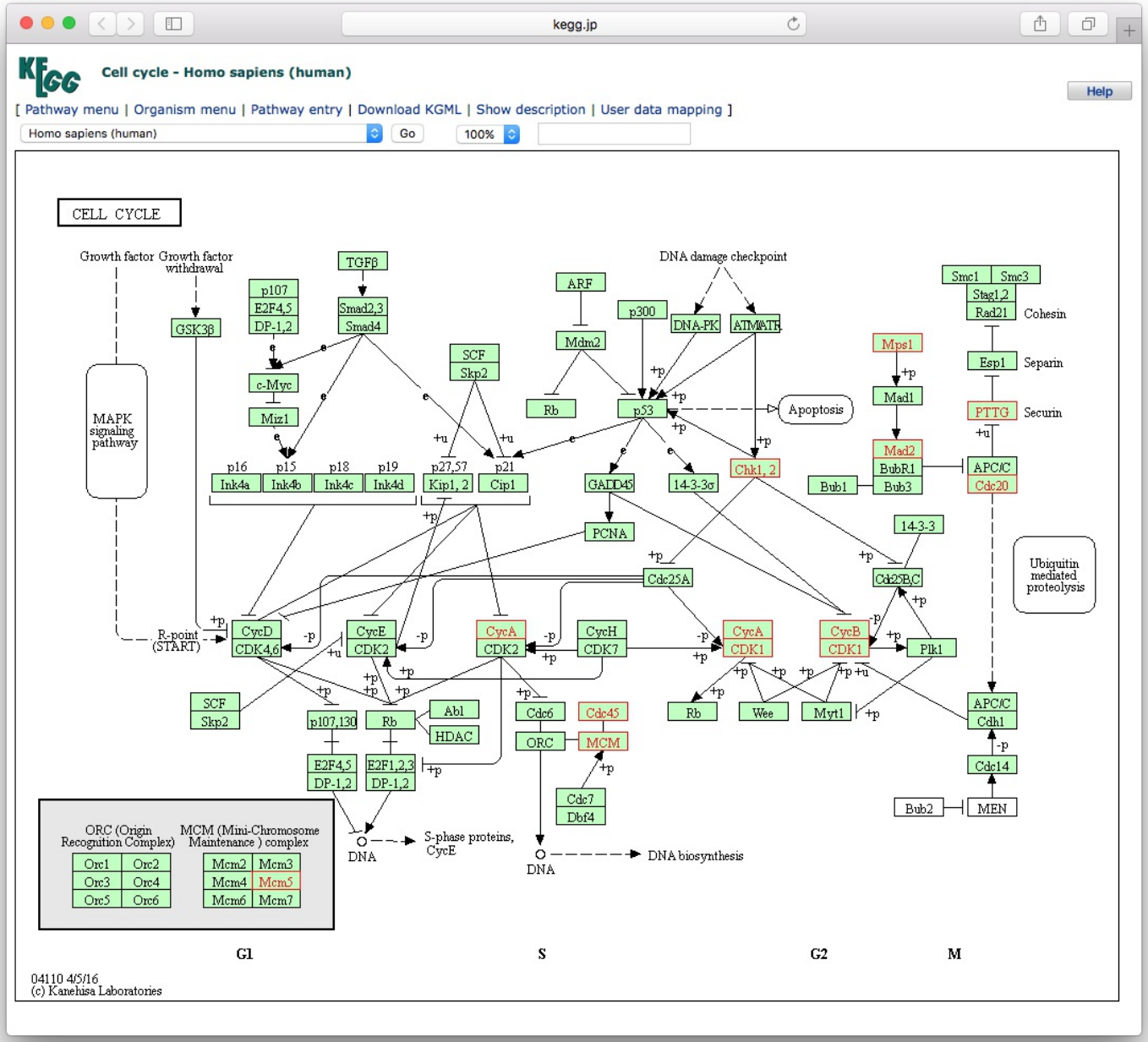


plotting gsea result

12.7 browseKEGG

To view the KEGG pathway, user can use `browseKEGG` function, which will open web browser and highlight enriched genes.

```
browseKEGG(kk, 'hsa04110')
```

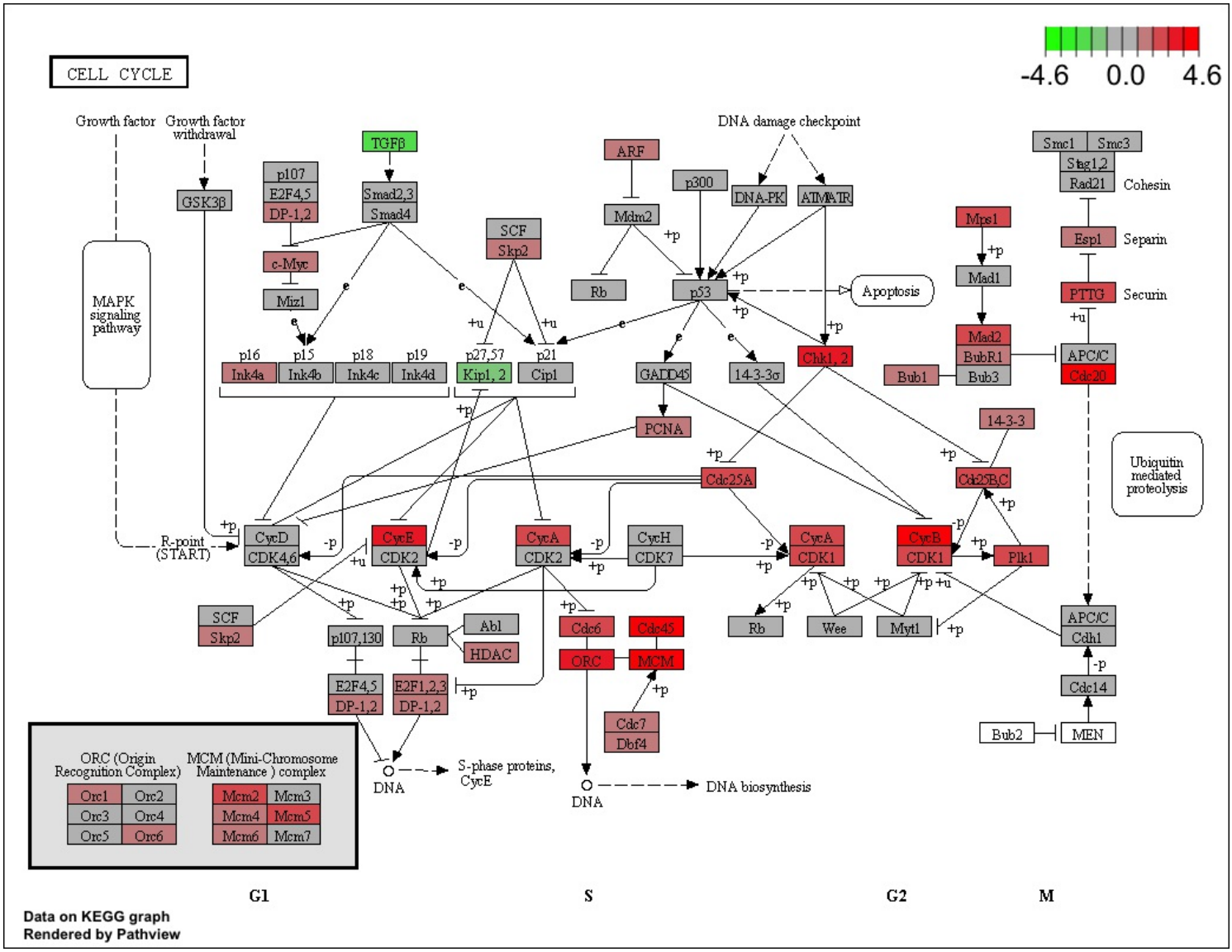


12.8 pathview from pathview package

clusterProfiler users can also use `pathview` from the *pathview*[12] to visualize KEGG pathway.

The following example illustrate how to visualize "hsa04110" pathway, which was enriched in our previous analysis.

```
library("pathview")
hsa04110 <- pathview(gene.data  = geneList,
                     pathway.id = "hsa04110",
                     species    = "hsa",
                     limit      = list(gene=max(abs(geneList)), cpd=1))
```



For further information, please refer to the vignette of *pathview*[12].

13 Biological theme comparison

clusterProfiler was developed for biological theme comparison[2], and it provides a function, `compareCluster`, to automatically calculate enriched functional categories of each gene clusters.

```
data(gcSample)
lapply(gcSample, head)
```

```
## $X1
## [1] "4597"  "7111"  "5266"  "2175"  "755"   "23046"
##
## $X2
## [1] "23450" "5160"  "7126"  "26118" "8452"  "3675"
##
## $X3
## [1] "894"   "7057"  "22906" "3339"  "10449" "6566"
##
## $X4
## [1] "5573"  "7453"  "5245"  "23450" "6500"  "4926"
##
## $X5
## [1] "5982" "7318" "6352" "2101" "8882" "7803"
##
## $X6
## [1] "5337"  "9295"  "4035"  "811"   "23365" "4629"
##
## $X7
## [1] "2621" "2665" "5690" "3608" "3550" "533"
##
## $X8
## [1] "2665" "4735" "1327" "3192" "5573" "9528"
```

The input for *geneCluster* parameter should be a named list of gene IDs. To speed up the compilation of this document, we set `use_internal_data = TRUE`.

```
ck <- compareCluster(geneCluster = gcSample, fun = "enrichKEGG")
head(as.data.frame(ck))
```

```
##   Cluster       ID                  Description GeneRatio  BgRatio
## 1      X2 hsa04110                   Cell cycle    18/355 124/7215
## 2      X2 hsa05340     Primary immunodeficiency     8/355  37/7215
## 3      X2 hsa05200           Pathways in cancer    35/355 395/7215
## 4      X2 hsa04064 NF-kappa B signaling pathway    13/355  95/7215
## 5      X3 hsa04512     ECM-receptor interaction     9/168  82/7215
## 6      X4 hsa04110                   Cell cycle    20/378 124/7215
##         pvalue    p.adjust      qvalue
## 1 3.166880e-05 0.008550575 0.008100545
## 2 3.488642e-04 0.041268238 0.039096225
## 3 4.585360e-04 0.041268238 0.039096225
## 4 7.093644e-04 0.047882099 0.045361988
## 5 1.107710e-04 0.026252736 0.024253027
## 6 5.765456e-06 0.001377171 0.001129318
##
## 1                                                                              991/1869/890/18
## 2
## 3 3675/1956/1869/324/3480/1871/113/1902/2261/1909/637/355/5888/9134/5915/3908/2246/5154/7704/443
## 4
## 5
## 6                                                                      6500/9184/4172/994/4175/
##   Count
## 1    18
## 2     8
## 3    35
## 4    13
## 5     9
## 6    20
```

13.1 Formula interface of compareCluster

`compareCluster` also supports passing a formula (the code to support formula has been contributed by Giovanni Dall'Olio) of type \(Entrez \sim group\) or \(Entrez \sim group + othergroup\).

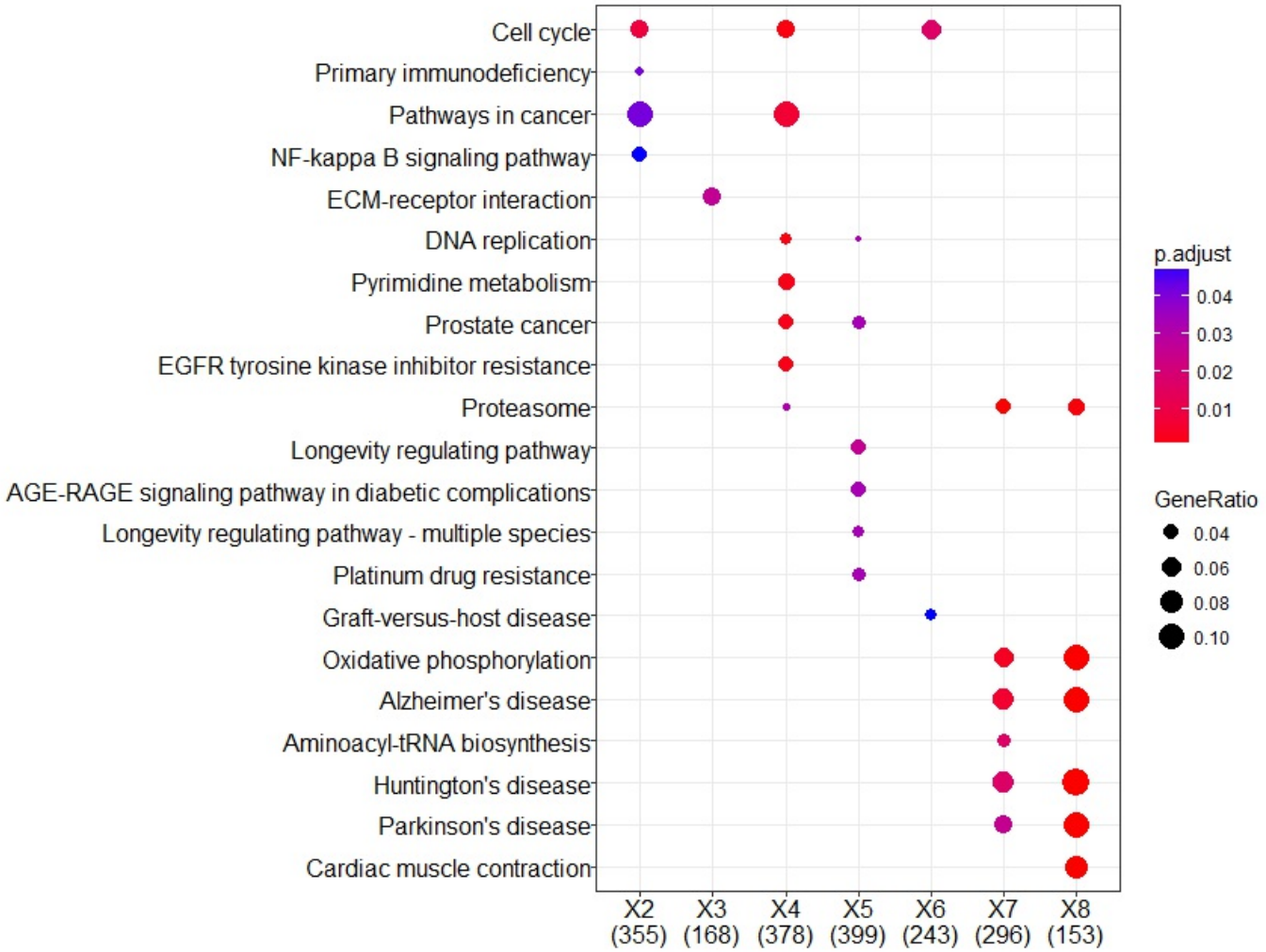
```
mydf <- data.frame(Entrez=names(geneList), FC=geneList)
mydf <- mydf[abs(mydf$FC) > 1,]
mydf$group <- "upregulated"
mydf$group[mydf$FC < 0] <- "downregulated"
mydf$othergroup <- "A"
mydf$othergroup[abs(mydf$FC) > 2] <- "B"

formula_res <- compareCluster(Entrez~group+othergroup, data=mydf, fun="enrichKEGG")

head(as.data.frame(formula_res))
```

```
##           Cluster         group othergroup       ID
## 1 downregulated.A downregulated          A hsa04974
## 2 downregulated.A downregulated          A hsa04510
## 3 downregulated.A downregulated          A hsa04512
## 4 downregulated.A downregulated          A hsa05414
## 5 downregulated.B downregulated          B hsa03320
## 6   upregulated.A   upregulated          A hsa04110
##                         Description GeneRatio  BgRatio       pvalue
## 1 Protein digestion and absorption    15/276  90/7215 1.299669e-06
## 2                   Focal adhesion    20/276 199/7215 6.801713e-05
## 3         ECM-receptor interaction    11/276  82/7215 2.625194e-04
## 4           Dilated cardiomyopathy    11/276  90/7215 5.949151e-04
## 5           PPAR signaling pathway      5/39  72/7215 3.802971e-05
## 6                       Cell cycle    20/210 124/7215 3.234258e-10
##       p.adjust       qvalue
## 1 3.197186e-04 2.955037e-04
## 2 8.366108e-03 7.732474e-03
## 3 2.152659e-02 1.989621e-02
## 4 3.658728e-02 3.381622e-02
## 5 4.981892e-03 4.723690e-03
## 6 6.824285e-08 6.162113e-08
##
## 1                              1281/50509/1290/477/1294/1360/1289/1292/23428/1359/1300/1287/6505/
## 2 55742/2317/7058/25759/56034/3693/3480/5159/857/1292/3908/3909/63923/3913/1287/3679/7060/3479/10
## 3                                                    7058/3693/3339/1292/3908/3909/63923/3913/1287/
## 4                                                    55799/27092/6444/3693/775/3908/5350/7043/3679/
## 5                                                                                  9370/5105/2167/
## 6          4171/993/990/5347/701/9700/898/23594/4998/9134/4175/4173/10926/6502/994/699/4609/5111/
##   Count
## 1    15
## 2    20
## 3    11
## 4    11
## 5     5
## 6    20
```

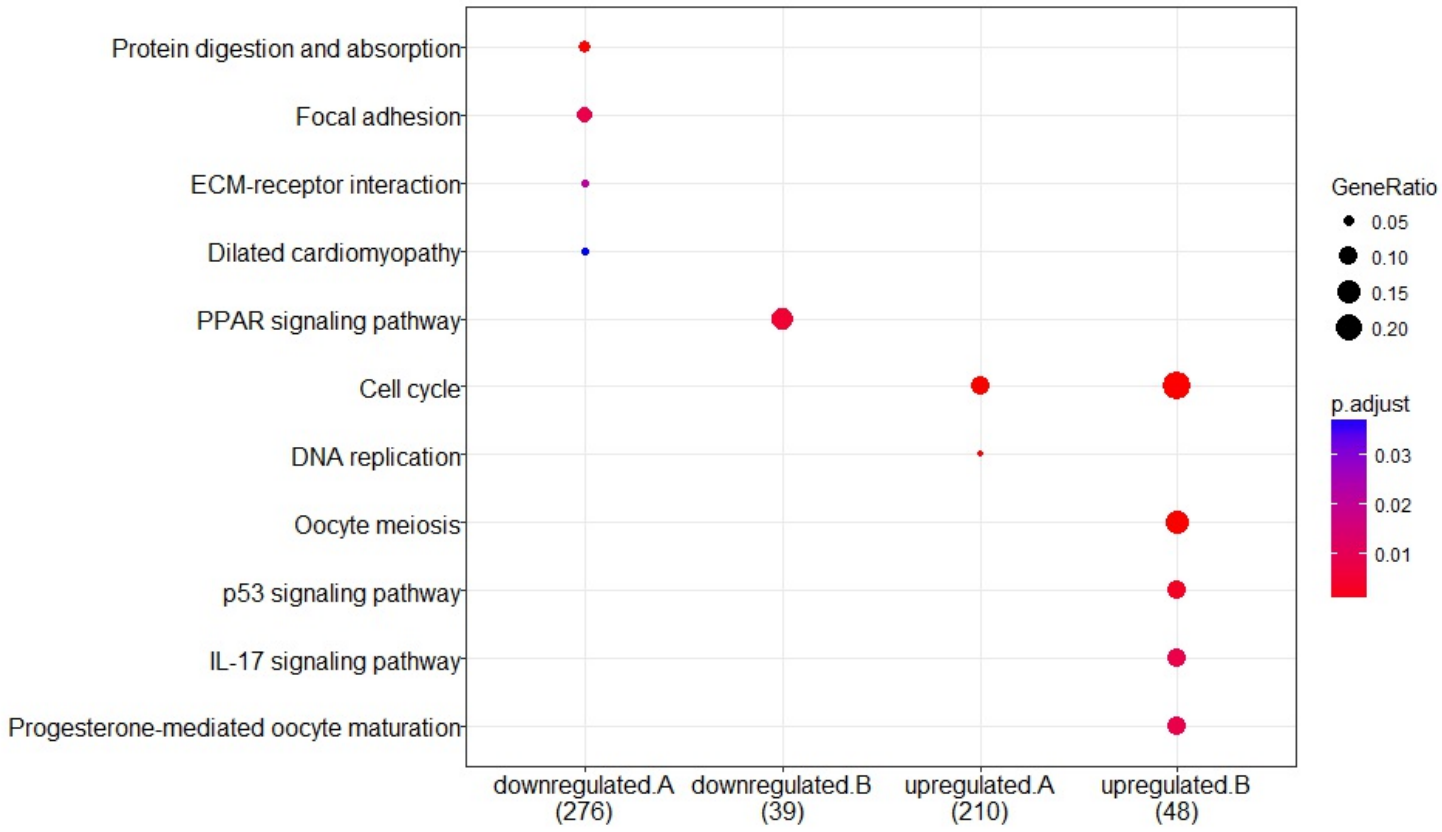
13.2 Visualization of profile comparison

We can visualize the result using `dotplot` method.

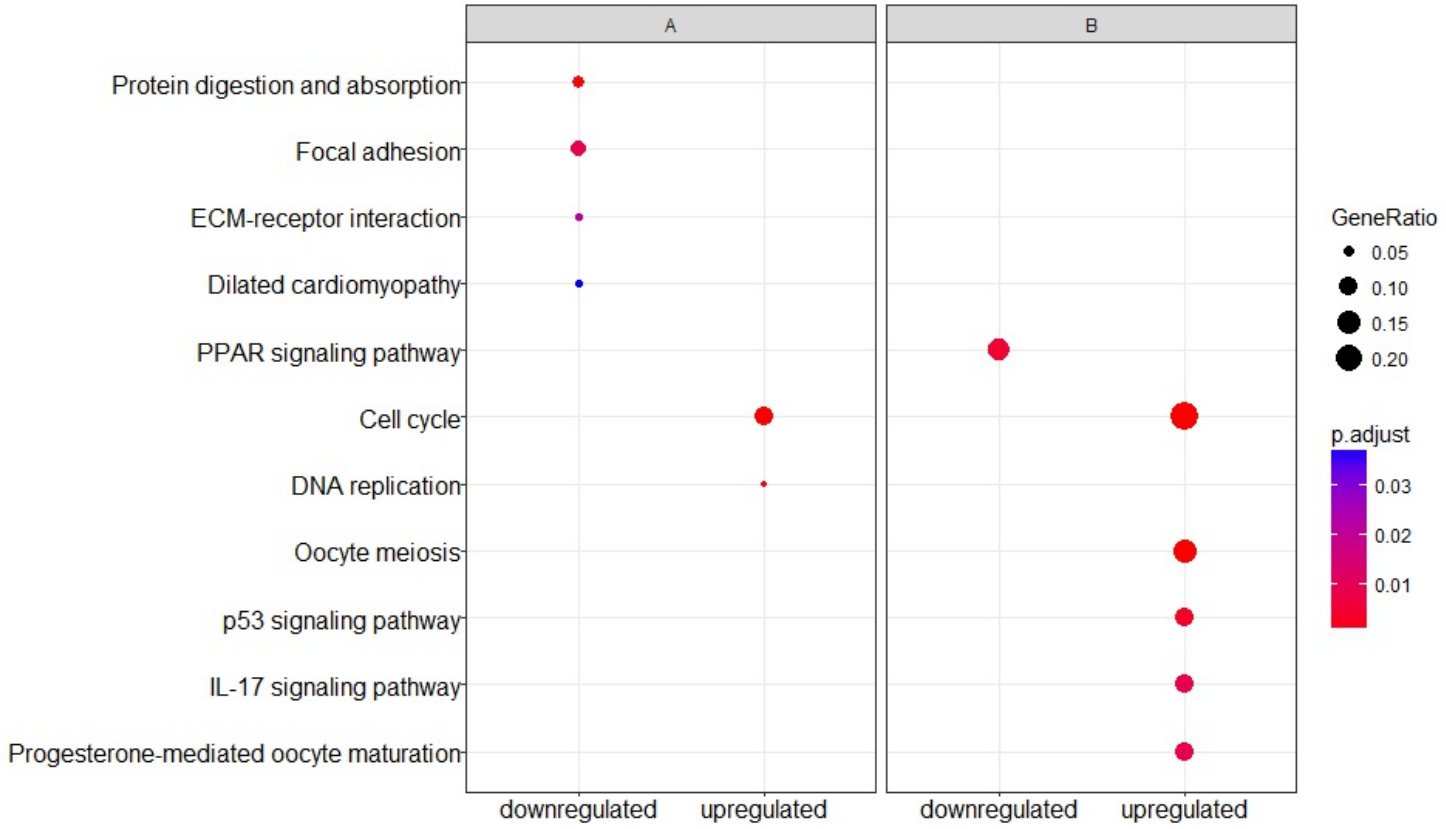
```
dotplot(ck)
```




```
dotplot(formula_res)
```



```
dotplot(formula_res, x=~group) + ggplot2::facet_grid(~othergroup)
```



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

The *plot* function accepts a parameter *by* for setting the scale of dot sizes. The default parameter *by* is setting to "geneRatio", which corresponding to the "GeneRatio" column of the output. If it was setting to *count*, the comparison will be based on gene counts, while if setting to *rowPercentage*, the dot sizes will be normalized by *count/(sum of each row)*

To provide the full information, we also provide number of identified genes in each category (numbers in parentheses) when *by* is setting to *rowPercentage* and number of gene clusters in each cluster label (numbers in parentheses) when *by* is setting to *geneRatio*, as shown in Figure 3. If the dot sizes were based on *count*, the row numbers will not shown.

The p-values indicate that which categories are more likely to have biological meanings. The dots in the plot are color-coded based on their corresponding p-values. Color gradient ranging from red to blue correspond to in order of increasing p-values. That is, red indicate low p-values (high enrichment), and blue indicate high p-values (low enrichment). P-values and adjusted p-values were filtered out by the threshold giving by parameter *pvalueCutoff*, and FDR can be estimated by *qvalue*.

User can refer to the example in Yu (2012)[2]; we analyzed the publicly available expression dataset of breast tumour tissues from 200 patients (GSE11121, Gene Expression Omnibus)[13]. We identified 8 gene clusters from differentially expressed genes, and using `compareCluster` to compare these gene clusters by their enriched biological process.

The comparison function was designed as a framework for comparing gene clusters of any kind of ontology associations, not only `groupGO`, `enrichGO`, `enrichKEGG` and `enricher` provided in this package, but also other biological and biomedical ontologies, for instance, `enrichDO` from *DOSE*[5] and `enrichPathway` from *ReactomePA* work fine with `compareCluster` for comparing biological themes in disease and reactome pathway perspective. More details can be found in the vignettes of *DOSE*[5] and *ReactomePA*.

14 Homepage

Please visit clusterProfiler homepage for more information.

15 Session Information

Here is the output of `sessionInfo()` on the system on which this document was compiled:

```
## R version 3.3.2 (2016-10-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows Server 2012 R2 x64 (build 9600)
##
## 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      parallel  stats4    stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] Rgraphviz_2.18.0       clusterProfiler_3.2.14 GSEABase_1.36.0
##  [4] annotate_1.52.1        XML_3.98-1.5           topGO_2.26.0
##  [7] SparseM_1.74           graph_1.52.0           org.Hs.eg.db_3.4.0
## [10] GO.db_3.4.0            AnnotationDbi_1.36.2   IRanges_2.8.1
## [13] S4Vectors_0.12.1       Biobase_2.34.0         BiocGenerics_0.20.0
## [16] DOSE_3.0.10            BiocStyle_2.2.1
##
## loaded via a namespace (and not attached):
##  [1] qvalue_2.6.0       fgsea_1.0.2        reshape2_1.4.2
##  [4] splines_3.3.2      lattice_0.20-34    colorspace_1.3-2
##  [7] htmltools_0.3.5    yaml_2.1.14        DBI_0.5-1
## [10] BiocParallel_1.8.1 matrixStats_0.51.0 plyr_1.8.4
## [13] stringr_1.2.0      munsell_0.4.3      GOSemSim_2.0.4
## [16] gtable_0.2.0       memoise_1.0.0      evaluate_0.10
## [19] labeling_0.3       knitr_1.15.1       highr_0.6
## [22] Rcpp_0.12.9        xtable_1.8-2       scales_0.4.1
## [25] backports_1.0.5    DO.db_2.9          gridExtra_2.2.1
## [28] fastmatch_1.1-0    ggplot2_2.2.1      digest_0.6.12
## [31] stringi_1.1.2      rprojroot_1.2      tools_3.3.2
## [34] bitops_1.0-6       magrittr_1.5       lazyeval_0.2.0
## [37] RCurl_1.95-4.8     tibble_1.2         RSQLite_1.1-2
## [40] tidyr_0.6.1        data.table_1.10.4  assertthat_0.1
## [43] rmarkdown_1.3      igraph_1.0.1
```

References

1. Yu, G. *et al.* GOSemSim: An r package for measuring semantic similarity among go terms and gene products. *Bioinformatics* **26,** 976–978 (2010).

2. Yu, G., Wang, L.-G., Han, Y. & He, Q.-Y. ClusterProfiler: An r package for comparing biological themes among gene clusters. *OMICS: A Journal of Integrative Biology* **16,** 284–287 (2012).

3. Boyle, E. I. *et al.* GO::TermFinder–open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. *Bioinformatics (Oxford, England)* **20,** 3710–3715 (2004).

4. Subramanian, A. *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America* **102,** 15545–15550 (2005).

5. Yu, G., Wang, L.-G., Yan, G.-R. & He, Q.-Y. DOSE: An r/bioconductor package for disease ontology semantic and enrichment analysis. *Bioinformatics* **31,** 608–609 (2015).

6. A., O., D., G. M., M., T. P. & C., F. D. NCG 5.0: Updates of a manually curated repository of cancer genes and associated properties from cancer mutational screenings. *Nucleic Acids Research* **44,** D992–D999 (2016).

7. Janet, P. *et al.* DisGeNET: A discovery platform for the dynamical exploration of human diseases and their genes. *Database* **2015,** bav028 (2015).

8. Yu, G. & He, Q.-Y. ReactomePA: An r/bioconductor package for reactome pathway analysis and visualization. *Mol. BioSyst.* **12,** 477–479 (2016).

9. Huang, D. *et al.* The DAVID gene functional classification tool: A novel biological module-centric algorithm to functionally analyze large gene lists. *Genome Biology* **8,** R183 (2007).

10. Fresno, C. & Fernández, E. A. RDAVIDWebService: A versatile r interface to DAVID. *Bioinformatics* **29,** 2810–2811 (2013).

11. Yu, G., Wang, L.-G. & He, Q.-Y. ChIPseeker: An r/bioconductor package for chip peak annotation, comparison and visualization. *Bioinformatics* **31,** 2382–2383 (2015).

12. Luo, W. & Brouwer, C. Pathview: An R/Bioconductor package for pathway-based data integration and visualization. *Bioinformatics* **29,** 1830–1831 (2013).

13. Schmidt, M. *et al.* The humoral immune system has a key prognostic impact in node-negative breast cancer. *Cancer Research* **68,** 5405–5413 (2008).