**clusterProfiler: universal enrichment tool for functional and comparative study**

Guangchuang Yu

State Key Laboratory of Emerging Infectious Diseases and Centre of Influenza Research, School of Public Health, The University of Hong Kong, 21 Sassoon Road, Pokfulam, Hong Kong SAR, China.

**I present a tool (clusterProfiler; accessible at https://github.com/GuangchuangYu/clusterProfiler) for functional enrichment analysis. clusterProfiler supports using functional annotation data provided by users or obtained from online databases. It enables comparative analysis and offers comprehensive visualization tools for result interpretation.**

Functional enrichment analysis is one of the most widely use technique for interpreting gene lists or genome-wide regions of interest (ROIs) derived from high-throughput sequencing (HTS). Although tools are proliferating to perform gene-centric or epigenomic enrichment analysis, most of them are designed for model organisms or specific domains (*e.g.* fungi[1] or plant[2] *etc.*) embedded with particular annotations (*e.g.* Gene Ontology (GO) or Kyoto Encyclopaedia of Genes and Genomes (KEGG)). Non-model organisms and functional annotations other than GO and KEGG are rarely supported. Moreover, an increasing concern upon the quality of gene annotation has raised an alarm in biomedical research, as reported by previous study[3] that about 42% of the tools were outdated by more than five years and functional significance were severely underestimated with only 26% of biological processes or pathways were captured compare to using up-to-date annotation. Such negative impacts can be propagated for years and harm follow-up studies.

The clusterProfiler package was designed by considering the supports of multiple ontologies/pathways, up-to-date gene annotation, multiple organisms, user's annotation data and comparative analysis. The package employs modular design and supports disease analyses (Disease Ontology[4], Network of Cancer Gene[5], gene-disease and variant-disease associations[6]), Reactome pathway and Medical Subject Headings (MeSH) analysis via its sub-packages DOSE[4], ReactomePA[7] and meshes (https://github.com/GuangchuangYu/meshes). The clusterProfiler package internally supports GO and KEGG. GO annotation data can be obtained directly from Bioconductor OrgDb packages, or retrieved from web resources (*e.g.*

AnnotationHub and biomaRt[8] *etc.*). KEGG Pathway and Module were directly obtained using KEGG API and supports more than five thousand genomes (http://www.genome.jp/kegg/catalog/org_list.html) as well as KEGG Orthology (KO) which is particular useful for metagenomic functional studies. In addition, clusterProfiler provides general functions, enricher and GSEA, to perform fisher's exact test and gene set enrichment analysis using user defined annotations, making it possible to support novel functional annotation of newly sequenced species (*e.g.* electronic annotation using Blast2GO[9] and KAAS[10]), unsupported ontologies/pathways (*e.g.* InterPro Domain, Clusters of Orthologous Groups, Mouse phenotype ontology *etc.*) or customized annotation. clusterProfiler provides parser functions to import GMT file so that gene sets downloaded from Molecular Signatures Database can be directly supported as well as to retrieve whole genome GO annotations from UniProt database using taxonomic ID. Epigenomic enrichment analysis is also supported with user-provided ROIs in BED format by combining with in-house annotation package ChIPseeker[11]. GO semantic similarity measurement implemented in another in-house package GOSemSim[12] allows calculating GO term similarity using several methods based on information content and graph structure was also incorporated in clusterProfiler to remove redundancy of enriched GO terms. This feature simplifies result and assists in interpretation as well as against annotation/interpretation bias.

Albeit functional enrichment analysis is commonly used in downstream analysis to decipher key biological processes, visualization of enriched result is mainly depicted by bar chart, which maybe dominate by redundant terms and cannot reveal the complex of gene-pathway associations. The clusterProfiler sub-package, enrichplot (https://github.com/GuangchuangYu/enrichplot), is designed to visualize enrichment result by integrating expression data (Fig. 1). These methods allow users without programming skills to generate effective visualization to explore and interpret results as well as to find patterns within the data. In addition, all these visualization methods were implemented based on ggplot2, which allows customization using grammar of graphics.

## AUTHOR CONTRIBUTIONS

G. Y. conceived and designed this project, implemented the algorithms and software packages and wrote the manuscript.

## COMPLETING FINANCIAL INTERESTS

None.


1. Priebe, S., Kreisel, C., Horn, F., Guthke, R. & Linde, J. FungiFun2: a comprehensive online resource for systematic analysis of gene lists from fungal species. *Bioinformatics* **31,** 445–446 (2015).
2. Yi, X., Du, Z. & Su, Z. PlantGSEA: a gene set enrichment analysis toolkit for plant community. *Nucleic Acids Res.* **41,** W98–W103 (2013).
3. Wadi, L., Meyer, M., Weiser, J., Stein, L. D. & Reimand, J. Impact of outdated gene annotations on pathway enrichment analysis. *Nat. Methods* **13,** 705 (2016).
4. 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).
5. An, O., Dall'Olio, G. M., Mourikis, T. P. & Ciccarelli, F. D. NCG 5.0: updates of a manually curated repository of cancer genes and associated properties from cancer mutational screenings. *Nucleic Acids Res.* **44,** D992–D999 (2016).
6. Piñero, J. *et al.* DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. *Database J. Biol. Databases Curation* **2015,** (2015).
7. Yu, G. & He, Q.-Y. ReactomePA: an R/Bioconductor package for reactome pathway analysis and visualization. *Mol. Biosyst.* **12,** 477–479 (2016).
8. Durinck, S., Spellman, P. T., Birney, E. & Huber, W. Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt. *Nat. Protoc.* **4,** 1184–1191 (2009).
9. Conesa, A. *et al.* Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics* **21,** 3674–3676 (2005).
10. Moriya, Y., Itoh, M., Okuda, S., Yoshizawa, A. C. & Kanehisa, M. KAAS: an automatic genome annotation and pathway reconstruction server. *Nucleic Acids Res.* **35,** W182–W185 (2007).
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. Yu, G. *et al.* GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics* **26,** 976–978 (2010).

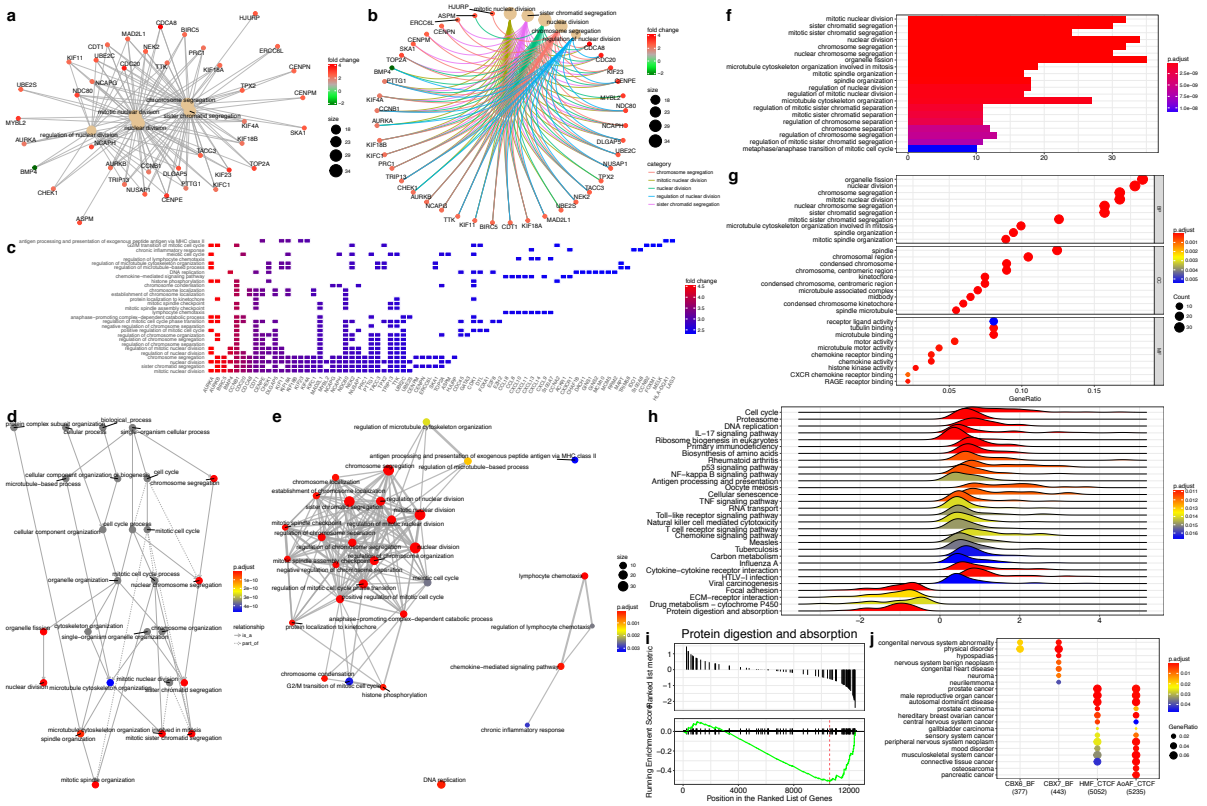


**Figure 1.** Visualization methods for enrichment result. (a) Gene-concept network depicts the linkages of genes and biological concepts as a network. (b) Circular version of gene-concept network. (c) Displaying gene-concept relationship as a heatmap with gene expression data integrated. (d) Directed acyclic graph induced by most significant GO terms. (e) Enrichment map organizes enriched terms into a network with edges weighted by the ratio of overlapping gene sets. Mutually overlapping gene sets are tending to cluster together, making it easy to identify functional modules. (f) Bar chart to display gene count or ratio as bar height and coloured by enrichment scores (e.g. p.adjust). (g) Dot chart that similar to bar chart with capability to encode another score as dot size (also known as bubble plot). Both bar and dot charts support faceting to visualize sub-ontologies simultaneously. (h) Ridge line plot for expression distribution of GSEA result. By default, only display the distribution of core enriched genes (also known as leading edge) and can be switched to visualize the distribution of whole gene sets. (i) Running score and pre-ranked list of GSEA result. (j) Comparing enriched results across multiple experiments. This example compared disease associations of multiple genome-wide ROIs (data from GSM1295076, GSM1295077, GSM749665, GSM749666).

## Source code to reproduce Figure 1.

```
devtools::install_github(
    c("guangchuangyu/enrichplot",
    "guangchuangyu/DOSE",
    "guangchuangyu/clusterProfiler",
    "guangchuangyu/ChIPseeker"))

library(DOSE)
```

```r
library(enrichplot)
library(clusterProfiler)

data(geneList)
de <- names(geneList)[abs(geneList) > 2]
ego <- enrichGO(de, OrgDb = "org.Hs.eg.db", ont="BP", readable=TRUE)

p1 <- goplot(ego)
p2 <- barplot(ego, showCategory=20)
go <- enrichGO(de, OrgDb = "org.Hs.eg.db", ont="all")
p3 <- dotplot(go, split="ONTOLOGY") + facet_grid(ONTOLOGY~., scale="free")
ego2 <- simplify(ego)
p4 <- cnetplot(ego2, foldChange=geneList)
p5 <- cnetplot(ego2, foldChange=geneList, circular = TRUE, colorEdge = TRU
E)
p6 <- heatplot(ego2, foldChange=geneList)
p7 <- emapplot(ego2)
kk <- gseKEGG(geneList, nPerm=10000)
p8 <- ridgeplot(kk)
p9 <- gseaplot(kk, geneSetID = 1, title = kk$Description[1])

require(ChIPseeker)
ff <- getSampleFiles()
downloadGSMbedFiles("GSM749665")
downloadGSMbedFiles("GSM749666")
list.files(pattern="gz")[c(2,4)] -> bed
bedfile <- c(ff[4:5], HMF_CTCF=bed[1], AoAF_CTCF=bed[2])
TxDb <- ChIPseeker:::loadTxDb(NULL)
seq <- lapply(bedfile, ChIPseeker:::loadPeak)
x <- lapply(seq, function(i) seq2gene(i, c(-1000, 3000), 3000, TxDb=TxDb))
xx <- compareCluster(x, fun="enrichDO")
p10 <- dotplot(xx, showCategory=8)


library(cowplot)

d = data.frame(x=1, y=1)
space = ggplot(d, aes(x, y)) + geom_blank() + theme_void()

ss = 30
p456 = plot_grid(plot_grid(p4, p5, labels=c('a', 'b'), label_size=ss),
    plot_grid(space, p6, ncol=2, rel_widths=c(.02, 1)),
    ncol=1, rel_heights=c(1.1, .8), labels=c("", "c"), label_size=ss)
p17 = plot_grid(plot_grid(space, p1, ncol=1, rel_heights=c(.015, 1)), p7,
        labels=c("d", "e"), rel_widths=c(.85, 1.2), label_size=ss)
p23 = plot_grid(p2, p3, ncol=1, rel_heights=c(.8, 1.2), labels=c("f", "g
"), label_size=ss)
p910 = plot_grid(p9, p10, ncol=2, rel_widths=c(.8, 1.2), labels=c("i", "j
"), label_size=ss)
p8910 = plot_grid(p8, p910, ncol=1, labels=c("h", ""), rel_heights=c(1, .
8), label_size=ss)
p178910 = plot_grid(p17, p8910, rel_widths=c(1.2, 1.05))
p45623 = plot_grid(p456, p23, ncol=2, rel_widths=c(1.2, .8))
```

```r
pp = plot_grid(p45623, p178910, ncol=1)
ggsave(pp, file="fig1.pdf", width=30, height=20)
```