04_enrichplot_visualization

2024-05-30

目录

1 Visualization of functional enrichment result 1

1.1 Bar Plot 2

1.2 Dot plot 9

1.3 Gene-Concept Network 11

1.4 Heatmap-like functional classification 15

1.5 Tree plot 15

1.6 Enrichment Map 18

1.7 UpSet Plot 18

1.8 ridgeline plot for expression distribution of GSEA result . . . 20

1.9 running score and preranked list of GSEA result 20

1.10 pubmed trend of enriched terms 28

1 Visualization of functional enrichment result

```
##
```

```
## clusterProfiler v4.12.0  For help: https://yulab-smu.top/biomedical-knowledge-mining
##
## If you use clusterProfiler in published research, please cite:
## T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu,

##
## 载入程序包：'clusterProfiler'

## The following object is masked from 'package:stats':
##
##     filter
```

The `enrichplot` package implements several visualization methods to help interpreting enrichment results. It supports visualizing enrichment results obtained from `DOSE` [@yu_dose_2015], `clusterProfiler` [@yu2012; @wu_clusterprofiler_2021], `ReactomePA` [@yu_reactomepa_2016] and `meshes` [@yu_meshes_2018]. Both over representation analysis (ORA) and gene set enrichment analysis (GSEA) are supported.

Note: Several visualization methods were first implemented in `DOSE` and rewrote from scratch using `ggplot2`. If you want to use the old methods, you can use the doseplot package.

1.1 Bar Plot

Bar plot is the most widely used method to visualize enriched terms. It depicts the enrichment scores (*e.g.* p values) and gene count or ratio as bar height and color (Figure 1A). Users can specify the number of terms (most significant) or selected terms (see also the FAQ) to display via the `showCategory` parameter.

enrichDGN() 仅针对人类

```
organisms <- "org.Rn.eg.db"
library(organisms, character.only = T)
```

```
## 载入需要的程序包：AnnotationDbi
```

```
## 载入需要的程序包：stats4
```

```
## 载入需要的程序包：BiocGenerics
```

```
##
## 载入程序包：'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, table,
##     tapply, union, unique, unsplit, which.max, which.min
```

```
## 载入需要的程序包：Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## 载入需要的程序包：IRanges
## 载入需要的程序包：S4Vectors
```

```
##
## 载入程序包：'S4Vectors'
```

```
## The following object is masked from 'package:clusterProfiler':
##
##     rename
```

```
## The following object is masked from 'package:utils':
##
##     findMatches
```

```
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```
##
## 载入程序包：'IRanges'
```

```
## The following object is masked from 'package:clusterProfiler':
##
##     slice
```

```
## The following object is masked from 'package:grDevices':
##
##     windows
```

```
##
## 载入程序包：'AnnotationDbi'
```

```
## The following object is masked from 'package:clusterProfiler':
##
##     select
```

```
##
```

```
geneList <- readRDS("outputs/geneList.rds")
gene <- names(geneList)

# gene.df <- bitr(gene, fromType = "SYMBOL", toType = "ENTREZID", OrgDb = organisms)
# names(geneList) <- gene.df$ENTREZID
geneList <- geneList[!duplicated(names(geneList))]

de <- names(geneList)

ego <- enrichGO(gene          = de,
                OrgDb         = organisms,
                keyType       = "SYMBOL",
                ont           = "ALL",
                pAdjustMethod = "BH",
                pvalueCutoff  = 0.01,
                qvalueCutoff  = 0.05,
                readable      = T)
head(ego)
```

```
##            ONTOLOGY         ID
## GO:0006457       BP GO:0006457
## GO:0061077       BP GO:0061077
## GO:0072594       BP GO:0072594
## GO:1904851       BP GO:1904851
## GO:1900182       BP GO:1900182
## GO:0070203       BP GO:0070203
##                                                                           Description
## GO:0006457                                                           protein folding
## GO:0061077                                        chaperone-mediated protein folding
## GO:0072594                        establishment of protein localization to organelle
## GO:1904851 positive regulation of establishment of protein localization to telomere
## GO:1900182                    positive regulation of protein localization to nucleus
## GO:0070203          regulation of establishment of protein localization to telomere
##            GeneRatio   BgRatio       pvalue     p.adjust       qvalue
## GO:0006457    19/123 192/18668 2.643361e-17 6.563466e-14 4.844308e-14
## GO:0061077    12/123  67/18668 1.709584e-14 2.122449e-11 1.566519e-11
## GO:0072594    19/123 465/18668 2.099182e-10 1.737423e-07 1.282342e-07
## GO:1904851     5/123  10/18668 2.808366e-09 1.575457e-06 1.162800e-06
## GO:1900182    10/123 112/18668 3.421039e-09 1.575457e-06 1.162800e-06
## GO:0070203     5/123  11/18668 5.121557e-09 1.575457e-06 1.162800e-06
##
## GO:0006457     Hspe1/Pdia6/Cct6a/Pdcd5/Hsp90aa1/Hspa8/Hspd1/Hspa9/Dnaja1/Sdf2l1/Hsp9
## GO:0061077                                                    Hspe1/Cct6a/Pdcd5/Hspa
## GO:0072594 Tomm22/Sec61b/Sec61g/Nolc1/Cct6a/Pdcd5/Hsp90aa1/Hspa8/Hspd1/Dnaja1/Timm13
## GO:1904851
## GO:1900182                                                              Park7/Cct6a/
## GO:0070203
##            Count
## GO:0006457    19
## GO:0061077    12
## GO:0072594    19
## GO:1904851     5
## GO:1900182    10
## GO:0070203     5
```

```r
# KEGG 适合全集分析，过少无结果
de <- names(geneList)
gene.df <- bitr(de, fromType = "SYMBOL",
                toType = c("ENTREZID"),
                OrgDb = organisms)
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
## Warning in bitr(de, fromType = "SYMBOL", toType = c("ENTREZID"), OrgDb =
## organisms): 1.59% of input gene IDs are fail to map...
```

```
kk <- enrichKEGG(gene         = gene.df$ENTREZID,
                 organism     = "rno",
                 pvalueCutoff = 0.05)
```

```
## Reading KEGG annotation online: "https://rest.kegg.jp/link/rno/pathway"...

## Reading KEGG annotation online: "https://rest.kegg.jp/list/pathway/rno"...
```

```
kk <- setReadable(kk, OrgDb = organisms, keyType = "ENTREZID")
head(kk)
```

```
##                                      category
## rno05166                       Human Diseases
## rno05012                       Human Diseases
## rno05020                       Human Diseases
## rno05222                       Human Diseases
## rno04141       Genetic Information Processing
## rno04061 Environmental Information Processing
##                                  subcategory       ID
## rno05166           Infectious disease: viral rno05166
## rno05012           Neurodegenerative disease rno05012
## rno05020           Neurodegenerative disease rno05020
## rno05222              Cancer: specific types rno05222
## rno04141    Folding, sorting and degradation rno04141
## rno04061 Signaling molecules and interaction rno04061
##                                                            Description
## rno05166                       Human T-cell leukemia virus 1 infection
## rno05012                                             Parkinson disease
## rno05020                                                 Prion disease
## rno05222                                        Small cell lung cancer
## rno04141                   Protein processing in endoplasmic reticulum
## rno04061 Viral protein interaction with cytokine and cytokine receptor
##          GeneRatio  BgRatio       pvalue     p.adjust       qvalue
## rno05166     12/87 253/9971 1.798610e-06 0.0001340588 0.0000985562
## rno05012     13/87 302/9971 1.906219e-06 0.0001340588 0.0000985562
## rno05020     13/87 305/9971 2.127918e-06 0.0001340588 0.0000985562
## rno05222      7/87 100/9971 2.504899e-05 0.0010982637 0.0008074119
## rno04141      9/87 183/9971 2.905459e-05 0.0010982637 0.0008074119
## rno04061      6/87  84/9971 8.742220e-05 0.0025807628 0.0018973026
##
## rno05166                   Ranbp1/Vdac1/Myc/Csf2/Lta/Cdk4/Ran/Bcl2l1/Slc25a5/Calr/Il
## rno05012 Ndufab1/Vdac1/Psmb3/Park7/Tubb4b/Tuba1c/Psmb6/Cycs/Bcl2l1/Slc25a5/Ndufv2/Tu
## rno05020  Stip1/Ndufab1/Vdac1/Psmb3/Tubb4b/Tuba1c/Psmb6/Cycs/Hspa8/Slc25a5/Ndufv2/Tu
## rno05222                                               Myc/Traf4/Cycs/Cdk4/Bcl2l1/Tr
## rno04141                         Sec61b/Sec61g/Pdia6/Hsp90aa1/Hspa8/Dnaja1/Hsp90ab1/
## rno04061                                                        Ccl4/Xcl1/Lta/Ccl20/
##          Count
## rno05166    12
## rno05012    13
## rno05020    13
## rno05222     7
## rno04141     9
## rno04061     6
```

```r
geneList2 <- geneList
names(geneList2) <- gene.df$ENTREZID
geneList2 <- geneList2[!duplicated(names(geneList2))]
kk2 <- gseKEGG(geneList     = geneList2,
               organism     = "rno",
               minGSSize    = 20,
               pvalueCutoff = 1)
```

```
## using 'fgsea' for GSEA analysis, please cite Korotkevich et al (2019).
```

```
## preparing geneSet collections...
```

```
## GSEA analysis...
## no term enriched under specific pvalueCutoff...
```

```
head(kk2)
```

```
## [1] ID              Description     setSize         enrichmentScore
## [5] NES             pvalue          p.adjust        qvalue
## <0 行> (或0-长度的row.names)
```

Other variables that derived using mutate can also be used as bar height or color as demonstrated in Figure 1B.

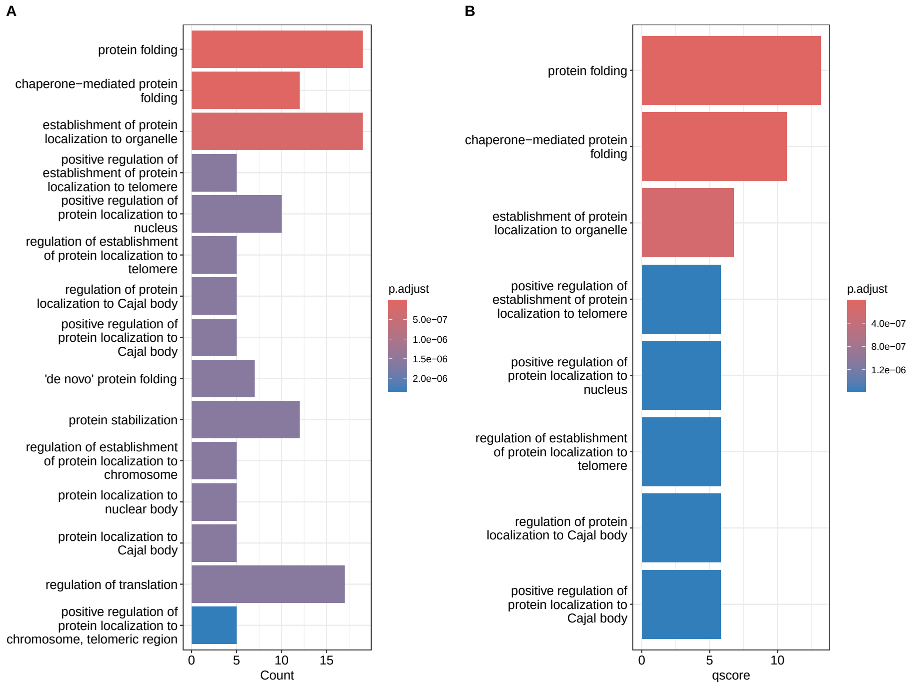


图 1: **Bar plot of enriched terms.**

1.2 Dot plot

Dot plot is similar to bar plot with the capability to encode another score as dot size.

Note: The `dotplot()` function also works with `compareCluster()` output.

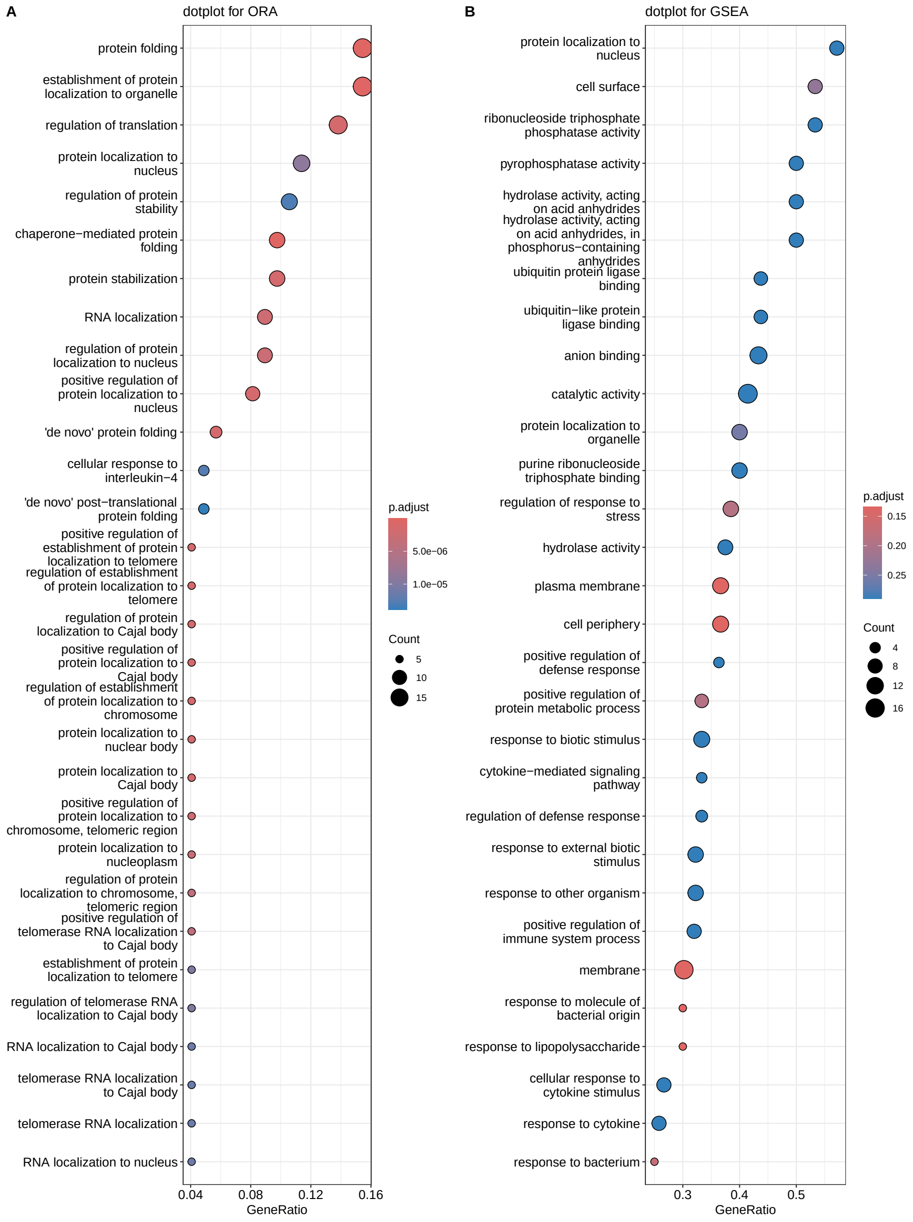


图 2: **Dot plot of enriched terms.**

1.3 Gene-Concept Network

Both the `barplot()` and `dotplot()` only displayed most significant or selected enriched terms, while users may want to know which genes are involved in these significant terms. 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 the `cnetplot()` function to extract the complex association. The `cnetplot()` depicts the linkages of genes and biological concepts (*e.g.* GO terms or KEGG pathways) as a network. GSEA result is also supported with only core enriched genes displayed.

```
## convert gene ID to Symbol
# egox <- setReadable(ego, 'org.Rn.eg.db', 'ENTREZID')
egox <- ego
p1 <- cnetplot(egox, foldChange = geneList)
```

```
## Warning in cnetplot.enrichResult(x, ...): Use 'color.params = list(foldChange = your
##  The foldChange parameter will be removed in the next version.

## Scale for size is already present.
## Adding another scale for size, which will replace the existing scale.
```

```
## categorySize can be scaled by 'pvalue' or 'geneNum'
p2 <- cnetplot(egox, categorySize = "pvalue", foldChange = geneList)
```

```
## Warning in cnetplot.enrichResult(x, ...): Use 'color.params = list(foldChange = your
##  The foldChange parameter will be removed in the next version.

## Scale for size is already present.
## Adding another scale for size, which will replace the existing scale.
```

```r
p3 <- cnetplot(egox, foldChange = geneList, circular = TRUE, colorEdge = TRUE)
```

```
## Warning in cnetplot.enrichResult(x, ...): Use 'color.params = list(foldChange = your
##  The foldChange parameter will be removed in the next version.

## Warning in cnetplot.enrichResult(x, ...): Use 'color.params = list(edge = your_value
##  The colorEdge parameter will be removed in the next version.

## Scale for size is already present.
## Adding another scale for size, which will replace the existing scale.
```

```r
cowplot::plot_grid(p1, p2, p3, ncol=3, labels=LETTERS[1:3], rel_widths=c(.8, .8, 1.2))
```

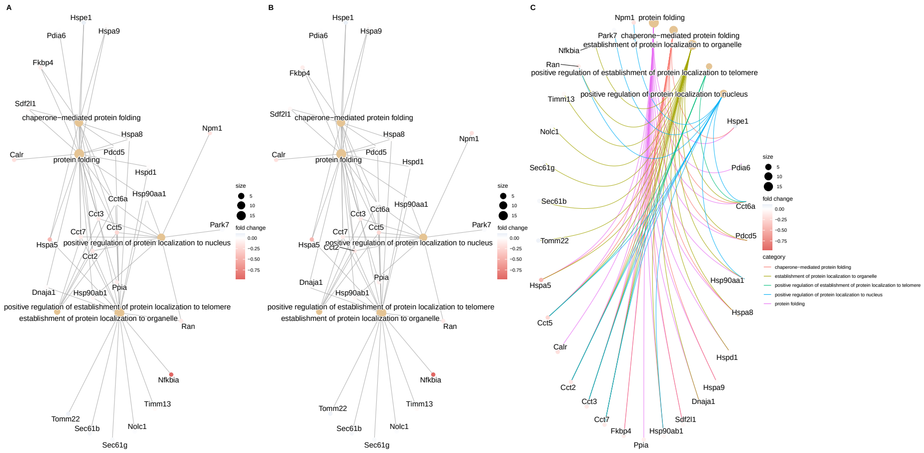


图 3: **Network plot of enriched terms.**

If you would like label subset of the nodes, you can use the `node_label` parameter, which supports 4 possible selections (i.e. “category”, “gene”, “all” and “none”), as demonstrated in Figure 4. The size of category and gene label can be specified via the `cex_label_category` and `cex_label_gene` parameters. The color of the categories and genes can be specified via the `color_category` and `color_gene` parameters.

```
p1 <- cnetplot(egox, node_label = "category",
        cex_label_category = 1.2)
```

```
## Warning in cnetplot.enrichResult(x, ...): Use 'cex.params = list(category_label = yo
##  The cex_label_category parameter will be removed in the next version.
```

```
p2 <- cnetplot(egox, node_label = "gene",
        cex_label_gene = 0.8)
```

```
## Warning in cnetplot.enrichResult(x, ...): Use 'cex.params = list(gene_label = your_v
##  The cex_label_gene parameter will be removed in the next version.
```

```
p3 <- cnetplot(egox, node_label = "all")
p4 <- cnetplot(egox, node_label = "none",
        color_category='firebrick',
        color_gene='steelblue')
```

```
## Warning in cnetplot.enrichResult(x, ...): Use 'color.params = list(category = your_v
##  The color_category parameter will be removed in the next version.
```

```
## Warning in cnetplot.enrichResult(x, ...): Use 'color.params = list(gene = your_value
##  The color_gene parameter will be removed in the next version.
```

```
cowplot::plot_grid(p1, p2, p3, p4, ncol=2, labels=LETTERS[1:4])
```

```
## Warning: Removed 28 rows containing missing values or values outside the scale range
## (`geom_text_repel()`).
```

```
## Warning: Removed 5 rows containing missing values or values outside the scale range
## (`geom_text_repel()`).
```

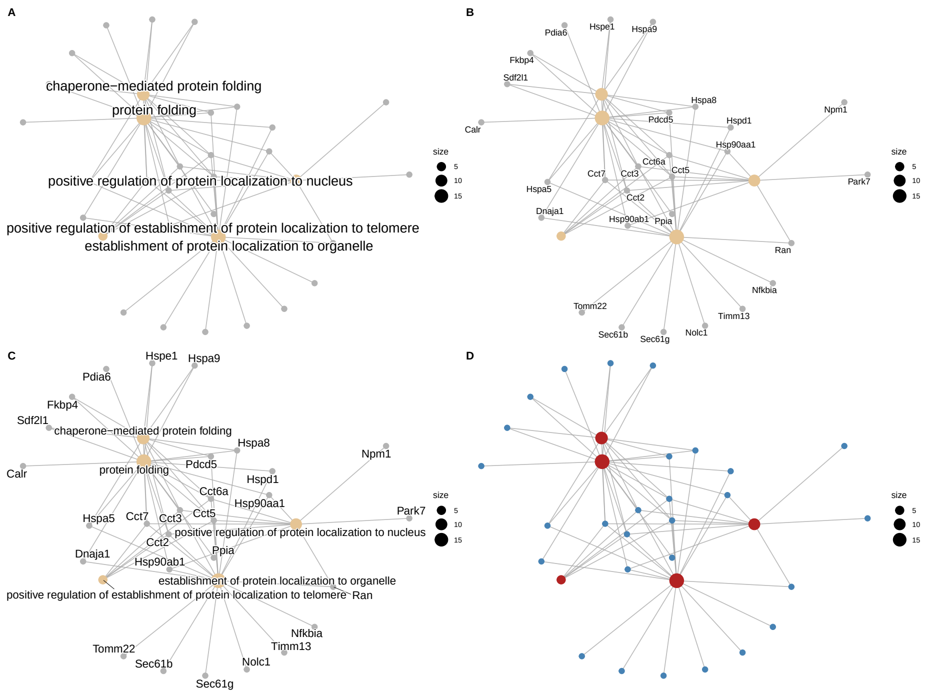

A
chaperone−mediated protein folding
protein folding
positive regulation of protein localization to nucleus
positive regulation of establishment of protein localization to telomere
establishment of protein localization to organelle
size
5
10
15
B
Pdia6
Hspe1
Hspa9
Fkbp4
Sdf2l1
Calr
Hspa8
Pdcd5
Hspd1
Npm1
Hsp90aa1
Cct6a
Cct7
Cct3
Cct5
Cct2
Hspa5
Dnaja1
Hsp90ab1
Ppia
Park7
Ran
Nfkbia
Tomm22
Timm13
Sec61b
Sec61g
Nolc1
size
5
10
15
C
Hspe1
Hspa9
Pdia6
Fkbp4
Sdf2l1
chaperone−mediated protein folding
Hspa8
Calr
protein folding
Pdcd5
Npm1
Hspd1
Cct6a
Hsp90aa1
Hspa5
Cct7
Cct3
Cct5
Park7
positive regulation of protein localization to nucleus
Cct2
Ppia
Dnaja1
Hsp90ab1
establishment of protein localization to organelle
positive regulation of establishment of protein localization to telomere
Ran
Nfkbia
Tomm22
Timm13
Sec61b
Nolc1
Sec61g
size
5
10
15
D
size
5
10
15

图 4: **Labelling nodes by selected subset.** gene category (A), gene name (B), both gene category and gene name (C, default) and not to label at all (D).

1.4 Heatmap-like functional classification

The `heatplot` is similar to `cnetplot`, while displaying the relationships as a heatmap. The gene-concept network may become too complicated if user want to show a large number significant terms. The `heatplot` can simplify the result and more easy to identify expression patterns.

```r
p1 <- heatplot(egox, showCategory = 5)
p2 <- heatplot(egox, foldChange = geneList, showCategory = 5)
cowplot::plot_grid(p1, p2, ncol = 1, labels = LETTERS[1:2])
```

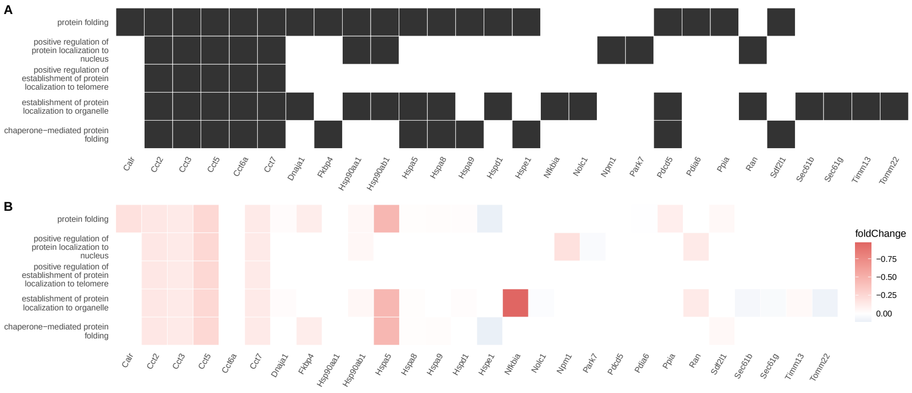


图 5: **Heatmap plot of enriched terms.** default (A), `foldChange=geneList` (B)

1.5 Tree plot

The `treeplot()` function performs hierarchical clustering of enriched terms. It relies on the pairwise similarities of the enriched terms calculated by the `pairwise_termsim()` function, which by default using Jaccard's similarity index (JC). Users can also use semantic similarity values if it is supported (*e.g.*, GO, DO and MeSH).

The default agglomeration method in `treeplot()` is `ward.D` and users can specify other methods via the `hclust_method` parameter (*e.g.*, 'average', 'complete', 'median', 'centroid', *etc.*, see also the document of the `hclust()` function). The `treeplot()` function will cut the tree into several subtrees (specify by the `nCluster` parameter (default is 5)) and labels subtrees using high-frequency words. This will reduce the complexity of the enriched result and improve user interpretation ability.

```
egox2 <- pairwise_termsim(egox)
p1 <- treeplot(egox2)
p2 <- treeplot(egox2, hclust_method = "average")
```

```
## Warning in treeplot.enrichResult(x, ...): Use 'cluster.params = list(method = your_v
##  The hclust_method parameter will be removed in the next version.
```

```
p1; p2
```

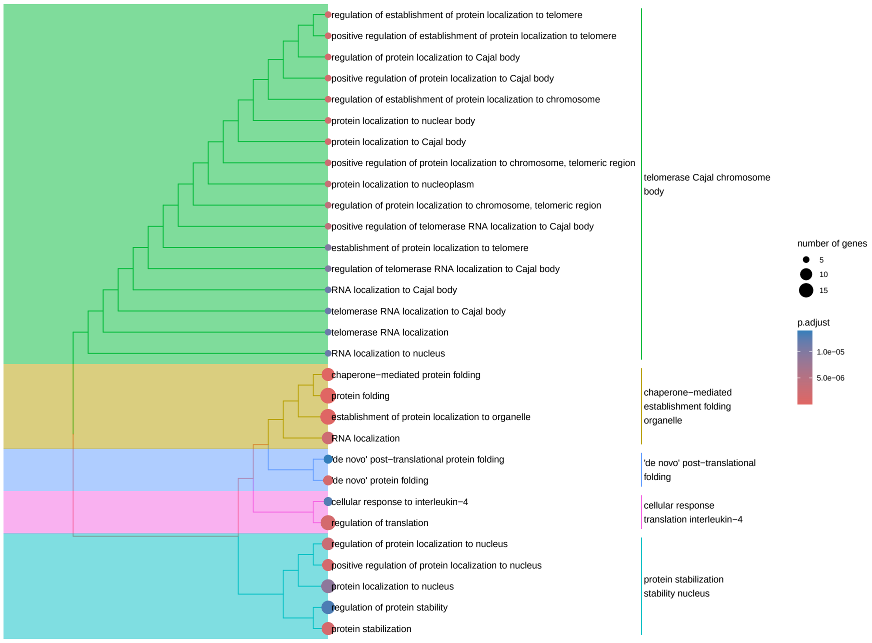


图 6: **Tree plot of enriched terms.** default (A), `hclust_method = "average"` (B)

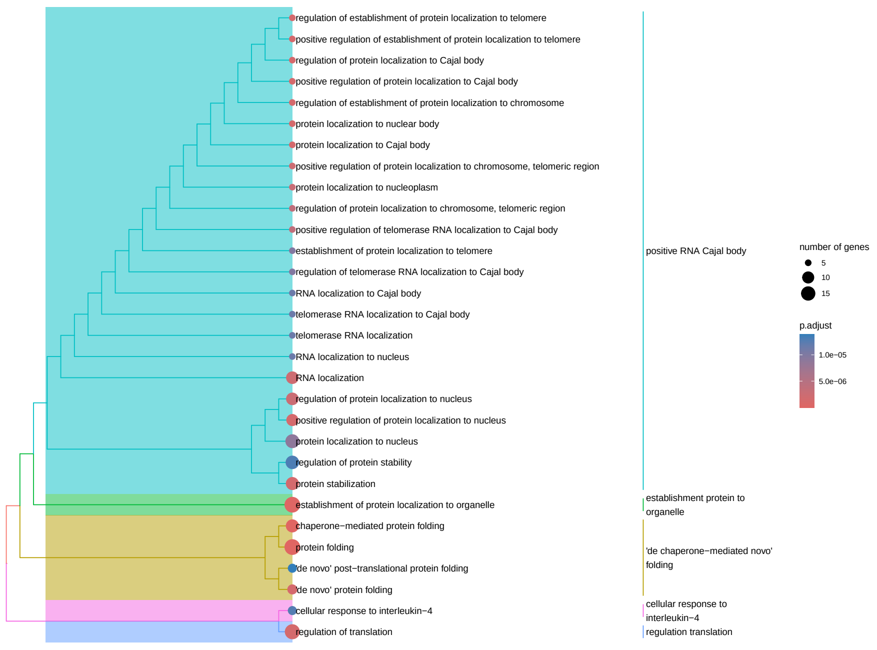


图 7: **Tree plot of enriched terms.** default (A), `hclust_method = "average"` (B)

```
# aplot::plot_list(p1, p2, tag_levels='A')
```

1.6 Enrichment Map

Enrichment map organizes enriched terms into a network with edges connecting overlapping gene sets. In this way, mutually overlapping gene sets are tend to cluster together, making it easy to identify functional module.

The `emapplot` function supports results obtained from hypergeometric test and gene set enrichment analysis. The `cex_category` parameter can be used to resize nodes, as demonstrated in Figure 8 B, and the `layout` parameter can adjust the layout, as demonstrated in Figure 8 C and D.

```
ego <- pairwise_termsim(ego)
p1 <- emapplot(ego, showCategory = 10)
p2 <- emapplot(ego, cex.params = list(category_node = 1.5), showCategory = 10)
p3 <- emapplot(ego, layout.params = list(layout = "kk"), showCategory = 10)
p4 <- emapplot(ego, cex.params = list(category_node = 1.5), layout.params = list(layout
cowplot::plot_grid(p1, p2, p3, p4, ncol=2, labels=LETTERS[1:4])
```

1.7 UpSet Plot

The `upsetplot` is an alternative to `cnetplot` for visualizing the complex association between genes and gene sets. It emphasizes the gene overlapping among different gene sets.

```
upsetplot(ego)
```

For over-representation analysis, `upsetplot` will calculate the overlaps among different gene sets as demonstrated in Figure 9. For GSEA result, it will plot the fold change distributions of different categories (e.g. unique to pathway, overlaps among different pathways).

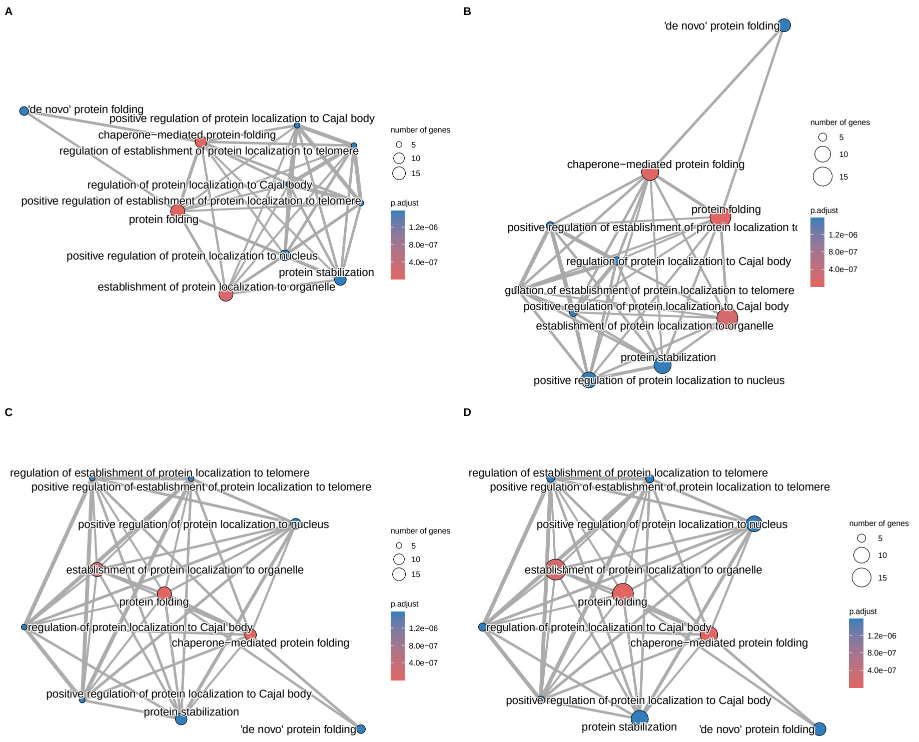


图 8: **Plot for results obtained from hypergeometric test and gene set enrichment analysis.** default (A), `cex_category=1.5` (B), `layout="kk"` (C) and `cex_category=1.5,layout="kk"` (D).

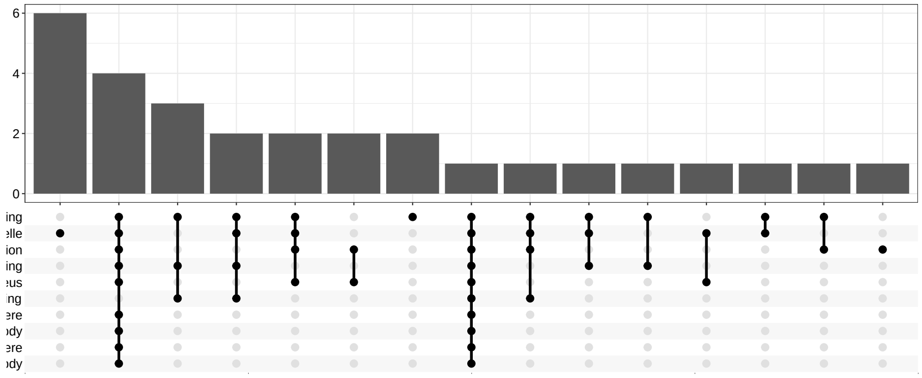


图 9: **Upsetplot for over-representation analysis.**

```
# upsetplot(kk2) ## 由于 kk2 中并无结果导致报错，故注释
```

1.8 ridgeline plot for expression distribution of GSEA result

The `ridgeplot` will visualize expression distributions of core enriched genes for GSEA enriched categories. It helps users to interpret up/down-regulated pathways.

```
ridgeplot(ego2)
```

1.9 running score and preranked list of GSEA result

Running score and preranked list are traditional methods for visualizing GSEA result. The `enrichplot` package supports both of them to visualize the distribution of the gene set and the enrichment score.

```
p1 <- gseaplot(ego2, geneSetID = 1, by = "runningScore", title = ego2$Description[1])
p2 <- gseaplot(ego2, geneSetID = 1, by = "preranked", title = ego2$Description[1])
p3 <- gseaplot(ego2, geneSetID = 1, title = ego2$Description[1])

plot_list(p1, p2, p3, ncol=1, labels=LETTERS[1:3])
```

Another method to plot GSEA result is the `gseaplot2` function:

```
gseaplot2(ego2, geneSetID = 1, title = ego2$Description[1])
```

```
gseaplot2(ego2, geneSetID = "GO:0005886")
```

The `gseaplot2` also supports multile gene sets to be displayed on the same figure:

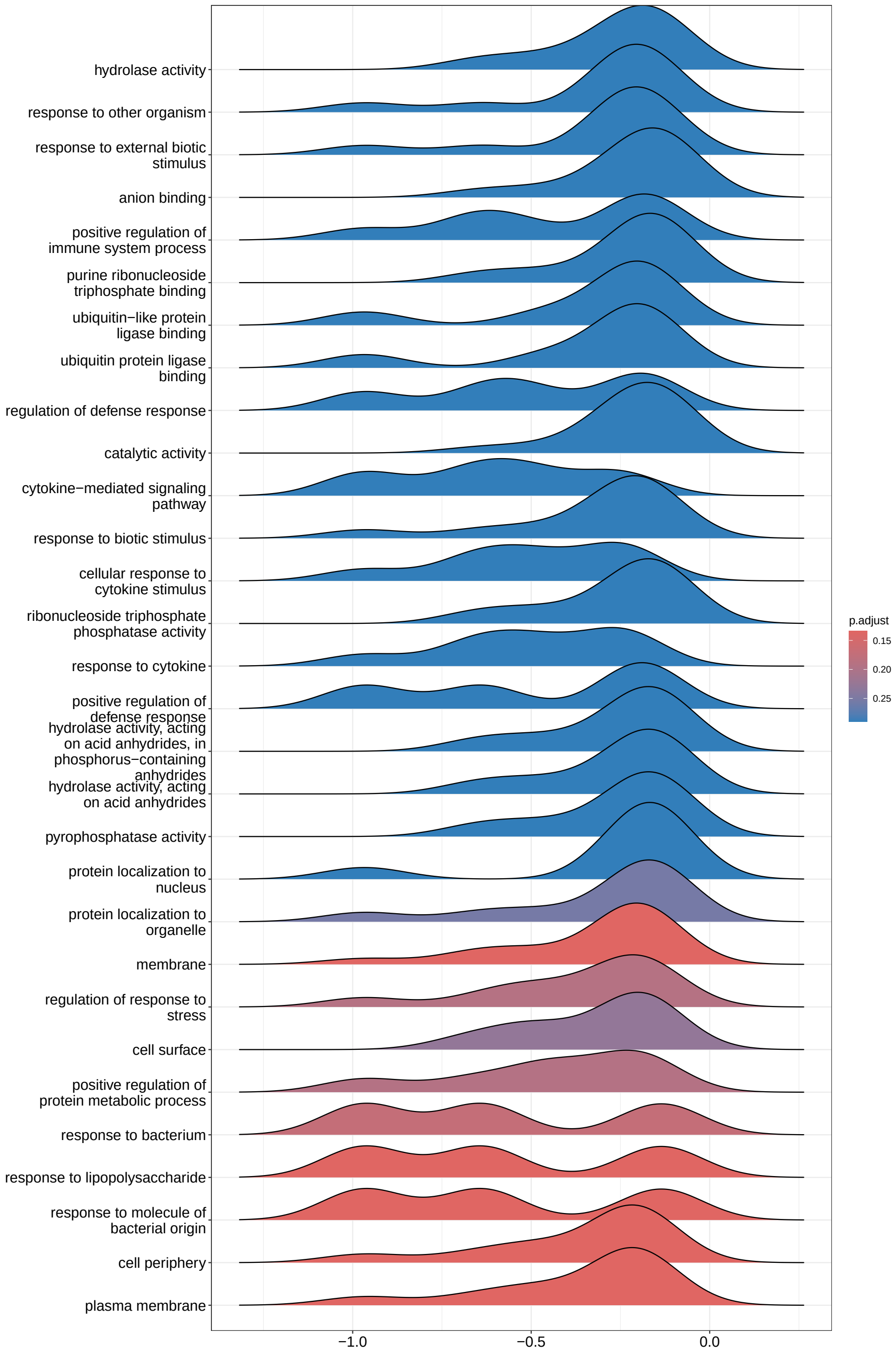


图 10: **Ridgeplot for gene set enrichment analysis.**

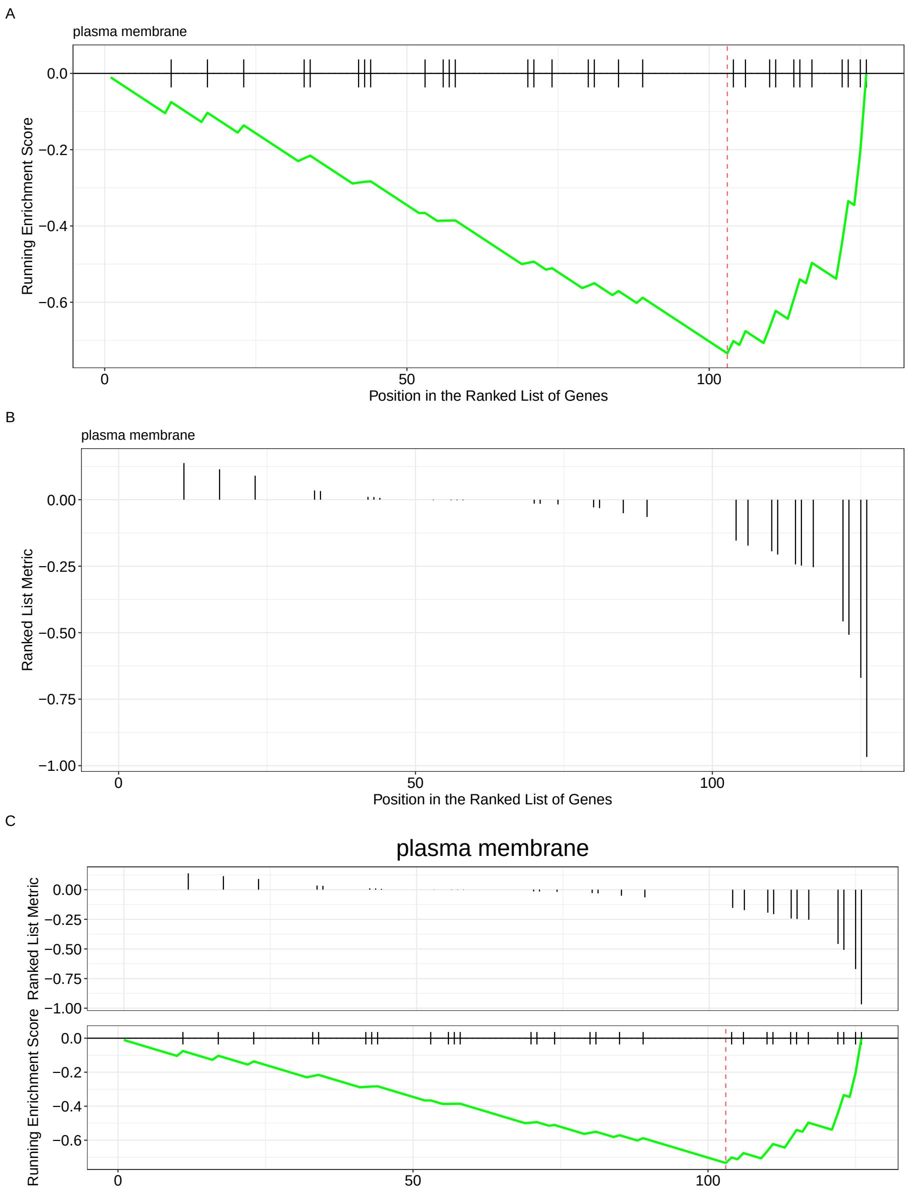


图 11: **gseaplot for GSEA result(`by = "runningScore"`).** `by = "runningScore"` (A), `by = "preranked"` (B), default (C)

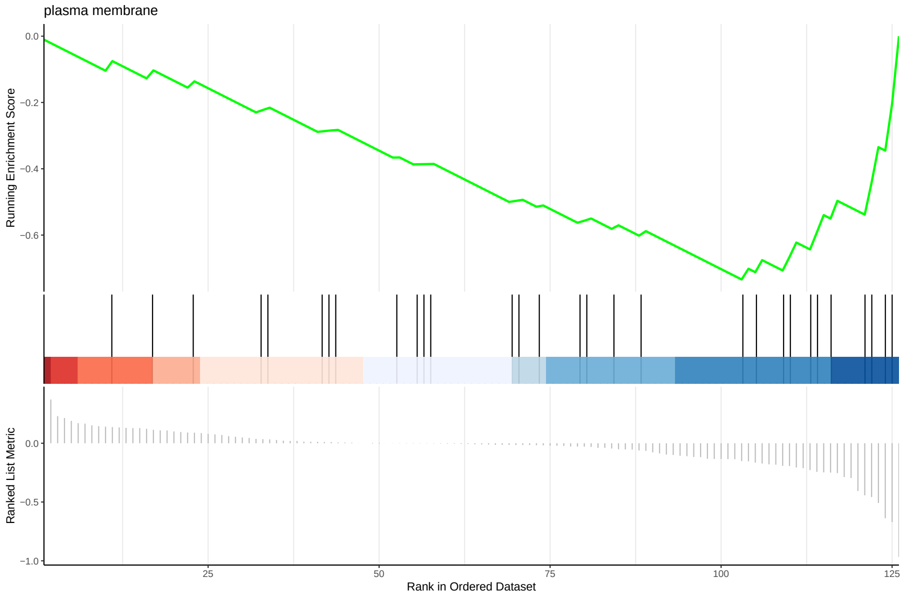


图 12: **Gseaplot2 for GSEA result.**

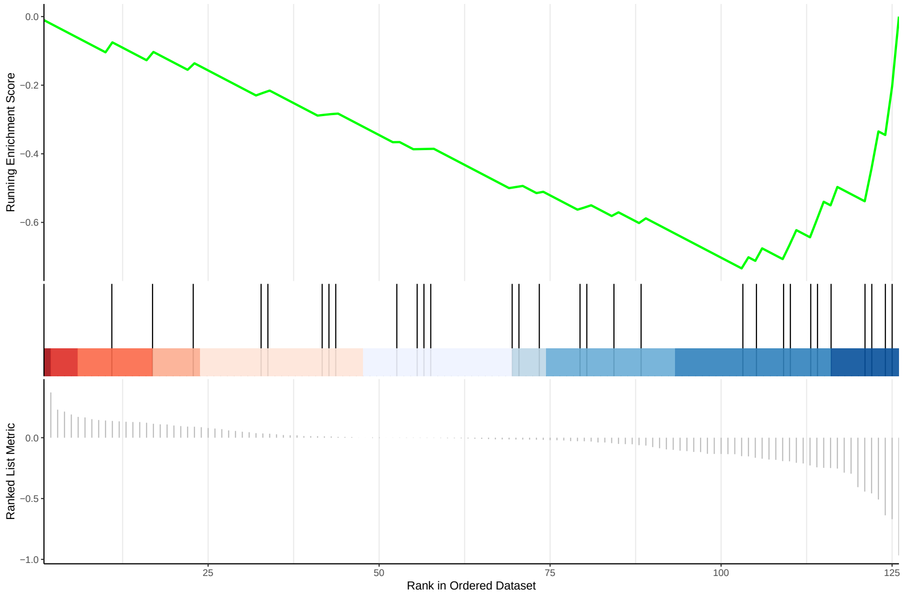


图 13: **Gseaplot2 for GSEA result.**

```
gseaplot2(ego2, geneSetID = c(1,3))
```

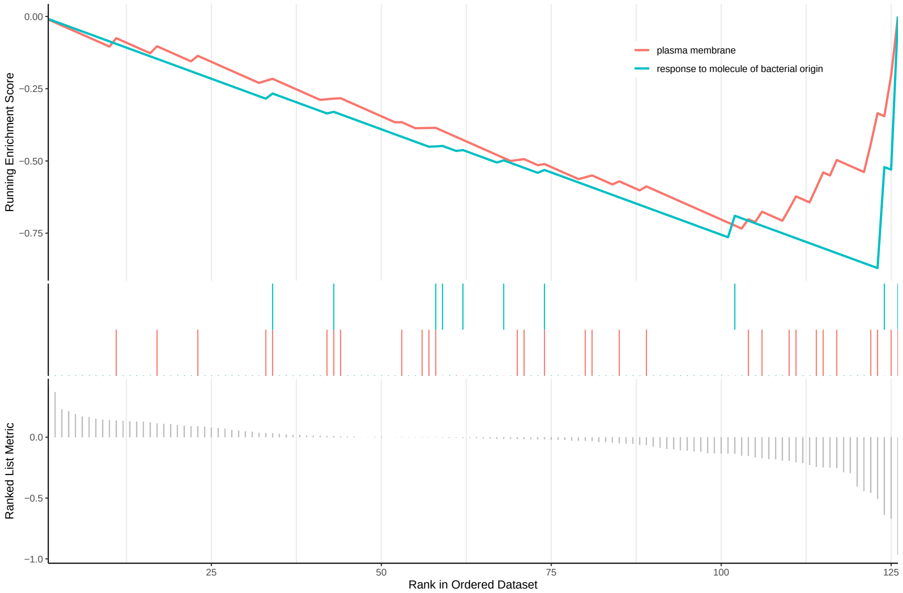


图 14: **Gseaplot2 for GSEA result of multile gene sets.**

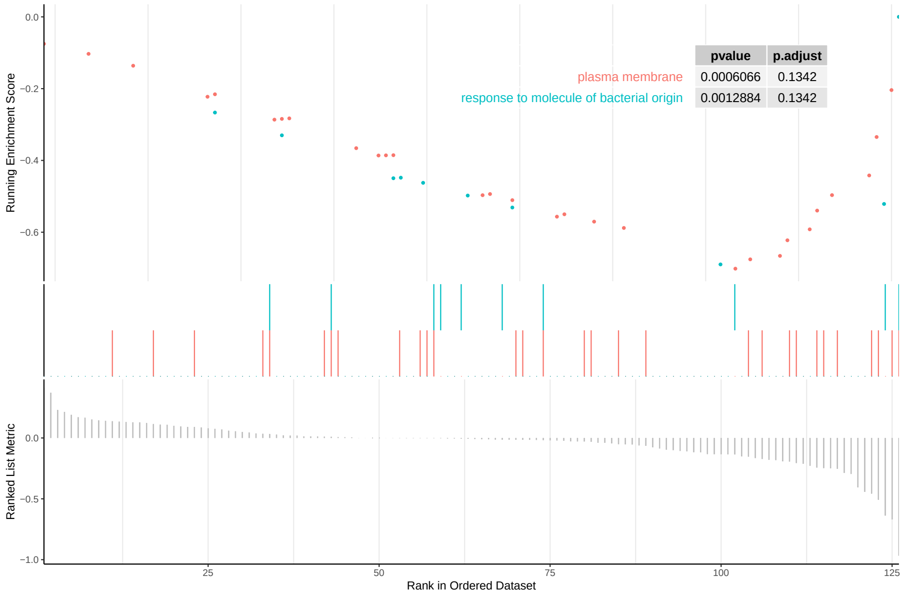
User can also displaying the pvalue table on the plot via `pvalue_table` parameter:

```
gseaplot2(ego2, geneSetID = c(1,3), pvalue_table = TRUE,
          color = c("#E495A5", "#86B875", "#7DB0DD"), ES_geom = "dot")
```

User can specify `subplots` to only display a subset of plots:

```
p1 <- gseaplot2(ego2, geneSetID = c(1, 3), subplots = 1)
p2 <- gseaplot2(ego2, geneSetID = c(1, 3), subplots = 1:2)
p3 <- gseaplot2(ego2, geneSetID = c(1, 3), subplots = 3)
plot_list(p1, p2, p3, ncol=1, labels=LETTERS[1:3])
```

The `gsearank` function plot the ranked list of genes belong to the specific gene set.



	pvalue	p.adjust
plasma membrane	0.0006066	0.1342
response to molecule of bacterial origin	0.0012884	0.1342

图 15: **Gseaplot2 for GSEA result of multile gene sets(add pvalue_table).**

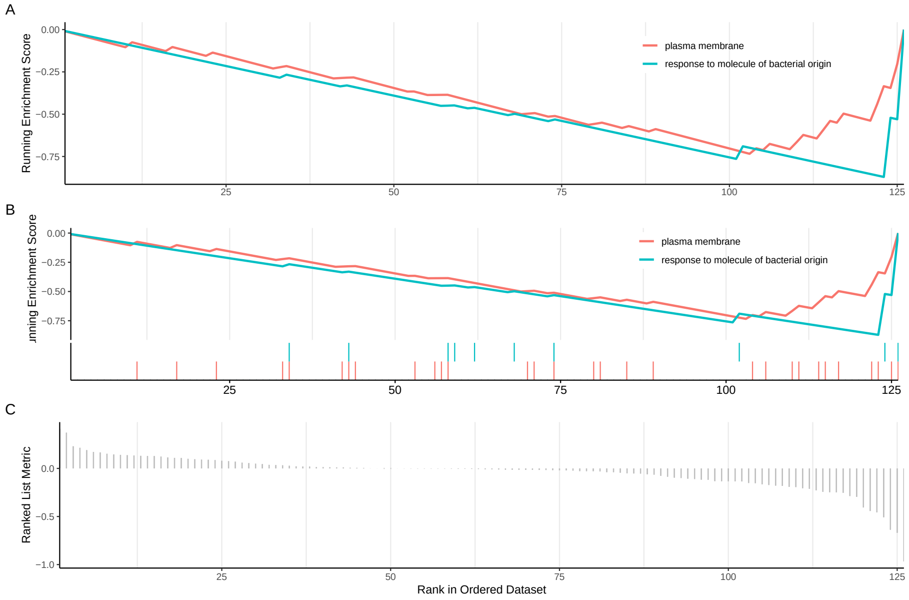


图 16: **Gseaplot2 for GSEA result of multile gene sets(add subplots).** `subplots = 1` (A),`subplots = 1:2` (B)

```
gsearank(ego2, 1, title = ego2[1, "Description"])
```

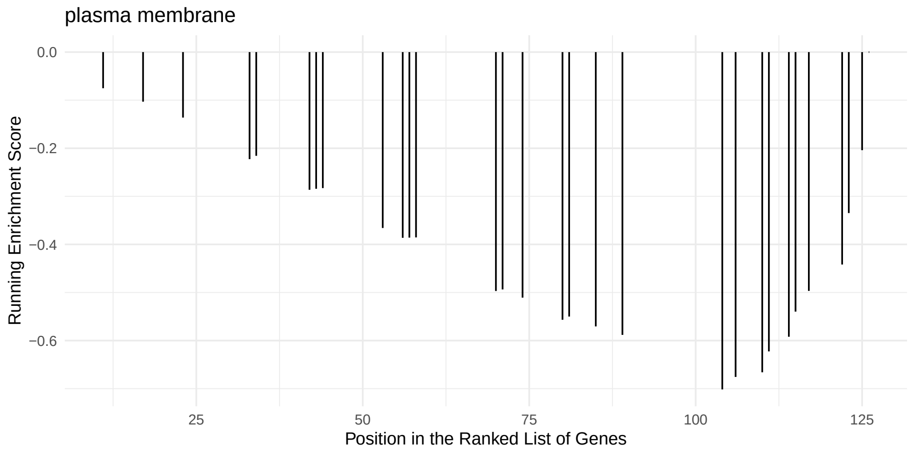


图 17: **Ranked list of genes belong to the specific gene set.**

Multiple gene sets can be aligned using `cowplot`:

```
library(ggplot2)
library(cowplot)

pp <- lapply(1:3, function(i) {
    anno <- ego2[i, c("NES", "pvalue", "p.adjust")]
    lab <- paste0(names(anno), "=",  round(anno, 3), collapse="\n")

    title <- paste0(ego2[i, 2], "\n", ego2[i, 3])
    gsearank(ego2, i, title) + xlab(NULL) +ylab(NULL) +
        annotate("text", 150, ego2[i, "enrichmentScore"] * .75, label = lab, hjust=0, v
})
plot_grid(plotlist=pp, ncol=1)
```

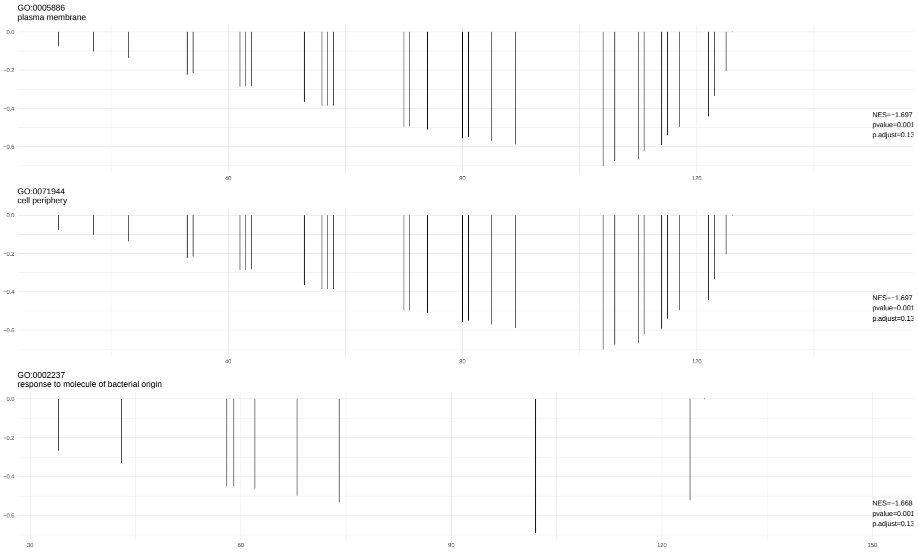


图 18: **Gsearank for multiple gene sets.**

1.10 pubmed trend of enriched terms

One of the problem of enrichment analysis is to find pathways for further investigation. Here, we provide `pmcplot` function to plot the number/proportion of publications trend based on the query result from PubMed Central. Of course, users can use `pmcplot` in other scenarios. All text that can be queried on PMC is valid as input of `pmcplot`.

```r
terms <- head(ego@result$Description)
p <- pmcplot(terms, 2013:2023)
p2 <- pmcplot(terms, 2013:2023, proportion=FALSE)
cowplot::plot_grid(p, p2, ncol=1)
```

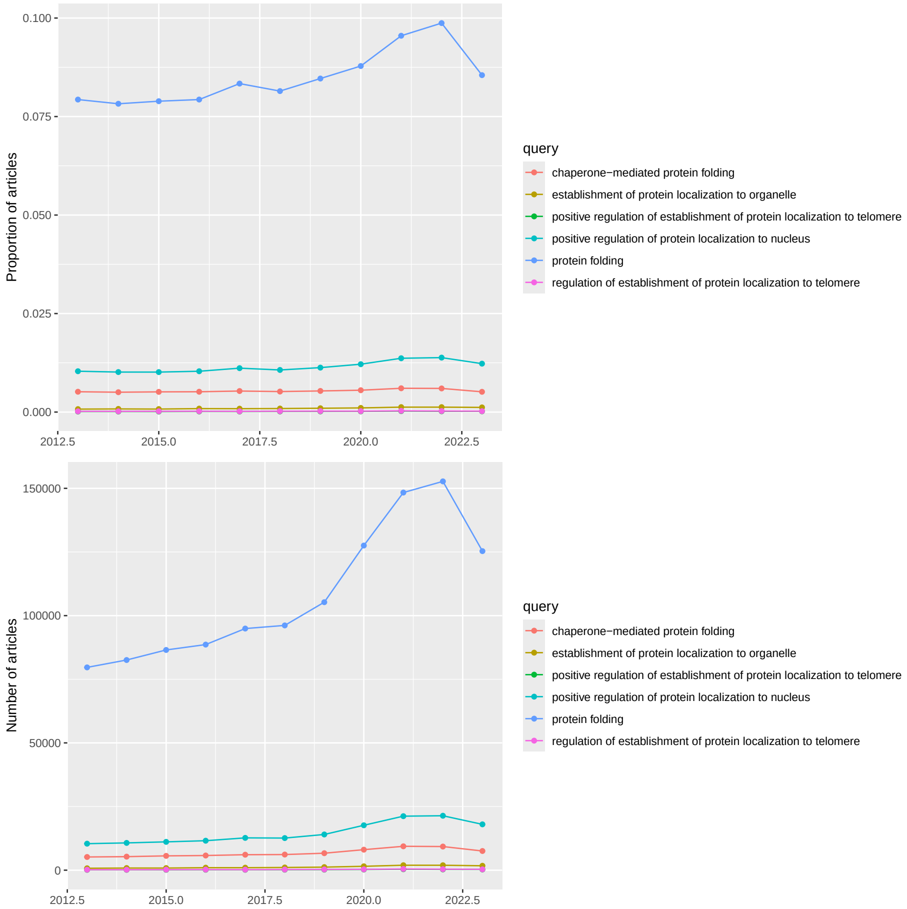


图 19: **Pmcplot of enrichment analysis.**