通路分析pipeline（GO+GSEA+KEGG）

卢建璋

一、总览：作用及数据背景

通路分析是常见的二代测序数据分析的下游分析方式，诸如RNA-seq，ChIP-seq，蛋白质组学等数据均可通过通路分析来获得更准确的结果。本文将以实验室RNA-seq数据（小鼠Pten/LKB1 KO：实验组 VS 小鼠Pten KO：对照组）为例演示通路分析的做法和如何生成有意义的图。

在我们进行基因差异表达分析的时候，我们会得到很多的DEG。我们很难找到一个规律去探究基因之间的联系。而通路分析则可以帮助我们获得差异通路，对RNA-seq数据进行进一步探索。

1. **数据准备：通过RNA-seq分析得到的基因表达矩阵。（如何从RNA-seq原数据得到表达矩阵请参考RNA-seq pipeline）。**

   注意：表达矩阵必须有以下三列：

   a. gene symbol或者gene id

   b. padj或者q.value

   c. Fold Change或者log(Fold Change)

gene_name	padj	FC
Aagab	0.0067925	1.612667548
AB124611	0.000154605	0.628886504
Abca1	0.000972927	0.432502743
Abca4	0.005908344	0.366970389
Abca4	4.01E-05	0.335148652
Abca5	0.000515771	0.631147085
Abcb10	0.001770376	0.70883929
Abcc12	0.008967195	0.569545216
Abcc5	0.00025144	1.633197706

2. R语言资源包准备

```r
library(DOSE)
library(org.Hs.eg.db) # For Human data
library(org.Mm.eg.db) # For Mouse data
library(topGO)
library(clusterProfiler)
library(pathview)
library(tidyverse)
library(RColorBrewer)
library(enrichplot)
library(R.utils)
library(DO.db)
library(msigdf)
library(GseaVis)
```

3. 数据导入 (如果原数据是Fold Change，最好转成logFC)

```r
# 3. 导入数据
data <- read.csv('Pten_LKB1 vs Pten.csv')
data$FC <- log2(data$FC) # 如果数据不是logFC，转化成logFC
colnames(data) <- c('gene_name', 'padj', 'log2FoldChange') # 规范格式，方便余下的分
```

	gene_name	padj	log2FoldChange
1	Aagab	0.006792500	0.6894491
2	AB124611	0.000154605	-0.6691284
3	Abca1	0.000972927	-1.2092188
4	Abca4	0.005908344	-1.4462644
5	Abca4	0.000040100	-1.5771270

注意：如果是microarray数据，极大概率会出现gene_name列有相同值的情况，这个时候选择保留logFC最显著的那一行数据。

```r
# 2. 去重
data$sign <- case_when(
  data$log2FoldChange >= 0 ~ "+",
  data$log2FoldChange < 0 ~ "-",
)
data$log2FoldChange <- abs(data$log2FoldChange)
data <- data %>% arrange(data[,1], data[,3])
data <- data[-which(duplicated(data[, 1], fromLast = T)), ]
data[which(data$sign == '-'), 'log2FoldChange'] <- -data[which(data$sign == '-')
data$sign <- NULL

# 3. 修改格式
rownames(data) <- data$gene_name
data$gene_name <- NULL
data <- na.omit(data)
```

最后的结果应该如图所示：

	padj	log2FoldChange
Aagab	0.006792500	0.6894491
AB124611	0.000154605	-0.6691284
Abca1	0.000972927	-1.2092188
Abca4	0.000040100	-1.5771270
Abca5	0.000515771	-0.6639518

二、Gene Ontology（GO）

GO分析要求输入所有的DEG，所以第一步要找出符合条件的DEG（筛选标准自定）

4. 找到DEG

```r
# 4. 找到DEG
data$group <- case_when(
  data$log2FoldChange > 1 & data$padj < 0.05 ~ "up",
  data$log2FoldChange < -1 & data$padj < 0.05 ~ "down",
  abs(data$log2FoldChange) <= 1 ~ "none",
  data$padj >= 0.05 ~ "none"
)

up <- rownames(data)[data$group == "up"]
down <- rownames(data)[data$group == "down"]
diff <- c(up,down)
```

5. 基因Symbol转化基因ID

对于GO分析，最好先将gene_symbol转化成gene_id。

```r
# 5. ID转换（使用clusterProfiler包自带ID转换函数bitr）
## 如果是人则用org.Hs.eg.db
up_entrez <- bitr(up,
                  fromType = "SYMBOL",
                  toType = "ENTREZID",
                  OrgDb = "org.Mm.eg.db")

down_entrez <- bitr(down,
                    fromType = "SYMBOL",
                    toType = "ENTREZID",
                    OrgDb = "org.Mm.eg.db")

diff_entrez <- bitr(diff,
                    fromType = "SYMBOL",
                    toType = "ENTREZID",
                    OrgDb = "org.Mm.eg.db")
```

基因map失败是正常现象，但是如果失败比例太高则需要注意。

```
'select()' returned 1:1 mapping between keys and columns
Warning message:
In bitr(diff, fromType = "SYMBOL", toType = "ENTREZID", OrgDb = "org.Mm.eg.db") :
  10.06% of input gene IDs are fail to map...
```

6. 开始GO分析

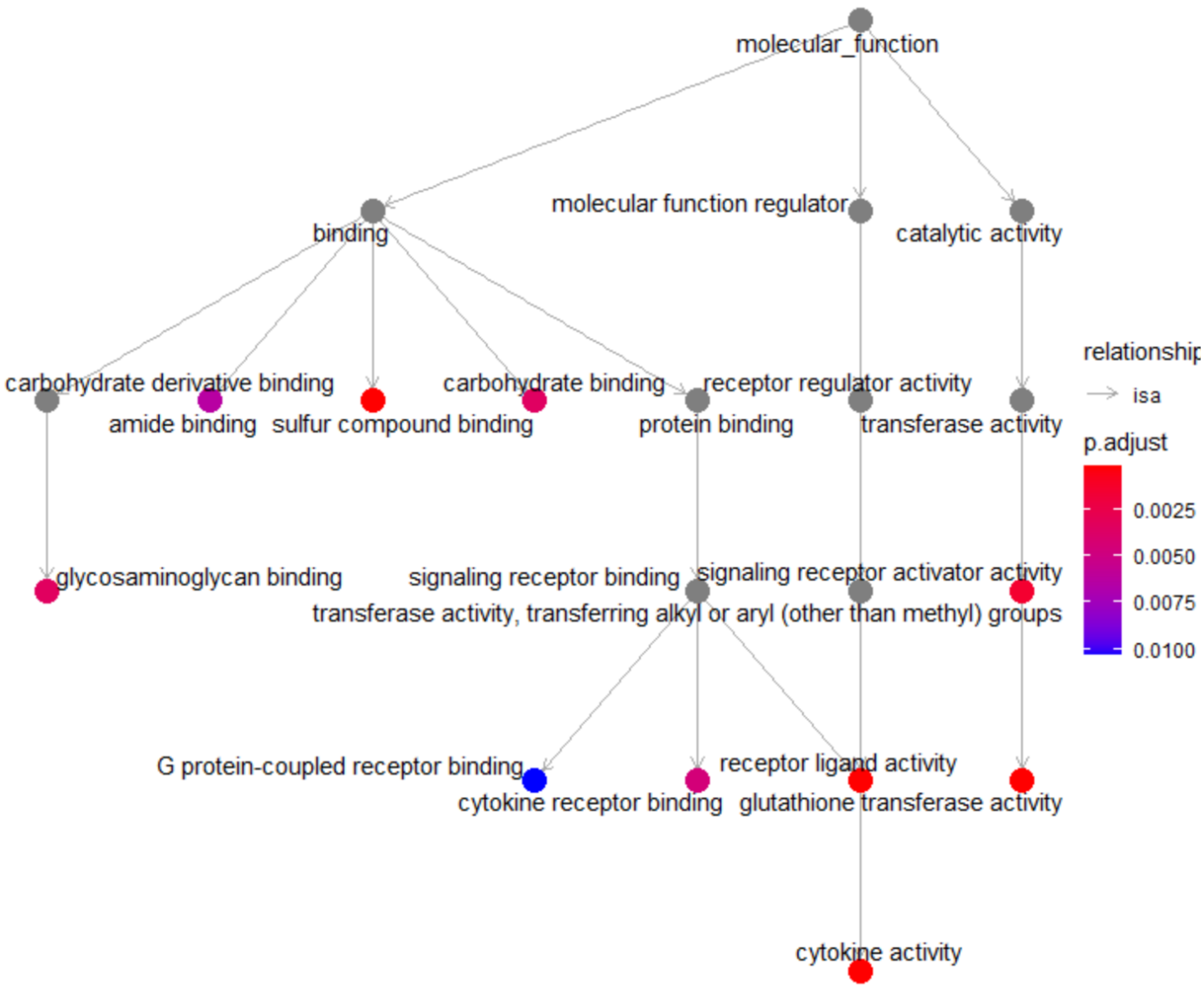
GO一共分为三个类别：Molecular Function (MF)，Cellular Component (CC)，Biological Process (BP)。本文只以其中一类作为例子，其他类别的操作同理。

```r
# 6. GO分析
GO_MF_diff <- enrichGO(gene = diff_entrez$ENTREZID, ## 分析所有DEG（up+down）
                       OrgDb = org.Mm.eg.db, ## human：org.Hs.eg.db
                       ont = "MF", ## 可选‘CC’或者‘BP’
                       pAdjustMethod = "BH",
                       pvalueCutoff = 0.05,
                       qvalueCutoff = 0.05,
                       readable = TRUE) ## 是否将gene ID映射到gene name
GO_MF_result <- GO_MF_diff@result
```

	ID	Description	GeneRatio	BgRatio	pvalue	p.adjus
GO:0048018	GO:0048018	receptor ligand activity	27/405	499/22669	3.862462e-07	0.0(
GO:0004364	GO:0004364	glutathione transferase activity	7/405	30/22669	7.876469e-07	0.0(
GO:0005125	GO:0005125	cytokine activity	17/405	231/22669	9.869832e-07	0.0(
GO:1901681	GO:1901681	sulfur compound binding	18/405	260/22669	1.152661e-06	0.0(
GO:0016765	GO:0016765	transferase activity, transferring alkyl or aryl (other than met...	8/405	60/22669	1.100499e-05	0.0(

7. 网络关系图

```r
# 7. GO plot
goplot(GO_MF_diff)
```

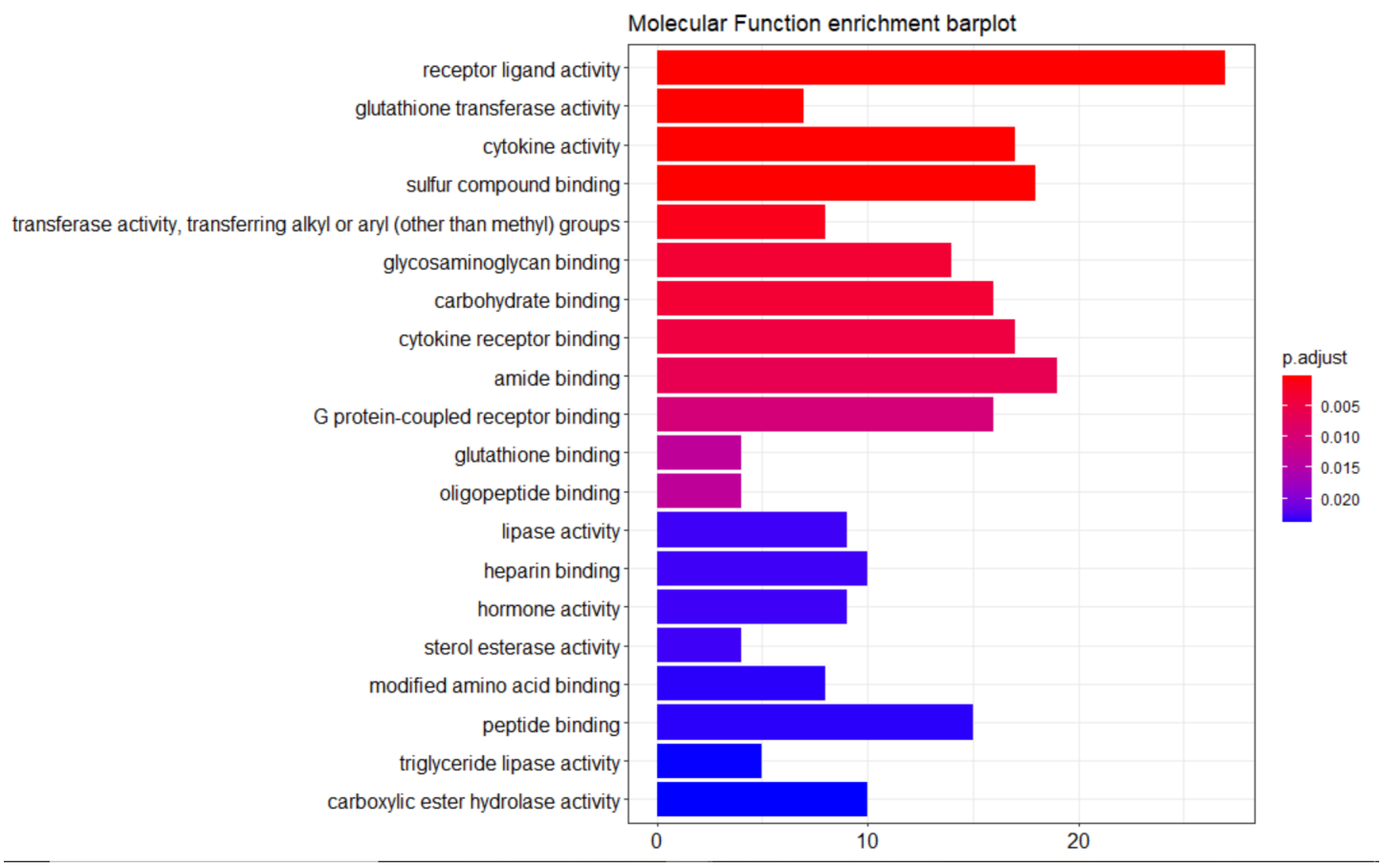


8. Barplot

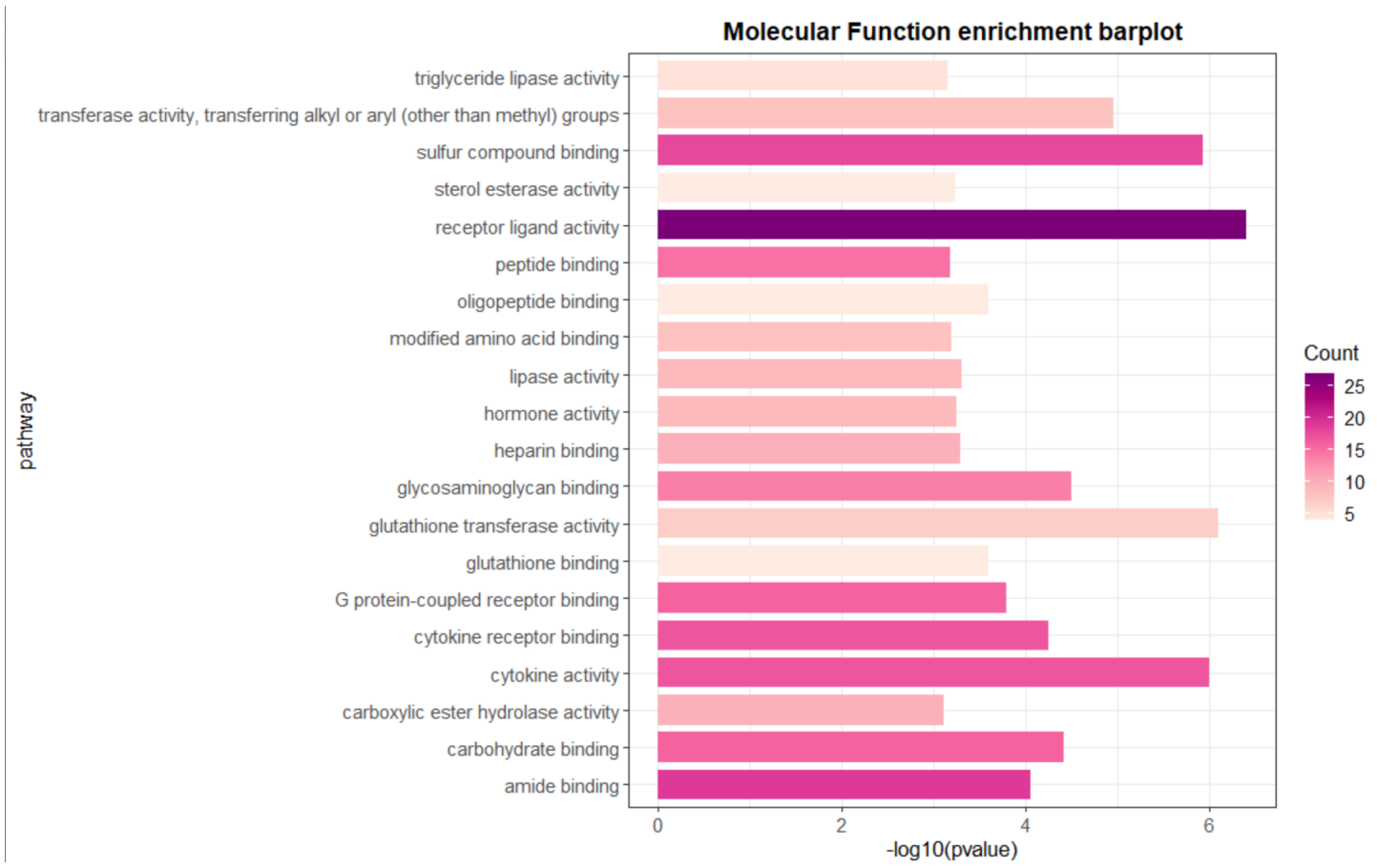
接下来将以两种方式演示barplot的画法，第一种是现成的函数画图（便捷但是不灵活），第二种是ggplot自行画图（可根据需求自行调整）

```r
# 8. Barplot —— 现成函数
barplot(
  GO_MF_diff,
  x = "Count", # 或者"GeneRatio"
  color = "p.adjust", # "pvalue" and "qvalue"
  showCategory = 20, # 显示前20(enrichResult按照pvalue排序)
  font.size = 12, #字体大小
  title = "Molecular Function enrichment barplot",
  label_format = 30 # 超过30个字符串换行
```

```
)
```

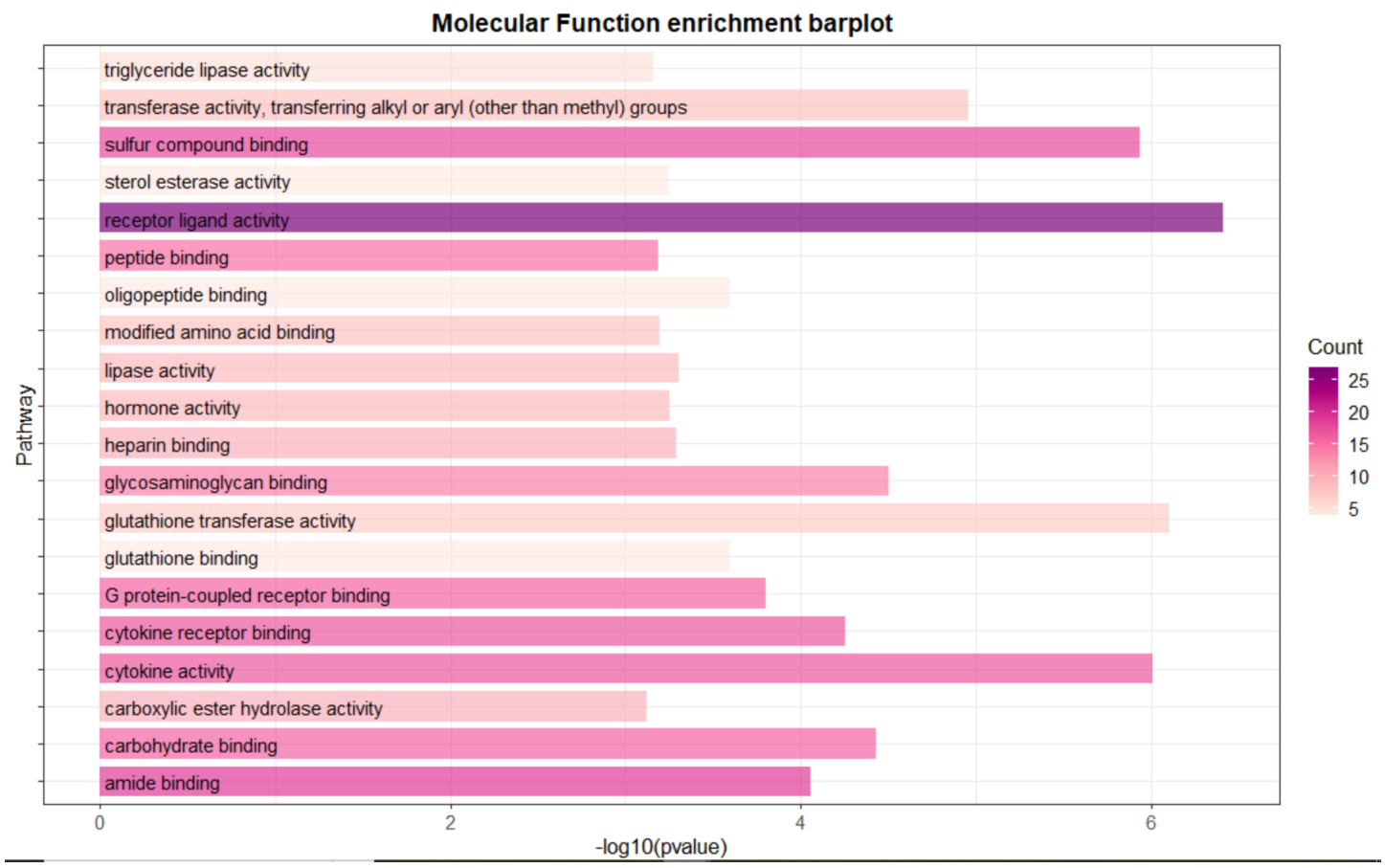



```r
# 8. Barplot —— ggplot
mytheme <- theme(axis.title = element_text(size = 13),
                 axis.text = element_text(size = 11),
                 plot.title = element_text(size = 14,
                                           hjust = 0.5,
                                           face = "bold"),
                 legend.title = element_text(size = 13),
                 legend.text = element_text(size = 11))

MF_top20 <- GO_MF_result[1:20, ]
ggplot(MF_top20,
       aes(x = -log10(pvalue), y = Description, fill = Count)) +
  scale_fill_distiller(palette = "RdPu",direction = 1) +
  geom_bar(stat = "identity", width = 0.8) +
  labs(x = "log10(pvalue)", y = "pathway",
       title = "Molecular Function enrichment barplot") +
  theme_bw() +
  mytheme
```



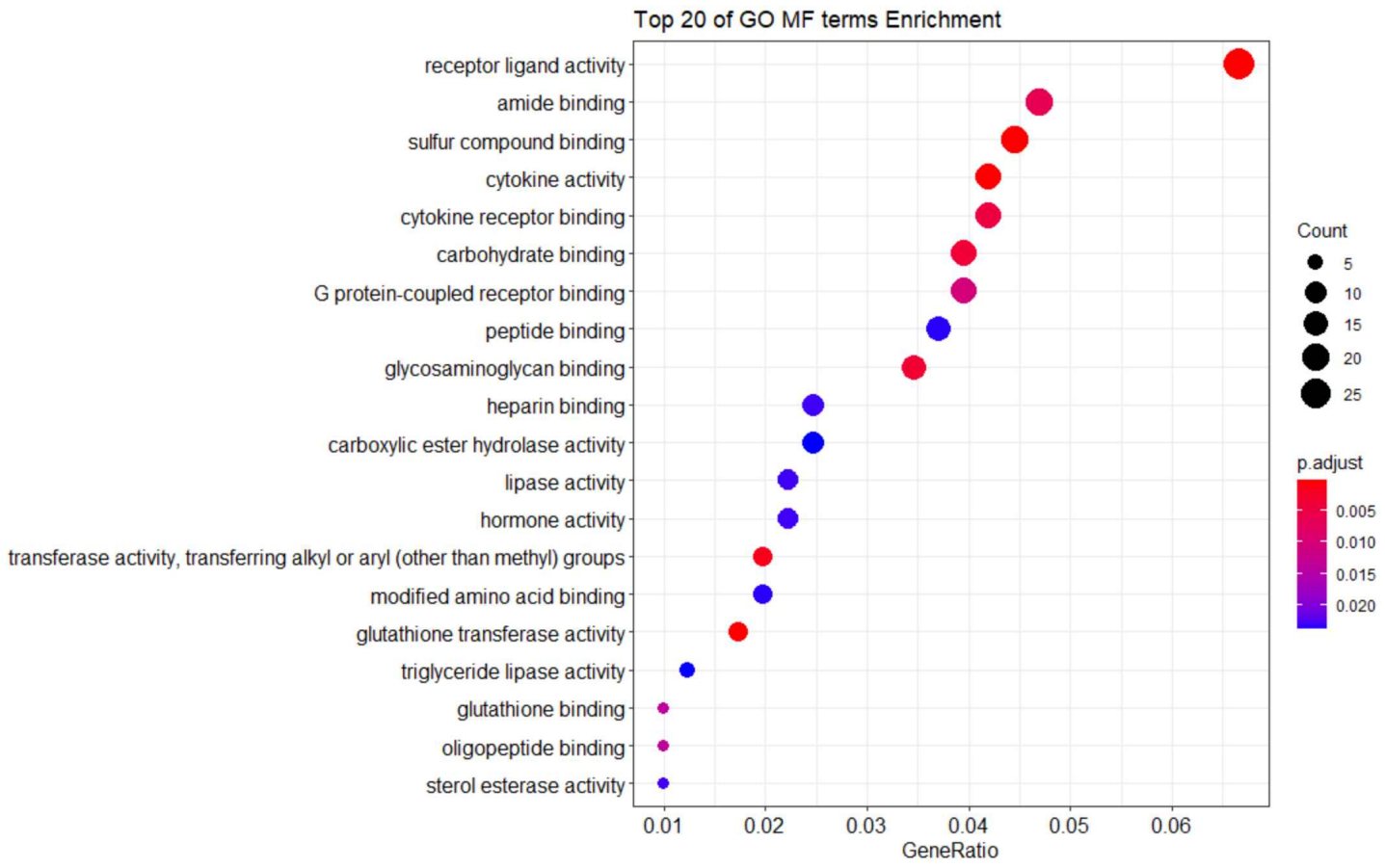
个人比较喜欢的一种方式：将pathway移到柱状图内部:

```
mytheme2 <- mytheme + theme(axis.text.y = element_blank())
MF_top20$text_x <- rep(0.03,20)
ggplot(data = MF_top20,
       aes(x = -log10(pvalue), y = Description)) +
  geom_bar(aes(fill = Count), stat = "identity", width = 0.8, alpha = 0.7) +
  scale_fill_distiller(palette = "RdPu", direction = 1) +
  labs(x = "-log10(pvalue)", y = "Pathway",
       title = "Molecular Function enrichment barplot") +
  geom_text(aes(x = text_x, label = Description),
            hjust = 0) +
  theme_bw() +
  mytheme2
```



9. Dotplot

```r
# 9. Dotplot
dotplot(
  GO_MF_diff,
  x = "GeneRatio",
  color = "p.adjust",
  title = "Top 20 of GO MF terms Enrichment",
  showCategory = 20,
  label_format = 30
)
```



三、Gene Set Enrichment Analysis (GSEA)

10. 总览

不同于GO，GSEA的输入则要求是排序好的gene_symbol，排序规则建议是按照logFC从大到小。

GSEA相比于GO的优势如下：

1. 分析所有基因，避免了GO中主动筛选DEG的影响。

2. GSEA分类众多（甚至可以分析GO数据集），可以根据不同需求进行不同类别的分析。

GSEA分类如下（想要查看具体的基因集可以去官网：https://www.gsea-msigdb.org/gsea/msigdb）

Human Collections

H	**hallmark gene sets** are coherently expressed signatures derived by aggregating many MSigDB gene sets to represent well-defined biological states or processes.
C1	**positional gene sets** corresponding to human chromosome cytogenetic bands.
C2	**curated gene sets** from online pathway databases, publications in PubMed, and knowledge of domain experts.
C3	**regulatory target gene sets** based on gene target predictions for microRNA seed sequences and predicted transcription factor binding sites.
C4	**computational gene sets** defined by mining large collections of cancer-oriented microarray data.
C5	**ontology gene sets** consist of genes annotated by the same ontology term.
C6	**oncogenic signature gene sets** defined directly from microarray gene expression data from cancer gene perturbations.
C7	**immunologic signature gene sets** represent cell states and perturbations within the immune system.
C8	**cell type signature gene sets** curated from cluster markers identified in single-cell sequencing studies of human tissue.

Mouse Collections

MH	**mouse-ortholog hallmark gene sets** are versions of gene sets in the MSigDB Hallmarks collection mapped to their mouse orthologs.
M1	**positional gene sets** corresponding to mouse chromosome cytogenetic bands.
M2	**curated gene sets** from online pathway databases, publications in PubMed, and knowledge of domain experts.
M3	**regulatory target gene sets** based on gene target predictions for microRNA seed sequences and predicted transcription factor binding sites.
M5	**ontology gene sets** consist of genes annotated by the same ontology term.
M8	**cell type signature gene sets** curated from cluster markers identified in single-cell sequencing studies of mouse tissue.

11. 处理数据

```
genelist <- data$log2FoldChange
# names(genelist) <- toupper(rownames(data))
names(genelist) <- rownames(data)
genelist <- sort(genelist, decreasing = T)
```

```
> genelist
     DMBT1      OOSP1       FBP1      THEM5     GM13880
 7.4338866  5.9688772  5.6502578  5.4405683  4.2910346
    PLAC1L        LTF        HAL     MFSD2A      FXYD2
 3.9255050  3.8595259  3.6917530  3.6683282  3.6010390
    HSD3B2     GM8909       CAP2   SLC22A19     GPRC5B
 3.4470780  3.3992282  3.2912731  3.2760984  3.2591520
     PTPRN      KRT85      NPHS2       SOX9       PAX7
 3.2390031  3.2339350  3.1809946  3.1609471  3.0756974
        IL6       IDI2       DPCR1       PMVK     GM15698
 3.0200912  2.9006113  2.8748999  2.8666734  2.8475421
```

正常来说，我们需要从官网下载pathway注释文件（.gmt）并进行处理（比较麻烦）。感兴趣的同学可以研究研究以下处理的代码：

```
x <- readLines("c6.all.v7.5.1.symbols.gmt")
res <- strsplit(x, "\t")
names(res) <- vapply(res, function(y) y[1], character(1))
res <- lapply(res, "[", -c(1:2))
```

好在R语言中GSEA数据库被打包进了msigdf这个包，因此我们无需再下载gmt文件，只需要对msigdf中数据库进行提取即可。**注意：该数据库中没有gene_id列，只有gene_symbol列，所以无需像GO一样将symbol转化成ID。且在human数据库中symbol列中的值均为大写，因此处理human数据时需要进行大小写转化（mouse数据则不需要）。**

```
> msigdf.human
# A tibble: 3,850,745 x 4
   category_code category_subcode geneset symbol
   <chr>         <chr>            <chr>   <chr>
 1 c1            all              chr1p12 VTCN1
 2 c1            all              chr1p12 LINC01525
 3 c1            all              chr1p12 MAN1A2
 4 c1            all              chr1p12 VPS25P1
 5 c1            all              chr1p12 TENT5C
 6 c1            all              chr1p12 VDAC2P3
 7 c1            all              chr1p12 PNRC2P1
 8 c1            all              chr1p12 GDAP2
 9 c1            all              chr1p12 WDR3
10 c1            all              chr1p12 SPAG17
```

```
> msigdf.mouse
# A tibble: 3,387,971 x 4
   category_code category_subcode geneset mouse.symbol
   <chr>         <chr>            <chr>   <chr>
 1 c1            all              chr1p12 Vtcn1
 2 c1            all              chr1p12 Man1a2
 3 c1            all              chr1p12 Tent5c
 4 c1            all              chr1p12 Gdap2
 5 c1            all              chr1p12 Wdr3
 6 c1            all              chr1p12 Spag17
 7 c1            all              chr1p12 Tbx15
 8 c1            all              chr1p12 Wars2
 9 c1            all              chr1p12 Hao2
10 c1            all              chr1p12 Hsd3b1
# ... with 3,387,961 more rows
# i Use `print(n = ...)` to see more rows
```

12. 开始GSEA

为了方便分析，可以根据自己需求写一个小函数（以下仅是我的写法）。其中返回的是一个列表，列表第一个值是通路汇总矩阵，方便人为查看，列表第二个值是GSEA分析出的大类，方便软件包后续画图。

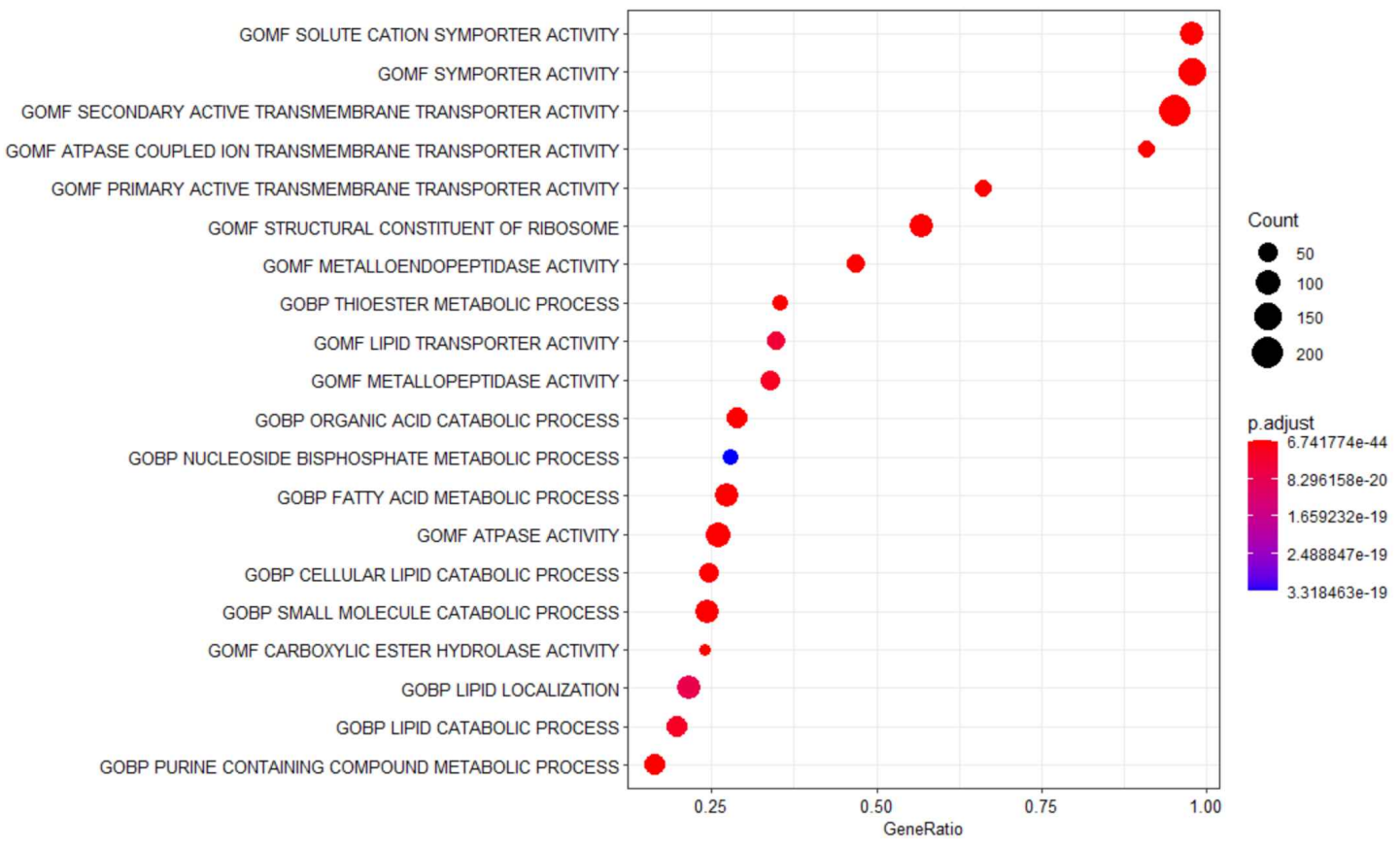
```
```

```r
myGSEA <- function(c){
  class <- msigdf.mouse %>%  # msigdf.human
    filter(category_code == c) %>%
    select(geneset, mouse.symbol) %>%
    as.data.frame
  Class_GSEA <- GSEA(genelist,
                     TERM2GENE = class,
                     minGSSize = 10,
                     maxGSSize = 500,
                     pvalueCutoff = 0.05,
                     pAdjustMethod = "BH",
                     verbose = FALSE,
                     eps = 0)

  result <- Class_GSEA@result
  # Class_GSEA@result$Description <-
  #   str_trunc(Class_GSEA@result$Description, width = 30, side = "right")
  return(list(result, Class_GSEA))
}

H_result_list <- myGSEA("h")
c5_result_list <- myGSEA("c5")

H_result <- H_result_list[[1]]
c5_result <- c5_result_list[[1]]
```

可根据函数提取出的矩阵（如本例中的c5_result）直接查看想要得知的通路信息。

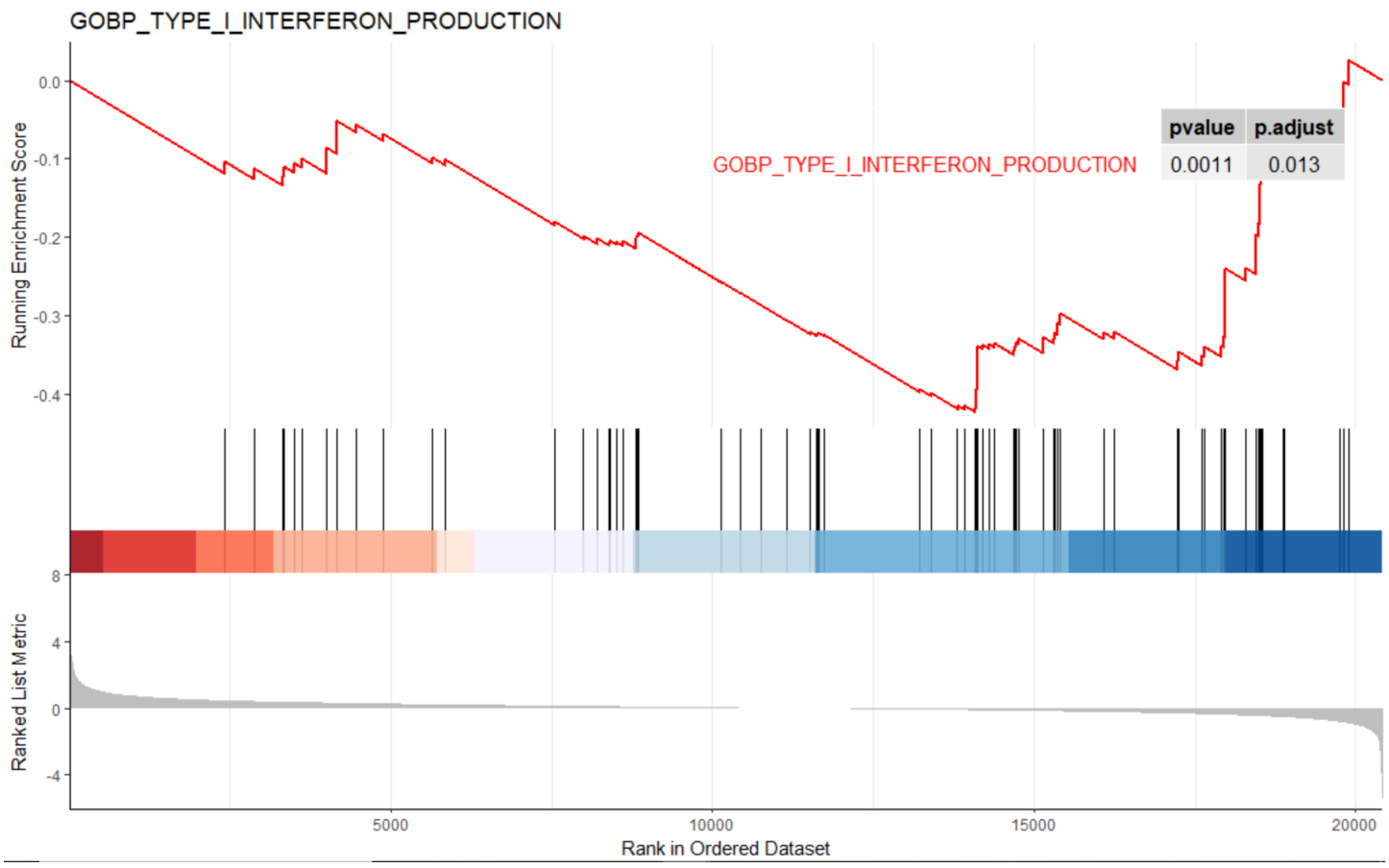
13. Dotplot

```r
# dotplot
dotplot(c5_result_list[[2]], showCategory = 20, font.size = 10)
```



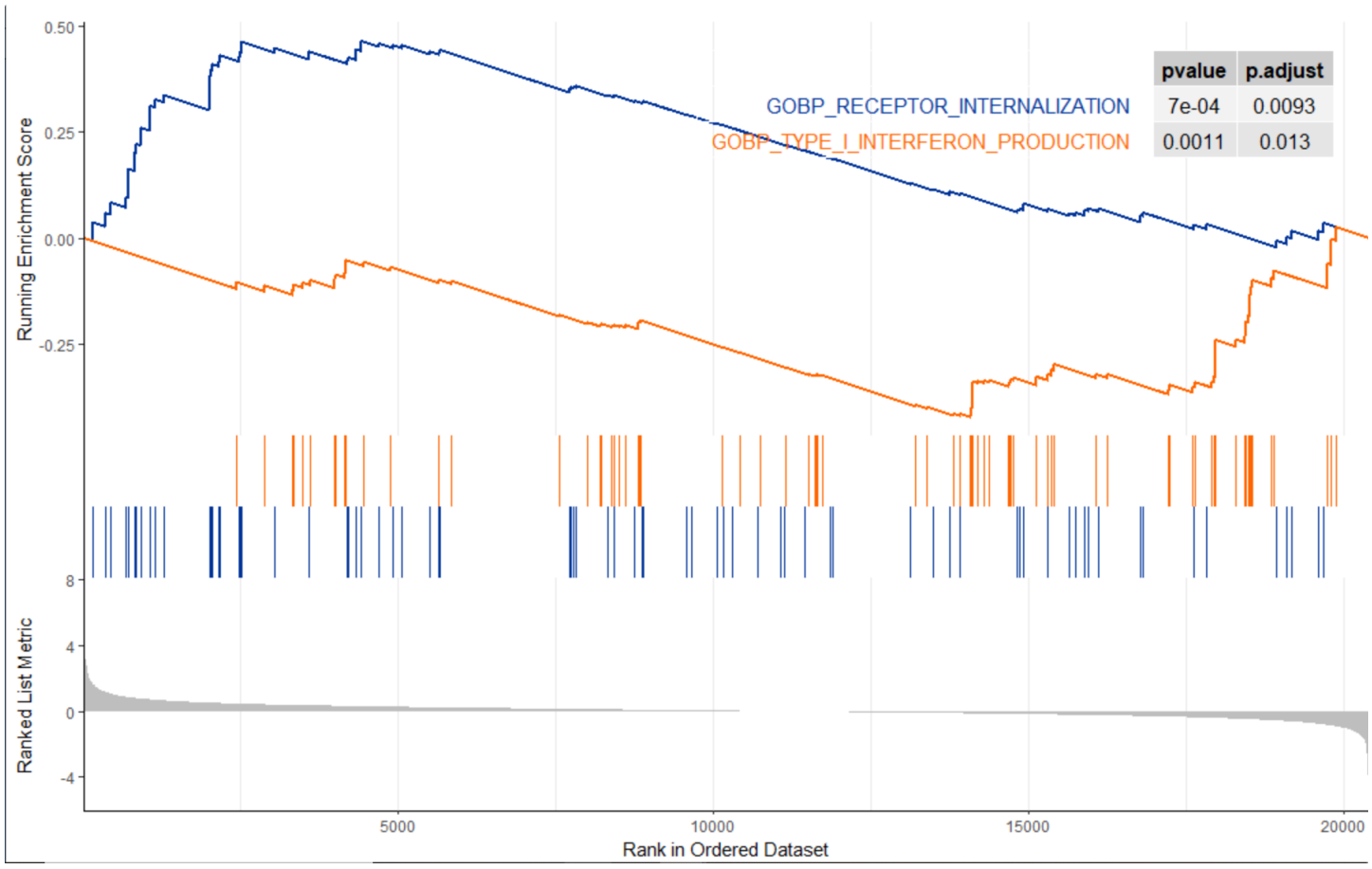
14. 富集图

```r
# 寻找某一特定通路
which(c5_result$ID == 'GOBP_TYPE_I_INTERFERON_PRODUCTION') # 848
gseaplot2(c5_result_list[[2]],
          geneSetID = 848,
          color = "red",
          rel_heights = c(1.5, 0.5, 1), # 子图高度
          subplots = 1:3, # 显示哪些子图
          pvalue_table = T, # 是否显示pvalue表
          title = c5_result_list[[1]]$Description[848],
          ES_geom = "line") # or "dot"
```



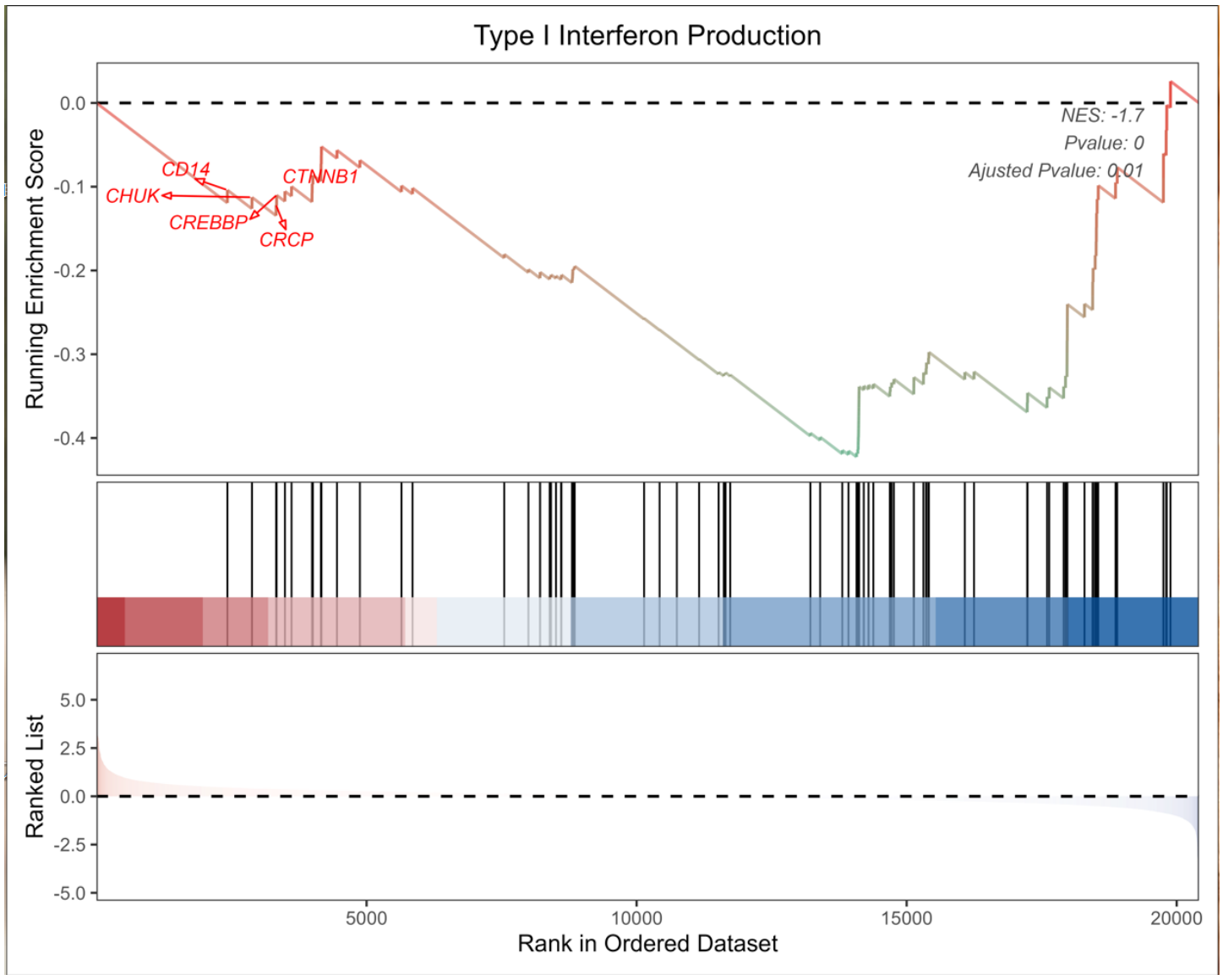
当然，也可以在一张图中同时显示多条通路

```r
gseaplot2(c5_result_list[[2]],
          geneSetID = c(772, 848),
          color = c("#003399", "#FF6600"),
          pvalue_table = TRUE,
          ES_geom = "line")
```



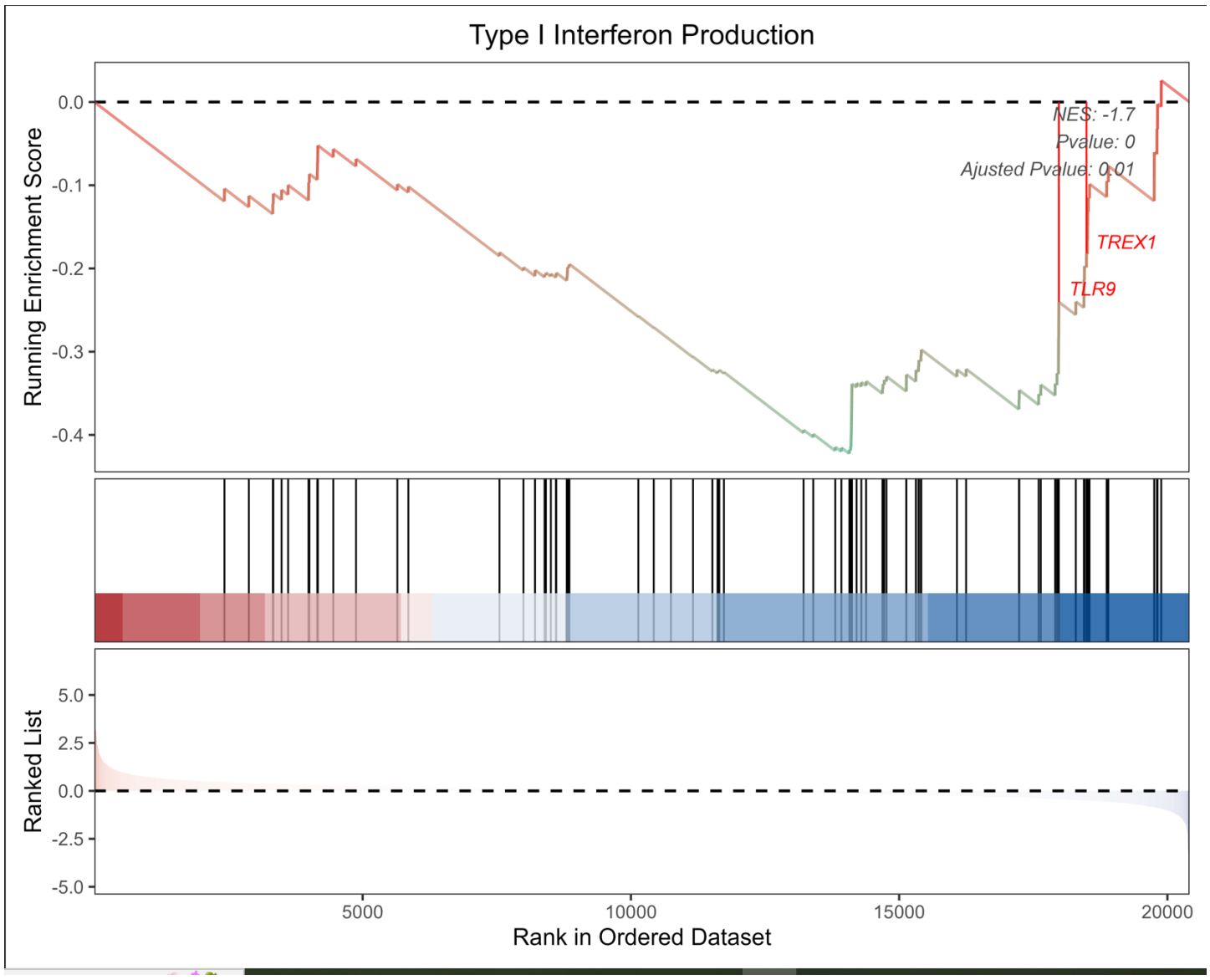
一个最近发现的新包：GseaVis，可以在图上注释基因（虽然有些乱）

```r
# 标注前5名的基因名字
gseaNb(
  object = c5_result_list[[2]],
  geneSetID = c5_result_list[[2]]@result$ID[848],
  addPval = T,
  pvalX = 0.95,
  pvalY = 0.8,
  addPoint = F,
  newCurveCol = c("#001871","#b9b4ad", "#f99f1c"),
  newHtCol = c("#001871", "white", "#f99f1c"),
  addGene = T,
  markTopgene = T, # 是否标注Top基因
  topGeneN = 5, # 标注前多少个gene
  geneCol = 'red', # 基因名标签颜色更改
  rmSegment = T
)
```



也可以按需求标注出基因及其位置。

```
gseaNb(
  object = c5_result_list[[2]],
  geneSetID = c5_result_list[[2]]@result$ID[848],
  addPval = T,
  pvalX = 0.95,
  pvalY = 0.8,
  addPoint = F,
  newCurveCol = c("#001871","#b9b4ad", "#f99f1c"),
  newHtCol = c("#001871", "white", "#f99f1c"),
  addGene = c('TREX1', 'TLR9'),
  geneCol = 'red',
  rmSegment = F
)
```



四、 Kyoto Encyclopedia of Genes and Genomes（KEGG）

KEGG分析流程与GO十分相似，只不过用了不同的数据库。因此不做过多说明。

事先说明：R语言连接KEGG十分不稳定，我经查询和多次尝试仍未得到结果，网上大多有的方法是该连接随机性太强（有的时候能得到结果，有的时候则不行）。另一种方法是在进行KEGG分析前进行一下设置。

```
R.utils::setOption("clusterProfiler.download.method", "auto")
```

15. 分析+画图

```
# KEGG
KEGG_diff <- enrichKEGG(gene = diff_entrez$ENTREZID,
                        organism = "mmu",
                        pvalueCutoff = 0.05,
                        qvalueCutoff = 0.05)

KEGG_result <- KEGG_diff@result
```

其中organism参数的设置可参考：https://www.genome.jp/kegg/catalog/org_list.html

（人hsa，小鼠mmu.....）

```
# 探索/调出选定的KEGG通路
# 直接跳转到对应网页，红色为富集到该通路的差异基因
```

```
browseKEGG(KEGG_diff, "hsa04610")
```

```
## Figure 1
barplot(
  KEGG_diff,
  x = "Count",
  color = "p.adjust",
  showCategory = 20,
  font.size = 12,
  title = "KEGG enrichment barplot",
  label_format = 30
)

## Figure 2
dotplot(
  KEGG_diff,
  x = "GeneRatio",
  color = "p.adjust",
  title = "Top 20 of Pathway Enrichment",
  showCategory = 20,
  label_format = 30
)
```

如果有任何疑问（包括但不限于不清晰的地方，解释/代码错误，更有意义的作图方式），麻烦联系卢建璋进行修改。祝大家科研顺利。