

LZ 文档

作者：荔枝 earth_farmer@outlook.com

更新：2023-11-20

目录

前言	5
为何阅读本书	5
本书结构	5
致谢	5
1 安装	7
1.1 安装 R 及 Rstudio 环境	7
1.2 安装 LZ 包	7
2 RNAseq 差异分析	9
2.1 差异分析	9
2.2 火山图	12
2.3 热图	13
3 富集分析	15
3.1 GO & KEGG 富集分析	15
3.2 GSEA 分析	18
4 差异及富集分析可视化专题	19
5 RNAseq 上游流程	21
6 多组学	23

7	CUT&TAG	25
8	单细胞分析	27
9	空间转录组	29

前言

本包致力于简化生信分析流程和批量分析。目前主要为 RNAseq 分析流程，后期会加入多组学联合分析流程。

为何阅读本书

本书是为 LZ R package 写的使用文档。旨在让完全没有编程基础或 R 基础的人学会使用 LZ 包来进行一些生信分析。LZ 包致力于简化生信分析流程和批量分析。目前 LZ 包已经完成了 RNAseq 下游分析流程的大部分，后期完善后还有加入更多的功能。例如 RNAseq 的上游分析流程，多组学联合分析流程，单细胞分析流程。通过学习完本书，您将会在不需系统学习 R 语言的情况下快速分析测序数据，如有疑问请发 email 至 earth_farmer@outlook.com，尽量有答复但不予保证。

本书结构

致谢

Chapter 1

安装

1.1 安装 R 及 Rstudio 环境

LZ R 包可以从 Github 上安装。先安装 R 及 Rstudio, 前者是核心, 后者是编辑器 (写代码的地方)。1. 安装 R 最新版, 根据系统自行选择版本, win 用户可以直接点R-4.3.2下载。2. 安装Rstudio 最新版, win 用户可直接点Rstudio Desktop下载 3. Win 电脑可以考虑安装 R 版本对应的Rtools (可选项, 新手可以不安装) 4. 重要提示: 请卸载或者至少退出一切杀毒软件 (微软自带的不用退出), 否则安装包时可能会出现难以解决的奇怪 bug。

1.2 安装 LZ 包

1. 安装 LZ 包所需的依赖包

```
# 安装 bioconductor
if (!require("BiocManager", quietly = TRUE)) {
  install.packages("BiocManager")
  # R 4.2.0
  BiocManager::install(version = "3.18") # 4.3 == 3.18
}

# 设置镜像 (可选)
options("repos" = c(CRAN = "https://mirrors.tuna.tsinghua.edu.cn/CRAN/"))
options(Bioc_mirror="https://mirrors.tuna.tsinghua.edu.cn/bioconductor")

# 安装 LZ 依赖包
```

```
# 安装 cran 包
cran_pack <- c('devtools', 'prettydoc', 'Hmisc', 'markdown',
              'Hmisc', 'tidyverse')
for (p in cran_pack) { if (!requireNamespace(p, quietly = T)) install.packages(p) }
# 安装 bioconductor 包
bioc_pack <- c("HPO.db", "DOSE", "clusterProfiler", 'DESeq2', 'edgeR',
              'limma', "topGO", 'Rgraphviz', 'org.Hs.eg.db')
for (p in bioc_pack) {
  cat(p, '=====\n')
  if (!requireNamespace(p, quietly = T)) BiocManager::install(p, update = F, ask = F)
}
```

2. 安装 LZ 包 (此包会不定时更新, 后续更新只需要重新运行这句即可, 上面的包不需要重新安装)

```
# 安装 LZ 包
devtools::install_github("ArronLZ/LZ", upgrade = "never", force = T,
                        build_vignettes = T)
# 查看文档 (点击 LZ Documents 查看网页版)
vignette('LZ')
```


Chapter 2

RNAseq 差异分析

2.1 差异分析

2.1.1 加载包

```
rm(list = ls());gc()
library(LZ)
library(tibble)
library(data.table)
library(DESeq2)
library(parallel)
library(BiocParallel)
library(ggplot2)
cat(" 您电脑线程为:", detectCores())
# 如果是 12 代以后的 interCPU, 建议最高不超过 6 或 8。服务器可加大设置, 但不能大于线程总数。
# 此处如果电脑性能一般, 建议直接使用 n=4 或者 2。
n = 4
# register(MulticoreParam(n)) 苹果和 linux 电脑使用这句替代下句
register(SnowParam(n))
```

2.1.2 数据准备

RNAseq 下游分析必须准备两个文件：表达矩阵表格文件，分组表格文件将 gene_count.csv, group.csv 放在工作目录下

- gene_count.csv 矩阵数据格式 (数值型, 整型)

ID	row1	row2	row3	row4
gene1	34	23	56	23
gene2	35	23	12	23
gene3	12	78	78	78

- group.csv 分组数据格式: 需要组的行名包含于表达谱的列名 rownames(group) %in% colname(eset)

Sample	Type	BATCH
rowname1	tumor	1
rowname2	tumor	1
rowname3	normal	2
rowname4	normal	2

- 注意: 表达文件中的基因名是 SYMBOL 还是 ESembleID。如为 EsembleID, 要注意是有小数点的 ID 还是没有小数点的。有小数点的形式为这样: ESEM00000123.34, 没有点的是 ESEM00000123。还有记住基因名这列的列名, 建议统一设置为 ID。

2.1.3 文件夹准备

本文中有时也将文件夹称为目录, 这两者等价。建议每个项目新建一个文件夹, 例如本项目新建了一个名为 LZexample 的文件夹, 然后再在这个文件夹下建了一个 data 文件夹, 以后 data 目录专门用来存原始文件, 例如 RNAseq 分析所需的 eset.csv, group.csv 或者更加原始的文件。

目录结构建议: 本项目的目录初始结构, 建议每个项目按着这个形式来。项目文件夹 LZexample 这个文件夹名建议改成有意义的名称, 一眼便能看出这个项目是什么数据或者什么目的, 而 data 文件夹名不建议更改。图片

2.1.4 差异分析预设置

```
# 设置工作目录, 即之后所有的操作如果不指定文件夹, 都将会在这个文件夹下进行
setwd("C:/data/LZexample") # 按需更改成你的项目文件夹
getwd() # 检查是否更改工作目录成功了?
# Windows 系统下默认的文件夹路径是 "C:/data/LZexample" 这中斜杆分隔文件夹,
```

```
# 如果是直接从 win 复制而来的，请将斜杠\改成反斜杠/，就如下面设置的这样。（改成\\也行）

# 设置此次分析的标记
mark <- "T_C" # 此次分析的标记（记录谁比谁或和筛选阈值，建议设置的有意义）
# 设置结果输出的文件夹，按需设定，可保持默认，如果修改只修改此处即可，下面无需修改。
# 第一次分析可以不用改，但如是第二分析，必须改这个，否则会覆盖第一次的结果。
outdir <- "result"
outdirsub <- paste0(outdir, "/", mark)
outdirsub.gsea <- paste0(outdirsub, "/gsea")
outdirsub.rich <- paste0(outdirsub, "/rich")
# 差异分析阈值设定
ffdr <- 0.1
fpval <- 0.05
flogfc <- 1
```

2.1.5 差异分析

```
# 读取并整理数据（如果都是按照上面的要求来的，不需要改这里的参数）
glist <- DEG_prepareData(eset_file = "data/eset.csv", # 表达数据的相对路径
                        group_file = "data/group.csv", # 分组文件的相对路径
                        id_dot = F, # ESEM 是否有点，有点设为 T
                        col.by = "ID", # 基因名列的列名
                        annot_trans = F, # 是否要注释，如果是 EsemblID 就需要设置为 T
                        f_mark = mark)

# 差异分析 deseq2 三部曲
dds <- DEG_DESeq2.dds(exprset.group=glist, batch = F)
DEG_DESeq2.pca(dds, outdir = outdirsub) # 此处有 warning 信息，不用管。
dds_list <- DEG_DESeq2.ana(dds)

# 差异后分析
if ( dir.exists(outdirsub.gsea) ) {
  # 如果 outdirsub.gsea 文件夹存在，清空该文件夹下所有文件
  file.remove(list.files(outdirsub.gsea, full.names = T))
}
# 构建 GSEA 官网软件分析所需格式文件
```

```
DEGres_ToGSEA(diffan.obj = dds_list, outdir = outdirsub.gsea) # 此处有 warning, 不用管。
# all_father 中记录了
#           差异分析的总表, 默认阈值的差异基因表,
#           上调基因列表, 下调基因列表
#           以及 R-GSEA 分析所需要的所有 mRNA 的表达排序列表。
# 是我们后续各种分析的万恶之源 (因此命名 all_father)
# 上述数据同时被保存在本地硬盘的到一个多 sheet 的 xls 表中【outdirsub.rich 目录中】
all_father <- DEGres_ToRICH(diffan.obj = dds_list, p=fpval, q=ffdr, f=flogfc,
                           mark=mark, outdir = outdirsub.rich)
save.image(file = paste0(outdirsub, "/1.diff.img.RDATA")) # 保存中间数据
```

2.2 火山图

需要修改的是以下几个值: df_valcano: 文件读取时的文件路径 ffdr: FDR 阈值 fpval: PValue 阈值 flogfc: logFC 阈值 filterc: 火山图展示模式 (p,ffdr 均考虑模式, 仅考虑 p 值模式, 仅考虑 ffdr 值模式, 默认为第一种"fpadj") label_gene: 展示基因列表不清楚建议默认: ffdr=0.1, pval=0.05, p,ffdr 均考虑模式。仅修改 label_gene: 展示基因列表即可。

```
# 本流程中不需要运行, 后续想再次分析时可从此步开始
# load("./result/T_C/1.diff.img.RDATA")
# library(LZ)
library(ggpubr);library(ggrepel);library(ggsci);library(scales)
library(tidyverse);library(dplyr);library(pheatmap);library(RColorBrewer)
# 导入火山图需要的数据, 即差异分析后的未筛选表格 (我们也称这个对象为 resdf,
# resdf 文件涵盖差异分析的所有结果信息, 可以做后续所有基于差异分析或者基因列表
# 的所有分析, 如果后续分析时使用其它数据, 请按这个 resdf 的格式改数据, 主要
# 就是把数据的列名改成和 resdf 的列名相同, 即可用此包的函数分析画图)
# 即 df_valcano <- readxl::read_xlsx("xxx.xlsx", sheet = 1)
df_valcano <- all_father$DIFF.ALL
names(df_valcano) # 对应的列名必须为 Gene, log2FC, PValue, FDR
# 差异分析阈值设定
ffdr <- 0.1
fpval <- 0.05
flogfc <- 1
# 模式设定
filterc <- "fpadj" # pvalue, padj 均考虑模式 ("fpadj": 仅考虑 ffdr 值模式, "other": 仅考虑 p 值模式)
# 设定需要标记的 marker gene
```

```

label_gene <- c('TFRC', 'ACSL1', 'LPCAT3', 'PCBP1', 'FTH1', 'SLC11A2',
               'SLC39A8', 'SAT1', 'FTL', 'GSS')
# 查看想展示的基因在不在差异分析总表中
# label_gene %in% df_valcano$Gene %>% all
# pic_data %>% filter(Row.names %in% label_gene)
# 火山图数据预处理
pic_data <- DEGp_prepareVolcano(df_valcano = df_valcano, filterc = filterc)
# 火山图 无标记
DEGp_Volcano(result = pic_data, logFC = flogfc,
              adj_P = ffdr, label_geneset = NULL)
ggsave(paste0(outdirsub, "/valcano.pdf"), width = 7, height = 7) # 保存
# 火山图 有标记
DEGp_Volcano(result = pic_data, logFC = flogfc, # log2(2)
              adj_P = ffdr, label_geneset = label_gene) %>%
  ggplotGrob() %>% cowplot::plot_grid()
ggsave(paste0(outdirsub, "/valcano.mark.gene.pdf"), width = 7, height = 7) # 保存

```

2.3 热图

On the way ...

Chapter 3

富集分析

3.1 GO & KEGG 富集分析

3.1.1 一键脚本 (批量处理)

这是一个一键脚本，请新建一个单独的文件写这段脚本，然后按这个脚本的顶部注释修改 `resdf` `outd` `fc.list` 处即可，运行即可批量出不同 FC 的富集分析结果。

```
# 此脚本为 GO、KEGG 分析（需要一个输入文件即可，为差异分析流程后的 resdf 文件）
# 即为第一步（或 1 脚本）的结果的一个结果文件（DIFF_an_***.xlsx）
# 即为 resdf 文件，此文件是差异分析后的总表
# 注意如果采用了其他的分析方法得到差异分析后表，运行这个脚本时可能需要更改列名
# 即我们的 resdf 对象的列名为 Gene, log2FC, PValue, FDR，需要与这些个列名保持一致。
# 此脚本中的需要修改的位于 /// *** /// 行中，另外还有一个 LZ::setproxy() 行，
# 如果没有代理工具，或者代理工具不支持 http 代理，或者端口不通，请不要运行。
rm(list = ls());gc() # 清空所有对象，慎用，必要时用
suppressMessages({ suppressWarnings({
  library(LZ)
  library(tidyverse);library(data.table)
  library(clusterProfiler);library(enrichplot)
  library(topGO);library(Rgraphviz)
  library(RColorBrewer);library(ggsci);library(pheatmap)
  library(xlsx);library(readxl)
}) })
# 若无代理工具，切勿运行
```

```

# LZ::setproxy() # 高危!!! 新手不要运行此行, 会使当前窗口断网!!!
# Sys.getenv('http_proxy') Sys.setenv('http_proxy='') Sys.setenv('https_proxy='')

# 读取数据 resdf 存放目录
# resdf <- readxl::read_xlsx("result/rnaseqOR-NC/rich/DIFF.an_OR-NC.xlsx",
#                             sheet = 1) %>% as.data.frame()
# 此处可能需要插入修改列名的代码, 需要为标准的 resdf 格式
# 标准 resdf 格式, 用列名 Gene, log2FC, PValue, FDR 来表示 gene, log2fc, p, q/fdr
resdf <- all_father$DIFF.ALL
# 输出目录
outd = "result/xx/rich"
# logFC 阈值, 多个阈值的话, 写成 fc.list <- list('1.2' = log2(1.2), '2' = log2(2))
# 注意!!!!!! : 括号里 log2(2) 的 2, 和引号里 '2' 的 2 都要需同步要改。!!!
# 否则可能会覆盖结果
# logFC 阈值, 多个阈值
fc.list <- list('1.5' = log2(1.5), '2' = log2(2), '4' = log2(4) )
# logFC 阈值, 单个阈值
# fc.list <- list('2'=log2(2))
# 设置物种为人类 (如是人类则不需要更改)
GO_database <- 'org.Hs.eg.db' # keytypes(org.Hs.eg.db)
KEGG_database <- 'hsa'

# 预处理数据符合 GOKEGG 分析的要求
# # 不同 fc 条件下的 GOgenelist list(ALL, UP, DOWN)
gogenelist <- lapply(fc.list, function(x) DEG_prepareGOglist(resdf, logfc = x,
                                                             p = 0.05, fdr = 0.1))
# gogenelist %>% length()
# 对 logFC 迭代, 每个 FC 新建一个目录, 用来存 upgene, downgene, allgene 的 GO 结果
enrich <- DEG_runENRICH(genelist = gogenelist, outdir = outd, glist.save = T,
                       rungo = T, runkegg = T, rapid = T)

```

3.1.2 简易 GO,KEGG 一次分析

如果已经得到了差异基因列表, 且无需批量分析, 可以进行这个简易分析。数据格式: head(genelist.lh)

```
[1] "AARS1" "AATF" "ABCB7" "ABCE1" "ABHD11" "ABHD12"
```



```

# 简易 GO,KEGG 一次分析 (即: 已经得到了差异基因列表)
# LZ::setproxy() # 代理设置, 新手别碰, 会断网
# 差异基因列表
genelist.lh <- pic.list$sig.data$Gene
# 转换 ID
gene_df <- bitr(genelist.lh, fromType = "SYMBOL", toType = c("ENTREZID", "UNIPROT"),
                OrgDb = 'org.Hs.eg.db')
# GO 分析
go.lh <- DEG_GO(gene_df, orgdb = "org.Hs.eg.db", sigNodes = 20,
                resultdir = "./result/proteinOR-NC", filemark = "p1.5_g_2")
go.lhdf <- sapply(go.lh, function(x) x@result, simplify = T)
write_xlsx(go.lhdf, path = "./result/xx/lh_go.all.xlsx")
# KEGG 分析
kegg.lh <- DEG_KEGG(gene_df)
write_xlsx(kegg.lh$pSigDF, path = "./result/xx/lh_kegg.all.xlsx")

```

3.1.3 GO、KEGG 分析结果可视化 {#enrich-visual}

```

# dotplot go
# 读取 go 分析保存的表格
# dotData <- go$GODF$ 倍数 "$ 变化趋势 (BP)
# dotData <- readxl::read_xlsx("kegg.xlsx", sheet = 1) 自定义挑选想要通路后的表格
dotData <- enrich$GODF$'2'$all
# 筛选数据 (按需配合其他筛选)
dotData <- DEGp_prepareDotplot(dotData, head = 30, delete = NULL)
pic.dot <- DEGp_Dotplot(dotData, title = 'TOP of GO',
                       resultdir = "./result/proteinOR-NC", filemark = 'GO_top',
                       pic.save = T)

# dotplot kegg
# 读取 kegg 分析保存的表格
# dotDataK <- readxl::read_xlsx("./result/proteinOR-NC/lh_kegg.all.xlsx", sheet = 1)
# 读取 kegg 分析保存的表格
dotDataK <- enrich$KEGGDF$'2'$up
# 筛选数据 (按需配合其他筛选)
dotDataK <- DEGp_prepareDotplot(dotDataK, head = 30, delete = NULL)
pic.dotk <- DEGp_Dotplot(dotDataK, title = 'TOP of KEGGpathway',

```

```
resultdir = "./result/proteinOR-NC", filemark = 'KEGG_top',  
pic.save = F)  
  
# 组合图  
gh <- ggplotGrob(pic.dot)  
gd <- ggplotGrob(pic.dotk)  
cowplot::plot_grid(gh, gd, rel_widths = c(1, 1.25))  
ggsave(paste0(dir_out, "/GO_KEGG_top.pdf"), width = 16, height = 10)
```

3.2 GSEA 分析

3.2.1 R GSEA

On the way ...

3.2.2 GSEA 官方软件

On the way ...

Chapter 4

差异及富集分析可视化专题

On the way ...

Chapter 5

RNAseq 上游流程

On the way ...

Chapter 6

多组学

On the way

Chapter 7

CUT&TAG

On the way ...

Chapter 8

单细胞分析

On the way ...

Chapter 9

空间转录组

On the way ...