R包DESeq2差异基因分析

高通量测序分析（例如RNA-Seq、ChIP-Seq等）以计数数据的形式提供定量读数。为了良好地推断此类数据中的差异信号，选择适当的统计方法估算动态范围内的数据波动以及合适的误差模型，特别是在低重复、高离散程度以及存在离群值的情况时是必需的。DESeq2是一种基于负二项式分布的方法，使用局部回归推算均值和方差，通过离散度和倍数变化的收缩率估计以提高稳定性。定量分析关注的更多是差异表达的“强度”，而非“存在”。

本篇简介R语言DESeq2包的差异基因分析流程。

下文中使用的示例数据及相关R代码的百度盘链接：

<https://pan.baidu.com/s/1LpWCFY6BnveGxMjNDT1kuA>

## 安装DESeq2包

常规方法，通过Bioconductor安装DESeq2。

BiocManager::install('DESeq2')

用过DESeq2的同学们肯定深有感触，小数据量时倒也没啥，但是大数据量时会特别让人抓狂，一个字，慢……尽管可以并行运算，但如果没有服务器，小笔记本简直伤不起啊。

前段时间DESeq2做出了重大的更新，效率得到质的飞跃。目前的最新版本是v1.25.9，运行速率相对于先前的版本提升了几十倍！但是最新版尚未添加至Bioconductor中，也就是说通过Bioconductor安装的还是以前的版本。

如果想尝试新版DESeq2，源码的github链接：<https://github.com/mikelove/DESeq2>

或者，在R中通过以下命令行链接至github直接安装。

devtools::install\_github('mikelove/DESeq2@ae7c6bd')

#若中间提示有其它依赖 R 包的旧版包冲突的话，先删除旧包再安装新的

remova.packages('xxx')

BiocManager::install('xxx')

## DESeq2差异基因分析的一般过程

网盘示例数据“gene.txt”，是一个基因表达量数据矩阵，包含16列样本，已剔除了全为0值的行。16个样本中8个样本属于control组（c），8个样本属于treat组（t）。接下来展示DESeq2分析的一般过程。

差异基因分析需要指定比较分组的先后顺序，以确定谁相对于谁的表达量上调/或下调。第一种方式是在读取分组文件后，将分组列转换为因子类型（factor），并指定因子（分组）顺序，对照在前处理在后，如下所示。第二种方式是在后续使用results()获取差异结果时，指定比较的分组（详见下文）。

library(DESeq2)

#基因表达矩阵

gene <- read.delim('gene.txt', row.names = 1, sep = '\t', stringsAsFactors = FALSE, check.names = FALSE)

#指定分组因子顺序

#注意要保证表达矩阵中的样本顺序和这里的分组顺序是一一对应的

coldata <- data.frame(group = factor(rep(c('control', 'treat'), each = 8), levels = c('control', 'treat')))

**第一步，构建DESeqDataSet对象**

第1步是构建DESeqDataSet，标准化reads count值，并用于存储输入值、中间计算和差异分析的结果。

#第一步，构建 DESeqDataSet 对象，详见 ?DESeqDataSetFromMatrix

dds <- DESeqDataSetFromMatrix(countData = gene, colData = coldata, design = ~group)

#查看归一化后的 count 值分布

boxplot(log10(assays(dds)[['cooks']]), range = 0, las = 2)

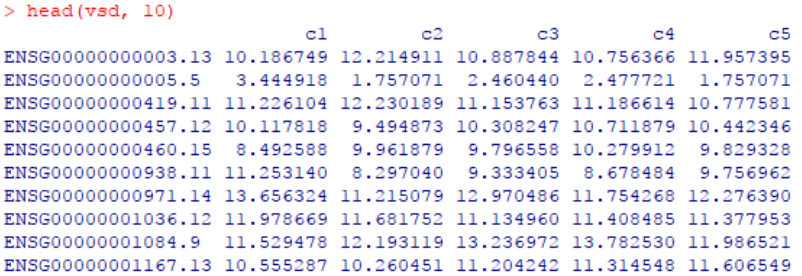
plotDispEsts(dds)

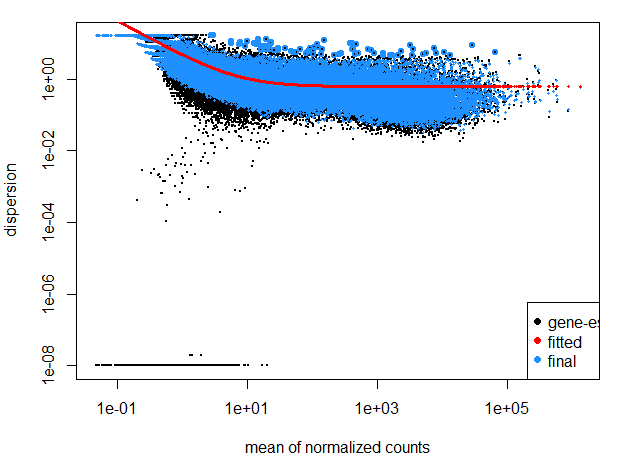
#获取归一化的基因表达矩阵

vsd <- assay(vst(dds, blind = FALSE))

head(vsd, 10)

#write.table(vsd, 'norm\_matrix.txt', sep = '\t', col.names = NA, quote = FALSE)





**第二步，差异分析**

之后直接运行默认的DESeq2差异分析流程就可以了。

函数DESeq()是一个包含因子大小估计、离散度估计、负二项模型拟合、Wald统计等多步在内的过程，结果将返回至DESeqDataSet对象。这步比较耗时，特别是数据量较大时，新旧版DESeq2的运算效率差距极为明显。

通过result()可获得最终计算的log2倍数变化和校正后p值等信息。contrast参数用于指定比较的分组顺序，即谁相对于谁的表达量上调/或下调；pAdjustMethod设定p值校正方法；alpha为显著性水平，这里0.05为校正后p值小于0.05即为显著。

#详见 ?DESeq 和 ?results

#标准方法

dds <- DESeq(dds, parallel = FALSE) #若 parallel = TRUE 将启用多线程模式

suppressMessages(dds)

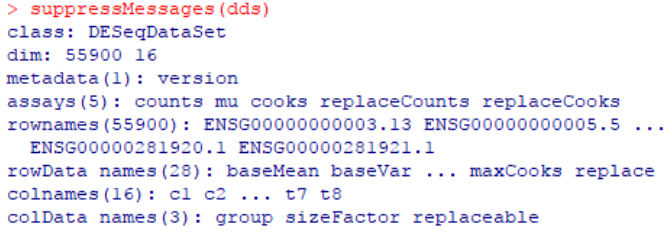
res <- results(dds, contrast = c('group', 'treat', 'control'), pAdjustMethod = 'fdr', alpha = 0.05)

#an alternate analysis: likelihood ratio test

ddsLRT <- DESeq(dds, test = 'LRT', reduced = ~ 1)

suppressMessages(ddsLRT)

resLRT <- results(ddsLRT, contrast = c('group', 'treat', 'control'), pAdjustMethod = 'fdr', alpha = 0.05)



#简要查看结果，例如

res

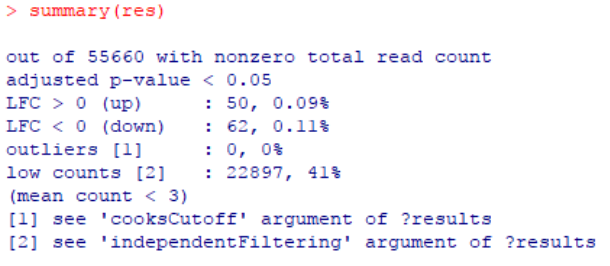
#再如

summary(res)

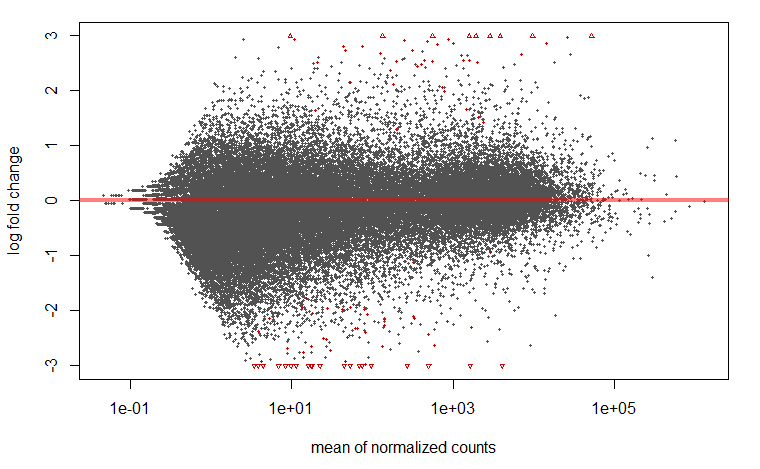
#再如

plotMA(res, alpha = 0.05, ylim = c(-3, 3))

通过summary()，可以根据预先设定的校正后p值<0.05水平（alpha=0.05，由results()指定），输出所比较两组间的上调/下调基因数量。这个结果可供参考，在后续也可以自己根据log2FC和校正后p值自定义作筛选。



通过内置函数plotMA()，可以根据预先设定的校正后p值<0.05水平（alpha=0.05），通过火山图展示差异基因。其中差异的基因默认展示为红色（无论上调或下调），由于通过ylim = c(-3, 3)参数指定了所展示的log2FC的边界区域，位于区域外的基因将以三角形展示在图中边界处。



差异分析结果保存在“res”中，可通过as.data.frame()直接转化为数据框类型。包含了基因id、标准化后的基因表达值的平均值、log2FoldChange值、显著性p值以及校正后p值等主要信息。通过该表，即可根据log2FC和校正后p值等信息自定义筛选差异表达基因了。

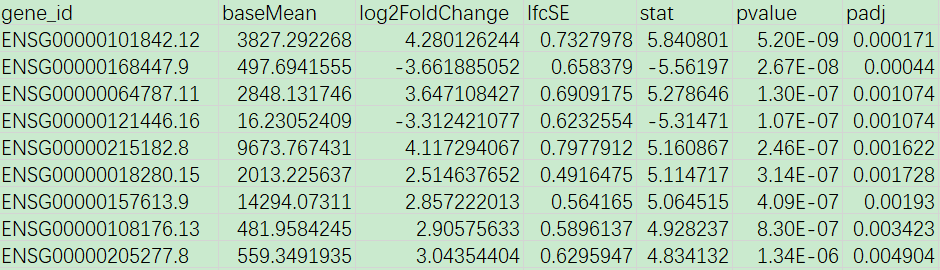
#可以先按校正和 p 值由小到大排个序，方便查看

deseq\_res <- as.data.frame(res[order(res$padj), ])

#输出

deseq\_res$gene\_id <- rownames(deseq\_res)

write.table(deseq\_res[c(7, 1:6)], 'DESeq2.txt', row.names = FALSE, sep = '\t', quote = FALSE)



## 自定义ggplot2作图示例

我们还可以根据获得的结果，通过其它作图包绘制差异火山图进行可视化展示。如下展示了ggplot2示例，横坐标为log2FoldChange，纵坐标为-log10 padj，差异基因展示为不同颜色。

library(ggplot2)

deseq\_res <- read.delim('DESeq2.txt', sep = '\t')

#例如这里根据 |log2FC| >= 1 & FDR p-value < 0.05 定义“差异”

deseq\_res[which(deseq\_res$padj %in% NA),'sig'] <- 'no diff'

deseq\_res[which(deseq\_res$log2FoldChange >= 1 & deseq\_res$padj < 0.05),'sig'] <- 'rich (p.adj < 0.05, log2FC >= 1)'

deseq\_res[which(deseq\_res$log2FoldChange <= -1 & deseq\_res$padj < 0.05),'sig'] <- 'down (p.adj < 0.05, log2FC <= -1)'

deseq\_res[which(abs(deseq\_res$log2FoldChange) < 1 | deseq\_res$padj >= 0.05),'sig'] <- 'no diff'

#纵轴为显著性 p 值

volcano\_p <- ggplot(deseq\_res, aes(log2FoldChange, -log(padj, 10))) +

geom\_point(aes(color = sig), alpha = 0.6, size = 1) +

scale\_color\_manual(values = c('blue2', 'gray30', 'red2')) +

theme(panel.grid = element\_blank(), panel.background = element\_rect(color = 'black', fill = 'transparent'), legend.position = c(0.26, 0.92)) +

theme(legend.title = element\_blank(), legend.key = element\_rect(fill = 'transparent'), legend.background = element\_rect(fill = 'transparent')) +

geom\_vline(xintercept = c(-1, 1), color = 'gray', size = 0.25) +

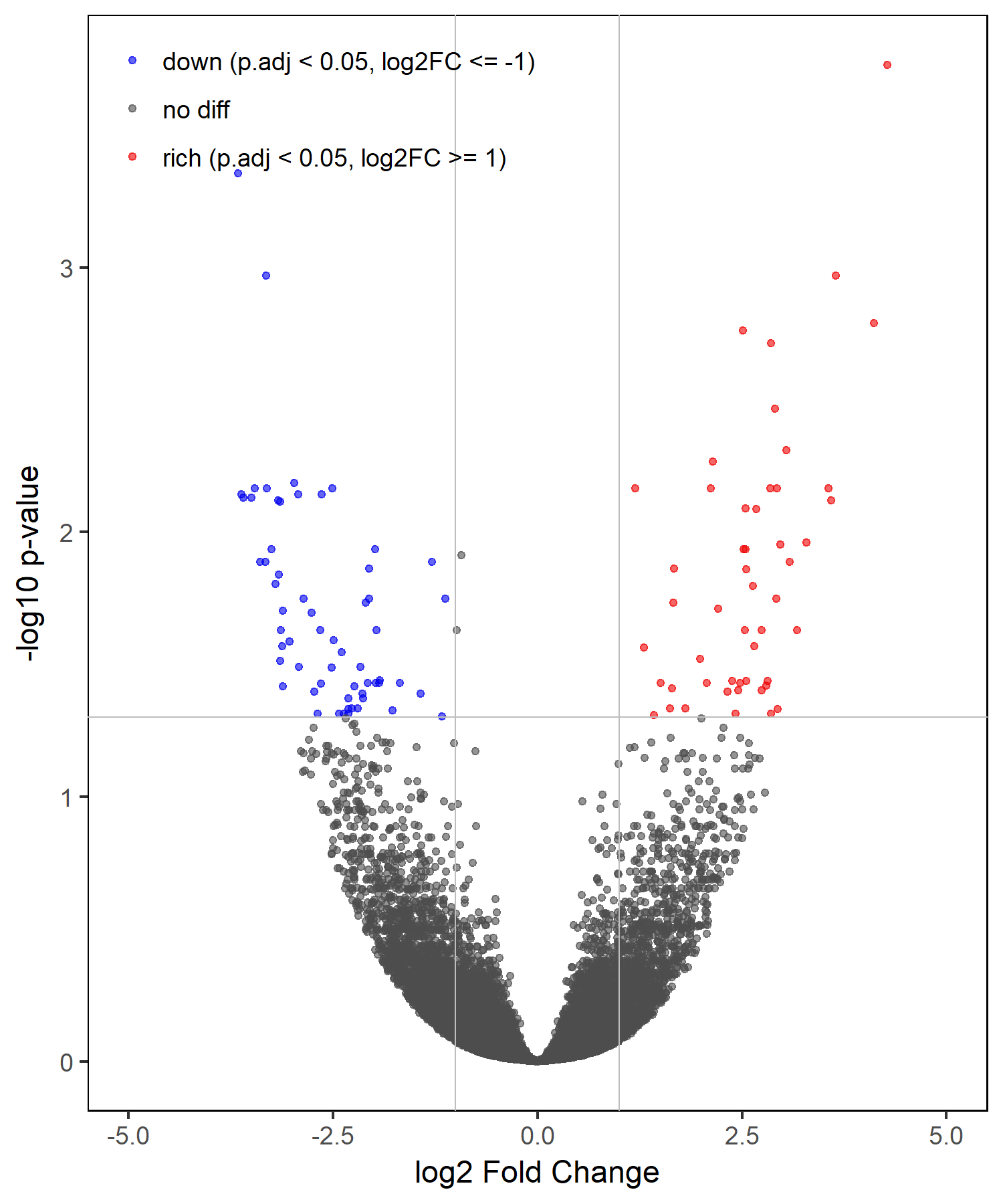
geom\_hline(yintercept = -log(0.05, 10), color = 'gray', size = 0.25) +

labs(x = 'log2 Fold Change', y = '-log10 p-value', color = NA) +

xlim(-5, 5)

#ggsave('volcano\_p.pdf', volcano\_p, width = 5, height = 6)

ggsave('volcano\_p.png', volcano\_p, width = 5, height = 6)



纵坐标为log2FoldChange，横坐标展示为标准化后的基因表达量的平均值（并log转化）。对于差异的基因，仍然以不同颜色表示。

#纵轴为基因表达值的 log10

volcano\_count <- ggplot(deseq\_res, aes(y = log2FoldChange, x = log10(baseMean))) +

geom\_point(aes(color = sig), alpha = 0.6, size = 1) +

scale\_color\_manual(values = c('blue2', 'gray30', 'red2')) +

theme(panel.grid = element\_blank(), panel.background = element\_rect(color = 'black', fill = 'transparent'), legend.position = c(0.2, 0.9)) +

theme(legend.title = element\_blank(), legend.key = element\_rect(fill = 'transparent'), legend.background = element\_rect(fill = 'transparent')) +

geom\_hline(yintercept = c(-1, 1), color = 'gray', size = 0.25) +

labs(y = 'log2 Fold Change', x = 'Average log10 baseMean') +

ylim(-5, 5)

#ggsave('volcano\_count.df', volcano\_count, width = 7, height = 5)

ggsave('volcano\_count.png', volcano\_count, width = 7, height = 5)

