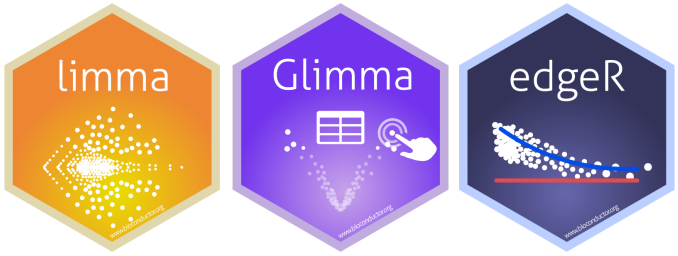
使用limma、Glimma和edgeR，RNA-seq数据分析易如反掌

***Charity Law*[1], *Monther Alhamdoosh*[2], *Shian Su*[3], *Xueyi Dong*[3], *Luyi Tian*[1], *Gordon K. Smyth*[4] *and Matthew E. Ritchie*[5]**

[1]The Walter and Eliza Hall Institute of Medical Research, 1G Royal Parade, Parkville, VIC 3052, Melbourne, Australia; Department of Medical Biology, The University of Melbourne, Parkville, VIC 3010, Melbourne, Australia
[2]CSL Limited, Bio21 Institute, 30 Flemington Road, Parkville, Victoria 3010, Australia
[3]The Walter and Eliza Hall Institute of Medical Research, 1G Royal Parade, Parkville, VIC 3052, Melbourne, Australia
[4]The Walter and Eliza Hall Institute of Medical Research, 1G Royal Parade, Parkville, VIC 3052, Melbourne, Australia; School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Melbourne, Australia
[5]The Walter and Eliza Hall Institute of Medical Research, 1G Royal Parade, Parkville, VIC 3052, Melbourne, Australia; Department of Medical Biology, The University of Melbourne, Parkville, VIC 3010, Melbourne, Australia; School of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Melbourne, Australia

2018年12月17日

Contents



1 摘要

简单且高效地分析RNA测序数据的能力正是Bioconductor的核心优势。 RNA-seq分析通常从基因水平的序列计数开始，涉及到数据预处理，探索性数据分析，差异表达检验以及通路分析，得到的结果可用于指导进一步实验和验证研究。 在这篇工作流程文章中，我们通过分析来自小鼠乳腺的RNA测序数据，示范了如何使用流行的**edgeR**包载入、整理、过滤和归一化数据，然后用**limma**包的*voom*方法、线性模型和经验贝叶斯调节（empirical Bayes moderation）来评估差异表达并进行基因集检验。通过使用**Glimma**包，此流程得到了增进，实现了结果的互动探索，使用户得以查看单个样本与基因。 这三个软件包提供的完整分析突出了研究人员可以使用Bioconductor轻松地从RNA测序实验的原始计数揭示生物学意义。

2 背景介绍

RNA测序（RNA-seq）已成为基因表达研究的主要技术。其中，基因组规模的多条件基因差异表达检测是研究者最常探究的问题之一。对于RNA-seq数据，来自Bioconductor项目(Huber et al. 2015)的 **edgeR** (Robinson, McCarthy, and Smyth 2010)和**limma**包 (Ritchie et al. 2015)提供了一套用于处理此问题的完善的统计学方法。

在这篇文章中，我们描述了一个用于分析RNA-seq数据的**edgeR** - **limma**工作流程，使用基因水平计数作为输入，经过预处理和探索性数据处理，然后得到差异表达（DE）基因和基因表达特征（gene signatures）的列表。**Glimma**包(Su et al. 2017)提供的交互式图表可以同时呈现整体样本和单个基因水平的数据，使得我们相对静态的**R**图表而言，可以探索更多的细节。

此工作流程中所分析的实验来自Sheridan等（2015）(Sheridan et al. 2015)，由三个细胞群组成，即基底（basal）、管腔祖细胞（liminal progenitor, LP）和成熟管腔（mature luminal, ML）。细胞群皆分选自雌性处女小鼠的乳腺，每种都设三个生物学重复。RNA样品分三个批次使用Illumina HiSeq 2000进行测序，得到长为100碱基对的单端序列片段。

本文所描述的分析假设从RNA-seq实验获得的序列片段已经与适当的参考基因组比对，并已经在基因水平上对序列进行了统计计数。在本文条件下，使用**Rsubread**包提供的基于**R**的流程将序列片段与小鼠参考基因组（mm10）比对（具体而言，先使用 `align` 函数(Liao, Smyth, and Shi 2013)，然后使用 `featureCounts` (Liao, Smyth, and Shi 2014)进行基因水平的总结，利用其内置的*mm10*基于RefSeq的注释）。

这些样本的计数数据可以从Gene Expression Omnibus (GEO)数据库http://www.ncbi.nlm.nih.gov/geo/ (http://www.ncbi.nlm.nih.gov/geo/)使用GEO序列登记号GSE63310下载。更多关于实验设计和样品制备的信息也可以在GEO使用该登记号查看。

3 初始配置

```
library(limma)
library(Glimma)
library(edgeR)
library(Mus.musculus)
```

4 数据整合

4.1 读入计数数据

为开始此分析，从https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE63310&format=file (https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE63310&format=file)在线下载文件*GSE63310_RAW.tar*，并从压缩包中解压出相关的文件。下方的代码将完成此步骤，或者您也可以手动进行这一步并继续后续分析。

```
url <- "https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE63310&for
mat=file"
utils::download.file(url, destfile="GSE63310_RAW.tar", mode="wb")
utils::untar("GSE63310_RAW.tar", exdir = ".")
files <- c("GSM1545535_10_6_5_11.txt", "GSM1545536_9_6_5_11.txt",
"GSM1545538_purep53.txt",
  "GSM1545539_JMS8-2.txt", "GSM1545540_JMS8-3.txt", "GSM1545541_JMS
8-4.txt",
  "GSM1545542_JMS8-5.txt", "GSM1545544_JMS9-P7c.txt", "GSM1545545_J
MS9-P8c.txt")
for(i in paste(files, ".gz", sep=""))
  R.utils::gunzip(i, overwrite=TRUE)
```

每一个文本文件均包含一个给定样品的原始基因水平计数。需要注意的是，我们的分析仅包含了此实验中的basal、LP和ML样品（请查看下方相关文件名）。

```
files <- c("GSM1545535_10_6_5_11.txt", "GSM1545536_9_6_5_11.txt",
   "GSM1545538_purep53.txt", "GSM1545539_JMS8-2.txt",
   "GSM1545540_JMS8-3.txt", "GSM1545541_JMS8-4.txt",
   "GSM1545542_JMS8-5.txt", "GSM1545544_JMS9-P7c.txt",
   "GSM1545545_JMS9-P8c.txt")
read.delim(files[1], nrow=5)
```

```
##    EntrezID GeneLength Count
## 1    497097       3634     1
## 2 100503874       3259     0
## 3 100038431       1634     0
## 4     19888       9747     0
## 5     20671       3130     1
```

尽管这九个文本文件可以分别读入**R**然后合并为一个计数矩阵，**edgeR**提供了更方便的途径，使用 `readDGE` 函数即可一步完成。得到的DGEList对象中包含一个计数矩阵，它的27179行分别对应唯一的Entrez基因标识（ID），九列分别对应此实验中的每个样品。

```
x <- readDGE(files, columns=c(1,3))
class(x)
```

```
## [1] "DGEList"
## attr(,"package")
## [1] "edgeR"
```

```
dim(x)
```

```
## [1] 27179     9
```

如果来自所有样品的计数存储在同一个文件中，数据可以首先读入**R**再使用 `DGEList` 函数转换为一个DGEList对象。

4.2 组织样品信息

为进行下游分析，与实验设计有关的样品水平信息需要与计数矩阵的列关联。这里需要包括各种对表达水平有影响的实验变量，无论是生物变量还是技术变量。例如，细胞类型（在这个实验中是basal、LP和ML），基因型（野生型、敲除），表型（疾病状态、性别、年龄），样品处理（用药、对照）和批次信息（如果样品是在不同时间点进行收集和分析的，记录进行实验的时间）等。

我们的DGEList对象中包含的 `samples` 数据框同时存储了细胞类型（ `group` ）和批次（测序泳道 `lane` ）信息，每种信息都包含三个不同的水平。需要注意的是，在 `x$samples` 中，程序会自动计算每个样品的文库大小，归一化系数会被设置为1。为了简单起见，我们从我们的DGEList对象 `x` 的列名中删去了GEO样品ID（GSM*）。

```
samplenames <- substring(colnames(x), 12, nchar(colnames(x)))
samplenames
```

```
## [1] "10_6_5_11" "9_6_5_11"  "purep53"   "JMS8-2"    "JMS8-3"
"JMS8-4"    "JMS8-5"
## [8] "JMS9-P7c"  "JMS9-P8c"


colnames(x) <- samplenames
group <- as.factor(c("LP", "ML", "Basal", "Basal", "ML", "LP",
                     "Basal", "ML", "LP"))
x$samples$group <- group
lane <- as.factor(rep(c("L004","L006","L008"), c(3,4,2)))
x$samples$lane <- lane
x$samples


##                             files group lib.size norm.factors l
ane
## 10_6_5_11 GSM1545535_10_6_5_11.txt    LP 32863052            1 L
004
## 9_6_5_11   GSM1545536_9_6_5_11.txt    ML 35335491            1 L
004
## purep53     GSM1545538_purep53.txt Basal 57160817            1 L
004
## JMS8-2       GSM1545539_JMS8-2.txt Basal 51368625            1 L
006
## JMS8-3       GSM1545540_JMS8-3.txt    ML 75795034            1 L
006
## JMS8-4       GSM1545541_JMS8-4.txt    LP 60517657            1 L
006
## JMS8-5       GSM1545542_JMS8-5.txt Basal 55086324            1 L
006
## JMS9-P7c   GSM1545544_JMS9-P7c.txt    ML 21311068            1 L
008
## JMS9-P8c   GSM1545545_JMS9-P8c.txt    LP 19958838            1 L
008
```

4.3 组织基因注释

我们的DGEList对象中的第二个数据框名为 genes ，用于存储与计数矩阵的行相关联的基因水平的信息。 为检索这些信息，我们可以使用包含特定物种信息的包，比如小鼠的**Mus.musculus** (Bioconductor Core Team 2016b)（或人类的**Homo.sapiens** (Bioconductor Core Team 2016a)）；或者也可以使用**biomaRt** 包 (Durinck et al. 2005, 2009)，它通过接入Ensembl genome数据库来进行基因注释。

可以检索的信息类型包括基因符号（gene symbols）、基因名称（gene names）、染色体名称和位置（chromosome names and locations）、Entrez基因ID（Entrez gene IDs）、Refseq基因ID（Refseq gene IDs）和Ensembl基因ID（Ensembl gene IDs）等。**biomaRt**主要使用Ensembl基因ID进行检索，而由于**Mus.musculus**包含多种不同来源的信息，它允许用户从多种不同基因ID中选取检索键。

我们使用**Mus.musculus**包，利用我们数据集中的Entrez基因ID来检索相关的基因符号和染色体信息。

```
geneid <- rownames(x)
genes <- select(Mus.musculus, keys=geneid, columns=c("SYMBOL", "TXC
HROM"),
                keytype="ENTREZID")
head(genes)
```

```
##    ENTREZID  SYMBOL TXCHROM
## 1    497097    Xkr4    chr1
## 2 100503874 Gm19938    <NA>
## 3 100038431 Gm10568    <NA>
## 4     19888     Rp1    chr1
## 5     20671   Sox17    chr1
## 6     27395  Mrpl15    chr1
```

与任何基因ID一样，Entrez基因ID可能不能一对一地匹配我们想获得的基因信息。在处理之前，检查重复的基因ID和弄清楚重复的来源非常重要。我们的基因注释中包含28个匹配到不同染色体的基因（比如基因Gm1987关联于染色体*chr4*和*chr4_JH584294_random*，小RNA Mir5098关联于*chr2*，*chr5*，*chr8*，*chr11*和*chr17*）。为了处理重复的基因ID，我们可以合并来自多重匹配基因的所有染色体信息，比如将基因Gm1987分配到*chr4 and chr4_JH584294_random*，或选取其中一条染色体来代表具有重复注释的基因。为了简单起见，我们选择后者，保留每个基因ID第一次出现的信息。

```
genes <- genes[!duplicated(genes$ENTREZID),]
```

在此例子中，注释与数据对象中的基因顺序是相同的。如果由于缺失和/或重新排列基因ID导致其顺序不一致，可以用 `match` 来正确排序基因。然后将基因注释的数据框加入数据对象，数据即被整洁地整理入一个DGEList对象中，它包含原始计数数据和相关的样品信息和基因注释。

```
x$genes <- genes
x
```

```
## An object of class "DGEList"
## $samples
##                              files group lib.size norm.factors l
ane
## 10_6_5_11 GSM1545535_10_6_5_11.txt    LP 32863052            1 L
004
## 9_6_5_11   GSM1545536_9_6_5_11.txt    ML 35335491            1 L
004
## purep53     GSM1545538_purep53.txt Basal 57160817            1 L
004
## JMS8-2       GSM1545539_JMS8-2.txt Basal 51368625            1 L
006
## JMS8-3       GSM1545540_JMS8-3.txt    ML 75795034            1 L
006
## JMS8-4       GSM1545541_JMS8-4.txt    LP 60517657            1 L
006
## JMS8-5       GSM1545542_JMS8-5.txt Basal 55086324            1 L
006
## JMS9-P7c   GSM1545544_JMS9-P7c.txt    ML 21311068            1 L
008
## JMS9-P8c   GSM1545545_JMS9-P8c.txt    LP 19958838            1 L
008
##
## $counts
##            Samples
## Tags        10_6_5_11 9_6_5_11 purep53 JMS8-2 JMS8-3 JMS8-4 JMS8
-5 JMS9-P7c JMS9-P8c
##   497097            1        2     342    526      3      3    5
35       2        0
##   100503874         0        0       5      6      0      0
5        0        0
##   100038431         0        0       0      0      0      0
1        0        0
##   19888             0        1       0      0     17      2
0        1        0
##   20671             1        1      76     40     33     14
98       18        8
## 27174 more rows ...
##
## $genes
##    ENTREZID  SYMBOL TXCHROM
## 1    497097    Xkr4    chr1
## 2 100503874 Gm19938    <NA>
## 3 100038431 Gm10568    <NA>
## 4     19888     Rp1    chr1
## 5     20671   Sox17    chr1
## 27174 more rows ...
```

5 数据预处理

5.1 原始数据尺度转换

由于更深的测序总会产生更多的序列，在差异表达相关的分析中，我们很少使用原始的序列数。在实践中，我们通常将原始的序列数进行归一化，来消除测序深度所导致的差异。通常被使用的方法有基于序列的CPM（counts per million）、log-CPM、FPKM（fragments per kilobase of transcript per million），和基于转录本数目的RPKM（reads per kilobase of transcript per million）。

尽管CPM和log-CPM转换并不像RPKM和FPKM那样考虑到基因长度区别的影响，但在我们的分析中经常会用到它们。虽然也可以使用RPKM和FPKM，但CPM和log-CPM只使用计数矩阵即可计算，且已足以满足我们所关注的比较的需要。假设不同条件之间剪接异构体（isoform）的使用没有差别，差异表达分析研究同一基因在不同条件下的表达差异，而不是比较多个基因之间的表达或测定绝对表达量。换而言之，基因长度在我们关注的比较中始终不变，且任何观测到的差异是来自于条件的变化而不是基因长度的变化。

在此处，使用**edgeR**中的 `cpm` 函数将原始计数转换为CPM和log-CPM值。如果可以提供基因长度信息，可以使用**edgeR**中的 `rpkm` 函数计算RPKM值，就像计算CPM值那样简单。

```
cpm <- cpm(x)
lcpm <- cpm(x, log=TRUE, prior.count=2)
```

对于一个基因，CPM值为1相当于在测序深度最低的样品中（JMS9-P8c, 序列数量约2千万）有20个计数，或者在测序深度最高的样品中（JMS8-3，序列数量约7.6千万）有76个计数。

log-CPM值将被用于探索性图表中。当设置 `log=TRUE` 时， `cpm` 函数会在进行log2转换前给CPM值加上一个弥补值。默认的弥补值是2/*L*，其中2是“预先计数”，而*L*是样本总序列数（以百万计）的平均值，所以log-CPM值是根据CPM值通过$\log_2(\text{CPM} + 2/L)$计算得到的。这样的计算方式可以确保任意两个具有相同CPM值的序列片段计数的log-CPM值也相同。弥补值的使用可以避免对零取对数，并能使所有样本间的log倍数变化（log-fold-change）向0推移而减小低表达基因间微小计数变化带来的巨大的伪差异性，这对于绘制探索性图表很有用。在这个数据集中，平均的样本总序列数是4.55千万，所以L约等于45.5，且每个样本中的最小log-CPM值为$\log_2(2/45.5) = -4.51$。换而言之，在加上了预先计数弥补值后，此数据集中的零表达计数对应的log-CPM值为-4.51：

```
L <- mean(x$samples$lib.size) * 1e-6
M <- median(x$samples$lib.size) * 1e-6
c(L, M)
```

```
## [1] 45.5 51.4
```

```
summary(lcpm)
```

```
##    10_6_5_11        9_6_5_11        purep53          JMS8-2
JMS8-3
##  Min.   :-4.51   Min.   :-4.51   Min.   :-4.51   Min.   :-4.51
Min.   :-4.51
##  1st Qu.:-4.51   1st Qu.:-4.51   1st Qu.:-4.51   1st Qu.:-4.51
1st Qu.:-4.51
##  Median :-0.68   Median :-0.36   Median :-0.10   Median :-0.09
Median :-0.43
##  Mean   : 0.17   Mean   : 0.33   Mean   : 0.44   Mean   : 0.41
Mean   : 0.32
##  3rd Qu.: 4.29   3rd Qu.: 4.56   3rd Qu.: 4.60   3rd Qu.: 4.55
3rd Qu.: 4.58
##  Max.   :14.76   Max.   :13.50   Max.   :12.96   Max.   :12.85
Max.   :12.96
##      JMS8-4          JMS8-5         JMS9-P7c        JMS9-P8c
##  Min.   :-4.51   Min.   :-4.51   Min.   :-4.51   Min.   :-4.51
##  1st Qu.:-4.51   1st Qu.:-4.51   1st Qu.:-4.51   1st Qu.:-4.51
##  Median :-0.41   Median :-0.07   Median :-0.17   Median :-0.33
##  Mean   : 0.25   Mean   : 0.40   Mean   : 0.37   Mean   : 0.27
##  3rd Qu.: 4.32   3rd Qu.: 4.43   3rd Qu.: 4.60   3rd Qu.: 4.44
##  Max.   :14.85   Max.   :13.19   Max.   :12.94   Max.   :14.01
```

在下游的线性模型分析中，使用**limma**的 voom 函数时也会用到log-CPM值，但 voom 会默认使用更小的预先计数重新计算自己的log-CPM值。

5.2 删除低表达基因

所有数据集中都混有表达的基因与不表达的基因。尽管我们想要检测在一种条件中表达但再另一种条件中不表达的基因，也有一些基因在所有样品中都不表达。实际上，这个数据集中19%的基因在所有九个样品中的计数都是零。

```
table(rowSums(x$counts==0)==9)
```

```
##
## FALSE  TRUE
## 22026  5153
```

对log-CPM值的分布绘制的图表显示每个样本中很大一部分基因都是不表达或者表达程度相当低的，它们的log-CPM值非常小甚至是负的（下图A部分）。

在任何样本中都没有足够多的序列片段的基因应该从下游分析中过滤掉。这样做的原因有好几个。 从生物学的角度来看，在任何条件下的表达水平都不具有生物学意义的基因都不值得关注，因此最好忽略。 从统计学的角度来看，去除低表达计数基因使数据中的均值 - 方差关系可以得到更精确的估计，并且还减少了在观察差异表达的下游分析中需要进行的统计检验的数量。

edgeR包中的 filterByExpr 函数提供了自动过滤基因的方法，可保留尽可能多的有足够表达计数的基因。

```
keep.exprs <- filterByExpr(x, group=group)
x <- x[keep.exprs,, keep.lib.sizes=FALSE]
dim(x)
```

```
## [1] 16624     9
```

此函数默认选取最小的组内的样本数量为最小样本数，保留至少在这个数量的样本中有10个或更多序列片段计数的基因。对基因表达量进行过滤时使用CPM值而不是表达计数，以避免对总序列数大的样本的偏向性。在这个数据集中，总序列数的中位数是5.1千万，且10/51约等于0.2，所以 `filterByExpr` 函数保留在至少三个样本中CPM值大于等于0.2的基因。此处，一个具有生物学意义的基因需要在至少三个样本中表达，因为三种细胞类型组内各有三个重复。所使用的阈值取决于测序深度和实验设计。如果样本总表达计数量增大，那么可以选择更低的CPM阈值，因为更大的总表达计数量提供了更好的分辨率来探究更多表达水平更低的基因。

使用这个标准，基因的数量减少到了16624个，约为开始时数量的60%。过滤后的log-CPM值显示出每个样本的分布基本相同（下图B部分）。需要注意的是，从整个DGEList对象中取子集时同时删除了被过滤的基因的计数和其相关的基因信息。过滤后的DGEList对象为留下的基因保留了相对应的基因信息和计数。

下方给出的是绘图所用代码。

```
lcpm.cutoff <- log2(10/M + 2/L)
library(RColorBrewer)
nsamples <- ncol(x)
col <- brewer.pal(nsamples, "Paired")
par(mfrow=c(1,2))
plot(density(lcpm[,1]), col=col[1], lwd=2, ylim=c(0,0.26), las=2, m
ain="", xlab="")
title(main="A. Raw data", xlab="Log-cpm")
abline(v=lcpm.cutoff, lty=3)
for (i in 2:nsamples){
den <- density(lcpm[,i])
lines(den$x, den$y, col=col[i], lwd=2)
}
legend("topright", samplenames, text.col=col, bty="n")
lcpm <- cpm(x, log=TRUE)
plot(density(lcpm[,1]), col=col[1], lwd=2, ylim=c(0,0.26), las=2, m
ain="", xlab="")
title(main="B. Filtered data", xlab="Log-cpm")
abline(v=lcpm.cutoff, lty=3)
for (i in 2:nsamples){
den <- density(lcpm[,i])
lines(den$x, den$y, col=col[i], lwd=2)
}
legend("topright", samplenames, text.col=col, bty="n")
```

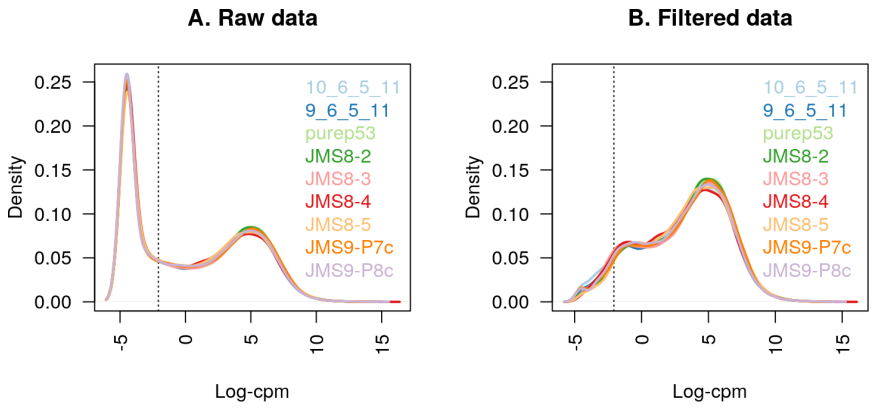


Figure 1: 每个样本过滤前的原始数据（**A**）和过滤后（**B**）的数据的**log-CPM**值密度。竖直虚线标出了过滤步骤中所用阈值（相当于**CPM**值为约**0**
2）。

5.3 归一化基因表达分布

在样品制备或测序过程中，不具备生物学意义的外部因素会影响单个样品的表达。比如说，在实验中第一批制备的样品会总体上表达高于第二批制备的样品。假设所有样品表达值的范围和分布都应当相似，需要进行归一化来确保整个实验中每个样本的表达分布都相似。

密度图和箱线图等展示每个样品基因表达量分布的图表可以用于判断是否有样品和其他样品分布有差异。在此数据集中，所有样品的log-CPM分布都很相似（上图B部分）。

尽管如此，我们依然需要使用 **edgeR** 中的 `calcNormFactors` 函数，用TMM(Robinson and Oshlack 2010)方法进行归一化。此处计算得到的归一化系数被用作文库大小的缩放系数。当我们使用DGEList对象时，这些归一化系数被自动存储在 `x$samples$norm.factors` 。对此数据集而言，TMM归一化的作用比较温和，这体现在所有的缩放因子都相对接近1。

```
x <- calcNormFactors(x, method = "TMM")
x$samples$norm.factors
```

```
## [1] 0.894 1.025 1.046 1.046 1.016 0.922 0.996 1.086 0.984
```

为了更好地可视化表现出归一化的影响，我们复制了数据并进行了调整，使得第一个样品的计数减少到了其原始值的5%，而第二个样品增大到了5倍。

```
x2 <- x
x2$samples$norm.factors <- 1
x2$counts[,1] <- ceiling(x2$counts[,1]*0.05)
x2$counts[,2] <- x2$counts[,2]*5
```

下图显示了没有经过归一化的与经过了归一化的数据的样本的表达分布，其中归一化前的分布显然不同，而归一化后比较相似。此处，第一个样品的TMM缩放系数0.06非常小，而第二个样品的缩放系数6.08很大，它们都并不接近1。

```
par(mfrow=c(1,2))
lcpm <- cpm(x2, log=TRUE)
boxplot(lcpm, las=2, col=col, main="")
title(main="A. Example: Unnormalised data",ylab="Log-cpm")
x2 <- calcNormFactors(x2)
x2$samples$norm.factors
```

```
## [1] 0.0577 6.0829 1.2202 1.1648 1.1966 1.0466 1.1505 1.2543 1.10
90
```

```
lcpm <- cpm(x2, log=TRUE)
boxplot(lcpm, las=2, col=col, main="")
title(main="B. Example: Normalised data",ylab="Log-cpm")
```

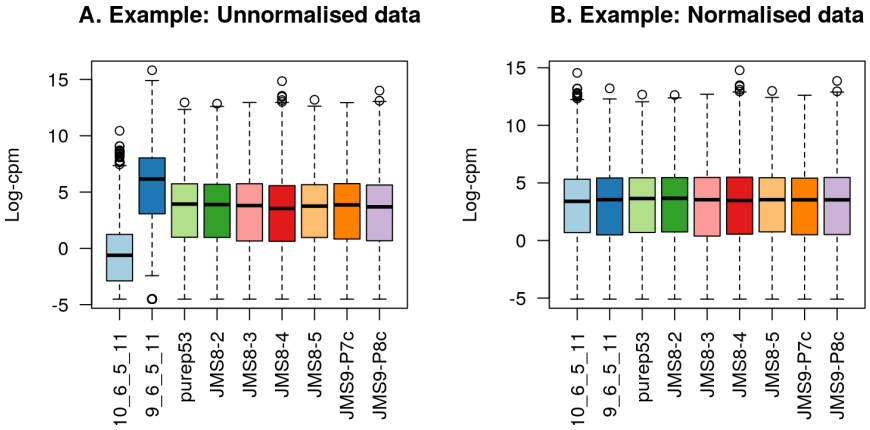


Figure 2: 样例数据：**log-CPM**值的箱线图展示了未经归一化的数据（**A**）及归一化后的数据（**B**）中每个样本的表达分布。数据集经过调整，样本**1**和**2**中的表达计数分别被缩放到其原始值的**5%**和**500%**。

5.4 对样本的无监督聚类

在我们看来，用于检查基因表达分析的最重要的探索性图表之一便是MDS图或其余类似的图。这种图表使用无监督聚类方法展示出了样品间的相似性和不相似性，能让我们在进行正式的检验之前对于能检测到多少差异表达基因有个大致概念。理想情况下，样本会在不同的实验组内很好的聚类，且可以鉴别出远离所属组的样本，并追踪误差或额外方差的来源。如果存在技术重复，它们应当互相非常接近。

这样的图可以用**limma**中的 `plotMDS` 函数绘制。第一个维度表示能够最好地分离样品且解释最大比例的方差的引导性的倍数变化（leading-fold-change），而后续的维度的影响更小，并与之前的维度正交。当实验设计涉及到多个因子时，建议在多个维度上检查每个因子。如果在其中一些维度上样本可按照某因子聚类，这说明该因子对于表达差异有影响，在线性模型中应当将其包括进去。反之，没有或者仅有微小影响的因子在下游分析时应当被剔除。

在这个数据集中，可以看出样本在维度1和2能很好地按照实验分组聚类，随后在维度3按照测序道（样品批次）分离（如下图所示）。请记住，第一维度解释了数据中最大比例的方差，需要注意到，当我们向高维度移动，维度上的取值范围会变小。

尽管所有样本都按组聚类，在维度1上最大的转录差异出现在basal和LP以及basal和ML之间。因此，预期在basal样品与其他之间的成对比较中能够得到大量的DE基因，而在ML和LP之间的比较中得到的DE基因数量略少。在其他的数据集中，不按照实验组聚类的样本可能在下游分析中只表现出较小的或不表现出差异表达。

为绘制MDS图，我们为不同的因子赋予不同的色彩组合。维度1和2使用以细胞类型定义的色彩组合进行检查。

维度3和4使用以测序泳道（批次）定义的色彩组合进行检查。

```
lcpm <- cpm(x, log=TRUE)
par(mfrow=c(1,2))
col.group <- group
levels(col.group) <-  brewer.pal(nlevels(col.group), "Set1")
col.group <- as.character(col.group)
col.lane <- lane
levels(col.lane) <-  brewer.pal(nlevels(col.lane), "Set2")
col.lane <- as.character(col.lane)
plotMDS(lcpm, labels=group, col=col.group)
title(main="A. Sample groups")
plotMDS(lcpm, labels=lane, col=col.lane, dim=c(3,4))
title(main="B. Sequencing lanes")
```

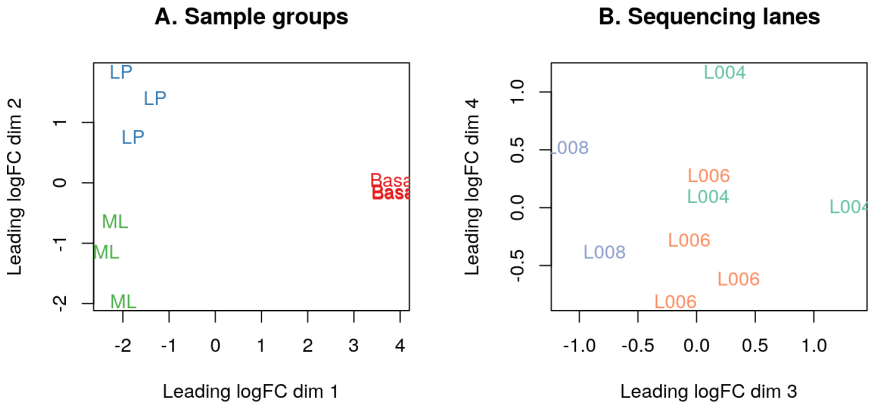


Figure 3: **log-CPM**值在维度**1**和**2**的**MDS**图，以样品分组上色并标记（**A**）和维度**3**和**4**的**MDS**图，以测序道上色并标记（**B**）。图中的距离对应于最主要的倍数变化（**fold change**），默认情况下也就是前**500**个在每对样品之间差异最大的基因的平均（均方根）**log2**倍数变化。

作为另一种选择，**Glimma**包也提供了便于探索多个维度的交互式MDS图。其中的 `glMDSPlot` 函数可生成一个html网页（如果设置 `launch=TRUE` ，将会在浏览器中打开），其左侧面板含有一张MDS图，而右侧面板包含一张展示了各个维度所解释的方差比例的柱形图。点击柱形图中的柱可切换MDS图绘制时所使用的维度，且将鼠标悬浮于单个点上可显示相应的样本标签。也可切换配色方案，以突显不同细胞类型或测序泳道（批次）。此数据集的交互式MDS图可以从 http://bioinf.wehi.edu.au/folders/limmaWorkflow/glimma-plots/MDS-Plot.html (http://bioinf.wehi.edu.au/folders/limmaWorkflow/glimma-plots/MDS-Plot.html) 看到。

```
glMDSPlot(lcpm, labels=paste(group, lane, sep="_"),
          groups=x$samples[,c(2,5)], launch=FALSE)
```

交互式MDS图链接 (glimma-plots/MDS-Plot.html)

6 差异表达分析

6.1 创建设计矩阵和对比

在此研究中，我们想知道哪些基因在我们研究的三组细胞之间以不同水平表达。在我们的分析中，假设基础数据是正态分布的，为其拟合一个线性模型。在此之前，需要创建一个包含细胞类型以及测序泳道（批次）信息的设计矩阵。

```
design <- model.matrix(~0+group+lane)
colnames(design) <- gsub("group", "", colnames(design))
design
```

```
##   Basal LP ML laneL006 laneL008
## 1     0  1  0        0        0
## 2     0  0  1        0        0
## 3     1  0  0        0        0
## 4     1  0  0        1        0
## 5     0  0  1        1        0
## 6     0  1  0        1        0
## 7     1  0  0        1        0
## 8     0  0  1        0        1
## 9     0  1  0        0        1
## attr(,"assign")
## [1] 1 1 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
## attr(,"contrasts")$lane
## [1] "contr.treatment"
```

对于一个给定的实验，通常有几种等价的方法可以创建一个合适的设计矩阵。 比如说， `~0+group+lane` 去除了第一个因子 `group` 的截距，但第二个因子 `lane` 的截距被保留。 此外也可以使用 `~group+lane` ，来自 `group` 和 `lane` 的截距均被保留。 此处的关键是理解如何解释给定模型中估计得到的系数。 我们在此分析中选取第一种模型，因为在没有 `group` 的截距的情况下能更直截了当地设定模型中的对比。用于细胞群之间成对比较的对比可以在**limma**中用 `makeContrasts` 函数设定。

```
contr.matrix <- makeContrasts(
   BasalvsLP = Basal-LP,
   BasalvsML = Basal - ML,
   LPvsML = LP - ML,
   levels = colnames(design))
contr.matrix
```

```
##           Contrasts
## Levels     BasalvsLP BasalvsML LPvsML
##   Basal            1         1      0
##   LP              -1         0      1
##   ML               0        -1     -1
##   laneL006         0         0      0
##   laneL008         0         0      0
```

limma的线性模型方法的核心优势之一便是其适应任意实验复杂程度的能力。简单的设计，比如此工作流程中关于细胞类型和批次的实验设计，直到更复杂的因子设计和含有交互作用项的模型，都能够被相对简单地处理。当实验或技术效应可被随机效应模型（ random effect model ）模拟时，**limma** 中的另一种可能性是使用 `duplicateCorrelation` 函数来估计交互作用，这需要在此函数以及 `lmFit` 的线性建模步骤均指定一个 `block` 参数。

6.2 从表达计数数据中删除异方差

据显示对于RNA-seq计数数据而言，当使用原始计数或当其被转换为log-CPM值时，方差并不独立于均值(Law et al. 2014)。使用负二项分布来模拟计数的方法假设均值与方差间具有二次的关系。在**limma**中，假设log-CPM值符合正态分布，并使用由 voom 函数计算得到的精确权重来调整均值与方差的关系，从而对log-CPM值进行线性建模。

当操作DGEList对象时， voom 从 x 中自动提取文库大小和归一化因子，以此将原始计数转换为log-CPM值。在 voom 中，对于log-CPM值额外的归一化可以通过设定 normalize.method 参数来进行。

下图左侧展示了这个数据集log-CPM值的均值-方差关系。通常而言，方差是测序实验中的技术差异和不同细胞类型的重复样本之间的生物学差异的结合，而*voom图会*显示出一个在均值与方差之间递减的趋势。 生物学差异高的实验通常会有更平坦的趋势，其方差值在高表达处稳定。 生物学差异低的实验更倾向于急剧下降的趋势。

不仅如此，voom图也提供了对于上游所进行的过滤水平的可视化检测。如果对于低表达基因的过滤不够充分，在图上表达低的一端，受到非常低的表达计数的影响，可以观察到方差水平的下降。如果观察到了这种情况，应当回到最初的过滤步骤并提高用于该数据集的表达阈值。

当前面观察的MDS图中具有明显的样本水平的差异时，可以用 voomWithQualityWeights 函数来同时合并样本水平的权重和 voom (Liu et al. 2015)估算得到的丰度相关的权重。关于此种方式的例子参见Liu等(2016) (Liu et al. 2016)。

```
par(mfrow=c(1,2))
v <- voom(x, design, plot=TRUE)
v
```

```
## An object of class "EList"
## $genes
##   ENTREZID SYMBOL TXCHROM
## 1   497097   Xkr4    chr1
## 5    20671  Sox17    chr1
## 6    27395 Mrpl15    chr1
## 7    18777 Lypla1    chr1
## 9    21399  Tcea1    chr1
## 16619 more rows ...
##
## $targets
##                              files group lib.size norm.factors l
ane
## 10_6_5_11 GSM1545535_10_6_5_11.txt    LP 29387429        0.894 L
004
## 9_6_5_11   GSM1545536_9_6_5_11.txt    ML 36212498        1.025 L
004
## purep53     GSM1545538_purep53.txt Basal 59771061        1.046 L
004
## JMS8-2       GSM1545539_JMS8-2.txt Basal 53711278        1.046 L
006
## JMS8-3       GSM1545540_JMS8-3.txt    ML 77015912        1.016 L
006
## JMS8-4       GSM1545541_JMS8-4.txt    LP 55769890        0.922 L
006
## JMS8-5       GSM1545542_JMS8-5.txt Basal 54863512        0.996 L
006
## JMS9-P7c   GSM1545544_JMS9-P7c.txt    ML 23139691        1.086 L
008
## JMS9-P8c   GSM1545545_JMS9-P8c.txt    LP 19634459        0.984 L
008
##
## $E
##         Samples
## Tags     10_6_5_11 9_6_5_11 purep53 JMS8-2 JMS8-3 JMS8-4 JMS8-5
JMS9-P7c JMS9-P8c
##   497097     -4.29    -3.86   2.519  3.293  -4.46  -3.99  3.287
-3.210    -5.30
##   20671      -4.29    -4.59   0.356 -0.407  -1.20  -1.94  0.844
-0.323    -1.21
##   27395       3.88     4.41   4.517  4.562   4.34   3.79  3.899
4.340     4.12
##   18777       4.71     5.57   5.396  5.162   5.65   5.08  5.060
5.751     5.14
##   21399       4.79     4.75   5.370  5.122   4.87   4.94  5.138
5.031     4.98
## 16619 more rows ...
##
## $weights
##       [,1]  [,2] [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
## [1,]  1.08  1.33 19.8 20.27  1.99  1.40 20.49  1.11  1.08
## [2,]  1.17  1.46  4.8  8.66  3.61  2.63  8.76  3.21  2.54
## [3,] 20.22 25.57 30.4 28.53 31.35 25.74 28.72 21.20 16.66
## [4,] 26.95 32.51 33.6 33.23 34.23 32.35 33.33 30.35 24.26
## [5,] 26.61 28.50 33.6 33.21 33.57 32.00 33.31 25.17 23.57
## 16619 more rows ...
##
```

```
## $design
##   Basal LP ML laneL006 laneL008
## 1     0  1  0        0        0
## 2     0  0  1        0        0
## 3     1  0  0        0        0
## 4     1  0  0        1        0
## 5     0  0  1        1        0
## 6     0  1  0        1        0
## 7     1  0  0        1        0
## 8     0  0  1        0        1
## 9     0  1  0        0        1
## attr(,"assign")
## [1] 1 1 1 2 2
## attr(,"contrasts")
## attr(,"contrasts")$group
## [1] "contr.treatment"
##
## attr(,"contrasts")$lane
## [1] "contr.treatment"
```

```
vfit <- lmFit(v, design)
vfit <- contrasts.fit(vfit, contrasts=contr.matrix)
efit <- eBayes(vfit)
plotSA(efit, main="Final model: Mean-variance trend")
```

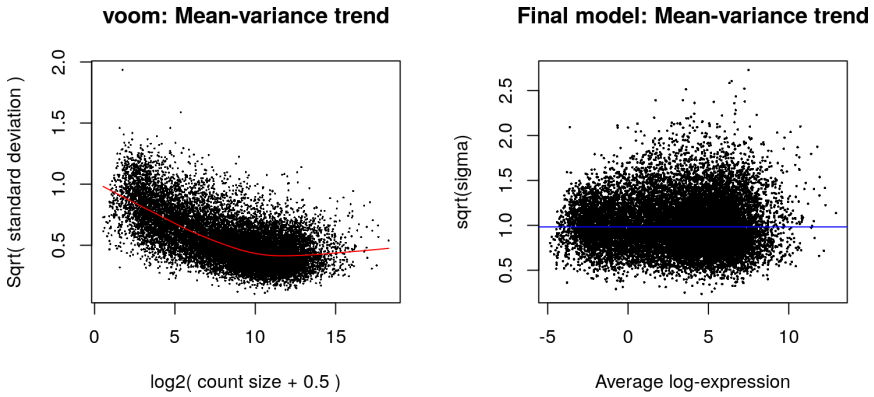


Figure 4: 图中绘制了每个基因的均值（**x**轴）和方差（**y**轴），显示了在该数据上使用 voom 前它们之间的相关性（左），以及当运用 voom 的精确权重后这种趋势是如何消除的（右）。左侧的图是使用 voom 函数绘制的，它为进行**log-CPM**转换后的数据拟合线性模型从而提取残差方差。然后，对方差取平方根（或对标准差取平方根），并相对每个基因的平均表达作图。均值通过平均计数加上**2**再进行**log2**转换计算得到。右侧的图使用 plotSA 绘制了**log2**残差标准差与**log-CPM**均值的关系。平均**log2**残差标准差由水平蓝线标出。在这两幅图中，每个黑点表示一个基因，红线为对这些点的拟合。

值得注意的是，DGEList对象中存储的另一个数据框，即基因和样本水平信息所存储之处，保留在了 voom 创建的 EList 对象 v 中。 v$genes 数据框等同于 x$genes ， v$targets 等同于 x$samples ，而 v$E 中所储存的表达值类似于 x$counts ，尽管它进行了尺度转换。此外， voom 的EList对象中还有一个精确权重的矩阵 v$weights ，而设计矩阵存储于 v$design 。

6.3 拟合线性模型以进行比较

limma的线性建模使用 lmFit 和 contrasts.fit 函数进行，它们原先是为微阵列而设计的。这些函数不仅可以用于微阵列数据，也可以用于RNA-seq数据。它们会单独为每个基因的表达值拟合一个模型。然后，通过利用所有基因的信息来进行经验贝叶斯

调整，这样可以获得更精确的基因水平的变异程度估计(Smyth 2004)。下一图为此模型的残差关于平均表达值的图。从图中可以看出，方差不再与表达水平均值相关。

6.4 检查DE基因数量

为快速查看差异表达水平，显著上调或下调的基因可以汇总到一个表格中。显著性的判断使用校正*p*值阈值的默认值5%。在basal与LP的表达水平之间的比较中，发现了4648个在basal中相较于LP下调的基因和4863个在basal中相较于LP上调的基因，即共9511个DE基因。在basal和ML之间发现了一共9598个DE基因（4927个下调基因和4671个上调基因），而在LP和ML中发现了一共5652个DE基因（3135个下调基因和2517个上调基因）。在包括basal细胞类型的比较中皆找到了大量的DE基因，这与我们在MDS图中观察到的结果相吻合。

```
summary(decideTests(efit))
```

```
##        BasalvsLP BasalvsML LPvsML
## Down        4648      4927   3135
## NotSig      7113      7026  10972
## Up          4863      4671   2517
```

一些研究中不仅仅需要使用校正*p*值阈值，更为严格定义的显著性可能需要差异倍数的对数（log-FCs）也高于某个最小值。*treat*方法(McCarthy and Smyth 2009)可以按照对最小log-FC值的要求，使用经过经验贝叶斯调整的*t*统计值计算*p*值。当我们的检验要求基因的log-FC显著大于1（等同于在原本的尺度上不同细胞类型之间差两倍）时，差异表达基因的数量得到了下降，basal与LP相比只有3684个DE基因，basal与ML相比只有3834个DE基因，LP与ML相比只有414个DE基因。

```
tfit <- treat(vfit, lfc=1)
dt <- decideTests(tfit)
summary(dt)
```

```
##        BasalvsLP BasalvsML LPvsML
## Down        1632      1777    224
## NotSig     12976     12790  16210
## Up          2016      2057    190
```

在多种比较中皆差异表达的基因可以从 `decideTests` 的结果中提取，其中的0代表不差异表达的基因，1代表上调的基因，-1代表下调的基因。共有2784个基因在basal和LP以及basal和ML的比较中都差异表达，其中的20个于下方列出。 `write.fit` 函数可用于将三个比较的结果提取并写入到单个输出文件。

```
de.common <- which(dt[,1]!=0 & dt[,2]!=0)
length(de.common)
```

```
## [1] 2784
```

```
head(tfit$genes$SYMBOL[de.common], n=20)
```

```
##  [1] "Xkr4"          "Rgs20"         "Cpa6"          "A830018L16
Rik" "Sulf1"
##  [6] "Eya1"          "Msc"           "Sbspon"        "Pi15"
"Crispld1"
## [11] "Kcnq5"         "Rims1"         "Khdrbs2"       "Ptpn18"
"Prss39"
## [16] "Arhgef4"       "Cnga3"         "2010300C02Rik" "Aff3"
"Npas2"
```

```
vennDiagram(dt[,1:2], circle.col=c("turquoise", "salmon"))
```

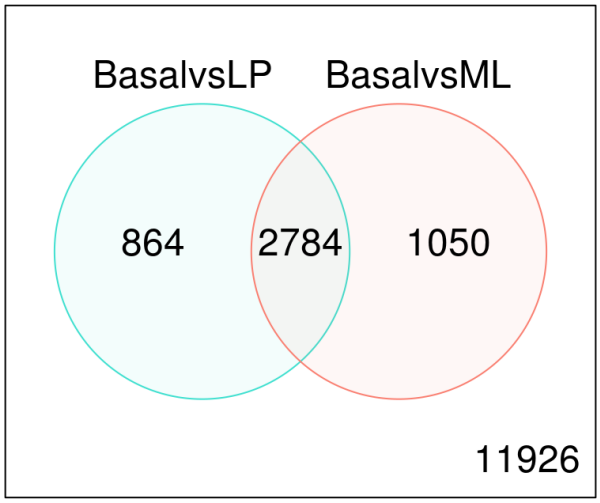


Figure 5: 韦恩图展示了仅**basal**和**LP**（左）、仅**basal**和**ML**（右）的对比的**DE**基因数量，还有两种对比中共同的**DE**基因数量（中）。在任何对比中均不差异表达的基因数量标于右下。

```
write.fit(tfit, dt, file="results.txt")
```

6.5 从上到下检查单个DE基因

使用 `topTreat` 函数可以列举出使用 `treat` 得到的结果中靠前的DE基因（对于 `eBayes` 的结果可以使用 `topTable` 函数）。默认情况下， `topTreat` 将基因按照校正*p*值从小到大排列，并为每个基因给出相关的基因信息、log-FC、平均log-CPM、校正*t*值、原始及经过多重假设检验校正的*p*值。列出前多少个基因的数量可由用户指定，如果设为 `n=Inf` 则会包括所有的基因。基因*Cldn7*和*Rasef*在basal与LP和basal于ML的比较中都位于DE基因的前几名。

```
basal.vs.lp <- topTreat(tfit, coef=1, n=Inf)
basal.vs.ml <- topTreat(tfit, coef=2, n=Inf)
head(basal.vs.lp)
```

```
##        ENTREZID SYMBOL TXCHROM logFC AveExpr     t  P.Value adj.
P.Val
## 12759     12759    Clu   chr14 -5.46    8.86 -33.6 1.72e-10  1.7
1e-06
## 53624     53624  Cldn7   chr11 -5.53    6.30 -32.0 2.58e-10  1.7
1e-06
## 242505   242505  Rasef    chr4 -5.94    5.12 -31.3 3.08e-10  1.7
1e-06
## 67451     67451   Pkp2   chr16 -5.74    4.42 -29.9 4.58e-10  1.7
4e-06
## 228543   228543   Rhov    chr2 -6.26    5.49 -29.1 5.78e-10  1.7
4e-06
## 70350     70350  Basp1   chr15 -6.08    5.25 -28.3 7.27e-10  1.7
4e-06
```

```
head(basal.vs.ml)
```

```
##        ENTREZID  SYMBOL TXCHROM logFC AveExpr     t  P.Value ad
j.P.Val
## 242505   242505   Rasef    chr4 -6.53    5.12 -35.1 1.23e-10  1.
24e-06
## 53624     53624   Cldn7   chr11 -5.50    6.30 -31.7 2.77e-10  1.
24e-06
## 12521     12521    Cd82    chr2 -4.69    7.07 -31.4 2.91e-10  1.
24e-06
## 20661     20661   Sort1    chr3 -4.93    6.70 -30.7 3.56e-10  1.
24e-06
## 71740     71740 Nectin4    chr1 -5.58    5.16 -30.6 3.72e-10  1.
24e-06
## 12759     12759     Clu   chr14 -4.69    8.86 -28.0 7.69e-10  1.
48e-06
```

6.6 差异表达结果的实用图形表示

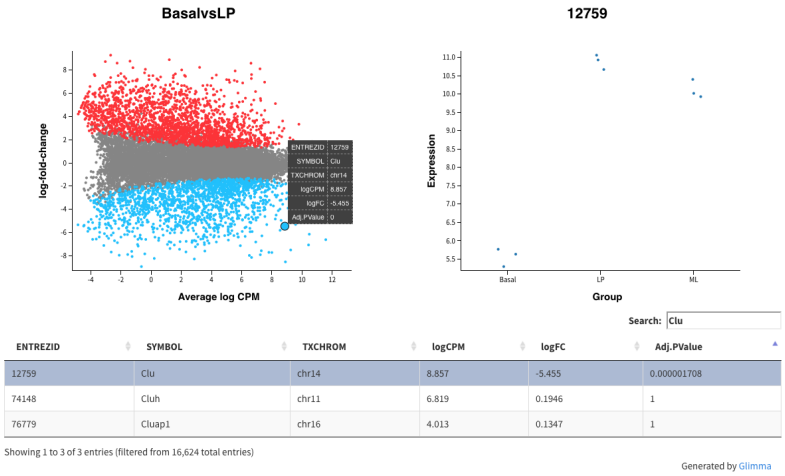
为可视化地总结所有基因的结果，可使用 `plotMD` 函数绘制均值-差异（MD）图，其中展示了线性模型拟合所得到的log-FC与log-CPM平均值间的关系，而差异表达的基因会被重点标出。

```
plotMD(tfit, column=1, status=dt[,1], main=colnames(tfit)[1],
       xlim=c(-8,13))
```

Glimma的 `glMDPlot` 函数提供了交互式的均值-差异图，拓展了这种图表的功能性。此函数的输出为一个html页面，左侧面板为结果的总结性图表（与 `plotMD` 的输出类似），右侧面板包含各个样本的log-CPM值，下方为结果的表格。 这种交互式展示允许用户使用提供的注释（比如基因名标识）搜索特定基因，而这在**R**统计图中是做不到的。

```
glMDPlot(tfit, coef=1, status=dt, main=colnames(tfit)[1],
         side.main="ENTREZID", counts=lcpm, groups=group, launch=FA
LSE)
```

交互式MD图链接 (glimma-plots/MD-Plot.html)



使用**Glimma**生成的均值-差异图。左侧面板显示了总结性数据（log-FC与log-CPM值的关系），其中选中的基因在每个样本中的数值显示于右侧面板。下方为结果的表格，其搜索框使用户得以使用可行的注释信息查找某个特定基因，如基因符号*Clu*。

上方指令生成的均值-差异图可以在线访问（详见http://bioinf.wehi.edu.au/folders/limmaWorkflow/glimma-plots/MD-Plot.html (http://bioinf.wehi.edu.au/folders/limmaWorkflow/glimma-plots/MD-Plot.html)）。**Glimma**提供的交互性使得单个图形窗口内能够呈现出额外的信息。**Glimma**是以**R**和Javascript实现的，使用**R**代码生成数据，并在之后使用Javascript库D3（https://d3js.org (https://d3js.org)）转换为图形，使用Bootstrap库处理界面并生成互动性可搜索的表格的数据表。这使得图表可以在任何现代的浏览器中查看，对于从Rmarkdown分析报告中将其作为关联文件而附加而言十分方便。

前文所展示的图表中，一些展示了在任意一个条件下表达的所有基因（比如共同DE基因的韦恩图或均值-差异图），而另一些展示单独的基因（交互性均值-差异图右边面板中所展示的log-CPM值）。而热图使用户得以查看一部分基因的表达。这对于查看单个组或样本的表达很有用，而不至于在关注于单个基因时失去对于研究整体的注意，也不会造成由于上千个基因所取平均值而导致的失去分辨率。

使用**gplots**包的`heatmap.2`函数，我们为basal与LP的对照中前100个DE基因（按调整p值排序）绘制了一幅热图。热图中正确地将样本按照细胞类型聚类，并重新排序了基因，形成了表达相似的块状。从热图中，我们观察到对于basal与LP之间的前100个DE基因，ML和LP样本的表达非常相似。

```
library(gplots)
basal.vs.lp.topgenes <- basal.vs.lp$ENTREZID[1:100]
i <- which(v$genes$ENTREZID %in% basal.vs.lp.topgenes)
mycol <- colorpanel(1000,"blue","white","red")
heatmap.2(lcpm[i,], scale="row",
   labRow=v$genes$SYMBOL[i], labCol=group,
   col=mycol, trace="none", density.info="none",
   margin=c(8,6), lhei=c(2,10), dendrogram="column")
```

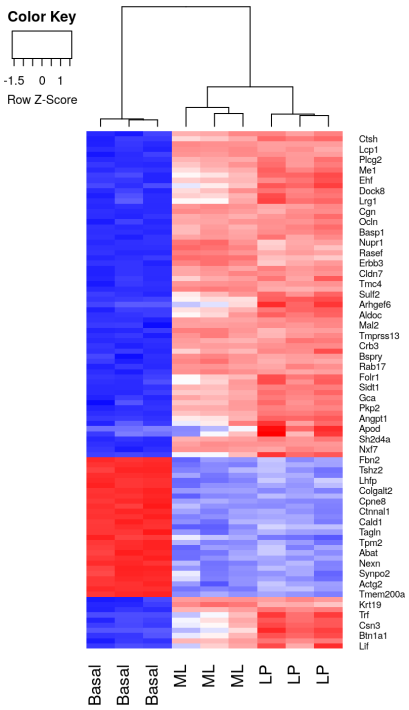


Figure 6: 在**basal**和**LP**的对比中前**100**个**DE**基因**log-CPM**值的热图。经过缩放调整后，每个基因（每行）的表达均值为**0**，并且标准差为**1**。给定基因相对高表达的样本被标记为红色，相对低表达的样本被标记为蓝色。浅色和白色代表中等表达水平的基因。样本和基因已通过分层聚类的方法重新排序。图中显示有样本聚类的树状图。

7 使用camera的基因集检验

在此次分析的最后，我们要进行一些基因集检验。为此，我们将*camera*方法(Wu and Smyth 2012)应用于Broad Institute的MSigDB c2中的(Subramanian et al. 2005)中适应小鼠的*c2*基因表达特征，这可从http://bioinf.wehi.edu.au/software/MSigDB/(http://bioinf.wehi.edu.au/software/MSigDB/)以RData对象格式获取。此外，对于人类和小鼠，来自MSigDB的其他有用的基因集也可从此网站获取，比如标志（hallmark）基因集。C2基因集的内容收集自在线数据库、出版物以及该领域专家，而标志基因集的内容来自MSigDB，从而获得具有明确定义的生物状态或过程。

```
load(system.file("extdata", "mouse_c2_v5p1.rda", package = "RNAseq1
23"))
idx <- ids2indices(Mm.c2,id=rownames(v))
cam.BasalvsLP <- camera(v,idx,design,contrast=contr.matrix[,1])
head(cam.BasalvsLP,5)
```

```
##                                                NGenes Direction    P
Value      FDR
## LIM_MAMMARY_STEM_CELL_UP                         791        Up 1.7
7e-18 8.36e-15
## LIM_MAMMARY_STEM_CELL_DN                         683      Down 4.0
3e-14 8.69e-11
## ROSTY_CERVICAL_CANCER_PROLIFERATION_CLUSTER      170        Up 5.5
2e-14 8.69e-11
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP                 94      Down 2.7
4e-13 3.23e-10
## SOTIRIOU_BREAST_CANCER_GRADE_1_VS_3_UP           190        Up 5.1
6e-13 4.87e-10
```

```
cam.BasalvsML <- camera(v,idx,design,contrast=contr.matrix[,2])
head(cam.BasalvsML,5)
```

```
##                                      NGenes Direction   PValue
FDR
## LIM_MAMMARY_STEM_CELL_UP                791        Up 1.68e-22 7.
92e-19
## LIM_MAMMARY_STEM_CELL_DN                683      Down 7.79e-18 1.
84e-14
## LIM_MAMMARY_LUMINAL_MATURE_DN           172        Up 9.74e-16 1.
53e-12
## LIM_MAMMARY_LUMINAL_MATURE_UP           204      Down 1.15e-12 1.
36e-09
## NAKAYAMA_SOFT_TISSUE_TUMORS_PCA2_UP     137        Up 2.24e-12 1.
88e-09
```

```
cam.LPvsML <- camera(v,idx,design,contrast=contr.matrix[,3])
head(cam.LPvsML,5)
```

```
##                                       NGenes Direction   PValu
e      FDR
## LIM_MAMMARY_LUMINAL_MATURE_DN            172        Up 6.73e-1
4 2.35e-10
## LIM_MAMMARY_LUMINAL_MATURE_UP            204      Down 9.97e-1
4 2.35e-10
## LIM_MAMMARY_LUMINAL_PROGENITOR_UP         94        Up 1.32e-1
1 2.08e-08
## REACTOME_RESPIRATORY_ELECTRON_TRANSPORT   94      Down 7.01e-0
9 8.28e-06
## REACTOME_RNA_POL_I_PROMOTER_OPENING       46      Down 2.04e-0
8 1.93e-05
```

camera 函数通过比较假设检验来评估一个给定基因集中的基因是否相对于不在集内的基因而言在差异表达基因的排序中更靠前。 它使用**limma**的线性模型框架，并同时采用设计矩阵和对比矩阵（如果有的话），且在测试的过程中会使用来自*voom*的观测水平权重。 在通过基因间相关性（默认设定为0.01，但也可通过数据估计）和基因集的规模得到方差膨胀因子（variance inflation factor），并使用它调整基因集检验统计值的方差后，将会返回根据多重假设检验进行了校正的*p*值。

此实验是与Lim等人(2010)(Lim et al. 2010)的数据集等价的RNA-seq，而他们使用Illumina微阵列分析了相同的分选细胞群，因此该早期文献中的基因表达特征出现在每种对比的列表顶部正符合我们的预期。在LP和ML的对比中，我们为Lim等人（2010）的成熟管腔基因集（上调及下调）绘制了条码图（barcodeplot）。需要注意的是，由于我们的对比是将LP与ML相比而不是相反，这些基因集的方向在我们的数据集中是反过来的（如果将对比反过来，基因集的方向将会与对比一致）。

```
barcodeplot(efit$t[,3], index=idx$LIM_MAMMARY_LUMINAL_MATURE_UP,
            index2=idx$LIM_MAMMARY_LUMINAL_MATURE_DN, main="LPvsML"
)
```

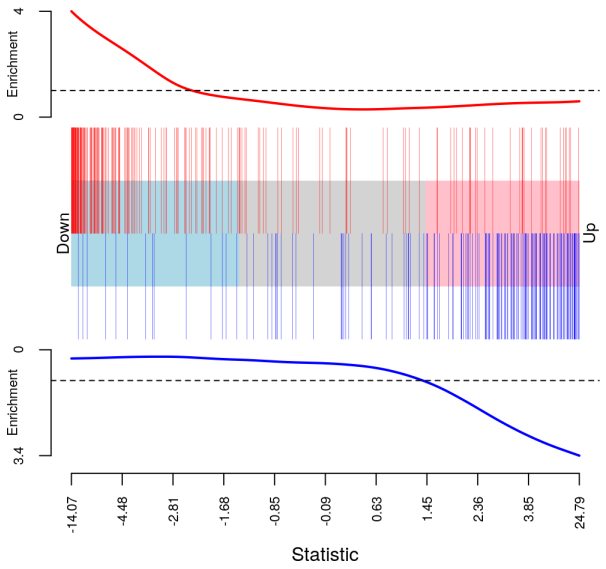


Figure 7: `LIM_MAMMARY_LUMINAL_MATURE_UP`（红色条形，图表上方）和 `LIM_MAMMARY_LUMINAL_MATURE_DN`（蓝色条形，图表下方）基因集在**LP**和**ML**的对比中的条码图，每个基因集都有一条富集线展示了竖直条形在图表每部分的相对富集程度。**Lim等人的实验(Lim et al 2010)**非常类似于我们的，用了相同的分选方式来获取不同的细胞群，只是他们使用的是微阵列而不是RNA-seq来测定基因表达。需要注意的是，上调基因集发生下调而下调基因集发生上调的逆相关性来自于对比的设定方式（LP相比于ML），如果将其对调，方向性将会吻合。

limma也有其他的基因集检验，比如*mroast*(Wu et al. 2010)的自包含检验。虽然*camera*非常适合检验基因集的大型数据库并观察其中哪些相对于其他的在排序上位次更高（如前文所示），自包含检验更善于集中检验一个或少个选中的集合是否本身差异表达。换句话说，*camera*更适用于搜寻具有意义的基因集，而*mroast*测试的是已经确定有意义的基因集的显著性。

8 使用到的软件和代码

此RNA-seq工作流程使用了Bioconductor项目3.8版本中的多个软件包，运行于*R* 3.5.1或更高版本。除了本文中重点提到的软件（**limma**、**Glimma**以及**edgeR**），亦需要一些其他软件包，包括**gplots**和**RColorBrewer**还有基因注释包**Mus.musculus**。 此文档使用**knitr**编译。所有用到的包的版本号如下所示。 Bioconductor工作流程包**RNAseq123**（可访问 https://bioconductor.org/packages/RNAseq123 (https://bioconductor.org/packages/RNAseq123)查看）内包含此文章的英文和简体中文版以及进行整个分析流程所需要的代码。安装此包即可管理以上提到的所有需要的包。对于RNA-seq数据分析实践培训而言，此包也是非常有用的资源。

```
sessionInfo()
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.2 LTS
##
## Matrix products: default
## BLAS:   /home/biocbuild/bbs-3.10-bioc/R/lib/libRblas.so
## LAPACK: /home/biocbuild/bbs-3.10-bioc/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C               LC_TI
ME=en_US.UTF-8
##  [4] LC_COLLATE=C               LC_MONETARY=en_US.UTF-8    LC_ME
SSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C                  LC_AD
DRESS=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8 LC_ID
ENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils
datasets  methods
## [9] base
##
## other attached packages:
##  [1] knitr_1.22                                gplots_3.0.1.1
##  [3] RColorBrewer_1.1-2                        Mus.musculus_1.3.1
##  [5] TxDb.Mmusculus.UCSC.mm10.knownGene_3.4.7 org.Mm.eg.db_3.8.2
##  [7] GO.db_3.8.2                               OrganismDbi_1.27.0
##  [9] GenomicFeatures_1.37.0                    GenomicRanges_1.3
7.0
## [11] GenomeInfoDb_1.21.0                      AnnotationDbi_1.4
7.0
## [13] IRanges_2.19.0                           S4Vectors_0.23.0
## [15] Biobase_2.45.0                           BiocGenerics_0.31.
0
## [17] edgeR_3.27.0                             Glimma_1.13.0
## [19] limma_3.41.0                             BiocStyle_2.13.0
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.0                  bit64_0.9-7                jso
nlite_1.6
##  [4] R.utils_2.8.0               gtools_3.8.1               ass
ertthat_0.2.1
##  [7] BiocManager_1.30.4          highr_0.8                  RBG
L_1.61.0
## [10] blob_1.1.1                  GenomeInfoDbData_1.2.1     Rsa
mtools_2.1.0
## [13] yaml_2.2.0                  progress_1.2.0             RSQ
Lite_2.1.1
## [16] lattice_0.20-38             digest_0.6.18              XVe
ctor_0.25.0
## [19] htmltools_0.3.6             Matrix_1.2-17              R.o
o_1.22.0
## [22] XML_3.98-1.19               pkgconfig_2.0.2            bio
maRt_2.41.0
## [25] bookdown_0.9                zlibbioc_1.31.0            gda
ta_2.18.0
```

```
## [28] BiocParallel_1.19.0         SummarizedExperiment_1.15.0 mag
rittr_1.5
## [31] crayon_1.3.4                memoise_1.1.0               eva
luate_0.13
## [34] R.methodsS3_1.7.1           graph_1.63.0                too
ls_3.6.0
## [37] prettyunits_1.0.2           hms_0.4.2                   mat
rixStats_0.54.0
## [40] stringr_1.4.0               locfit_1.5-9.1              Del
ayedArray_0.11.0
## [43] Biostrings_2.53.0           compiler_3.6.0              caT
ools_1.17.1.2
## [46] rlang_0.3.4                 grid_3.6.0                  RCu
rl_1.95-4.12
## [49] bitops_1.0-6                rmarkdown_1.12              DBI
_1.0.0
## [52] R6_2.4.0                    GenomicAlignments_1.21.0    rtr
acklayer_1.45.0
## [55] bit_1.1-14                  KernSmooth_2.23-15          str
ingi_1.4.3
## [58] Rcpp_1.0.1                  xfun_0.6
```

参考文献

Bioconductor Core Team. 2016a. *Homo.sapiens: Annotation package for the Homo.sapiens object*. https://bioconductor.org/packages/release/data/annotation/html/Homo.sapiens.html (https://bioconductor.org/packages/release/data/annotation/html/Homo.sapiens.html).

———. 2016b. *Mus.musculus: Annotation package for the Mus.musculus object*. https://bioconductor.org/packages/release/data/annotation/html/Mus.musculus.html (https://bioconductor.org/packages/release/data/annotation/html/Mus.musculus.html).

Durinck, S., Y. Moreau, A. Kasprzyk, S. Davis, B. De Moor, A. Brazma, and W. Huber. 2005. “BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis.” *Bioinformatics* 21:3439–40.

Durinck, S., P. Spellman, E. Birney, and W. Huber. 2009. “Mapping identifiers for the integration of genomic datasets with the R/Bioconductor package biomaRt.” *Nature Protocols* 4:1184–91.

Huber, W., V. J. Carey, R. Gentleman, S. Anders, M. Carlson, B. S. Carvalho, H. C. Bravo, et al. 2015. “Orchestrating High-Throughput Genomic Analysis with Bioconductor.” *Nature Methods* 12 (2):115–21. http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html (http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html).

Law, C. W., Y. Chen, W. Shi, and G. K. Smyth. 2014. “Voom: Precision Weights Unlock Linear Model Analysis Tools for RNA-seq Read Counts.” *Genome Biology* 15:R29.

Liao, Y., G. K. Smyth, and W. Shi. 2013. “The Subread Aligner: Fast, Accurate and Scalable Read Mapping by Seed-and-Vote.” *Nucleic Acids Res* 41 (10):e108.

———. 2014. “featureCounts: an Efficient General-Purpose Program for Assigning Sequence Reads to Genomic Features.” *Bioinformatics* 30 (7):923–30.

Lim, E., D. Wu, B. Pal, T. Bouras, M. L. Asselin-Labat, F. Vaillant, H. Yagita, G. J. Lindeman, G. K. Smyth, and J. E. Visvader. 2010. “Transcriptome analyses of mouse and human mammary cell subpopulations reveal multiple conserved genes and pathways.” *Breast Cancer Research* 12 (2):R21.

Liu, R., K. Chen, N. Jansz, M. E. Blewitt, and M. E. Ritchie. 2016. “Transcriptional Profiling of the Epigenetic Regulator Smchd1.” *Genomics Data* 7:144–7.

Liu, R., A. Z. Holik, S. Su, N. Jansz, K. Chen, H. S. Leong, M. E. Blewitt, M. L. Asselin-Labat, G. K. Smyth, and M. E. Ritchie. 2015. “Why weight? Combining voom with estimates of sample quality improves power in RNA-seq analyses.” *Nucleic Acids Res* 43:e97.

McCarthy, D. J., and G. K. Smyth. 2009. “Testing significance relative to a fold-change threshold is a TREAT.” *Bioinformatics* 25:765–71.

Ritchie, M. E., B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth. 2015. “limma Powers Differential Expression Analyses for RNA-Sequencing and Microarray Studies.” *Nucleic Acids Res* 43 (7):e47.

Robinson, M. D., D. J. McCarthy, and G. K. Smyth. 2010. “edgeR: A Bioconductor Package for Differential Expression Analysis of Digital Gene Expression Data.” *Bioinformatics* 26:139–40.

Robinson, M. D., and A. Oshlack. 2010. “A Scaling Normalization Method for Differential Expression Analysis of RNA-seq data.” *Genome Biology* 11:R25.

Sheridan, J. M., M. E. Ritchie, S. A. Best, K. Jiang, T. J. Beck, F. Vaillant, K. Liu, et al. 2015. “A pooled shRNA screen for regulators of primary mammary stem and progenitor cells identifies roles for Asap1 and Prox1.” *BMC Cancer* 15 (1). BioMed Central:221.

Smyth, G. K. 2004. “Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments.” *Stat Appl Genet Mol Biol* 3 (1):Article 3.

Su, S., C. W. Law, C. Ah-Cann, M. L. Asselin-Labat, M. E. Blewitt, and M. E. Ritchie. 2017. “Glimma: Interactive Graphics for Gene Expression Analysis.” *Bioinformatics* 33:2050–52.

Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, et al. 2005. “Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles.” *Proc Natl Acad Sci U S A* 102 (43):15545–50.

Wu, D., E. Lim, F. Vaillant, M. L. Asselin-Labat, J. E. Visvader, and G. K. Smyth. 2010. “ROAST: rotation gene set tests for complex microarray experiments.” *Bioinformatics* 26 (17):2176–82.

Wu, D., and G. K. Smyth. 2012. “Camera: a competitive gene set test accounting for inter-gene correlation.” *Nucleic Acids Res* 40 (17):e133.