Overview of capabilities

edgeR可针对来源任何类型的counts数据，尤其是二代测序，检测差异表达，包含了两种分析模式，calssic和glm，都是采用empirical Bays methods评估gene-sepcific变异，即使是不存在重复的样本。**一般而言，glm模式比classic模式更流行，且glm更大灵活性，同时glm包含2种检测模式，likelihood ratio test和quasi-likelihood F-test，强烈推荐quasi-likelihood方式，在离散型评估过程中会给予更严格的错误控制；而likelihood ratio test在单细胞RNA-seq和没有重复样本检测中更有用。**

1. Aligning reads to a genome

推荐subread-featureCounts流程，快速有效；流行的流程还有STAR-featureCounts，STAR-htseq和Bowtie-TopHat-htseq流程。

同时findOverlaps (GenomicRanges)和htseq-counts(Python software)也可用于根据annotation和mapped reads计算counts table。

针对shRNA-seq和CRISPR-Cas9 genetic screens， 可使用processAmplicons直接从fastq文件获得counts table

2. Reading the counts from a file

读取且存储counts table及相关信息，DGEList()，可以想lsit一样处理，DGEList对象包含：

[Use with downstream Bioconductor DGE packages]
[https://github.com/mikelove/tximport/blob/master/vignettes/tximport.Rmd]

tximport建议二种导入估计用于差异基因表达分析方式：第一种，用于'edgeR'和'DESeq2'；从定量软件获得基因水平的计数，同时使用转录水平的分布估计来计算基因水平的offset，用于修正样本间的平均转录本长度的变化。对于'edgeR'，需要指定矩阵用于'y$offset'，而'DESeqDataSetFromTximport'可自动调用offset。第二种，使用'tximprot' 函数'countsFromAbundance="lengthScaledTPM"或"scaledTPM"，然后直接使用基因水平的计数矩阵'txi$counts'。

不建议将原始的基因水平的计数用于下游分析。除非，计数counts不存在长度偏差，例如3‘ tagged RNA-seq data，txi$counts可直接用于下游分析。

从tximport导入edgeR

```
cts <- txi$counts
```

```
normMat <- txi$length
```

根据观测得到的长度，调整避免改变了counts的数量级

```
normMat <- normMat/exp(rowMeans(log(normMat)))
```

```
normCts <- cts/normMat
```

根据sacled counts计算有效长度大小，得到样本间的组成偏差

```
eff.lib <- calcNormFactors(normCts) * colSums(normCts)
```

合并有效文库长度和长度因子，计算log-link GLM的offsets(sweep用于array, apply用于matrix)

`normMat <- sweep(normMat,2,eff.lib, "*")` #normMat * eff.lib

`normMat <- log(normMat)`

构建DGEList对象

`y <- DGEList(cts)`

`y <- scaleOffset(y, normMat)`

`scaleOffset` ensures that the scale of offsets are consistent with library sizes. This is done by ensuring that the mean offset for each gene is the same as the mean log-library size.

过滤低表达基因

`keep <- filterByExpr(y)`

`y <- y[keep,]`

接着y可用于评估离散度

或同下图

edgeR

An example of creating a `DGEList` for use with *edgeR* (Robinson, McCarthy, and Smyth 2010):

```
library(edgeR)
```

```
cts <- txi$counts
normMat <- txi$length
normMat <- normMat/exp(rowMeans(log(normMat)))
library(edgeR)
o <- log(calcNormFactors(cts/normMat)) + log(colSums(cts/normMat))
y <- DGEList(cts)
y <- scaleOffset(y, t(t(log(normMat)) + o))
# filtering
keep <- filterByExpr(y)
y <- y[keep, ]
# y is now ready for estimate dispersion functions see edgeR User's Guide
```

而DESeq2不需要上步骤转换，直接导入

counts矩阵，不是数据框，包含counts数值

samples数据框，包含样本或文库信息，另包含lib.size列，用于描述文库大小或测序深度，如果未指定，直接从counts列计算而来

genes数据框，可选的，包含genes或genomic feature的注释信息

`y <- DGEList(counts=x, group=group)`

3. Filtering

- 去重复transcript，保留最大counts

首先order排序，从大到小，然后再去重复，标记samller subsripts为重复

`o <- order(rowSums(y$counts), decreasing=T)`

`y <- y[o,]`

`d <- duplicated(y$genes$Symbol)`

`y <- y[!d]`

- filterByExpr(y)

CPM: 平均到每百万个read时的一个exon的read计数，用于样本间exon比较

RPKM: 平均到每百万个read时，以1kb为单位时，一个exon的read计数，用于样本间exon比较和样本内exon比较

read counts过滤，所有文库具有的低counts的genes将会为差异表达分析提供很小的贡献，此外，从生物学角度而言，一个gene必须达到一个最小的水平，才可能被转录为蛋白或表现出生物作用。**经验而言，假如一个gene在所有样本或者所有条件状态下都没有检测到，那么该gene就不会表达，一般设置单个文库的表达至少为5-10才认为该gene在该文库内表达。也可以定义count-per-million(CPM)，而不根据counts值，因为counts的直接过滤没有考虑样本间文库片段的大小问题。**

Roughly speaking,the strategy keeps genes that have at least 'min.count' reads in a worthwhile number samples. More precisely, the filtering keeps genes that have count-per-million (CPM) above *k* in *n* samples, where *k* is determined by 'min.count' and by the sample library sizes and *n* is determined by the design matrix.

filterByExpr函数默认选取最小的组内的样本数量为最小的样本数，保留至少在这个数量的样本中有10个或更多的序列片段计数的基因。根据数据集中的总序列数的中位数,`median(y$samples$lib.size)*1e-6`,例如为1.9, 那么10个count约等于10/1.9=5.26, 那么filterByExpr函数保留在至少3个样本(最小组内样本数)中CPM值大于等于5.26的基因；使用CPM过滤可避免对总序列大的样本的偏向性

返回对应的nrow(y)长度向量，表明哪些行可以保留

```
keep <- filterByExpr(y, group=group)
```

```
y_filtered <- y[keep,]
```

```
y <- DGEList(counts=y_filtered$counts, group=group)
```

针对过滤后的DGEList对象，建议重新计算lib.size。

```
y$samples$lib.size <- colSums(y$counts)
```

或使用手动方法：

cpm/rpkm, 计算每百万条read的基因上的count数目/每百万read中基因的每千碱基长度的read数目; CPM只对read count相对总reads数做了数量的均一化。当如果想进行表达量的基因间比较，则不得不考虑基因长度的不同。如果进一步做长度的均一化，就得到了下面的RPKM; RPKM法能消除基因长度和测序量差异对计算基因表达的影响，计算得到的基因表达量可直接用于比较不同样品间的基因表达差异和不同基因间表达高低的比较。

cpm(y, log=2,prior.count=2), 这里采用log2, prior.count为避免出现0, 默认使用normalized library sizes

例如：

```
rowSums(cpm(y) > 0.5) > =2
```

或者直接根据counts数目过滤:

```
rowSums(y$counts > 10) >= 2
```

- Normalization

edgeR关注的是差异表达分析，而不是量化差异表达水平，因此，所关心的是不同状态下表达水平的相对改变情况，而不是直接评估绝对表达水平。例如，read counts会与转录本的长度成正比例，同时每个RNA样本的gene都会有相同的read counts比例出现，因此不需要根据gene长度做校准。仅当技术因素会产生样本特异性的效果时，才需要做标准化。

相对与gene表达水平，测序深度是最明显的影响read counts的因素。edgeR会根据不同的测序深度(lib.size)来校准差异表的分析，同时该部分是fold-chang，p-value计算过程中自动完成，无需额外的分析处理。

RNA-seq提供了测量每个RNA样本每个gene的相对丰度，但是不能检测每个细胞内总共的RNA产出。因此，当小部分gene处于非常高表达时，该细胞内的另一部分gene会因为总libraray size被高表达gene所消耗掉，处于under-sample状态，会导致假的下调结果。因此，**calcNormFactors函数会根据library sizes来校准RNA的组成，从而最小化样本中大部分gene的log-fold changes**，默认计算scale factors使用trimmed mean of M-values(TMM) between each pair of samples。使用effective library size取代original library size用于下游分析。

```
y <- calcNormFactors(y,method="TMM")
```

```
> y_fil_nor$sample
         group lib.size norm.factors
NA18486      1  7750614    0.9294587
NA18498      1 13614927    1.0964857
NA18499      1  8570996    0.9576338
NA18501      1  8596932    1.1944060
NA18502      1 13377004    0.9422220
NA18504      1  9883172    0.9827715
NA18505      1 12302920    1.0259967
NA18507      1 14097355    1.0458764
NA18508      1 12610934    1.0225245
```

Effective library size等同于lib.size乘以norm.factors。当norm.factors小于1时，表明存在少量高表达genes垄断了测序数据，导致其他genes的counts数少于给定的library size，因此effective library size会小于original library size，类似于下调counts。

每个样本的gene的GC-content和长度都是一样的，因此不会给差异表达分析造成影响。**样本特异性的GC-content影响可以通过EDASeq和cqn包来修正GC-content带来的偏差。**

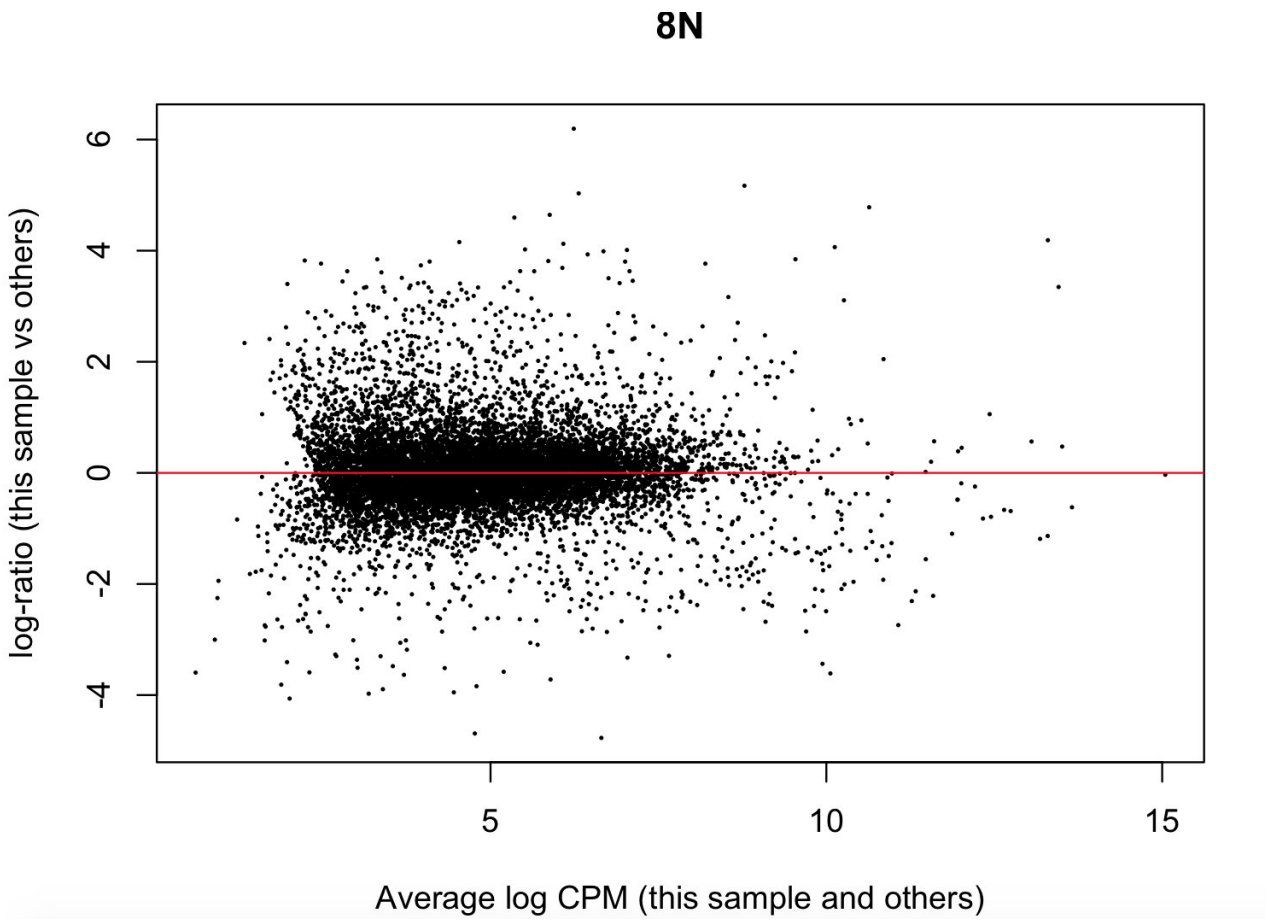
针对基于模型的标准化，可以设置correction factors用于calcNormFactors，计算effective library size。需要注意的是，标准化的过程，并不设计到原始read coutns的改变，因此在将其输入到edgeR之前，不应对read counts做任何的转换，例如 RPKM和FPKM值取代read counts用于edgeR，或者在counts上加入额外的值，此类转换会阻止edgeR评估数据的平均变异关系。

classic edgeR函数estimateCommonDisp和exactTest所生成的pseudo-count仅作为输出的部分，用于edgeR流程使用，不用其他用途。另外，pseudo-counts是一种标准化后的counts，而prior count是用于offset 小的counts的一个起始值。

绘制表达个体样本的表达图，能够更近低查看mean-difference(MD) plots。MD图展示了两个样本经过library-size adjusted后的log-fold改变。默认使用样本1(column=1)文库的比较其他所有样本均值的构成的参考样本文库。

```
plotMD(y, column=1)
```

```
abline(h=0, col="red", lty=2, lwd=2)
```



每个点代表一个gene，红线代表log-ratio为0，可见主要的点都是围绕红线聚集

4. 评估离散度及差异表达分析

BCV：biological coefficient of variation，是gene真实丰度在重复的RNA样本之间变化的系数，**它代表了biological replicates在测序深度无限制增加时依旧存在于样本之间的coefficient of variation，是负二项式模型的离散度参数的平方根，等同于负二项式模型的离散度。**

最简单的假设就是所有gene拥有相同的mean-variation关系，也是就是说所有gene的离散度都相同，"common dispersion"。

实际上gene的表达水平是不同的，在gene之间是独立存在的。因此，引入empirical bayes moderation进行genewise离散度评估，"tagwise diserpsion"。

- 针对单因素实验的(group的factor) 离散度评估，采用quantile-adjusted conditional maximum likelihood(qCML) 方式。

```
y <- estimateDisp(y)
```

等同于分步骤计算：

`y <- estimateCommonDisp(y)`：estimate common negative binomial dispersion by conditional maximum likelihood

`y <- estimateTagwiseDisp(y)`：estimate empirical bayes tagwise dispersion values

完成negative binomial modles fitted和dispersion estimates后，使用exact test检出差异表达，且exact test仅使用于单因素实验差异分析

```
et <- exactTest(y)
```

```
topTags(et)
```

- 针对复杂实验设计，设计到多个因素的关系，采用广义线性模型分析差异表达(quasi-likelihood (QL) F-test, glmQLFit, glmQLFTest/likelihood ratio test, glmFit, glmLRT)

edgeR采用Cox-Reid profile-adjusted likelihood(CR)方式来评估离散度

多因素实验设计，首先对实验样本进行分组，然后建立design matrix，最后根据design matrix评估离散度，以及检测差异表达。

```
group <- factor(c(1,1,2,2,3,3))
```

```
design <- model.matrix(~group)
```

根据design matrix评估离散度，robust=TRUE，用于保护empirical bays estimates对具有格外大或者小的离散的outlier genes的评估(Here robust=TRUE has been used to protect the empirical Bayes estimates against the possibility of outlier genes with exceptionally large or small individual dispersions)

`y <- estimateDisp(y, design, robust=T)`：Estimate Common, Trended and Tagwise Negative Binomial dispersions by weighted likelihood empirical Bayes

等同于分步计算：

`y <- estimateGLMCommonDisp(y, design)`：Estimates a common negative binomial dispersion parameter for a DGE dataset with a general experimental design

`y <- estimateGLMTreadedDisp(y, design)`：Estimates the abundance-dispersion trend by Cox-Reid approximate profile likelihood

`y <- estimateGLMTagwiseDisp(y, design)`：Compute an empirical Bayes estimate of the negative binomial dispersion parameter for each tag, with expression levels specified by a log-linear model

完成negative binomial modles fitted和dispersion estimates后，可使用quasi-likelihood(QL) F-test或likelihood ratio test评估差异表达(quasi-likelihood (QL) F-test, glmQLFit, glmQLFTest/likelihood ratio test, glmFit, glmLRT)

QLF-test能反应出评估每个gene离散度的不确定性，当每个样本的重复次数较小时，能提供更稳健和可信的错误控制率

根据design matrix检测表达差异，推荐使用robust=TRUE，允许gene-specific prior of estimates。减少了特别高或低的原始离散度的genes所带来的假阳性结果，同时增加了主体genes的差异表达检出能力(Setting robust=TRUE in glmQLFit is usually recommended. This allows gene-specific prior df estimates, with lower values for outlier genes and higher values for the main body of genes. This reduces the Chance of getting false positives from genes with extremely high or low raw dispersions, while at the same time increasing statistical power to detect differential expression for the main body of genes)

```
fit <- glmQLFit(y, design, robust=T)
```

```
qlf <- glmQLFTest(y, coef=2)
```

```
topTags(qlf)
```

采用likelihood ratio test时

根据design matrix检测表达差异

```
fit <- glmFit(y, design)
```

```
lrt <- glmLRT(fit, coef=2)
```

```
topTags(lrt)
```

5. 信号通路分析

gene ontology(GO)富集分析和KEGG通路分析时用于解释差异表达结果常用的下游分析步骤。针对上调或者下调的genes，GO或pathway富集分析将查找与这些genes相关的GO terms或pathway。

最简单且常用到的解释差异表达gene的方法就是，注释到可能的GO term上， 在一组gene中出现频率最高的GO term就可能是过表达或富集的term。

KEGG数据库远小于GO，记录了分子通路和疾病相关数据。默认的，kegga自动读取最近的KEGG数据信息用于注释。

goana()和kegga()会采用DGELRT或DGEExat数据进行分析，另外，两者均采用NCBI RefSeq 注释信息，因此输入的数据的gene行名称应该为Entrez Gene identifier(ID)，species对应研究物种。

```
go <- goana(qlf, species="Mm")
```

```
topGO(go, sort="up")
```

```
keg <- kegga(qlf, species="Mm")
```

```
topKEGG(keg, sort="up")
```

```
> topGO(go, n=15)
                                      Term Ont    N Up Down  P.Up    P.Down
GO:0022402              cell cycle process  BP  931 19  118 0.913 2.69e-23
GO:0000280                nuclear division  BP  460 10   78 0.789 3.58e-23
GO:1903047     mitotic cell cycle process  BP  628  8   92 0.995 1.56e-22
GO:0048285                organelle fission  BP  500 11   78 0.785 7.35e-21
GO:0007067       mitotic nuclear division  BP  376  4   66 0.991 1.51e-20
GO:0007049                     cell cycle  BP 1301 21  138 0.997 6.22e-20
GO:0000278             mitotic cell cycle  BP  736  8   96 0.999 7.97e-20
GO:0007059         chromosome segregation  BP  237  1   49 0.998 1.50e-18
GO:0051301                  cell division  BP  550  6   77 0.997 8.95e-18
GO:0000776                    kinetochore  CC  112  1   33 0.952 8.99e-18
GO:0000775 chromosome, centromeric region  CC  163  1   38 0.988 1.77e-16
GO:0098813 nuclear chromosome segregation  BP  171  1   38 0.990 9.64e-16
GO:0042254            ribosome biogenesis  BP  223  1   42 0.998 1.43e-14
GO:0098687             chromosomal region  CC  278  6   47 0.756 2.69e-14
GO:0005730                       nucleolus  CC  663  4   78 1.000 9.72e-14
```

默认GO分析差异gene需要满足阈值5%，行名为GO term，Term列为方便读识的解释，Ont表示GO term所属类别(BP, biological process, CC, cellular component, MF, molecular function)，N代表GO term所含的总gene数，Up和Down列代表属于该GO term的上调下调gene数，P.up和P.down对应其p-values，一般忽略p-values大于10[^-5] 的GO term

KEGG的topKEGG分析结果解释同topGO分析结果说明。

6. 可变剪切分析

egdeR可以对RNA-seq数据在exon水平检测差异剪切和isoform-specific差异表达。可变剪切通过检测每个gene的exon差异表达水平实现。

diffSpliceDGE()可同时检测exon-level和gene-level水平的可变剪切，exon水平的logFC用于多个exon的差异剪切，而gene水平的logFC用于检测一个gene少数几个exon的差异剪切。

```
sp <- diffSpliceDGE(fit, coef=4, geneid="GeneID", exonid="Start")
```

Simes方式适用于一个gene少数几个exon差异剪切的检测，而F-tests适用于检测一个gene多个exon的差异剪切检测。

```
topSliceDGE(sp, test="Simes", n=20)
```

或F-tests

```
topSpliceDGE(sp, test="gene", n=20)
```

图示

```
par(mfrow=c(1,2))
```

```
plotSpliceDGE(sp, geneid="trol", genecol="Symbol")
```

```
plotSpliceDGE(sp, geneid="lola", genecol="Symbol")
```

7. FRY gene set tests

GO和KEGG分析相对简单，仅依靠一组差异表达的gene，同时该组gene跨越了多个GO和KEGG注释项，最终结果依赖差异表达的统计显著性。

roast or fry gene set test用于检测个别gene表达或者个别pathway的显著性，评估整体的gene集作为一个set来评估，该gene set可以被选来代表任何感兴趣的phenotype或pathway。

- 从GO.db挑选感兴趣GO term，获得对应的genes，例如和cytokinesis相关的GO terms

```
library(GO.db)
```

```
cyt.go <- c("GO:0032465", "GO:0000281", "GO:0000920")
```

```
term <- select(GO.db, keys=cyt.go, columns="TERM")
```

```
> term
        GOID                                  TERM
1 GO:0032465          regulation of cytokinesis
2 GO:0000281                mitotic cytokinesis
3 GO:0000920 septum digestion after cytokinesis
```

提取对应GO term的genes信息

```
library(org.Mm.eg.db)
```

```
Rkeys(org.Mm.egGO2ALLEGS) <- cyt.go
```

```
cyt.go.genes <- as.list(org.Mm.egGO2ALLEGS)
```

- 构建比较group，检测GO terms在比较大group中是否差异表达

```
B.VvsL <- makeContrasts(B.virgin - B.lactating, levels=design)
```

```
fry( y_cal_disp, index=cyt.go.genes, design=design, contrast=B.VvsL)
```

```
> fry(y_cal_disp,index=cyt.go.genes,design=design, contrast=B.VvsL)
           NGenes Direction      PValue         FDR PValue.Mixed    FDR.Mixed
GO:0032465     67        Up 0.001152679 0.003458036 8.424911e-06 8.424911e-06
GO:0000920      7      Down 0.006614836 0.009922254 7.096017e-06 8.424911e-06
GO:0000281     77        Up 0.037963171 0.037963171 6.717914e-06 8.424911e-06
```

NGenes，对应GO term所含gene数目；Direction表示改变net方向；PValue列给定双尾检测作为整体的set gene是否差异高或者低表达；PValue.Mixed列给定set中的genes是否倾向于差异表达，不考虑方向；FDR计算错误率。

Miscellaneous

1. 当没有重复样本测序时

- **简单选择可靠的离散值，用于exactTest或glmFit。Typical values for the common BCV(square-root-dispersion) for datasets arising from well-controlled experiments are 0.4 for human data, 0.1 for data on genetically identical model organisams or 0.01 for technical replicates**。

```
bcv <- 0.2
```

```
counts <- matrix(rnbinom(40, size=1/bcv^2, mu=10), 20,2)
```

`y <- DGEList(counts=counts, group=1:2)`

`et <- exactTest(y, dispersion=bcv^2)`

- 假如存在相当大的不会发生差异表达的control 转录本数量，那么可以用它们评估disersion。例如，housekeeping genes不会根据实验条件而发生差异表达。

`y1 <- y`

`y1$samples$group <- 1`

然后将所有文库作为一个group，使用housekeeping genes评估common dispersion

`y0 <- estimateDisp(y1[housekeeping,], tread="none", tagwise=FASLE)`

再将其插入到数据中

`y$common.dispersion <- y0$common.dispersion`

`fit <- glmFit(y, design=design)` / `fit <- glmFit(y, design=design,dispersion=y0$common.dispersion)`

`lrt <- glmLRT(fit, coef=2)`

因此，需要大量的control transcripts，至少几十个，最好上百个。

2. 高于倍数阈值的差异表达

edgeR在GLM框架下提供了关于阈值假设的统计检，给定fold-change或log-fold-change阈值，输入来自glmFit或glmQLFit的DGEGLM对象，通过glmTreat函数阈值检测实现，假如lfc为0，那么等同于glmQLFTest

`fit <- glmQLFit(y, design)`

`tr <- glmTreat(fit, coef=2, lfc=1)`

`topTags(tr)`

lfc阈值不是限定大于fold-change的阈值才输出，而是大于该阈值的才会认为会有统计显著性。

3. 聚类和热图

针对每对RNA样本之间的leading log-fold-change，plotMDS绘制对应的多维度距离图。leading log-fold-change为，每对样本之间的最大距对log-fold-change的均值(root-mean-square)。同时该函数也提供了针对BCV的距离图。

针对filter和normalization后的DGEList对象

`plotMDS(y)`

针对RNA-seq样本，建议使用moderated log-counts-per-million，y为normalized DGEList对象。

`logcpm <- cpm(y, log=T, prior.count=2)`

`logrpkm <- rpkm(y, log=T,prior.count=2)`

`libary(pheatmap)`

```
pheatmap(logcpm)
```

4. matrix design

- classic edgeR approach is to make pairwise comparisions between groups

```
et <- exactTest(y, pair=c("A","B"))
```

```
topTags(et)
```

or B vs. C

```
et <- exactTest(y, pair=c("C","B"))
```

```
et <- exactTest(y, pair=c(3,2))
```

这里group的level是按照字母顺序排列的，也可以设定level的ref 或control level

```
y2 <- y
```

```
y2$samples$group <- relevel(y$samples$group, ref="C")
```

```
levels(y2$samples$group)
```

```
[1] "C" "A" "B"
```

- glm approach require a design matrix to describe the treatment conditons

```
design <- model.matrix( ~ 0 + group, data=y$samples)
```

```
colnames(design) <- levels(y$samples$group)
```

在model formula中加0就是为了消除回归方程中的截距值，这样也给了ref水平一个coefficient，方便理解

```
fit <- glmQLFit(y, design)
```

```
qlf <- glmQLFTest(fit, contrast=c(-1,1,0))
```

```
topTags(qlf)
```

```
qlf <- glmQLFTest(fit, contrast=c(-0.5, -0.5, 1))
```

constrast: 使用数字向量或矩阵来指定一个或多个用于比较的线性模型矩阵coefficient，检测coefficent是否为0

传统方式，公式不加0项

```
design <- model.matrix(~group, data=y$sampels)
```

此时第一个coefficient就是截距，表示第一个水平的treatment condition(group A)的baseline logCPM表达水平

```
fit <- glmQLFit(fit, design)
```

```
qlf <- glmQLFTest(fit, coef=2)
```

省略coef值，对应默认为colnames(design)最后一个系数值

检测B vs A，这里ref为A

```
qlf <- glmQLFTtest(fit, contrast=c(0,-1,1))
```

仍然检测B vs A

```
qlf <- glmQLFTest(fit, coef=1)
```

不应使用，因为此时检测第一个coefficient是否为0，这里比比较group A的logCPM为0没有意义。

检测任意两组之间差异

```
qlf <- glmQLFTest(fit, coef=2:3)
```

```
topTags(qlf)
```

用于检测B-A, C-A是否为0，也就是是否存在差异。

Nested interaction forumulas

```
design <- model.matrix(~Treat + Treat:Time, data=targets)
```

```
fit <- glmQLFit(y, design)
```

这里的Time作为一个交互项，提供了对Treat的一个额外限制分类而已，对应的glmQLFTest应用不变。

Interaction at any time

```
design <- model.matrix(~Treat * Time, data=targets)
```

等同于

```
design <- model.matrix(~Treat + Time + Treat:Time, data=targets)
```

对应glmQLFTest使用不变。

Case 1

寻找男性和女性之间差异表达对genes

- Loading the data

```
library(tweeDEseqCountData)
```

```
data(pickrell1)
```

```
Counts <- exprs(pickell1.eset)
```

比较男性和女性，查看样本性别

```
Gender <- pickrell1.eset$gender
```

```
table(Gender)
```

对应其gene注释文件

```
data(annotEnsembl63)
```

```
annot <- annotEnsembl63[,c("Symbol","Chr")]
```

创建DGEList对象

```
y <- DGEList(counts=Counts, genes=annot[rownames(Counts),])
```

- Filtering and normalization

使用filterByExpr过滤原始counts

```
isexpr <- filterByExpr(y, group=Gender)
```

过滤掉注释gene名称为NA的行，同时修改过滤后的lib.size值

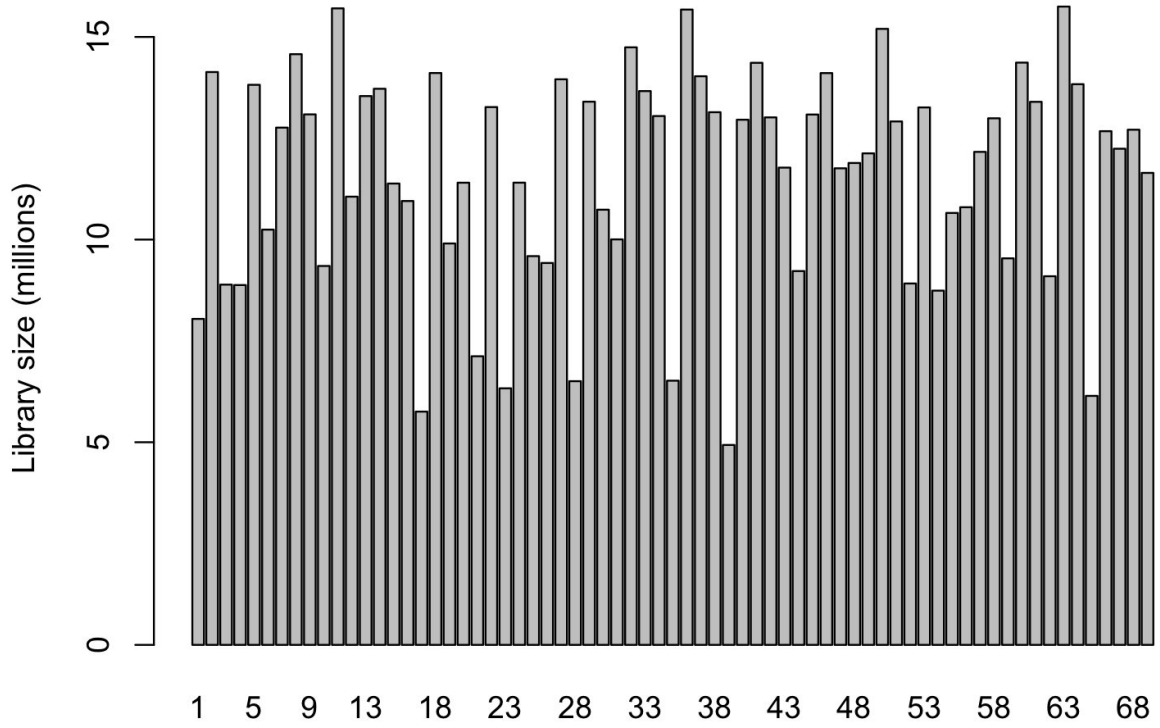
```
hasannot <- rowSums(is.na(y$genes)) == 0
```

```
y_filtered <- y[isexpr & hasannot, ]
```

```
y_filtered$samples$lib.size <- colSums(y_filtered$counts)
```

绘制条图展示每个样本reads大小

```
barplot(y$samples$lib.size*1e-6, names=1:69, ylab="Library size (millions)")
```



使用TMM normalization来计算标准化factors

```
y_filtered_nor <- calcNormFactors(y_filtered)
```

- Estimating the dispersion

这里感兴趣的是男女之间的表达差异，创建包含性别因子的design matrix

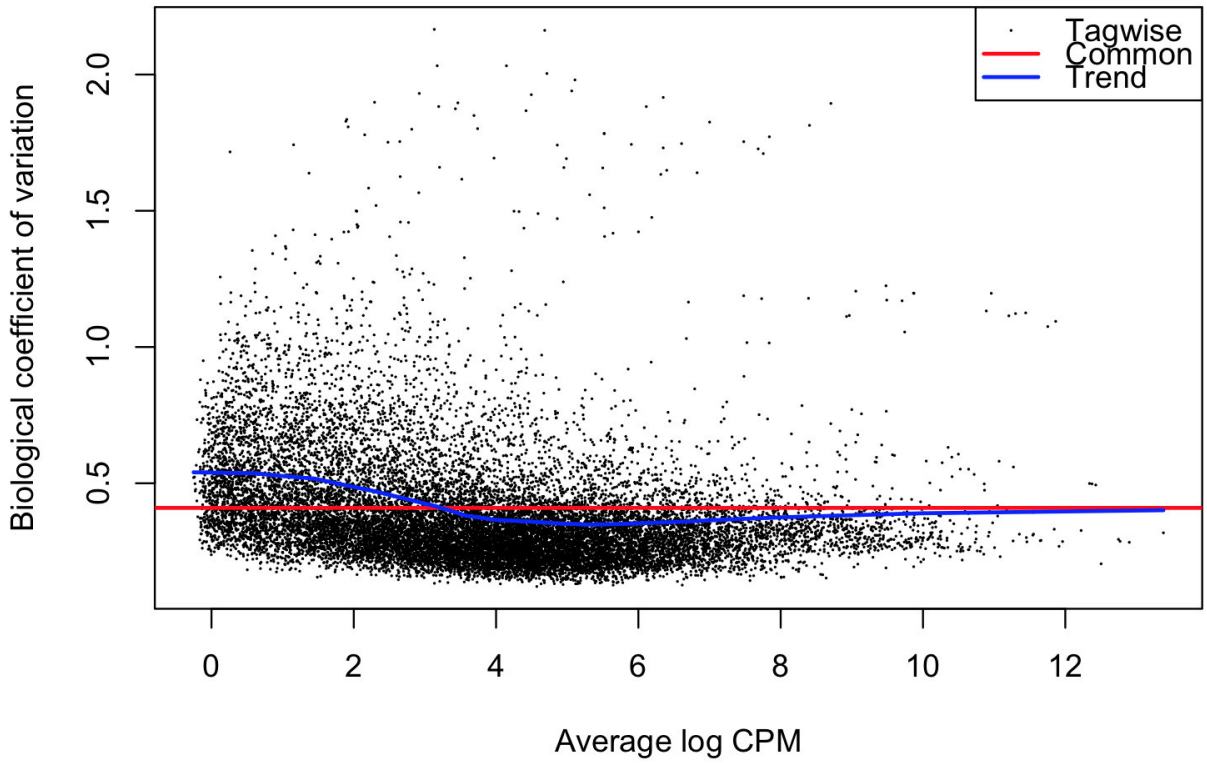
```
design <- model.matrix(~Gender)
```

选中robust：the estimation is robustified against potential outlier genes

```
y_filtered_nor_est <- estimateDisp(y, design, robust=T)
```

绘制genewise变异系数和gene 丰度(log2 counts per million)之间的关系图

```
plotBCV(y_filtered_nor_est)
```



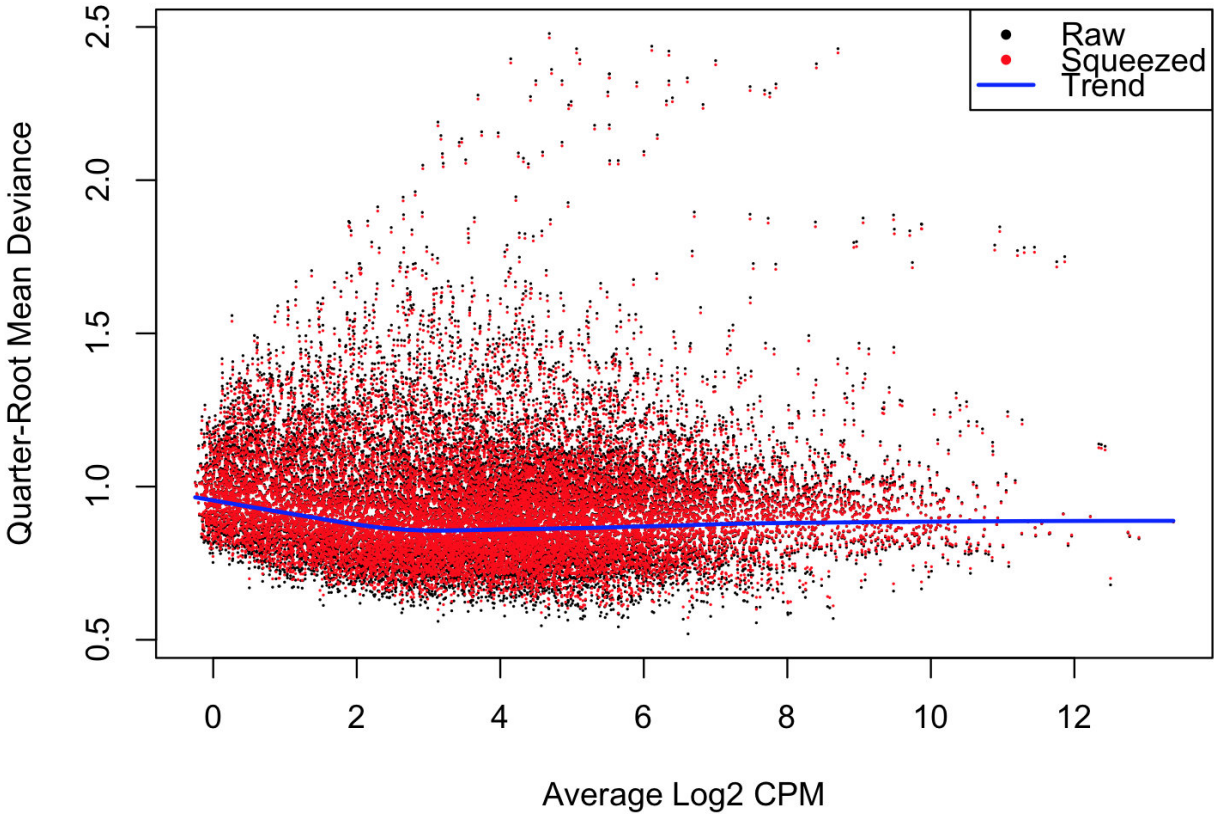
图中的点对应y_filtered_nor_est中对应值

使用glmQLFit评估dispersion trend的QL离散度

```
fit <- glmQLFit(y_filtered_nor_est, design, robust=T)
```

绘制genewise quasi-likelihood 离散度于gene丰度(log2 counts per million)之间关系

```
plotQLDisp(fit)
```



图中可见，glmQLFit回归模型是根据disepresion tread值建立的

- Differential expression

根据性别分组，因此最大的差异表达gene会出现在X或Y染色体上

```
colnames(design) : [1] "(Intercept)" "Gendermale"
```

```
qlf <- glmQLFTest(fit)
```

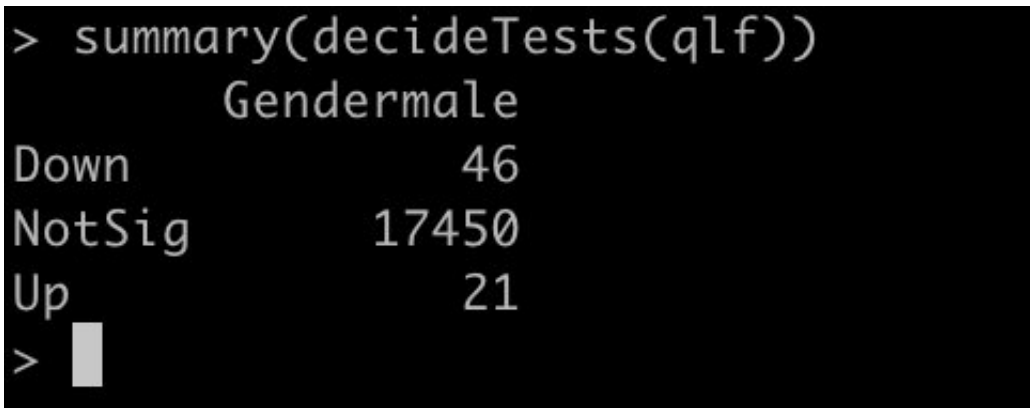
```
topTags(qlf, n=15)
```

```
> topTags(qlf,n=15)
Coefficient:  Gendermale
                     Symbol Chr     logFC    logCPM         F       PValue
ENSG00000229807          XIST   X -9.483930 7.2485638 1208.7498 1.111831e-46
ENSG00000099749      CYorf15A   Y  4.280995 1.7573264  857.5624 1.098064e-41
ENSG00000131002      CYorf15B   Y  5.625463 2.0561674  583.7136 3.130662e-36
ENSG00000157828        RPS4Y2   Y  3.174347 4.2077043  576.5347 4.653353e-36
ENSG00000233864        TTTY15   Y  4.839181 1.2541225  536.1408 4.713876e-35
ENSG00000198692        EIF1AY   Y  2.361532 3.2468360  376.0180 2.843761e-30
ENSG00000165246        NLGN4Y   Y  5.094421 1.6750613  305.4040 1.377885e-27
ENSG00000183878           UTY   Y  1.859205 3.1366773  254.3942 2.595209e-25
ENSG00000243209    AC010889.1   Y  2.660333 0.7974748  230.8797 3.856753e-24
ENSG00000129824        RPS4Y1   Y  2.528374 5.4013153  228.7910 5.203445e-24
ENSG00000012817         KDM5D   Y  1.467618 4.9490771  226.3579 6.638451e-24
ENSG00000213318  RP11-331F4.1  16  3.667248 3.6882542  214.3896 3.706763e-23
ENSG00000067048         DDX3Y   Y  1.623651 5.6207194  183.1320 1.894284e-21
ENSG00000146938        NLGN4X   X  3.938783 1.0468230  140.2532 1.519127e-18
ENSG00000232928 RP13-204A15.4   X  1.443493 3.5576682  112.2998 2.455676e-16
                         FDR
ENSG00000229807 1.947595e-42
ENSG00000099749 9.617393e-38
ENSG00000131002 1.827994e-32
ENSG00000157828 2.037820e-32
ENSG00000233864 1.651459e-31
ENSG00000198692 8.302360e-27
ENSG00000165246 3.448059e-24
ENSG00000183878 5.682535e-22
ENSG00000243209 7.506526e-21
ENSG00000129824 9.114874e-21
ENSG00000012817 1.057143e-20
ENSG00000213318 5.410947e-20
ENSG00000067048 2.552474e-18
ENSG00000146938 1.900753e-15
ENSG00000232928 2.867738e-13
```

检测显著表达的差异genes，**decideTestsDGE同decideTests识别FDR为5%的差异表达genes数目**，默认adjust.method="BH", p.value=0.05

```
summary(decideTestsDGE(qlf))
```

```
summary(decideTests(qlf))
```



抑或手动添加p.adjust值

```
qlf$table$P.adjust <- p.adjust(qlf$table$PValue,"BH")
```

```
$table
                    logFC    logCPM           F       PValue  P.adjust
ENSG00000127720 -0.01061925 0.8791829  0.00478902 0.945020418 0.9926199
ENSG00000051596  0.03205493 5.1848978  0.09309232 0.761163974 0.9596473
ENSG00000236211 -0.06280782 5.5688169  0.06969821 0.792541745 0.9661428
ENSG00000213697  0.59454817 1.1040161 11.60793646 0.001078923 0.1959965
ENSG00000135541  0.08888236 1.9507066  0.35621069 0.552493776 0.9067772
17512 more rows ...
```

- Gene set testing

tweeDEseqCountData包含了一系列男性和女性独有的genes信息，检验这些gene是否存在与差异表达中

```
data(genederGenes)
```

```
Ymale <- rownames(y_filtered_nor_est) %in% msYgenes
```

```
Xescape <- rownames(y_filtered_nor_est) %in% XiEgenes
```

使用fry()函数检测男性特有gene在与女性比较时显著性高表达，而女性特有gene显著性低表达

```
index <- list(Y=Ymale, X=Xescape)
```

```
fry(y_filtered_nor_est, index=index, design=design)
```

```
> fry(y_filtered_nor_est,index=index, design=design)
  NGenes Direction       PValue          FDR PValue.Mixed    FDR.Mixed
Y     12        Up 1.003371e-45 2.006742e-45 6.703818e-11 6.703818e-11
X     47      Down 6.929453e-17 6.929453e-17 1.264739e-68 2.529478e-68
>
```

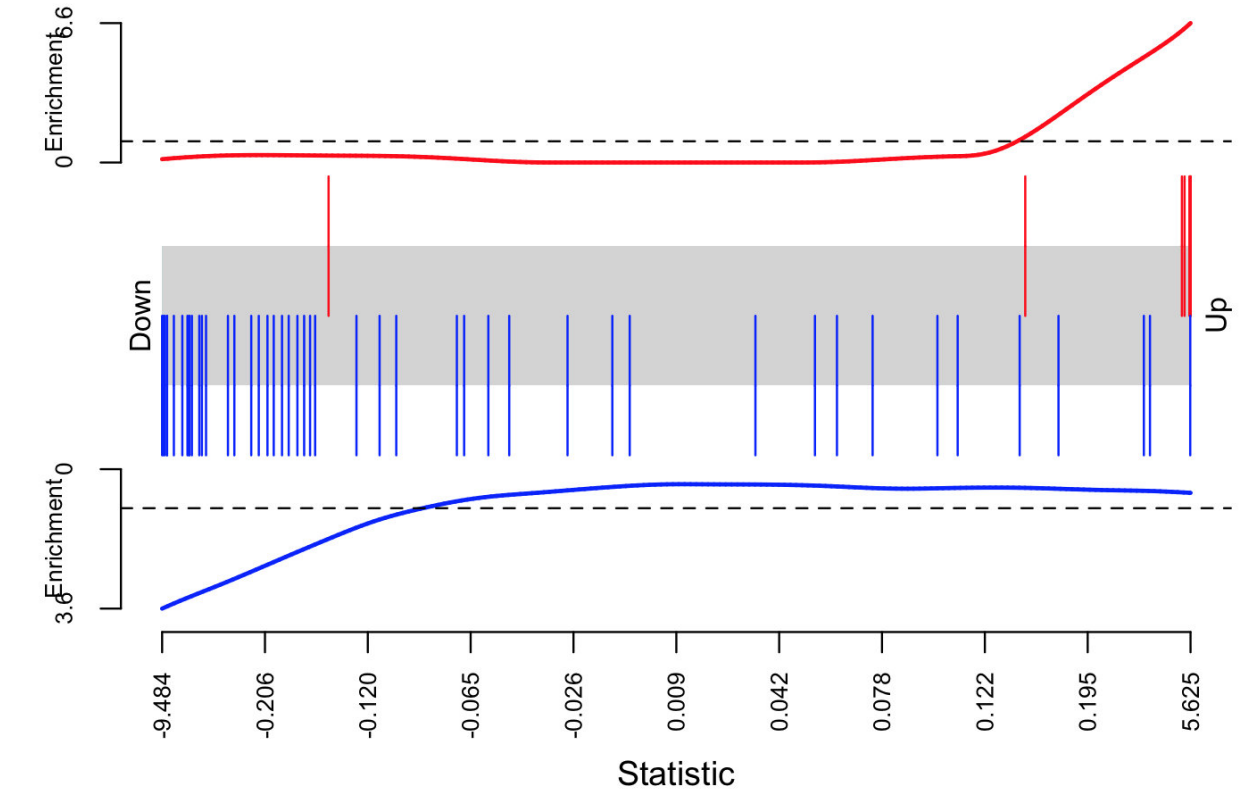
查看对应logFC值

```
> qlf$table[Xescape,]
                    logFC      logCPM           F       PValue     P.adjust
ENSG00000102032 -0.006444191  2.81216516 8.645995e-04 9.766241e-01 9.977028e-01
ENSG00000101846 -0.804760745  3.75194499 1.679037e+01 1.086825e-04 3.120970e-02
ENSG00000185753 -0.416561085  3.55876787 1.783928e+01 6.943087e-05 2.133720e-02
ENSG00000072501 -0.334580781  6.59386671 7.312000e+00 8.544627e-03 4.936466e-01
ENSG00000101849 -0.048998847  4.65926643 1.282012e-01 7.213539e-01 9.530316e-01
ENSG00000130985 -0.213600411  7.76132961 4.135477e+00 4.568607e-02 6.185026e-01
ENSG00000182378  0.061462983  5.73100586 1.733209e-01 6.784178e-01 9.416851e-01
ENSG00000131171  0.110177113  5.50607557 1.445150e+00 2.332536e-01 7.787781e-01
ENSG00000198910 -0.166808458  2.28602013 1.171745e+00 2.826614e-01 8.054329e-01
ENSG00000126012 -0.567477706  6.72794885 4.744159e+01 1.791927e-09 1.012554e-06
ENSG00000176896 -0.188272434  0.87064281 2.792973e+00 9.902751e-02 6.821597e-01
ENSG00000183943 -0.315691992  5.29109442 2.377747e+01 6.275577e-06 2.290193e-03
ENSG00000006756  0.287266274  2.25792020 1.833109e+00 1.800029e-01 7.451294e-01
ENSG00000180182 -0.071572798  4.97959648 7.135402e-01 4.010742e-01 8.483204e-01
```

```
> qlf$table[Ymale,]
                    logFC   logCPM          F       PValue     P.adjust
ENSG00000067048  1.6236508 5.620719 183.131989 1.894284e-21 2.552474e-18
ENSG00000114374  0.6293865 4.705861  37.320517 4.649703e-08 2.262468e-05
ENSG00000165246  5.0944207 1.675061 305.403972 1.377885e-27 3.448059e-24
ENSG00000099749  4.2809954 1.757326 857.562367 1.098064e-41 9.617393e-38
ENSG00000154620 -0.1473538 9.315790   2.540371 1.153573e-01 7.062348e-01
ENSG00000131002  5.6254629 2.056167 583.713615 3.130662e-36 1.827994e-32
ENSG00000099725  0.1447446 4.611111   5.356575 2.350547e-02 5.947392e-01
ENSG00000198692  2.3615316 3.246836 376.018007 2.843761e-30 8.302360e-27
ENSG00000129824  2.5283741 5.401315 228.791032 5.203445e-24 9.114874e-21
ENSG00000067646  0.5574185 3.154752  19.391755 3.638535e-05 1.225696e-02
ENSG00000157828  3.1743469 4.207704 576.534709 4.653353e-36 2.037820e-32
ENSG00000183878  1.8592053 3.136677 254.394193 2.595209e-25 5.682535e-22
```

barcode plot展示对应结果图，genes根据log-fold-change从左到右排列，红色表示msYgenes，蓝色表示XiEgenes。barcode上的线展示了图中各个部分vertical bars的相对局部富集图。

```
barcodeplot(qlf$table$logFC, index[[1]], index[[2]])
```



camera函数检测一组高表达gene相对于其它差异表达genes，是否能够解释内部gene的相关性(Test whether a set of genes is highly ranked relative to other genes in terms of differential expression, accounting for inter-gene correlation)

```
camera(y_filtered_nor_est, index, design)
```

```
> camera(y_filtered_nor_est,index,design)
  NGenes Direction        PValue           FDR
Y     12        Up 1.215533e-295 2.431067e-295
X     47      Down  7.336530e-25  7.336530e-25
>
```

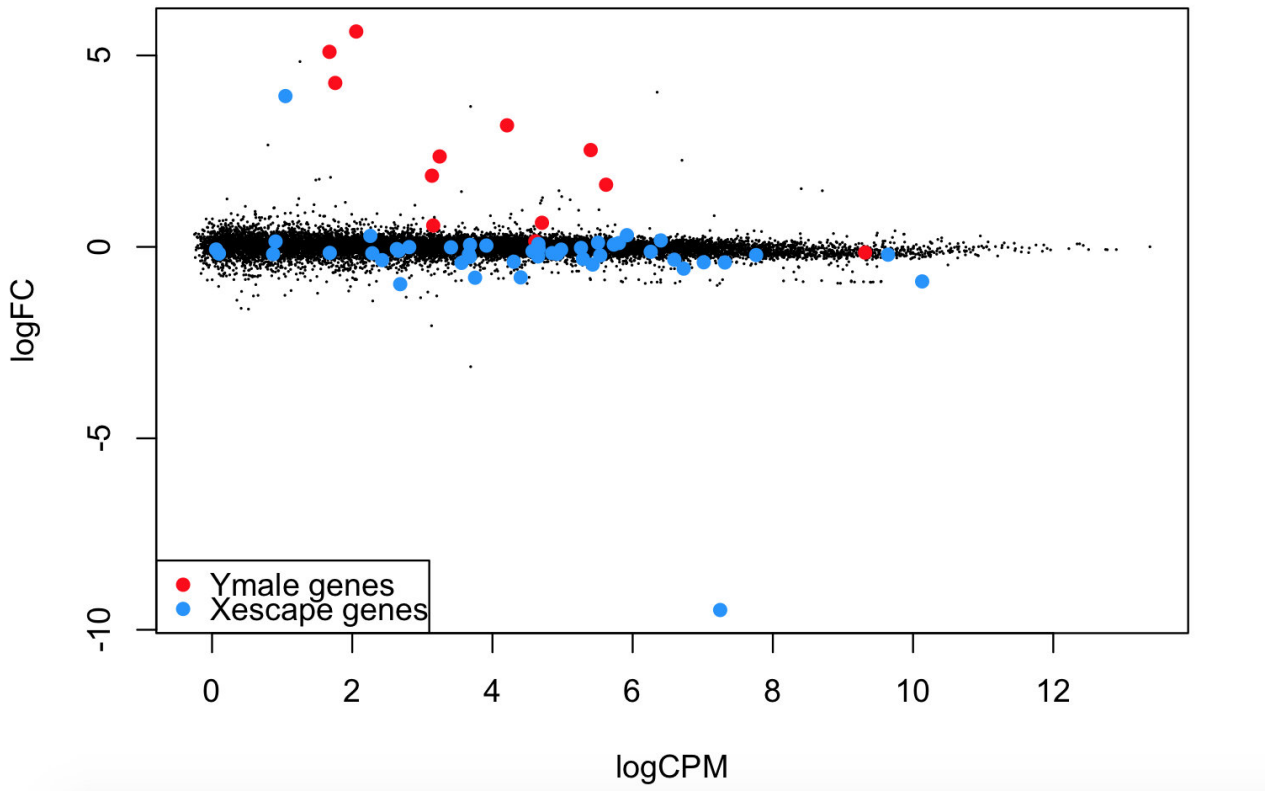
通过MA图查看X和Ygenes的落点位置分布

```
with(qlf$table, plot(logCPM, logFC, pch=16, cex=0.2))
```

```
with(qlf$table, points(logCPM[Ymale], logFC[Ymale], pch=16, col="red"))
```

```
with(qlf$table, points(logCPM[Xescape], logCPM[Xescape], pch=16,
col="dodgerblue"))
```

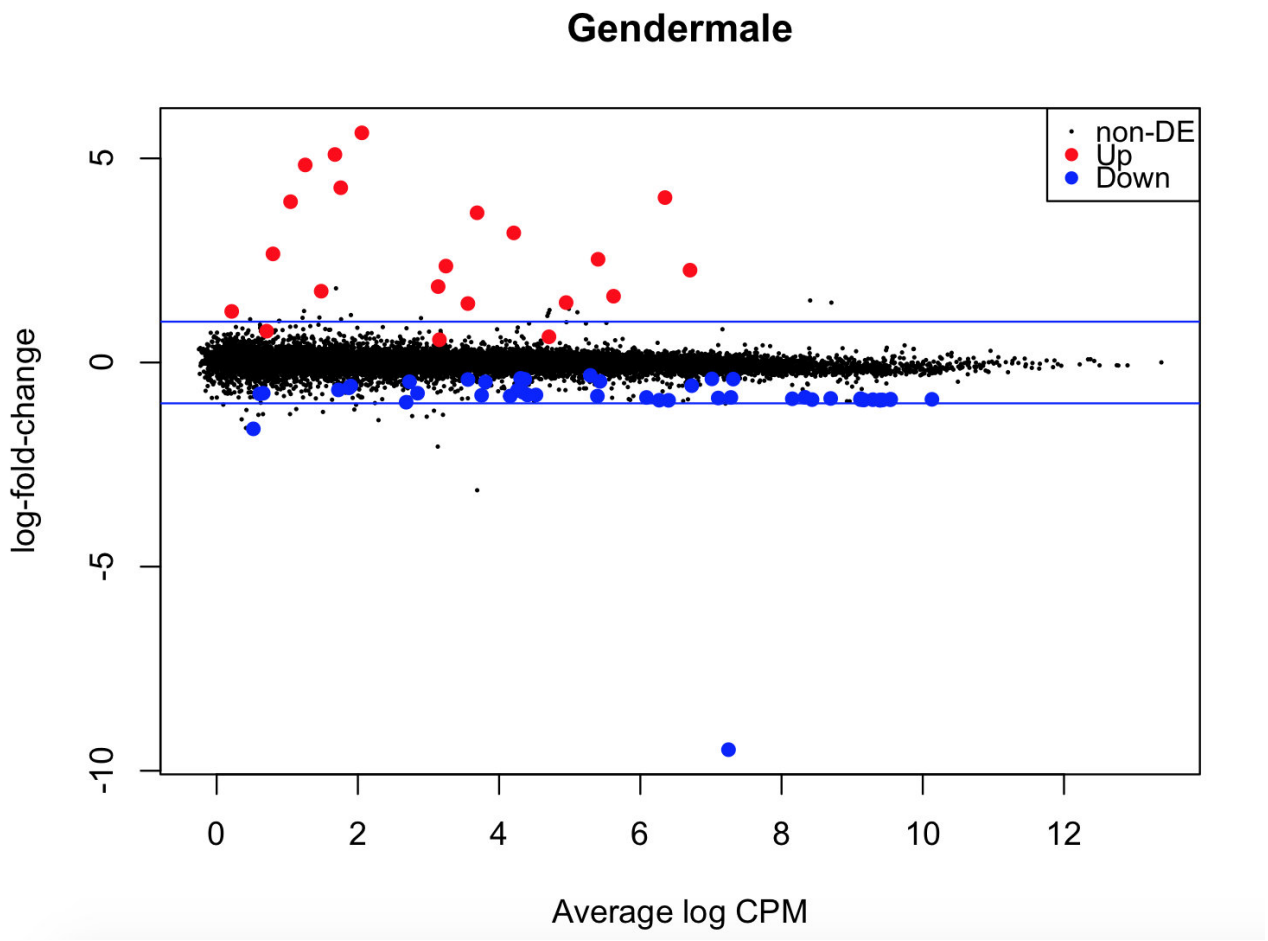
```
legend("bottomleft", legend=c("Ymale genes", "Escape genes"), pch=16,
col=c("red","dodgerblue"))
```



绘制log-fold change图，差异genes高亮

```
plotMD(qlf)
```

```
abline(h=c(-1,1), col="blue")
```



- 针对接下来的GO和pathway分析，对应输入的genes id需要为Entrez id，因此需要在分析开始时，现转换id名称

```
library(org.Hs.eg.db)
```

```
select(org.Hs.eg.db, key=rownames(Counts), column=("ENTREZID"),
keytype="ENSEMBL")
```

返回对应名称关系，添加进入DGEList后方可进行后续分析，略！！！