Introduce

RNA-seq的通过计数数据检测差异表达的基因，该数据呈现出每个样本的每个基因上所能比对到的测序片段数目。其重要的分析问题就是比较不同条件下的有规则变化的统计推导和量化。**DESeq2通过负二项式广义线性回归模型来检测差异表达，以及评估离散度和对数倍数改变。**

DESeq2模型先评估size factors(estimateSizeFactors)，用于得到样本真实的片段浓度；然后评估diseprsions(estimateDispersions)，用于定义观察到的变异与平均值之间的关系；最后通过负二项式广义线性回归 (nbinomWaldTest)检测样本基因log2倍数改变。

The Wald test (also called the Wald Chi-Squared Test)：卡方检验就是统计样本的实际观测值与理论推断值之间的偏离程度，实际观测值与理论推断值之间的偏离程度就决定卡方值的大小，如果卡方值越大，二者偏差程度越大；反之，二者偏差越小；若两个值完全相等时，卡方值就为0，表明理论值完全符合。 注意：卡方检验针对分类变量。

Input

输入数据要求是非标准化的计数统计(un-normalized counts)，数据矩阵的行和列分别对应基因和样本信息，用来表示多少个reads能比对到样本j的基因i上。由于DESeq2会内部校正文库片段大小，因此经过转换或者标准化的数据矩阵不能用于DESeq2的输入数据矩阵。

- SummarizedExperiment input

GenomicAlignemnts包的summarizeOverlaps(mode = 'Union")函数可用于得到基因比对数据矩阵。

`library("airway")`

`data("airday")`

`se <- airway`

`ddsSE <- DESeqDataSet(se, design = ~ cell + dex)`

注意，如果修改了design，那么接下来所有的差异过程都应该重做，因为design公式用于评估离散度以及模型的log2倍数改变。

- Count matrix input

Rsubread包的featureCounts函数可利用比对文件快速出数据矩阵，使用DESeqDataSetFromMatrix函数，以及额外的样本信息数据框。

`library("pasilla")`

`pasCts <- system.file("extdata", "pasilla_gene_counts.tsv", package="pasilla", mustWork=T)`

`pasAnno <- system.file("extdata", "pasilla_sample_annotation.csv", package="pasilla", mustWork=T)`

`countData <- as.matrix(read.csv(pasCts, sep="\t", row.names="gene_id"))` `colData <- read.csv(pasAnno, row.names=1)`

`colData <- colData[, c("condition", "type")]`

这里要保证colData的行名称和countData的列名称要顺序内容一致。

`rownames(colData) <- sub("fb", "", rownames(colData))`

`countData <- countData[, rownames(colData)]`

`dds <- DESeqDataSetFromMatrix(countData = countData, colData = colData, design = ~ condition)`

可以通过增加metadata columns向DESeqDataSet增加额外的feature data。

`featureData <- data.frame(gene=rownames(countData))`

`mcols(dds) <- DataFrame(mcols(dds), featureData)`

- tximport: transcript abundance summarized to gene-level

可以通过tximport包的将以下软件所产生的转录丰度评估数据导入给DESeq2包使用：sailfish，salmon，kallisto，RSEM。以上软件在进行转录本丰度评估时具有以下优势，1）该过程修正了不同样本中潜在的基因长度变化问题；2）sailfish，salmon，kallisto整体上速度更快使用内存更小，想比与基于比对的方式；3）同时可以避免舍弃那些比对到多个具有同源序列的基因的片段，增加了敏感性。

`library(tximport)`

`library(readr)`

`library(tximportData)`

`dir <- system.file("extdata", package="tximportData")`

`samples <- read.table(file.path(dir, "samples.txt"), header=T)`

`files <- file.path(dir, "salmon", samples$run, "quant.sf")`

`names(files) <- paste0("sample", 1:6)`

`tx2gene <- read.csv(file.path(dir, "tx2gene.csv"))`

`txi <- tximport(files, type="salmon", tx2gene=tx2gene, reader=read_csv)`

`coldata <- data.frame(condition = factor(rep(c("A","B"), each=3)))`

`rownames(coldata) <- colnames(txi$counts)`

`ddsTxi <- DESeqDataSetFromTximport(txi, colData=coldata, design = ~ condition)`

- HTSeq input

针对htseq-count数据，使用DESeqDataSetFromHTSeqCount函数。

`directory <- system.file("extdata", package="pasilla", mustWork=T)`

`sampleFiles <- grep("treated", list.files(directory), value=T)`

`sampleCondition <- sub("(.*treated).*", "\\1", sampleFiles)` `sampleTable <- data.frame(sampleName = sampleFiles, fileName = sampleFiles, condition = sampleCondition)`

`ddsHTSeq <- DESeqDataSetFromHTSeqCount(sampleTable = sampleTable, directory = directory, design = ~ condition)`

这里sampleTable对于htseq-cout而言，含有3个或以上的列信息，每一行代表一个样本，第一列为样本名称，第二列为heseq-count所生成的计数文件的文件名称，剩下的行为以后出现在colData中的样本的metadata信息；directory则为htseq-cout的文件路径信息，默认为当前目录。

Fintering

- Pre-filtering

在运行DESeq2前提前过滤low count的genes并不是必要的，但是过滤low count的genes(没有read或几乎没有read)可以控制dds文件大小，加速DESeq2接下来的处理速度。

`dds <- dds[rowSums(counts(dds)) > 1, ]`

其他相同函数见help(rowSums)。

- Note on factor levels

默认，R根据字母排列顺序选择第一个出现的变量为参考水平。因此，可以通过results函数的contrast来设定参考水平或着提前设定因子水平来设定参考水平。

`dds$condition <- factor(dds$condition, levels=c("untreated", "treated"))`

或者relevel

`dds$condition <- relevel(dds$condition, ref="untreated")`

同时使用droplevels来舍弃那些没有样本信息的水平(condition without samples)

`dds$condition <- droplevels(dds$condition)`

- Collapsing technical replicates

collapseReplicates函数将计数矩阵中technical replicates的counts为一列。technical replicates指的是相同样本的多个测序run数据，而biological replicates指的是不同的biological units的多个文库。

`collapseReplicates(object, groupby, run, renameCols=T)` 将technical replicates合并为一列(相加)

DE Analysis

- Differential expression analysis

标准的差异表达分析步骤包含与单个函数DESeq中，最终的results table由results函数生成，该results table包含了log2 倍数改变值，p 值以及adjusted p值。在不加参数情况下，results函数返回最后一个变量水平比上第一个变量水平的差异情况， 例如， conditon treatdvsuntreatd， 该对数倍数改变结果为log2(treated/untreated)

`dds <- DESeq(dds)`

DESeq函数为，基于负二项式分布的差异表达分析，其过程包含以下步骤：1）estimate of size factors: estimateSizeFactors；2）estimate of dispersion：estimatedispersions；3）Negative Binomial GLM fitting and Wald statistics：nbinomWaldTest。最终返回DESeqDataSet对象，然后使用results函数生成log2倍数改变及对应p值。

根据最小的adjustd p值来对res重排序

`resOrdered <- res[order(res$padj), ]`

使用summary函数得到res基本信息

`summary(res)`

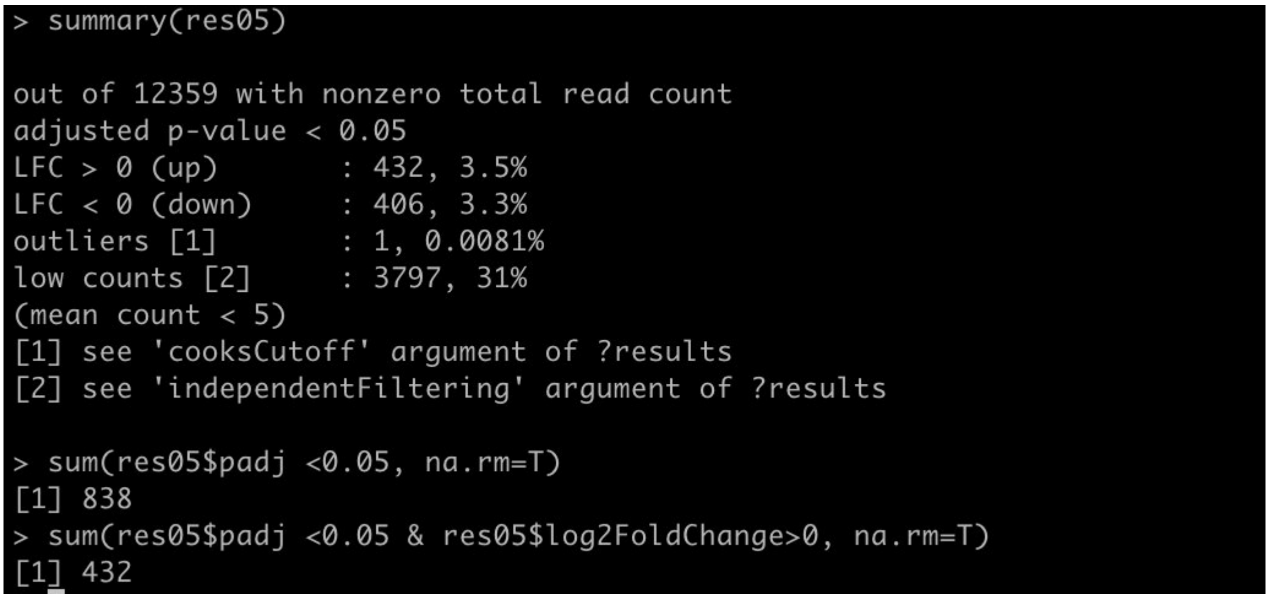
获得adjusted p值小于0.1(默认alpha值)的数目

`sum(res$padj < 0.1, na.rm=T)`

results函数包含了一系列变量指导输出最终results table。同时可指定alpha(对应padjust values)水平，来调整输出table。

`res05 <- results(dds, alpha=0.05)`

`summary(res05)`



```
> summary(res05)

out of 12359 with nonzero total read count
adjusted p-value < 0.05
LFC > 0 (up)       : 432, 3.5%
LFC < 0 (down)     : 406, 3.3%
outliers [1]       : 1, 0.0081%
low counts [2]     : 3797, 31%
(mean count < 5)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

> sum(res05$padj <0.05, na.rm=T)
[1] 838
> sum(res05$padj <0.05 & res05$log2FoldChange>0, na.rm=T)
[1] 432
```

A generalization of the idea of p value filtering is to weight hypotheses to optimize power. A new Bioconductor package, IHW , is now available that implements the method of Independent Hypothesis Weighting.

`library(IHW)`

`resIHW <- results(dds, filterFun=ihw)`

`sumamry(resIHW)`

`sum(resIHW$padj < 0.1, na.rm=T)`

`metadata(resIHW)$ihwResult`

Exploring and exporting results

- MA-plot

plotMA函数展示log2倍数随着标准化的counts均值改变情况。

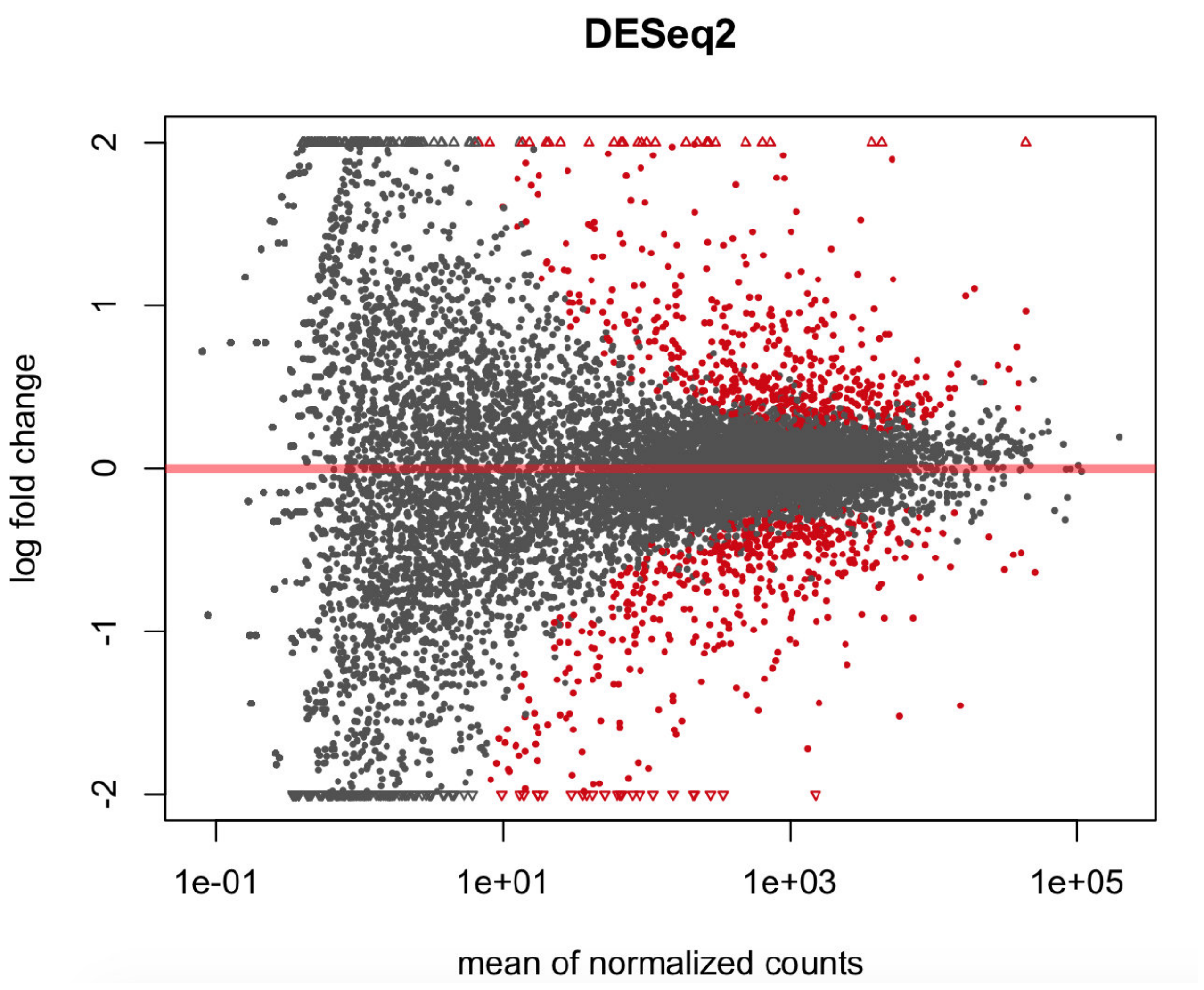
`plotMA(res, main="DESeq2", ylim=c(-2,2))`

完成plotMA图后，使用identify函数通过点击图像点获得对应基因的行坐标，会搜索最近点的xy坐标。

`idx <- identify(res$baseMean, res$log2FoldChange)`

`rownames(res)[idex]`

res[, idx]对应的是baseMean，log2FoldChange对应图上的点。



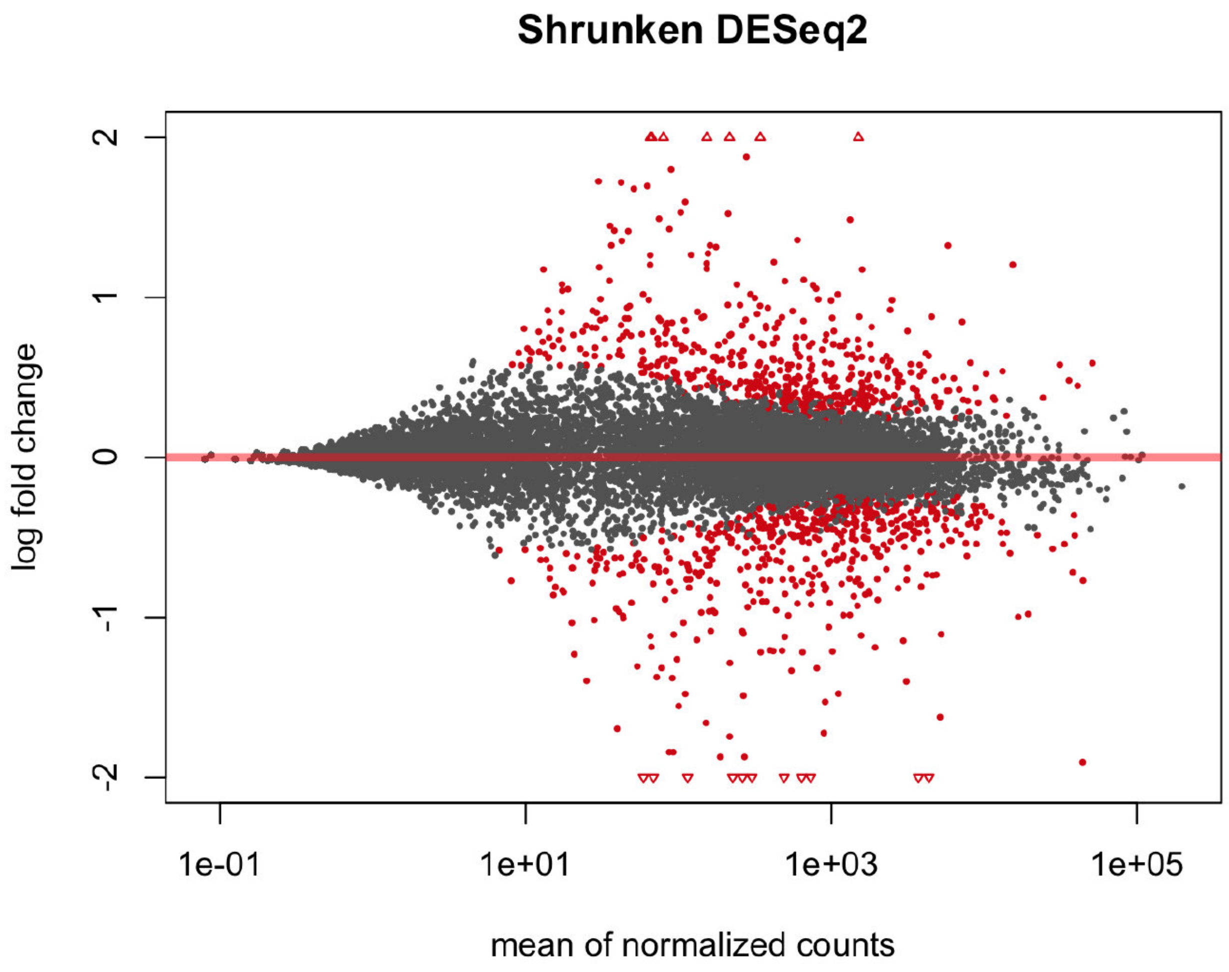
x轴为标准化的counts均值，也就是经过片段校准后的平均counts，y轴对应为log2倍数改变。可以看到没有经过shrinkage的图像显示，counts数越小，dispersion会越大。

目前推荐的流程使用lfcShrink函数生成shrunken MAP评估图。该参数值适用于contrast，指定三个用于比较的参数。**此过程不会改变最终差异结果，仅为绘图使用。**

`res_shrunken <- lfcShrink(dds, contrast=c("condition", "treated", "untreated"))`

或`res_shrunken <- lfcShrink(dds, coef=2)`

`plotMA(res_shrunken, main="shunken DESeq2", ylim=c(-2,2))`



图中红色点代表padj值小于0.1的genes(metadata(res)$alpha, 0.1)，超过窗口范围的点分别用向上或向下的三角点代表。

- Plot counts

使用plotCounts函数检查单个基因的count reads数，plotCounts函数使用测序深度来标准化counts，同时加上1/2以便接下来的log转换。使用参数intgroup对counts进一步做group的细分。

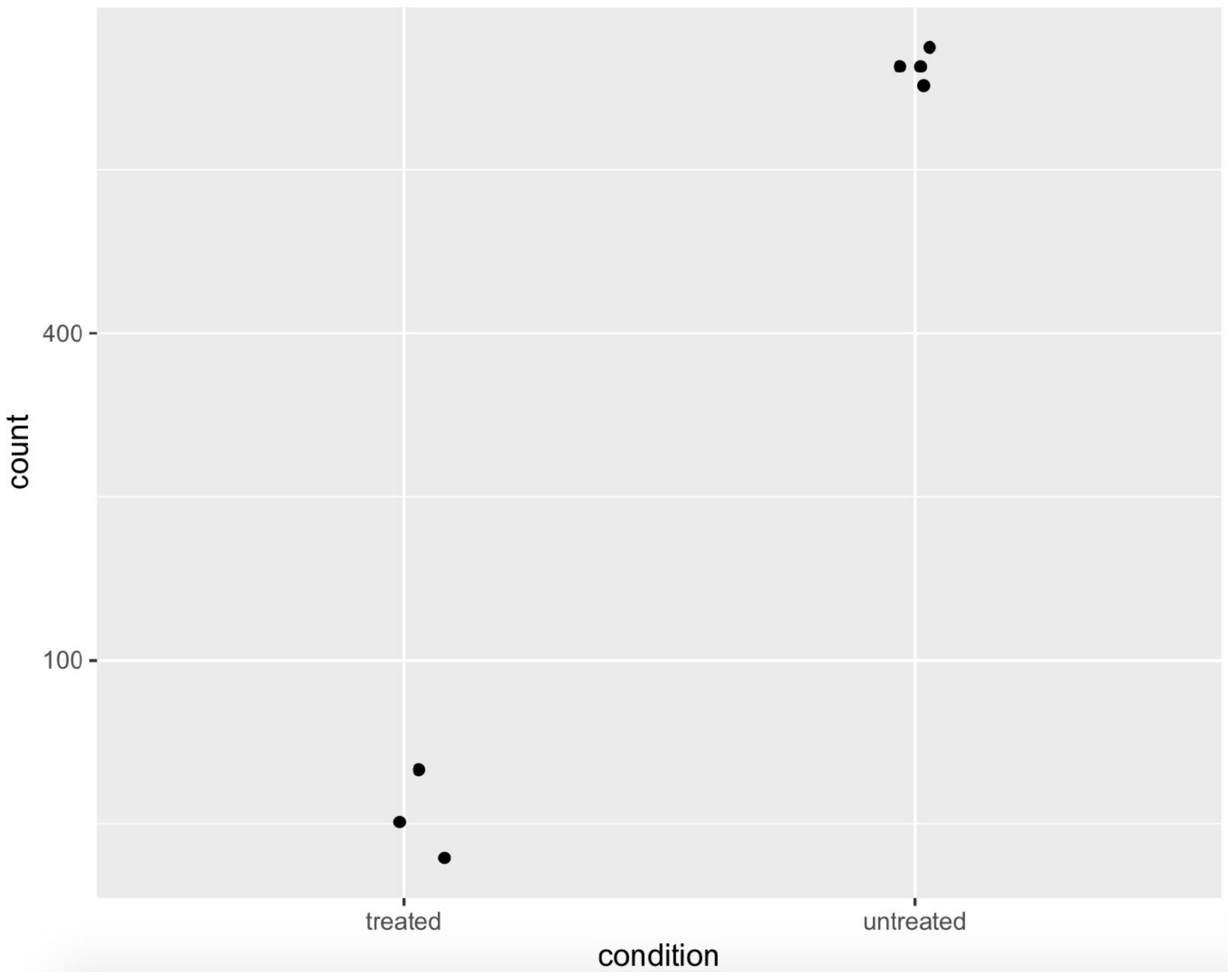
```
plotCounts(dds, gene=which.min(res$padj), intgroup="condition")
```

同样可以选择returnData=TRUE，然后使用ggplot2对返回的数据使用ggplot绘图。

```
d <- plotCounts(dds, gene=which.min(res$padj), intgroup="condition",
returnData=T)
```

```
ggplot(d ,aes(x=condition, y=count)) + geom_point(position=position_jitter(w=0.1,
h=0)) + scale_y_log10(breaks=c(25, 100, 400))
```

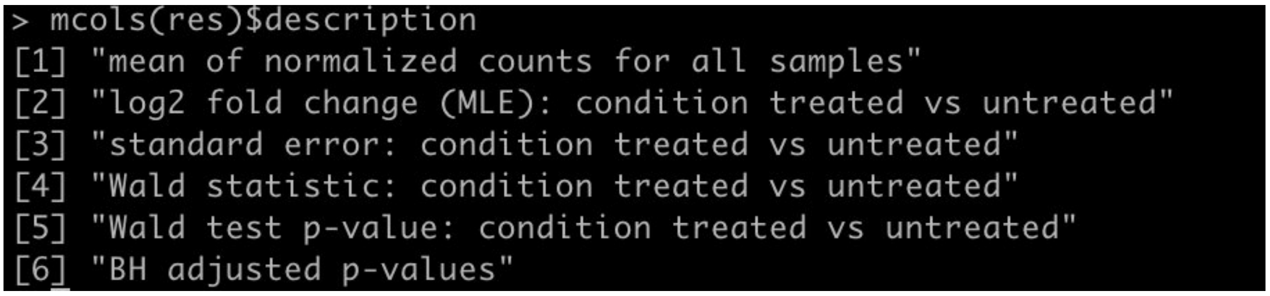
这里的widht，height向jitter同时增加了正负方向的距离，因此途中离散距离是指定值的两倍。



- More information on results columns

关于results返回结果描述。

`mcols(res)$description`



```
> mcols(res)$description
[1] "mean of normalized counts for all samples"
[2] "log2 fold change (MLE): condition treated vs untreated"
[3] "standard error: condition treated vs untreated"
[4] "Wald statistic: condition treated vs untreated"
[5] "Wald test p-value: condition treated vs untreated"
[6] "BH adjusted p-values"
```

`mean(counts(dds)[1,]/colData(dds)$sizeFactor) == res$baseMean`

对于gene而言，log2倍数改变-1，意味着treated vs. untreated时gene改变水平为2^{-1} = 0.5。

对于res中p值的NA, 遵循以下条件：

一行中，所有样本都只有0个计数，则baseMean栏为0，log2FoldChange，p值以及adjusted p值均为NA;

假如一行中某一样本含有极端的count outlier值，那么p值和adjustd p值将会为NA，这些outlier counts的检出依靠Cook's distance；

假如一行被自动独立过滤所删除了，针对低mean normalized count， 那么只有adjusted p值将被设为NA。

- Exporting results to CSV files

`write.csv(as.data.frame(resOrdered), file="condition_treated_results.csv")`

同时可以选择adjusted p值来选择输出文件

`resSig <- subset(resOrdered, padj<0.1)`

- Multi-factor designs

当counts被超过一个因素所影响时，可以设置多因素比对，默认参考水平为字母顺序，下面type的参考水平为paired-end，condition的参考水平为untreated。

`design(ddsMF) <- formula( ~ type + condition)`

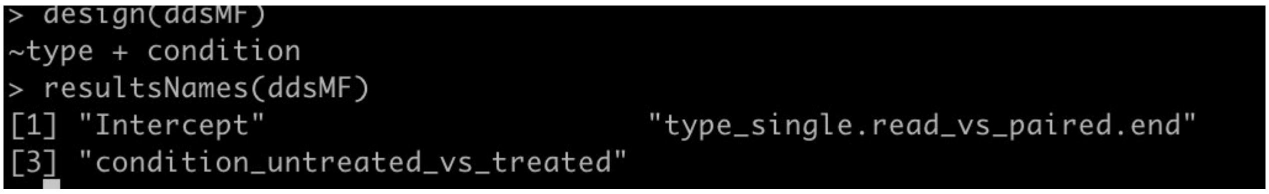
`ddsMF <- DESeq(ddsMF)`

`resMF <- results(ddsMF)`

同时可以在results中使用contrast选项指定比对来获得log2改变以及p，adjused p值。

`resMFType <- results(ddsMF, contrast=c("type", "single-read", "paired-end"))`

或



```
> design(ddsMF)
~type + condition
> resultsNames(ddsMF)
[1] "Intercept"                          "type_single.read_vs_paired.end"
[3] "condition_untreated_vs_treated"
```

`results(ddsMF, name="type_single.read_vs_paired.end")`

Data transformations and visualization

- Count data transformations

为检测差异性表达，对原始counts数据使用离散分布分析。但是为了其他类型下游分析，例如visulization和clustering，counts数据的转换会更有用。

最常见的数据转换操作就是取对数。由于count值有可能为0，因此要加上一个psedocounts来避免0的出现。

$y = \log_2 (n + 1)$ 或者 $y = \log_2 (n + n_0)$

这里n代表count值，n_0代表一个正数常数值。

这里展示了两种途径来提供更加理论的转换以及合理的方式来选择参数n_0，这两种转换，rlog和VST(variance stabilizing transformation)，都是消除对均值变异的依赖，尤其当均值很低时count数据的对数值会出现很高的变异情形。

Variance-stabilizing transformation的目的在于通过简单的函数y=f(x)来使得y值的变异与它们均值无关。

注：vst函数为varianceStabilizingTransformation的更快版本。

rlog和varianceStabilizingTransformation函数都有一个参数blind，默认情况下为TRUE，此时函数在评估离散度时仅依据一个截距值(design formula ~1)。This setting should be used in order to compare samples in a manner wholly unbiased by the information about experimental groups, for example to perform sample QA (quality assurance)。

但是当期待许多或者大部分基因将会根据实验设计而出现大的counts差异时，blind dispersion estimation时不适用的。这种情形，blind dispersion estimation 将会导致偏大的离散度评估，将源于实验设计导致的counts差异判定为噪音，同时会过分shrink转换后的counts值。

通过设置bind为FALSE，会将已经得到的dispersion用于数据转换，如果dispersion不存在，则会根据当前的design formula重新评估dispersion。

`help(rlog): This function transforms the count data to the log2 scale in a way which minimizes differences between samples for rows with small counts, and which normalizes with respect to library size. The rlog transformation produces a similar variance stabilizing effect as ‘varianceStabilizingTransformation’, though ‘rlog’ is more robust in the case when the size factors vary widely.`

`help(varianceStabilizingTransformation): This function calculates a variance stabilizing transformation (VST) from the fitted dispersion-mean relation(s) and then transforms the count data (normalized by division by the size factors or normalization factors), yielding a matrix of values which are now approximately homoskedastic (having constant variance along the range of mean values). The transformation also normalizes with respect to library size. The ‘rlog’ is less sensitive to size factors, which can be an issue when size factors vary widely. These transformations are useful when checking for outliers or as input for machine learning techniques such as clustering or linear discriminant analysis.`

- Extracting transformated values

`rld <- rlog(dds, blind=F)`

`vsd <- varianceStabilizingTransformation(dds, blind=F)`

`vsd.fast <- vst(dds, blind=F)`

无需是否经过DESeq()函数处理，都可以直接使用这些函数。

- Effects of transformations on the variance

`library(vsn)`

`notAllZero <- (rowSums(counts(dds))>0)`

`meanSdPlot(log2(counts(dds,normalized=T)[notAllZero,] + 1))`

`meanSdPlot(assay(rld[notAllZero,]))`

`meanSdPlot(assay(vsd[notAllZero,]))`

注: normalized, logical indicating whether or not to divide the counts by the size factors or normalization factors before returning (normalization factors always preempt size factors)

改图展示了经过直接变换或rlog函数变换后的row standard deviations随row means变化的情况。这里，标准差随均值变化越小，越能减少因均值不同而带来的差异。

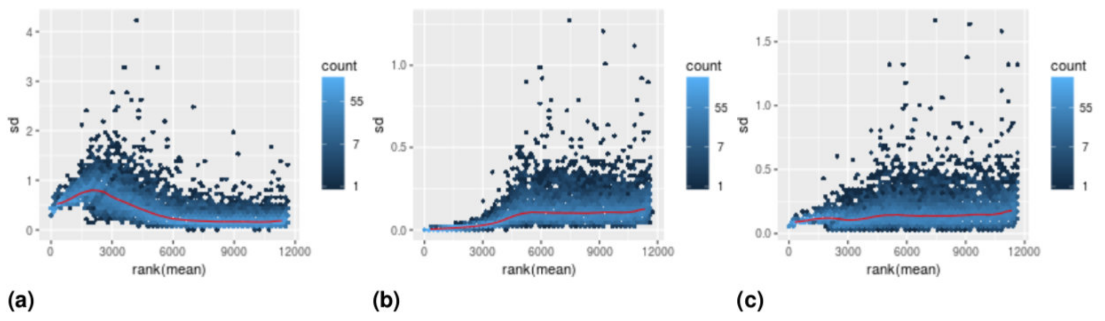


Figure 4: Per-gene standard deviation (taken across samples), against the rank of the mean. **(a)** for the shifted logarithm $\log_2(n+1)$, the regularized log transformation **(b)** and the variance stabilizing transformation **(c)**.

Data quality assessment by sample clustering and visualization

数据质量评估和控制(去除不够好的数据)是任何数据分析的必要步骤，这些步骤应该在数据分析的早期就要完成。

- Heatmap of the count matrix

热图为使用颜色图像展示矩阵数据的个体数值，这里展示经过转换后数据的热图。

`library(pheatmap)`

`select <- order(rowMeans(counts(dds, normalized=T)), decreasing=T)[1:20]`

`nt <- normTransform(dds)` # defaults to log2(x+1)

`log2.norm.counts <- assay(nt)[select,]`

`df <- as.data.frame(colData(dds)[, c("condition","type")]])`

`pheatmap(log2.norm.counts, cluster_rows=F, show_rownames=F, cluster_cols=F, annotation_col=df)`

`pheatmap(assay(rld)[selece, ], cluster_rows=F, show_rownames=F, cluster_cols=F, annotation_col=df)`

`pheatmap(assay(vsd)[select, ], cluster_rows=F, show_rownames=F, cluster_cols=F, annotation_col=df)`

这里的normTransform函数简单创建了DESeqTransform对象：normTransform(object, f=log2, pc=1)

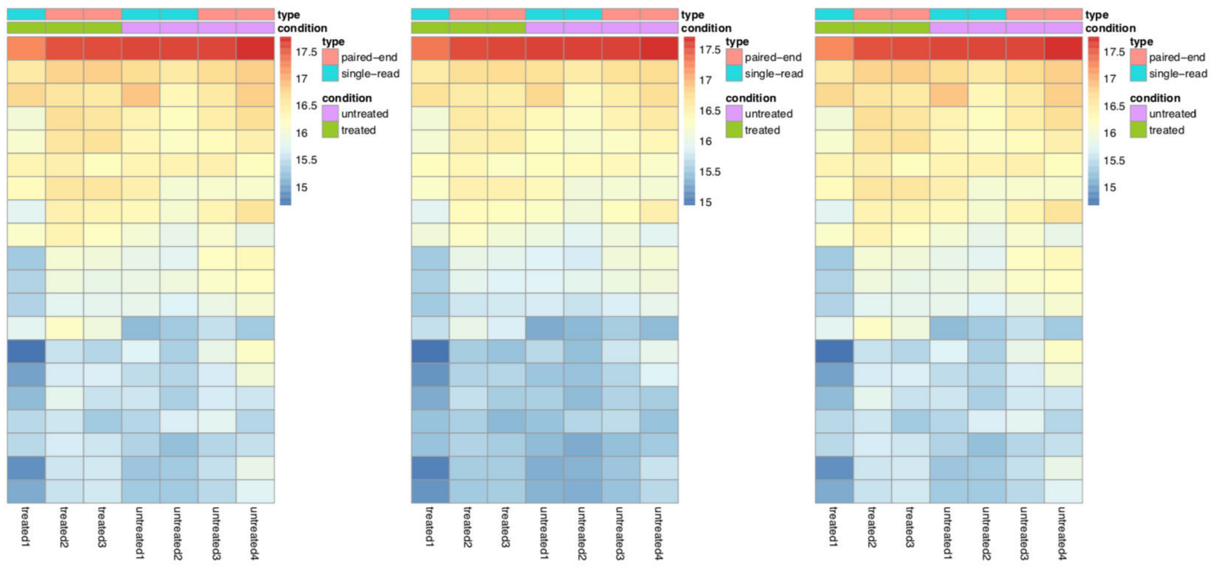


Figure 5: **Heatmaps showing the expression data of the 20 most highly expressed genes.** The data is of log2 normalized counts (left), from regularized log transformation (center) and from variance stabilizing transformation (right).

- Heatmap of the sample-to-sample distance

另一个转换后数据的用途是样本的clustering分析。首先使用dist函数处理转换后counts matrix获得样本与样本之间的距离。

`sampleDists <- dist(t(assay(rld)))`

dist函数使用特殊的方法计算返回data matrix的行之间距离: dist(x, method="euclidean", diag=F, upper=F, p=2)

也可以使用plot(hclust(dist(t(assay(rld)))))来获得基于距离的树状图。

`library(RcolorBrewer)`

`sampleDistMatrix <- as.matrix(sampleDists)`

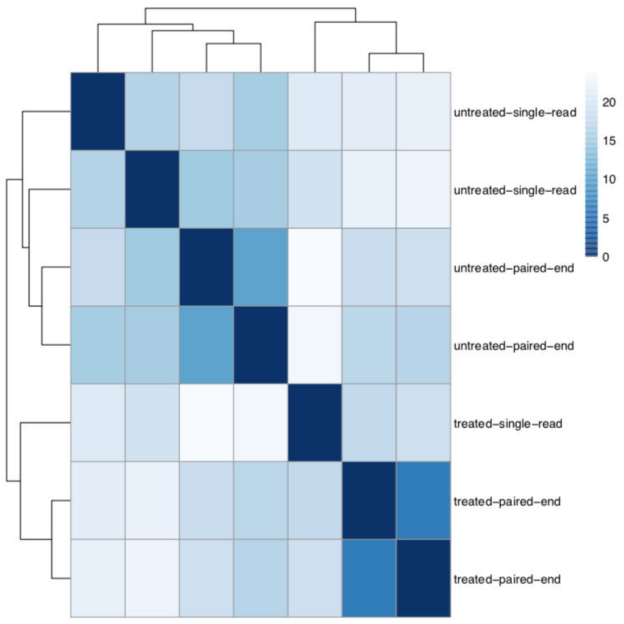
等同于dist(t(assay(rld)), upper=T, diag=T)

`rownames(sampleDistMatrix) <- paste(rld$condition, rld$type, sep="-")`

`colnames(sampleDistMatrix) <- NULL`

`colors <- colorRampPalette(rev(brewer,pal(9, "Blues")))(255)`

```
pheatmap(sampleDistMatrix, clustering_distance_rows=samleDists,
clustering_distance_cols=sampleDists, col=colors)
```



(古希腊数学家)欧几里得的,欧几里得几何学的

Figure 6: Sample-to-sample distances. Heatmap showing the Euclidean distances between the samples as calculated from the regularized log transformation.

- Principal component plot of the samples

与距离矩阵相关的图就是样本的PCA图。

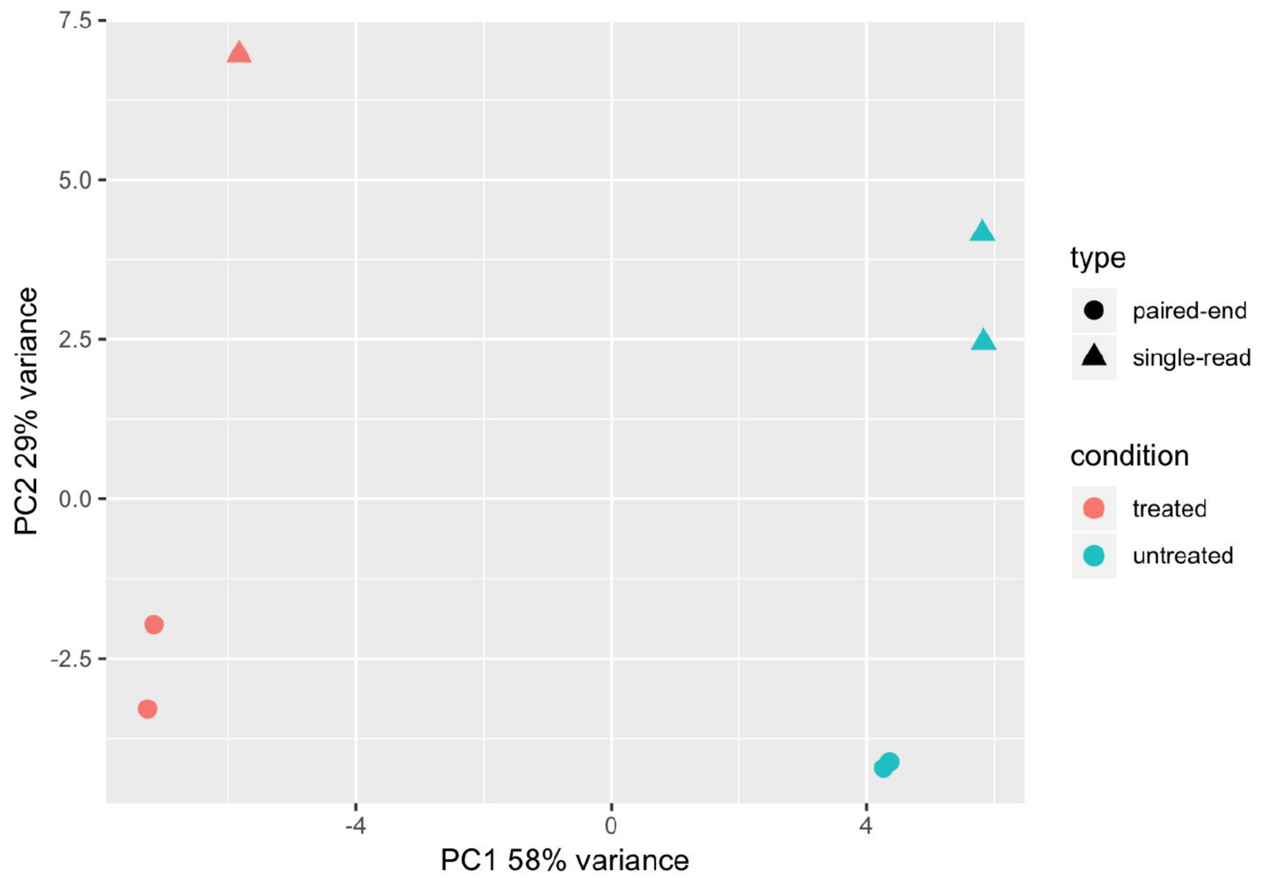
```
plotPCA(rld, intgroup=c("condition","type"))
```

同时也可以使用ggplot自主绘制PCA图。

```
data <- plotPCA(rld, intgroup=c("condition", "type"), returnData=T)
```

```
percentVar <- round(100 * attr(data, "percentVar"))
```

```
ggplot(data, aes(PC1, PC2, color=condition, shape=type)) + geom_point(size=3) +
xlab(paste0("PC1: ", percentVar[1], "% variance")) + ylab(paste0("PC2: ",
percentVar[2], "% variance")) + coord_fixed()
```



Variations to the standard workflow

- Wald test individual steps

DESeq函数依次运行一些步骤:

评估片段大小因子： dds <- estimateSizeFactors(dds)

评估离散度： dds <- estimateDispersions(dds)

负二项式glm检验： dds <- nbinomWaldTest(dds)

posterior log2 fold change estimation is not default setting for nbinomWaldTest

dds_shrunken <- lfcShrink(dds) : Adds shrunken log2 fold changes (LFC) and SE to a results table from 'DESeq' run without LFC shrinkage.

- Contrasts

constrast是评估log2倍数改变的线性组合，能用来检测组间差异是否为0。

`results(dds ,constrast=c("condition", "treated", "untreated"))`

或`results(dds, name="condition_untreated_vs_treated")`

`results(dds, name=resultsNames(dds)[2])`

- Interactions

交互项同样可以添加到design formula中。

`dds$group <- factor(paste0(dds$condition, dds$type))`

`design(dds) <- ~group`

`dds <- DESeq(dds)`

`resultsNames(dds)`

[1] "Intercept" [2] "group_treatedsingle.read_vs_treatedpaired.end" [3] "group_untreatedpaired.end_vs_treatedpaired.end" [4] "group_untreatedsingle.read_vs_treatedpaired.end"

`results(dds, contrast=c("group", "treatedsingle-read","treatedpaired-end"))`

- Likelihood ration test

DESeq2提供两种不同的假设检验，Wald test，使用log2倍数改变的standard error来检测是否为0，或相似性比率检测(LRT)。 LRT针对counts检测两个模型，full model和reduced model(some of the terms of the full model are removed)。该检测来判断使用full model中extra terms有所带来的相似性是否超过了预期，判断这些extra terms带来的影响是否真的为0(the test determines of the increased likelihood of the data using the extra terms in the full model is more than expected if those extra terms are truly zero)。

在DESeq函数中选择参数test='LRT'，同时提供reduced design formula。

`dds <- DESeq(dds, test="LRT", reduced=~condition)`

`res <- results(res)`

log2 fold change(MLE): condition untreated vs treated

LRT p-value : '~ type + condition' vs '~condition'

`dds <- DESeq(dds, test="LRT", reduced=~type)`

`res <- results(dds)`

log2 fold change (MLE): condition untreated vs treated

LRT p-value: '~type + condition' vs '~type'

reduce 这里是指定一个formula。此时：

stat LRT statistic: '~ condition + type' vs '~ type' pvalue LRT p-value: '~ condition + type' vs '~ type' padj BH adjusted p-values

而Walt Test:

stat Wald statistic: condition untreated vs treated pvalue Wald test p-value: condition untreated vs treated padj BH adjusted p-values

- Approach to count outliers

RNA测序数据常包含非常大大counts，明显看来与实验设计无关的结果，而这些则可悲考虑为outliers。针对这些counts，DESeq函数针对每个样本的每一个基因计算Cook's distance，Cook's distance is a measure of how much a single sample is influencing the fitted coefficients for a gene, and a large value of Cook's distance is intented to indicate an outlier count。距离越大越可能为outlier点。该距离矩阵为assays(dds)["cooks"]。(how much the fitted coefficients would change if an individual sample were removed)

results函数自动标记Cook's distance高于阈值同时包含3个或3个以上重复的基因，该基因的p值和adjusted p值均被设为NA。可以通过设置cooksCutoff=F来关闭该过程：results(dds, cooksCutoff=F)；而一个样本的拷贝数为7个或7个以上时， DESeq function will automatically replace counts with large Cook's distance with the trimmed mean over all samples, scaled up by the size factor or normalizetion factor for that sample。

假如在summary(res)中发现很多outlier值时(many hundreds or thousands)，这时就要考虑是不是一个或者几个样本由于低质量该被舍弃了。

`par(mar=c(8,5,2,2))`

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

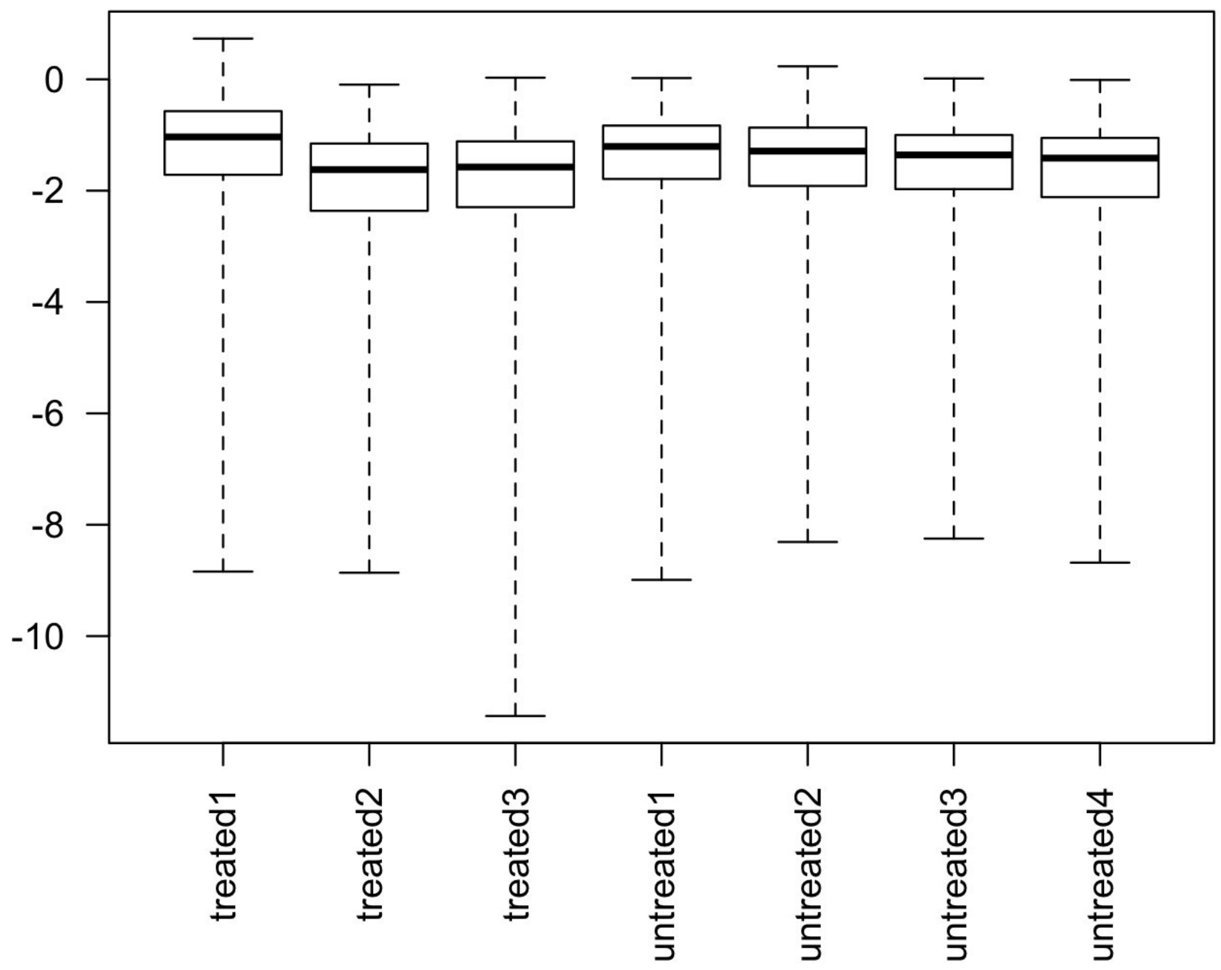
或

`test_cooks <- data.frame(assays(dds)[["cooks"]])`

`melt_cooks <- melt(test_cooks)`

`ggplot(melt_cooks, aes(variable, log10(value), color=variable))+geom_boxplot()`

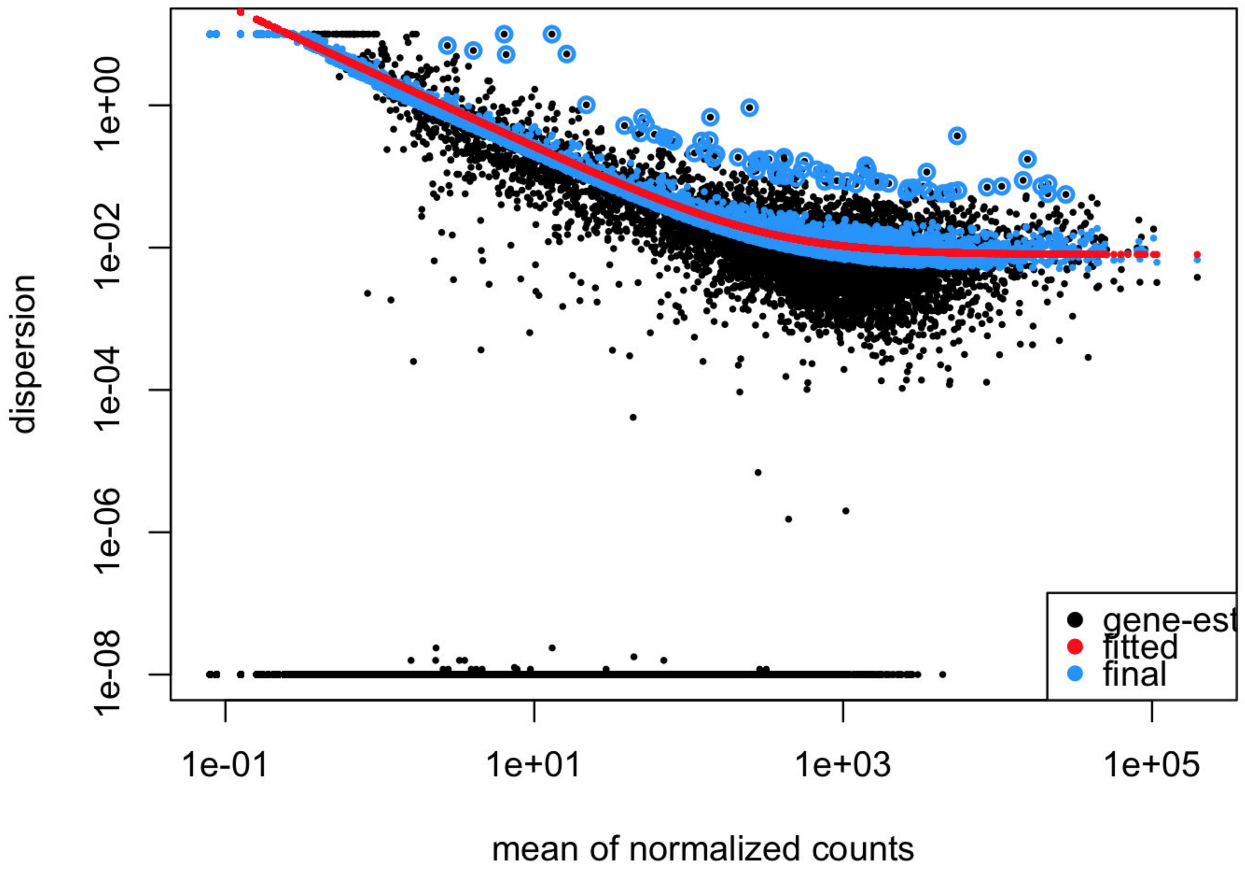
`ggplot(melt_cooks, aes(variable, value, color=variable))+geom_boxplot() + scale_y_log10()`



- Dispersion plot and fitting alternatives

dispersions estimates是一个很有用的诊断图

`dds <- makeExampleDESeqDataSet()` `dds <- estimateSizeFactors(dds)` `dds <- estimateDispersions(dds)` `plotDispEsts(dds)`



改图x轴为标准化的counts均值，y轴为离散度：The dispersion estimate plot shows the gene-wise estimates (black), the fitted values (red), and the final maximum a posteriori estimates used in testing (blue)。

- Independent filtering of results

采用genefilter包中的filtered_p函数过滤，且所有其参数都可在results中使用

- Tests of log2 fold change above or below a threshold

提供阈值构建Wald tests 显著性检测。results 函数包含两个参数选型可用于设置该检测：lfcThreshold，指定非负数阈值；altHypothesis，指定检测类型。

altHypothesis可选择以下4类值，β 值表示name参数指定的log2 fold change。

greaterAbs : β > lfcThreshold tests are two-tailed

lessAbs : β < lfcThreshold p values are the maximum of the upper and lower tests

greater : β > lfcThreshold

less : β < lfcThreshold

这里altHypothesis="lessAbs",要求在运行DESeq时选择参数betaPrior=F。以上四个无非就是指定了log2倍数改变值，然后求对应了p值。

`ddsNoPrior <- DESeq(dds, betaPrior=F)`

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

`yl <- c(-2.5, 2.5)`

`resGA <- results(dds, lfcThreshold=0.5, altHypothesis="greaterAbs")` #大于或小于正负0.5，同时满足p值小于alpha(标红)

`resLA <- results(ddsNoPrior, lfcThreshold=0.5, altHypothesis="lessAbs")` #处于正负0.5之间，同时满足p值小于alpha(标红)

`resG <- results(dds, lfcThreshold=0.5, altHypothesis="greater")`

`resL <- results(dds, lfcThreshold=0.5, altHypothesis="less")`

`plotMA(resGA, ylim=yl)`

`abline(h=c(-0.5,0.5), col="dodgerblue", lwd=2)`

`plotMA(resLA, ylim=yl)`

`abline(h=c(-0.5,0.5), col="dodgerblue", lwd=2)`

`plotMA(resG, ylim=yl)`

`abline(h=0.5, col="dodgerblue", lwd=2)`

`plotMA(resL, ylim=yl)`

`abline(h=-0.5, col="dodgerblue", lwd=2)`

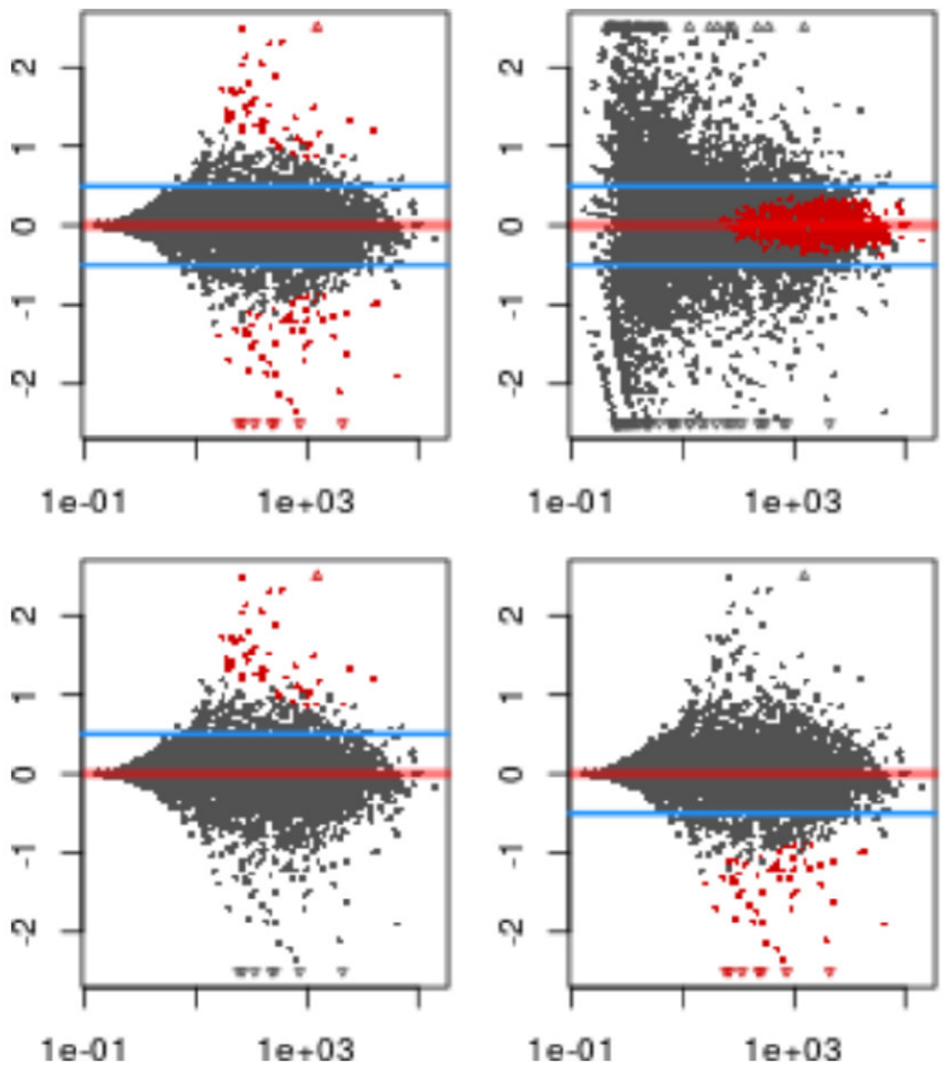


Figure 13: MA-plots of tests of log2 fold change with respect to a threshold value. Going left to right across rows, the tests are for `altHypothesis="greaterAbs"`, `"lessAbs"`, `"greater"`, and `"less"`.

- Access to all caclulated values

`mcols(dds)`

`substr(names(mcols(dds)),1,10)`

`mcols(mcols(dds))`

`head(assays(dds)[["cooks"]])`

`head(dispersions(dds))`

`sizeFactors(dds)`

`head(coef(dds))`

`dispersionFunction(dds)`

- Sample-/gene-denpendent normalization factors

针对gene-dependent dependencies， 不同样本之间会存在差异。例如来自不同实验室或不同处理时间的样本间的GC-content bias 或 length bias。

DEASeq and cqn packages can help correct the GC and length biases， they would help create matrices which can be used by DESeq2。

- Model matrix not full rank

参考《Expressing_design_formula_in_R_for_RNA_seq》

- Count outlier detection

We use Cook's distance, which is a measure of how much the fitted coefficients would change if an individual sample were removed.

`w <- res$stat`

`maxCooks <- apply(assays(dds)[["cooks"]],1,max)`

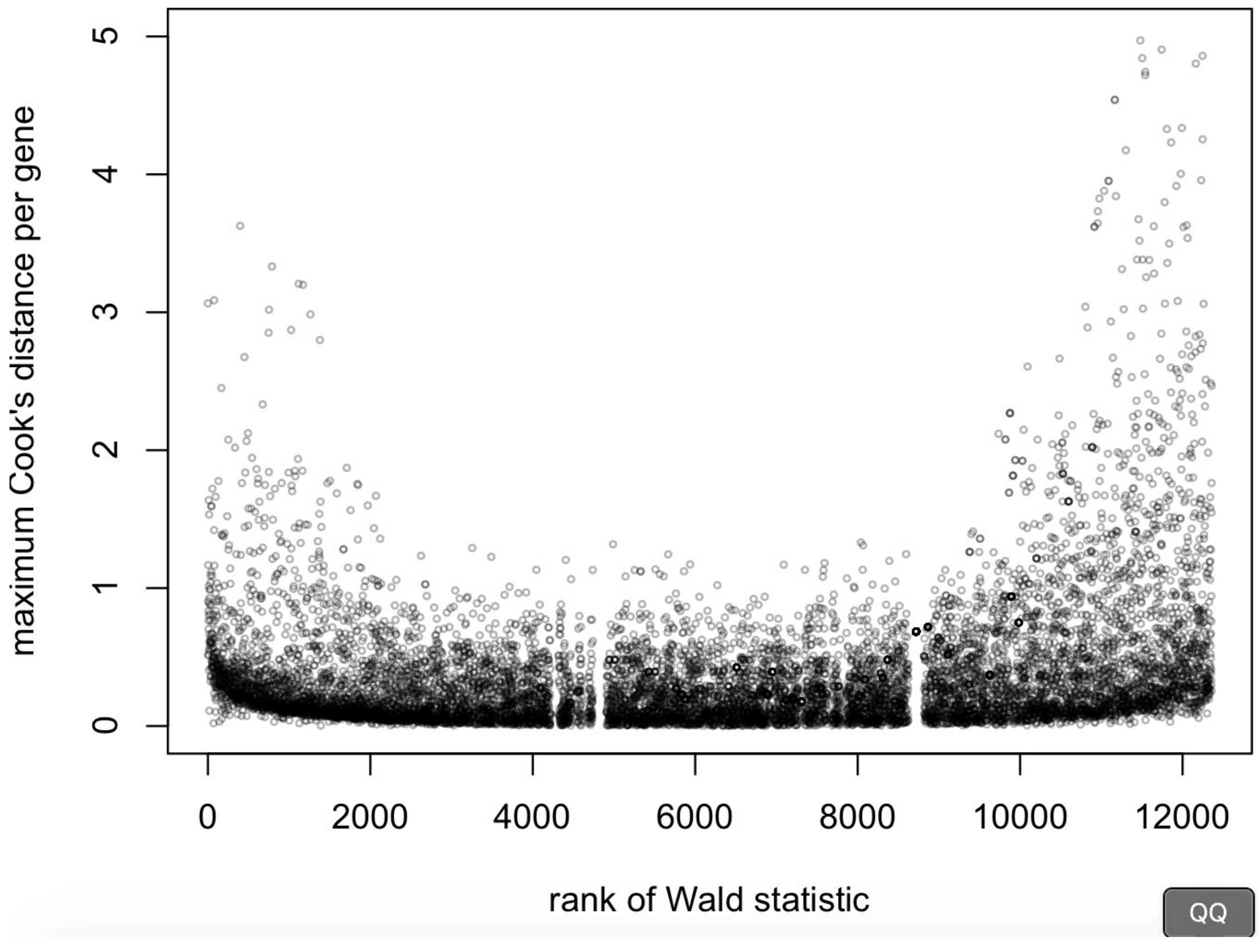
`idx <- ! is.na(w)`

`plot(rank(w[idx]), maxCooks[idx], xlab="rank of Wald statistic", ylab="maximum Cook's distance per gene", ylim=c(0.5), cex=0.4, col=rgb(0,0,0.3))`

`m <- ncol(dds)`

`p <- 3`

`abline(h=qf(0.99,p,m-p))`



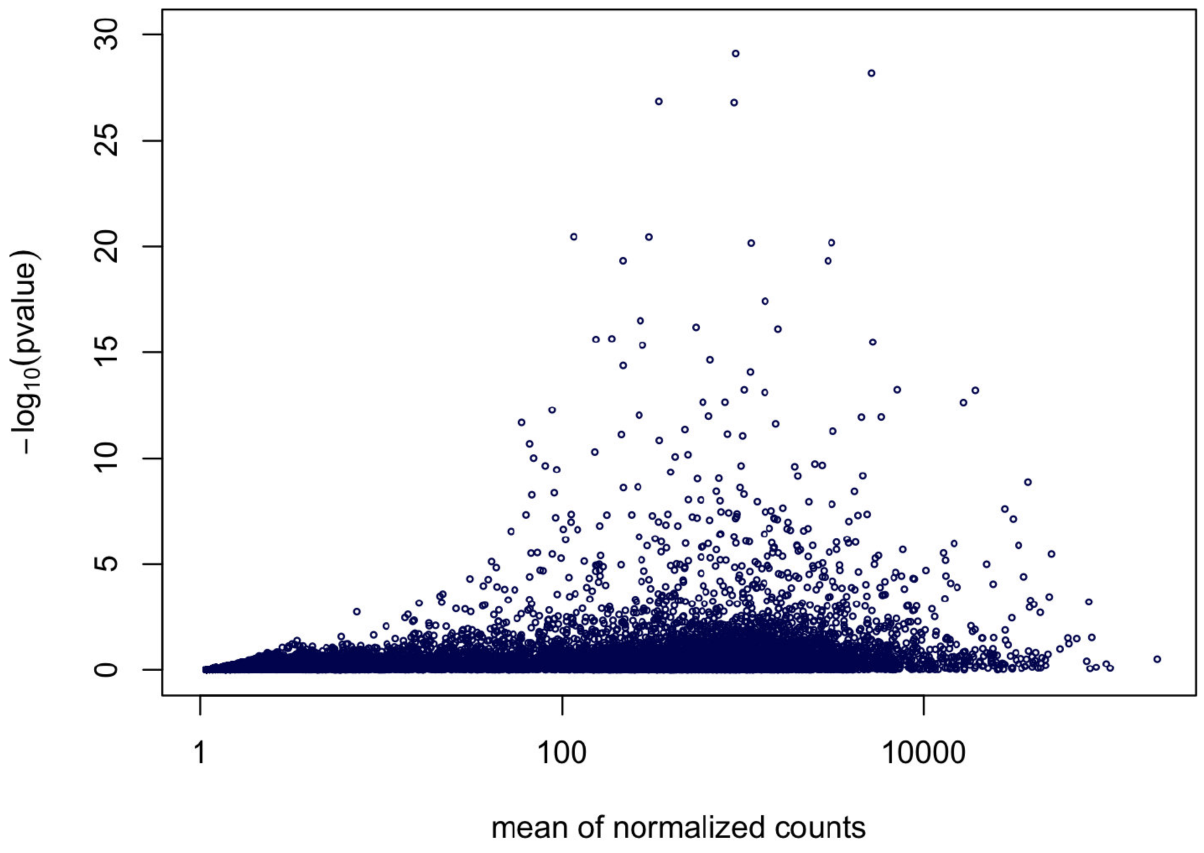
x轴为wald statistics for condition(经过rank排序), y轴为maximum cook's distance。

- Filtering cirteria

Independent filtering 是不依靠统计检验，过滤掉哪些没有或具有很小显著性意义的数据。

一个简单的过滤标准就是不考虑生物状态下对平均标准化后的counts的均值的过滤。由于低counts的genes不太可能会有显著性的差异。通过绘制x轴为平均counts，y轴为log10 p值，可得：

```
plot(res$baseMean+1, -log(res$pvalue), log="x", xlab="mean of normalized counts",
ylab=expression(-log[10](pvalue)), ylim=c(0,30), cex=0.4, col=rgb((0,0,0.3))
```



整体而言，p值不会随着counts数目变化而发生变化。

- Histogram of p values for all tests

```
use <- res$baseMean > metadata(res)$filterThershold
```

```
h1 <- hist(res$pvalue[!use], breaks=0:50/50, plot=F)
```

```
h2 <- hist(res$pvalue[use], breaks=0:50/50, plot=F)
```

colori <- c('do not pass' = "khaki",'pass' = "powderblue")

```
barplot(height=rbind(h1$counts, h2$counts), beside=F, col=colori, space=0,
main="", ylab="frequency")
```

```
text(x=c(0, length(h1$counts)), y=0, label=paste(c(0,1)),adj=c(0.5,1.7), xpd=NA)
```

```
legend("topright", fill=rev(colori), legend=rev(names(colori)))
```

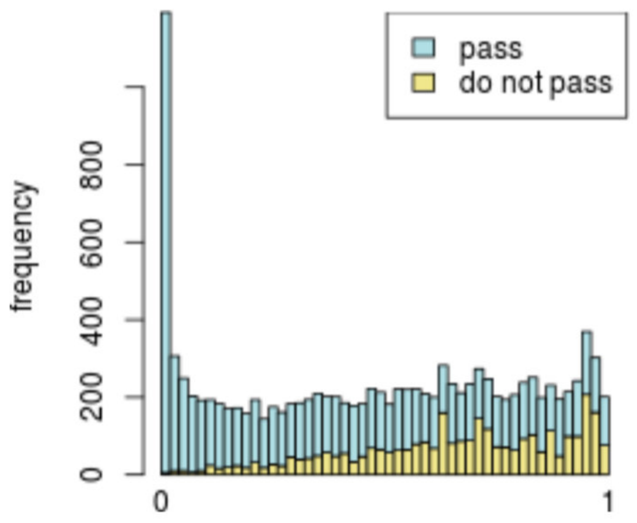


Figure 16: Histogram of p values for all tests. The area shaded in blue indicates the subset of those that pass the filtering, the area in khaki those that do not pass.

- Get unfiltered DESeq results

without removal of independent filtering

`dds <- DESeq(dds, minReplicatesForReplace=Inf)`

`res <- results(dds, cooksCutoff=F, independentFiltering=F)`

minReplicatesForReplace: the minimum number of replicates required in order to use ‘replaceOutliers’ on a sample. If there are samples with so many replicates, the model will be refit after these replacing outliers, flagged by Cook's distance. Set to ‘Inf’ in order to never replace outliers.

这是只有当所有counts都为0是，p values才设为NA。