How to use DESeq2 to analyse RNAseq data

Posted on **February 17, 2014**

News: *My colleagues at NZGL have developed an open source R based GUI for generating plots using cuffdiff data. One for deseq2 will be available soon! Feel free to check it out and get back to us with any suggestions. Only requires R-studio.*

There is only one thing better than DESeq and thats DESeq2! The updated version is out and I'm keen to give it a whirl. Like with my old DESeq post, once again I am really just following the excellent DESeq2 manual, thanks again to the authors for the great documentation!

Just a quick warning that I haven't tested this workflow extensively, let me know if things don't work. Also, DESeq2 is new so some of the function might have changed, so if you have problems makes sure you check what version you are using versus what I used (see sessionInfo below).

The files I used for this are found here if you want to give them a go or follow along. I've already got a blow by blow run through on how to do use DESeq and much of that applies to the new package, so here I'll just concentrate on some DESeq2 specific stuff as well as all the graphs. I've included some side by side comparisons between DESeq and DESeq2.

Installing is easy:

```r
source('http://bioconductor.org/biocLite.R')
biocLite('DESeq2')
```

One important change is that now you can directly create the count table using raw HT-Seq-count output, and I'll show you how to do that below. Remember HT-Seq-count will create a single file for each replicate of a condition (based on an SAM alignment file), so in my case with two conditions (control and treatment) and 3 reps each, that makes a total of 6 files. I called these files treated1.txt, treated2.txt, treated3.txt, untreated1 untreated2, untreated3 so that i can use grep to import them (explained below). Previously i would use a script to merge them all together, now DESeq2 allows you to import these files directory.

Below I set the directory where the files are located; use grep to catch all these files based on the string match "treated" that they all share (be carefully it doesn't catch anything else), this is stored in sampleFiles. If you like you could just directly specify the files using "sampleFiles<-c("treated1.txt",..etc.."untreated3.txt"). Importantly we need to setup the sampleConditions, with the same order as that found for the file names in sampleFiles (so it knows what each files represents). Finally we make a dataframe that becomes a deseq table. Note my "#" lines are just nonfunctional comment lines that I am using to print out the output from the screen!

```r
library('DESeq2')
directory<-'/home/dwheeler/Desktop/BLOG/Dec_post2'
#use grep to search for the 'treated' part of filename to collect file
sampleFiles<-grep('treated',list.files(directory),value=TRUE)
```

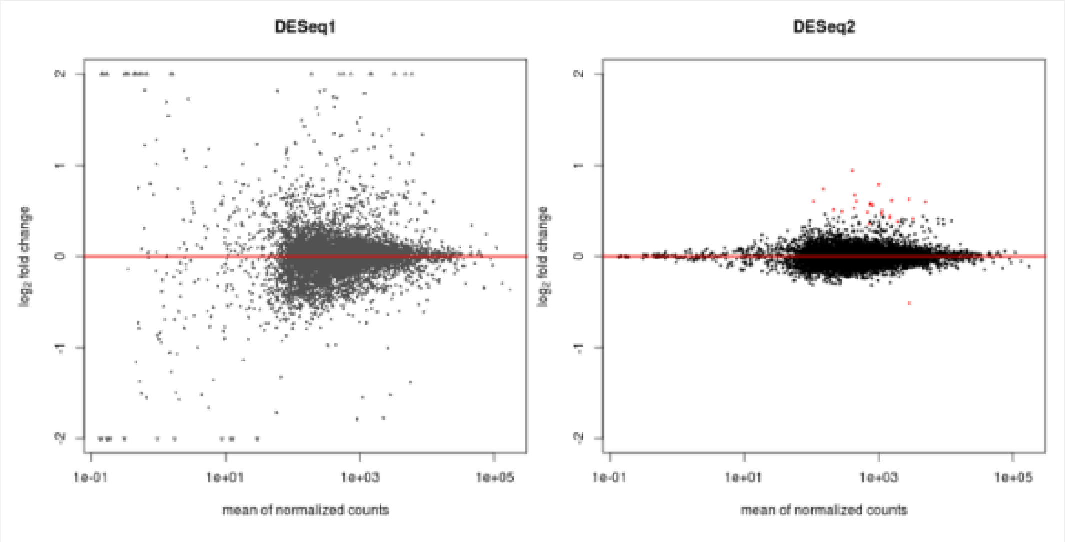
```
# sampleFiles
#[1] 'treated1.txt'   'treated2.txt' 'treated3.txt'  'untreated1.txt'
#[5] 'untreated2.txt' 'untreated3.txt'

sampleCondition<-c('treated','treated','treated','untreated','untreated',
sampleTable<-data.frame(sampleName=sampleFiles, fileName=sampleFiles, c
####
#sampleTable
#     sampleName       fileName condition
#1   treated1.txt   treated1.txt   treated
#2   treated2.txt   treated2.txt   treated
#3   treated3.txt   treated3.txt   treated
#4 untreated1.txt untreated1.txt untreated
#5 untreated2.txt untreated2.txt untreated
#6 untreated3.txt untreated3.txt untreated
######

ddsHTSeq<-DESeqDataSetFromHTSeqCount(sampleTable=sampleTable, directory=
#####
#ddsHTSeq
#class: DESeqDataSet
#dim: 7921 6
#exptData(0):
#assays(1): counts
#rownames(7921): seq_1 seq_2 ... seq_7920 seq_7921
#rowData metadata column names(0):
#colnames(6): treated1.txt treated2.txt ... untreated2.txt
#  untreated3.txt
#colData names(1): condition
#######
colData(ddsHTSeq)$condition<-factor(colData(ddsHTSeq)$condition, levels=
```

The levels in colData are important because they are used in the log calculations; it makes sense to set untreated or control first so that the direction of the logs fold changes doesn't confuse everyone (typically we do comparisons to the control)! Now for the guts of the DEseq2 analysis.

```
dds<-DESeq(ddsHTSeq)
res<-results(dds)
res<-res[order(res$padj),]
head(res)
#DataFrame with 6 rows and 6 columns
#          baseMean log2FoldChange      lfcSE      stat       pvalue
#
#seq_3146  997.5419      0.7894523 0.08297687  9.514125 1.832488e-21
#seq_1802  746.3972      0.5685789 0.08533961  6.662544 2.691282e-11
#seq_2146  406.1395      0.9424543 0.14108613  6.679993 2.389544e-11
#seq_7548  466.5453      0.6036683 0.10178158  5.931017 3.010637e-09
#seq_3240 1569.6556      0.6132326 0.11145966  5.501835 3.758596e-08
#seq_958   149.6504      0.7398193 0.14154162  5.226868 1.724055e-07
#                 padj
#
#seq_3146 1.299050e-17
#seq_1802 6.359498e-08
#seq_2146 6.359498e-08
#seq_7548 5.335601e-06
#seq_3240 5.328937e-05
#seq_958  2.036971e-04
```

Looking good, time for some plots. BTW I'm using the same dataset I used for the original DESeq blog post (links to data and that blog at top of page).

```
plotMA(dds,ylim=c(-2,2),main='DESeq2')
dev.copy(png,'deseq2_MAplot.png')
dev.off()
```



— MAPlot of DESeq1 (left) and DESeq2 (right) for the same data

As expected for this dataset there are not many differentially expressed genes (red). There certainly is a difference in the level of scatter with this dataset using DESeq and DESeq2. Also note that there is good reduction in scatter for low count reads (left hand side of the graph) in DESeq2 versus the original version. DESeq tends to be a conservative approach, I like that, and with that in mind the update uses a test called cooks distance to remove outliers from the analysis. Cooks distance looks to see how much each sample contributes to a genes overall value fold change, with samples that cause extreme effects removed. To be specific, the gene will not be analysed for differential expression if one of its samples is considered an outlier The idea being here that we want to see only DE genes that show a consistent pattern. The draw back of this approach is that there is a loss of power, as some genes that are truly DE will be removed before the statistical tests are performed.

We can save the table, and also print out some information on what the columns mean.

```
mcols(res,use.names=TRUE)
#DataFrame with 6 rows and 2 columns
#                       type

#baseMean       intermediate
#log2FoldChange      results
#lfcSE               results
#stat                results
#pvalue              results
#padj                results
                                                       #description

#baseMean                         the base mean over all rows
#log2FoldChange log2 fold change (MAP): condition treated vs untreated
#lfcSE           standard error: condition treated vs untreated
#stat            Wald statistic: condition treated vs untreated
#pvalue          Wald test p-value: condition treated vs untreated
#padj            BH adjusted p-values
#write the table to a csv file
write.csv(as.data.frame(res),file='sim_condition_treated_results_deseq2.
```

BTW take this with a pinch of salt because its only a simple sample dataset, but the difference in gene counts is that deseq only found a single differentially expressed gene (at padj 0.1), whilst deseq2 called this same gene plus 23 others. Also, reducing the cut-off multiple testing correction to 0.05 only removes 3 genes from the list with DESeq2.

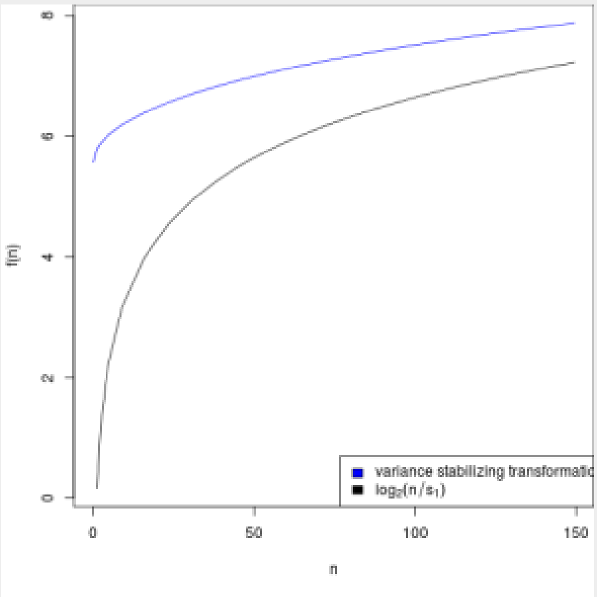
Now we want to transform the raw discretely distributed counts so that we can do clustering.

```
rld<- rlogTransformation(dds, blind=TRUE)
vsd<-varianceStabilizingTransformation(dds, blind=TRUE)
```

Here we choose blind so that the initial conditions setting does not influence the outcome, ie we want to see if the conditions cluster based purely on the individual datasets, in an unbiased way According to the documentation, the rlogTransformation method that converts counts to log2 values is apparently better than the old varienceStabilisation method when the data size factors vary by large amounts.

The code and plot below shows the [nice] effect of the transformation.

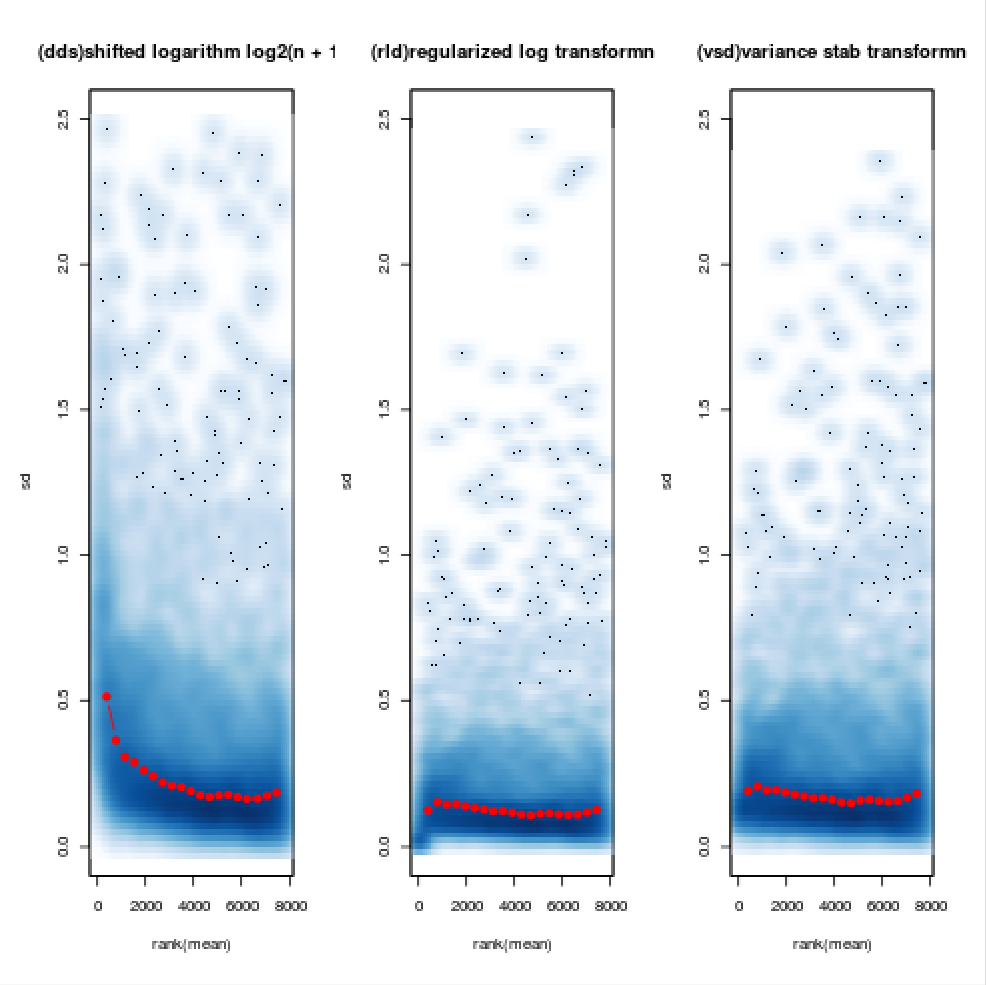
```
par(mai=ifelse(1:4 <= 2, par('mai'), 0))
px     <- counts(dds)[,1] / sizeFactors(dds)[1]
ord    <- order(px)
ord    <- ord[px[ord]<150]
ord    <- ord[seq(1, length(ord), length=50)]
last   <- ord[length(ord)]
vstcol <- c('blue', 'black')
matplot(px[ord], cbind(assay(vsd)[, 1], log2(px))[ord, ], type=l, lty=1,
legend('bottomright', legend = c(expression('variance stabilizing transf
dev.copy(png,'DESeq2_VST_and_log2.png')
```



— Graph showing variance stabilizing transformation for sample 1 (blue) and of the transformation f (n) = log2 (n/s1) (black)

The x axis is the square root of variance over the mean for all samples, so this will naturally included variance due to the treatment. The goal here is to flattern the curve so that there is consistent variance across the read counts, and that is what we got.

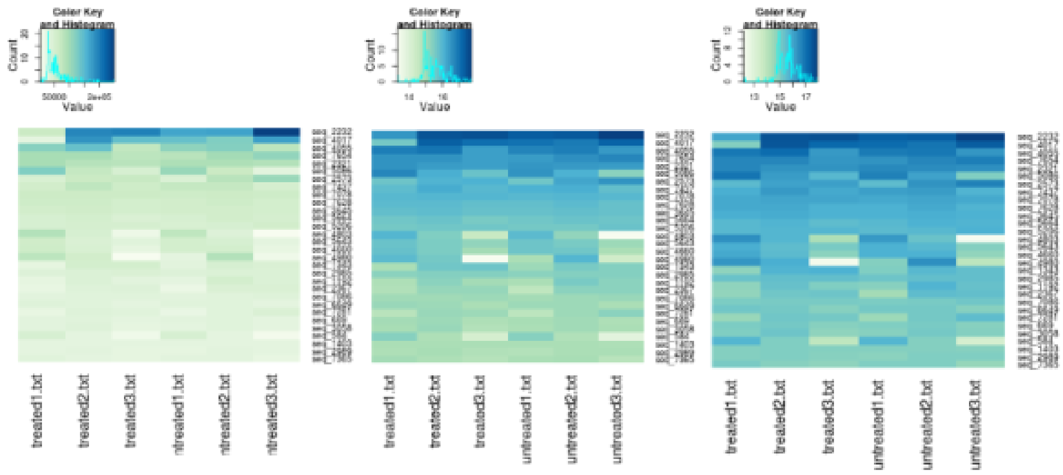
```
library('vsn')
par(mfrow=c(1,3))
notAllZero <- (rowSums(counts(dds))>0)
meanSdPlot(log2(counts(dds,normalized=TRUE)[notAllZero,] + 1), ylim =c(0,
meanSdPlot(assay(rld[notAllZero,]), ylim =c(0,2.5))
meanSdPlot(assay(vsd[notAllZero,]), ylim =c(0,2.5))
```



This interesting plot shows the standard deviation across all samples against the mean counts using three different methods of transformation. With this data you can see that the shifted logarithm method (left) seems to do pretty badly at for low count genes, with both regularized log (center) and DESeqs variance stabilisation (right) doing a much better job across the entire dynamic range of counts.

For some reason, everyone loves a good heat map!

```
library('RColorBrewer')
library('gplots')
select <- order(rowMeans(counts(dds,normalized=TRUE)),decreasing=TRUE)[1:
hmcol<- colorRampPalette(brewer.pal(9, 'GnBu'))(100)
heatmap.2(counts(dds,normalized=TRUE)[select,], col = hmcol,
Rowv = FALSE, Colv = FALSE, scale='none',
dendrogram='none', trace='none', margin=c(10,6))
dev.copy(png,'DESeq2_heatmap1')
dev.off()
heatmap.2(assay(rld)[select,], col = hmcol,
Rowv = FALSE, Colv = FALSE, scale='none',
dendrogram='none', trace='none', margin=c(10, 6))
dev.copy(png,'DESeq2_heatmap2')
dev.off()
heatmap.2(assay(vsd)[select,], col = hmcol,
Rowv = FALSE, Colv = FALSE, scale='none',
dendrogram='none', trace='none', margin=c(10, 6))
dev.copy(png,'DESeq2_heatmap3')
dev.off()
```

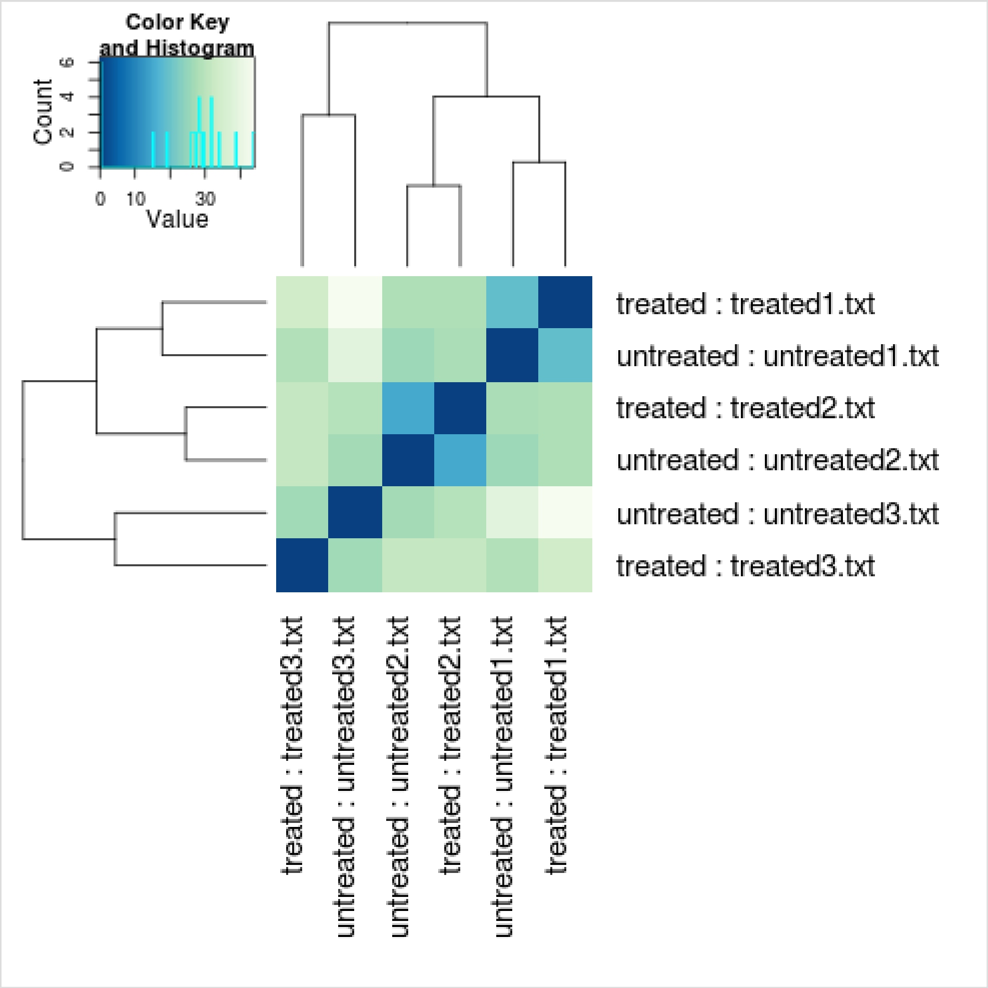


— heatmaps

The above shows heatmaps for 30 most highly expressed genes (not necessarily the biggest fold change). The data is of raw counts (left), regularized log transformation (center) and from variance stabilizing transformation (right) and you can clearly see the effect of the transformation has by shrinking the variance so that we don't get the squish effect shown in the left hand graph.

Now we calculate sample to sample distances so we can make a dendrogram to look at the clustering of samples.

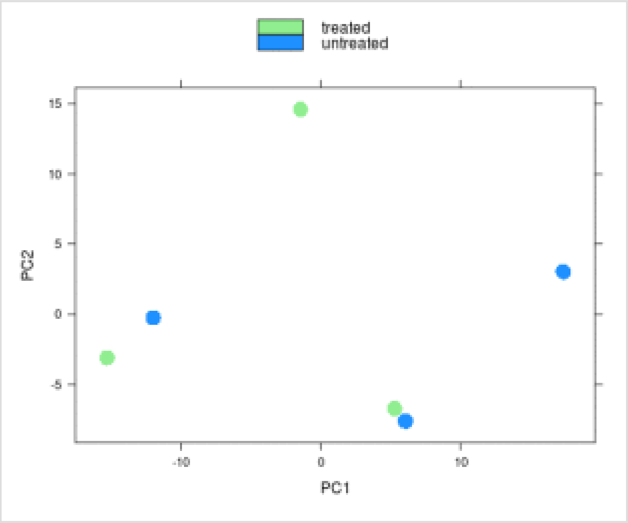
```
distsRL <- dist(t(assay(rld)))
mat<- as.matrix(distsRL)
rownames(mat) <- colnames(mat) <- with(colData(dds),
paste(condition,sampleFiles , sep=' : '))

#updated in latest vignette (See comment by Michael Love)
#this line was incorrect
#heatmap.2(mat, trace='none', col = rev(hmcol), margin=c(16, 16))
#From the Apr 2015 vignette
hc <- hclust(distsRL)
heatmap.2(mat, Rowv=as.dendrogram(hc),
symm=TRUE, trace='none',
col = rev(hmcol), margin=c(13, 13))
dev.copy(png,'deseq2_heatmaps_samplebysample.png')
dev.off()
```



Although this result looks terrible, as we would expect samples to cluster by treatment, in this case I’m actually happy by this result. Why? Well this was actually a control experiment to show that slightly different (and unavoidable) experimental setup for the diferent samples, wasn’t responsible for the observed expression differences, so seeing that there is little treatment effect makes me happy. Remember, always *try* and do what Fisher tells us to, replicate, randomised, block.

Similarly the pca.

```
print(plotPCA(rld, intgroup=c('condition')))
dev.copy(png,'deseq2_pca.png')
dev.off()
```



I hope your's looks better!

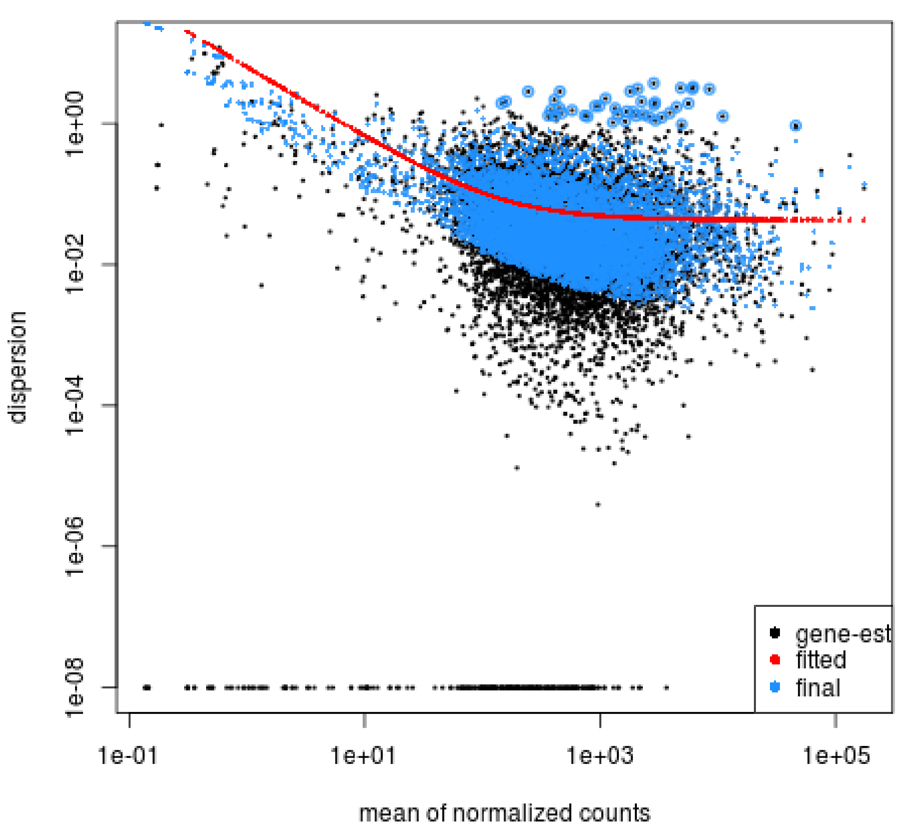
Previously we talked about the cooks distance treatment of outliers, in that a gene is thrown away if one of its samples is deemed to be an outlier. You may not want this to happen so DESeq2 we can take a different approach by replacing the outlier value with one estimated value as predicted by the distribution using the trimmed mean approach. DESeq2 recomends you only do this if you have several replicates per treatment, and indeed it automatically uses this feature if you have 7 or more replicates in your datatable.

```
ddsClean <- replaceOutliersWithTrimmedMean(dds)
ddsClean <- DESeq(ddsClean)
tab <- table(initial = results(dds)$padj < .1,
cleaned = results(ddsClean)$padj < .1)
addmargins(tab)
write.csv(as.data.frame(tab),file='sim_condition_treated_results_cleaned_
resClean <- results(ddsClean)
write.csv(as.data.frame(resClean),file='sim_condition_treated_results_cle
```

In my case it didn't really make much difference.

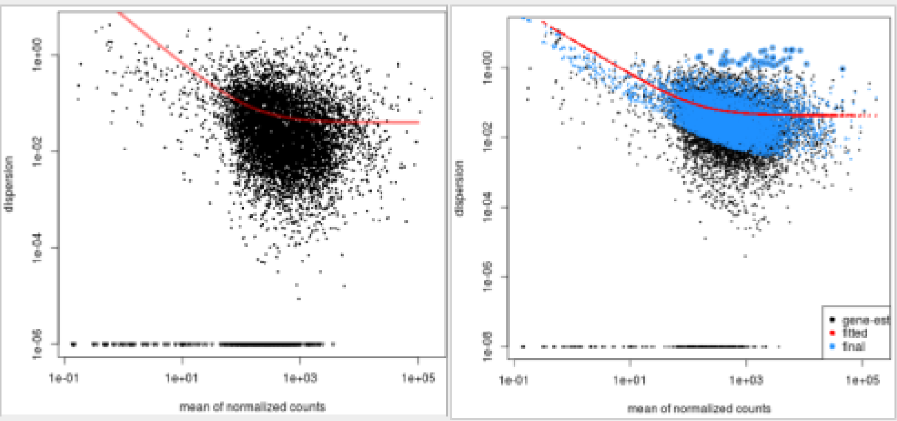
Dispersion plot shows how the estimates are shrunk from the gene wise values (black dots) toward the fitted estimates, with the final values used in testing being the blue dots.

```
plotDispEsts(dds)
```



— Dispersion plots deseq and deseq2 (right)

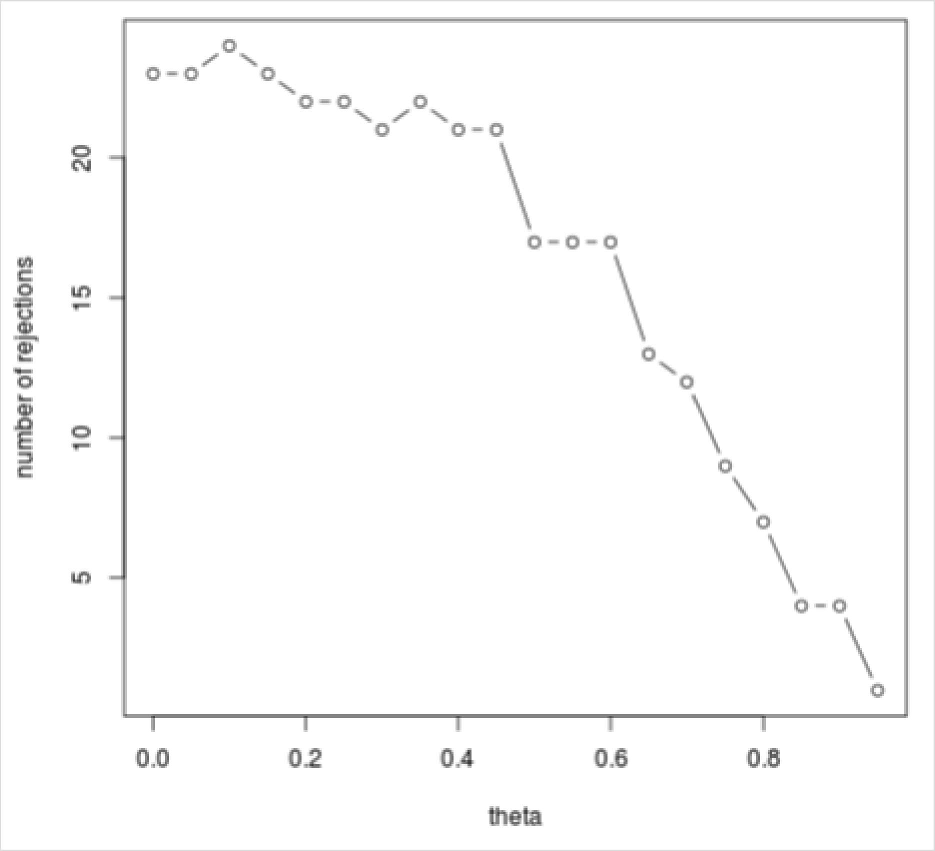
and compared DESeq1 and 2.



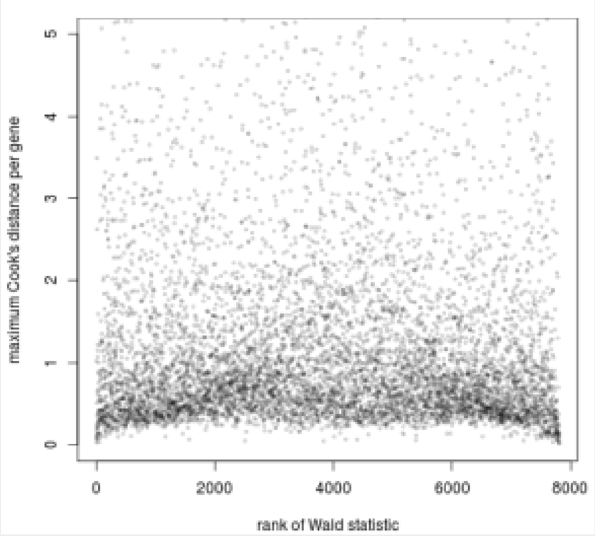
— dispersion plots Deseq1 Left and right deseq2

Now independent filtering to remove any tests that have little chance of pass to reduce the number of tests we have to perform, thus reducing the effects of multiple testing error. (false discovery). You can see how many genes are rejected based on different values of alpha (FDR)

```
#filtering threashold
attr(res,'filterThreshold')
#      10%
#91.48005
plot(attr(res,'filterNumRej'),type='b', ylab='number of rejections')
dev.copy(png,'deseq2_filtering_treshold.png')
dev.off()
```



```
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=.4, col=rgb(0,0,0,.3))
m <- ncol(dds)
p <- 3
abline(h=qf(.99, p, m - p))
dev.copy(png,'deseq2_cooksdist.png')
dev.off()
```

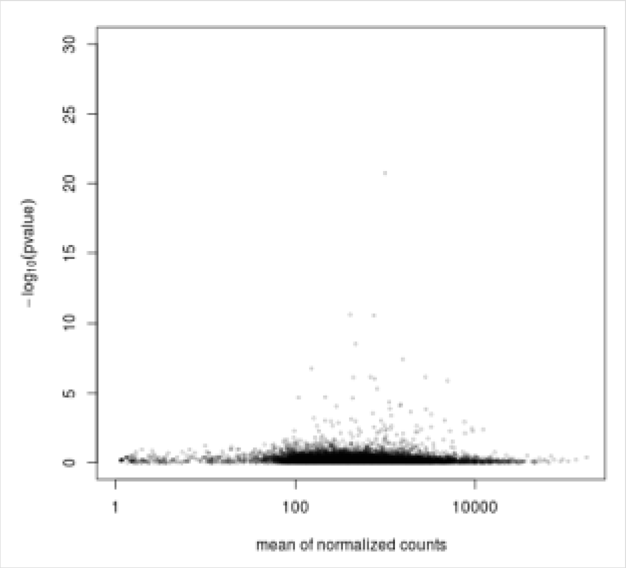


— cooks distance

Plot of the maximum Cook's distance per gene over the rank of the Wald statistics for the condition.

Here more about independent filtering. What it shows in genes with very low counts are unlikely to have a significant p-value due to excessive dispersion at the left side of the dynamic range of counts. The y-axis here is -log10, so bigger numbers are smaller p-values (better).

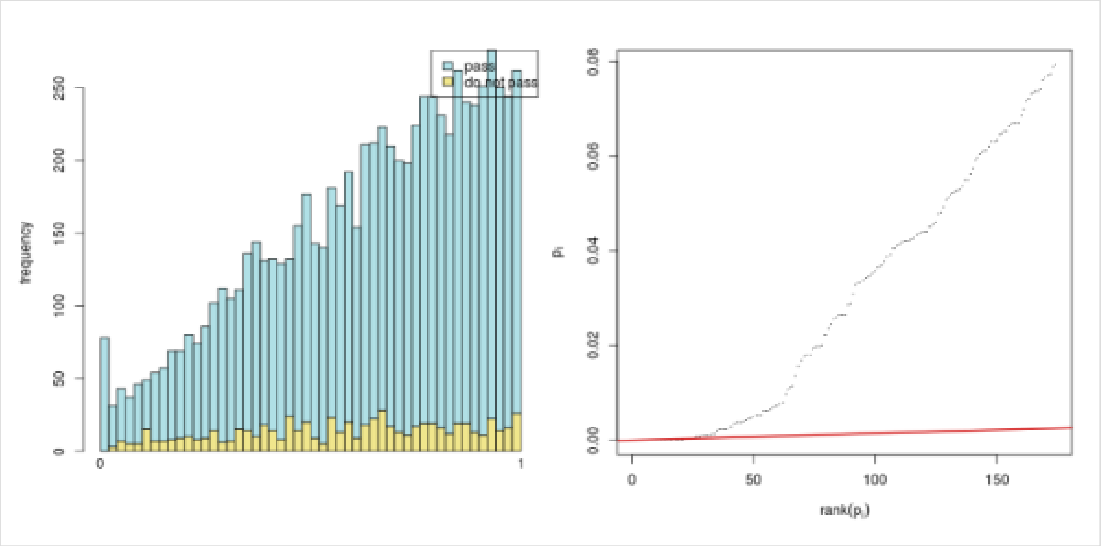
```
plot(res$baseMean+1, -log10(res$pvalue),
log='x', xlab='mean of normalized counts',
ylab=expression(-log[10](pvalue)),
ylim=c(0,30),
cex=.4, col=rgb(0,0,0,.3))
dev.copy(png,'deseq2_indep_filt.png')
dev.off()
```



All those dots on the left hand side the graph represent failed tests due to very low count values, thus we can really just get rid of them to reduce our chance of making a type I error

And again, you can see that only a few small (or no) p-values are discarded by the filtering. NOTE: You might only see blue lines [I've broken something??]

```
use <- res$baseMean > attr(res,'filterThreshold')
table(use)
h1 <- hist(res$pvalue[!use], breaks=0:50/50, plot=FALSE)
h2 <- hist(res$pvalue[use], breaks=0:50/50, plot=FALSE)
colori <- c('do not pass'='khaki', 'pass'='powderblue')
barplot(height = rbind(h1$counts, h2$counts), beside = FALSE,
col = colori, space = 0, main = '', ylab='frequency')
text(x = c(0, length(h1$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)))
```



The graph on the left ranks the p-values from smallest to biggest (x-axis) and plots them. The black line is the actual p-value numbers (remember only about 23 genes had a p-value lower than 0.05). The red line has a slope that represents the number of tests divided by the false discovery rate (0.1). The idea here is the FDR is controlled at the 0.1% value for all tests that occur to the left of the right-most intersection of the black and red line.

The code for the right hand plot above.

```
resFilt <- res[use & !is.na(res$pvalue),]
orderInPlot <- order(resFilt$pvalue)
showInPlot <- (resFilt$pvalue[orderInPlot] <= 0.08)
alpha <- 0.1

plot(seq(along=which(showInPlot)), resFilt$pvalue[orderInPlot][showInPlo
pch='.', xlab = expression(rank(p[i])), ylab=expression(p[i]))
abline(a=0, b=alpha/length(resFilt$pvalue), col='red3', lwd=2)
```

```
sessionInfo()
R version 3.0.2 (2013-09-25)
Platform: x86_64-pc-linux-gnu (64-bit)

locale:
[1] LC_CTYPE=en_US.UTF-8 LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8 LC_COLLATE=en_US.UTF-8
[5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8 LC_NAME=C
[9] LC_ADDRESS=C LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] parallel stats graphics grDevices utils datasets methods
[8] base

other attached packages:
[1] gplots_2.12.1 RColorBrewer_1.0-5 BiocInstaller_1.12.0
[4] DESeq2_1.2.8 RcppArmadillo_0.3.920.3 Rcpp_0.10.6
[7] GenomicRanges_1.14.3 XVector_0.2.0 IRanges_1.20.6
[10] BiocGenerics_0.8.0

loaded via a namespace (and not attached):
[1] affy_1.40.0 affyio_1.30.0 annotate_1.40.0
[4] AnnotationDbi_1.24.0 Biobase_2.22.0 bitops_1.0-6
[7] caTools_1.16 DBI_0.2-7 DESeq_1.14.0
[10] gdata_2.13.2 genefilter_1.44.0 geneplotter_1.40.0
[13] grid_3.0.2 gtools_3.1.1 KernSmooth_2.23-10
[16] lattice_0.20-24 limma_3.18.5 locfit_1.5-9.1
```

```
[19] preprocessCore_1.24.0 RSQLite_0.11.4 splines_3.0.2
[22] stats4_3.0.2 survival_2.37-4 tools_3.0.2
[25] vsn_3.30.0 XML_3.98-1.1 xtable_1.7-1
[28] zlibb
```

[Updated July '14: to fix errors with distance matrix plot, cooks distance, and the Benjamini-Hochberg multiple testing adjustment procedure (props to Stefan for pointing them out]

SHARE THIS:

Press This Twitter Facebook 8

Reblog ★ Like

Be the first to like this.

RELATED

How to use DESeq to analyse RNAseq data
In "Bioinformatic's"

Pandas, matplotlib and Ipython - all you need for great data anaylsis
In "Bioinformatic's"

Setting up a Lubuntu virtual machine with virtual box [and get guest additions working]
In "Linux"

This entry was posted in **Bioinformatic's**, **Programing** and tagged **DESeq**, **DESeq2**, **gene expression**, **R**, **RNAseq** by **Dave Wheeler** . Bookmark the **permalink [https://dwheelerau.com/2014/02/17/how-to-use-deseq2-to-analyse-rnaseq-data/]**

86 THOUGHTS ON " HOW TO USE DESEQ2 TO ANALYSE RNASEQ DATA"

Rupesh
on **February 19, 2014 at 11:49 am** said:

Dear sir,
Thanks sir for posting this great tutorial. Sir i would like to plot MA with padj =< 0.05. I know its by default deseq2 set it as 0.1….

could you help me to plot RED dots (significant DE at padj =< 0.05) in plotMA. I assumed i have to adjust this value in dds ???

Thanks and looking forward for your kind help

> **Dave Wheeler**
> on **February 19, 2014 at 7:52 pm** said:
>
> Hi There,

PlotMA is a function from DESeq, you can see all the info here at the biocondutor page listing all the doc material http://bioconductor.org/packages/2.13/bioc/manuals/DESeq/man/DESeq.pdf I haven’t checked the DESeq2 function, but I guess it would be the same.

the function is listed as (from the above):
plotMA(x, ylim,
col = ifelse(x$padj>=0.1, “gray32”, “red3”),
linecol = “#ff000080”,
xlab = “mean of normalized counts”, ylab = expression(log[2]~fold~change),
log = “x”, cex=0.45, …)

So, following that logic we just need to pass a col variable to override the default, so this should work for your needs:

plotMA(dds,col=ifelse(res$padj>0.05,”grey32″,”red”))

Rupesh
on **February 20, 2014 at 10:33 am** said:

Thanks sir but logic is different….from deseq to deseq2
i got it, it should be ….
plotMA(dds,pvalCutoff=.05,ylim=c(-2,2),main=”DESeq2″)

Dave Wheeler
on **February 20, 2014 at 7:02 pm** said:

Thanks for posting that, good to know

Will
on **February 27, 2014 at 10:48 pm** said:

Hi,
I am having trouble getting started and it is probably due to the fact my counts tables did not derive from HTSeq (I used a different software, CLC). I do, however, have a merged file containing the gene ID, followed by the counts of each of the different samples. I started following your DESeq tutorial and got to the point of conds<-factor(c(""untreated","treated"…..)) and then ran into problems because DESeq2

doesn't have newCountDataSet function. This, of course, led me to this tutorial but don't know how to proceed since I already created a merged count file.

Apologies for the basic question. I'd appreciate any assistance you can provide.

Dave Wheeler
on **February 27, 2014 at 11:58 pm** said:

The simplest (but rather unsatisfying) solution is to just split the table using a script or excel, col1 gene names, cold 2 counts for each replicate.

Going by this post (http://seqanswers.com/forums/showthread.php?t=39253)though it looks like the DESeqDataSetFromMatrix should get you started ("?DESeqDataSetFromMatrix") to see usage

Will
on **February 28, 2014 at 2:12 am** said:

Thanks! I figured out how to get it in. Here's the command lines:
>countsTablerownames(countsTable)countsTablehead(countsTable)
>countDatacolData<-data.frame(condition=factor(c("insert your treatments in proper order for each column of countsTable")))
ddscolData(dds)$condition<-factor(colData(dds)$condition,levels=c("your treatments"))
dds

This got me to the point where I can run DESeq and convert the dds file to res. Now I am trying to figure out what is wrong with the PlotMA function. I type:
plotMA(dds,ylim=c(-2,2),main="DESeq2")

and receive the following error'x' must be a data frame with columns named 'baseMean', 'log2FoldChange'.

Not sure why I am getting this. Will need to troubleshoot.

Will
on **March 4, 2014 at 5:08 pm** said:

Hi,

Just a general note for anyone going through this: make sure you have the updated packages before beginning. I know I ran into some problems with this.

Thanks Dave for the great tutorial. I found it to be very helpful!

A couple of things: I had trouble with plotting the “h1 <- hist(res$pvalue[!use], breaks=0:50/50, plot=FALSE)". My R didn't know the function of "use" or "!use". I compared sessionInfo() and didn't see anything stand out. I'm sure I am overlooking something simple here…

Lastly, your line, resClean
write.csv(as.data.frame(resClean),file="sim_condition_treated_results_cleaned_deseq2.csv")

gave me the error

Error: unexpected symbol in "resClean write.csv"

Other than that, very straightforward. Thanks again!

Dave Wheeler
on **March 5, 2014 at 5:55 am** said:

Thanks for the thanks.

I’ve been having trouble with the code being mushed up by wordpress. I was making a stupid mistake here, that hopefully is fixed. I’ve re-entered all the code (probably adding in some mistakes along the way) and the formating looks a lot better. Hopefully this will fix some of these silly errrors like “”resClean write.csv” which is just complete mushed code.

Super
on **March 3, 2014 at 12:00 pm** said:

Hi Dave, nice post.
Is edgeR/DESeq/DESeq2 only outputs the list of genes? If I want to find the DE transcripts or coding sequence CDS, could I do it by DESeq?

Dave Wheeler
on **March 3, 2014 at 7:34 pm** said:

Thanks for the comment! If you have a well annotated GTF file that contains isoform information and a splice aware mapper like tophat HTSeq count will use that information to generate isoform specific read counts (see http://www-huber.embl.de/users/anders/HTSeq/doc/count.html). My bug of interest does not have good gene models at the moment so I'm stuck with just using loci level information in all the example tables I put up here. Other options are to try DEXseq which was designed to deal with alt splice forms, but personally I have never used this. I think its correct to say the biggest difference between cufflinks and DESeq is that the former will (additionally) use the reads to generate novel gene models that are then merged with the genome gene models.

Super
on **March 4, 2014 at 11:01 am** said:

Thank you! Could I summarize the Cufflinks vs DESeq
1.Cufflinks is stronger for their efficiency and test rate (like the TPR?).?
2.Cufflinks could potentially discovering novel genes and transcripts? and output the DE transcripts/CDS/TSS which DESeq probably not (decided by HTseq output).
3. DESeq is more suitable for unbalanced-samples (like 3 samples in treated and 4 samples in untreated) because Cufflinks/Cuffdiff is only suitable for pairwise comparison.
Am I right? Do you have your revised/supplement comments?

Dave Wheeler
on **March 5, 2014 at 6:04 am** said:

I think its fair to say that cufflinks2 (the cuffdiff part) and deseq are now more similar than they used to be with regard to the DEG stats (see http://cufflinks.cbcb.umd.edu/manual.html#dispersion_meth). But the other big difference is normalisations based on FPKM versus counts, its perhaps an unsettled debate which is best.

Super
on **March 5, 2014 at 12:06 pm** said:

Thank you. I don't know counts or FPKM which is better as well.So what do you think? Compare the edgeR/DESeq/Cuffdiff? Do you have any suggestion or conclusions? (e.g. running time? statistic power? and so on) Thank you!

Will
on **March 5, 2014 at 4:08 pm** said:

I thought I would chime in my thoughts on this topic… There was a fairly recent (2013) paper that compares Cufflinks2 and DESeq2 (and EdgeR and a couple other methods). One of the conclusion of this study suggested Cufflinks FPKM may be a bit too conservative and overlooks many differential gene expressions. Personally, I think it is prudent to try out a couple of these programs and draw conclusions from both. For example, if both a conservative and liberal analysis produce the same set of genes as being differentially expressed then you have greater evidence than if only one analysis produces the results.

Dave Wheeler
on **March 7, 2014 at 10:31 pm** said:

I generally agree, although I always throught FPKM were actually too liberal at least based in this (excellent) paper
http://bib.oxfordjournals.org/content/early/2012/09/15/bib.bbs046.abstract
But yes trying different methods is important.I think we have to be realistic and remember that we are asking a lot of the models to accurately estimate expression based on just 3 samples. I guess when seq gets even cheaper we can start using many more biological replicates to get a better sample of the transcriptome under some condition or at least make sure we have enough coverage to get good readings for low expressed genes that are at the edge of the models statistical power.

Will
on **March 6, 2014 at 12:07 am** said:

Hi again,
I had a general question in interpreting the final results. I noticed in my data I have a long list of genes with an ever-increasing p-value. In fact, I don't start seeing "NA" under "padj" until row 22685. Is this to say that I have 22,684 significant differential expressed genes? If not, how can I justify statistically where to draw the minimum p-value threshold? I was looking through the DESeq2 manual, and although you don't plot it here, I was wondering the Benjamini-Hochberg multiple testing adjustment procedure (Fig.12 pg 30) could be used to reduce the number of genes to examine. If I understand it correctly, anything to

the left of the red and black line intersection is controlled for FDR at alpha 0.01. Thus, only 1481 genes (in the manual data) are of interest.

Your thoughts?

geneart
on **March 26, 2014 at 12:06 am** said:

Thanks for a great post DrWheeler ! Very much appreciate your time and effort in doing this which is a wonderful guide for a rookie like myself ! As I was following your analysis steps , I got an error while plotting plotMA. I had this same error while running DESeq as well.If I plot the dispersions against counts it works but not by the way you do.THis is what it says:
> plotMA(dseq,pvalCutoff=.05,ylim=c(-2,2))
Error in as.vector(data) :
no method for coercing this S4 class to a vector
I looked it up on the net , and it seems like this is an R related errors and not DEseq per sayt. However I am not sure. Please could you help me troubleshoot this?
THanks very much in advance !
geneart.

> **Dave Wheeler**
> on **March 26, 2014 at 3:53 am** said:
>
> Hi There, thanks for the kind words. Bit hard to work out whats wrong, but one easy check is to make sure you are using the same versions as me by looking at my sessionInfo() versus yours. Better yet, you can access the vignettes for the version of DESeq you are using with this, browseVignettes(package = "DESeq") then double check that function in the doc to see if it has changed at all. That would be a good place to start as often the functions change slightly between versions.
>
> geneart
> on **March 26, 2014 at 5:14 pm** said:
>
> Thanks for the reply DrWheeler. I checked on my sessionInfo and nothing seems to be drastically different from your sessionInfo, except that I am using a different version of R , not too far off though ! also I checked the vignette that shows the same function as u mention, so no problems there:
> > sessionInfo()
> R version 3.0.1 (2013-05-16)
> Platform: i386-w64-mingw32/i386 (32-bit)

```
locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] grDevices datasetsparallel stats graphics utils methods
[8] base

other attached packages:
[1] DESeq2_1.2.10 RcppArmadillo_0.4.100.2.1
[3] Rcpp_0.11.1 GenomicRanges_1.14.4
[5] XVector_0.2.0 IRanges_1.20.7
[7] DESeq_1.14.0 lattice_0.20-15
[9] locfit_1.5-9.1 Biobase_2.22.0
[11] BiocGenerics_0.8.0 BiocInstaller_1.12.0

loaded via a namespace (and not attached):
[1] annotate_1.40.1 AnnotationDbi_1.24.0 DBI_0.2-7
[4] genefilter_1.44.0geneplotter_1.40.0grid_3.0.1
[7] RColorBrewer_1.0-5RSQLite_0.11.4 splines_3.0.1
[10] stats4_3.0.1 survival_2.37-7tools_3.0.1
[13] XML_3.98-1.1xtable_1.7-3
```

However I did look for plotMA problems on web and one suggestion I got from the DESeq2 manual is to detach BiocGenerics or load limma package later as they both use the plotMA and somehow it interferes. I tried detaching the BiocGenerics but this is what I got:

```
> detach(“package:BiocGenerics”, unload=TRUE)
Error: package ‘BiocGenerics’ is required by ‘GenomicRanges’ so will not be detached
> detach(“package:limma”, unload=TRUE)
Error: package ‘limma’ is required by ‘edgeR’ so will not be detached
> detach(“package:edgeR”, unload=TRUE)
Warning messages:
1: In FUN(X[[2L]], …) :
Created a package name, ‘2014-03-26 12:55:56’, when none found
2: In FUN(X[[2L]], …) :
Created a package name, ‘2014-03-26 12:55:56’, when none found
3: In FUN(X[[2L]], …) :
Created a package name, ‘2014-03-26 12:55:56’, when none found
4: In FUN(X[[2L]], …) :
Created a package name, ‘2014-03-26 12:55:56’, when none found
> detach(“package:limma”, unload=TRUE)
There were 18 warnings (use warnings() to see them)
> plotMA()
Error in class(object) : ‘object’ is missing
```

```
> plotMA(dseq,pvalCutoff=.05,ylim=c(-2,2))
Error in (function (classes, fdef, mtable):
unable to find an inherited method for function ‘extractROWS’ for signature
‘”list”’
```

So at this point I am lost ! I would love to have this plotMA working for me , but cannot figure out the problem. Any suggestions? anyone ?Thanks🙂
geneart.

Dave Wheeler
on **March 26, 2014 at 7:29 pm** said:

The order is important, the last package to load will overwrite the previous function. I wouldn;t detact the other other library and DESeq needs it. I would load both and then explicitly call the DESeq2 function like this.

```
DESeq2::plotMA(bla bla)
```

```
#replacing bla bla with arguments.
```

Sorry I not really an R expert, but hopefully that helps

geneart
on **March 26, 2014 at 10:02 pm** said:

Thanks , will try this ang update again🙂
Geneart

geneart
on **April 1, 2014 at 10:43 pm** said:

Hello Dr.Wheeler,

```
> ddsHTSeq<-DESeqDataSetFromHTSeqCount(sampleTable=sampleTable,
directory=directory, design=~sampleCondition)
Error in `rownames<-.data.frame`(`*tmp*`, value = value) :
duplicate 'rownames' are not allowed
In addition: Warning message:
non-unique value when setting 'rownames': 'balt.txt'
```

does this indicate I have a gene_name duplicated in the balt.txt file? If I use HTSeqCOunt function instead of CountDataSet I get this errorIf I use the CountDataSet everything works fine. ANy suggestions? I am reading the files just like how you did.

geneart
on **April 12, 2014 at 8:05 pm** said:

Hello Dr.Wheeler,
Finally after some digging through I got to figure how to fix my plotMA and it is working. However I had one question though. In your demo you have used the treated first and then you control in the sample conditions. However in calling the level , later in ddsHTSEq u have called levels=c("untreated","treated"). I believe thi sshould nto be an issue. However wanted to double check on this if it would be an issue??
I have used control and treated as the order in my sample conditions an dhave called levels=c("control","treated"). when I plotted my DESeq2::plotMA(dseq,pvalCutoff=.05,ylim=c(-2,2)) I see only 2 red spots.
I was wondering if my order was an issue or just that I ahve two outliers ? Please could you help me? HEre is my plot.

> **Dave Wheeler**
> on **April 16, 2014 at 10:42 pm** said:
>
> The order is only important for the direction of the fold change, it won't change any of the statistics. It makes sense to use control vs treatment, as anything down in response to the treatment will be given a negative fold change, and visa versa. Most people would expect a 2 fold up to equal UP in response to treatment, for example. Only 2 significant DEG is better than none!

Gthm
on **April 14, 2014 at 3:40 am** said:

Should we filter the data before feeding in to DESEQ2 ? Like, remove entities that have zero across all the conditions and then feed to DESEQ ?

> **Dave Wheeler**
> on **April 16, 2014 at 10:51 pm** said:
>
> Filtering the zero rows will certainly help reduce the multiple testing issues by reducing the number of tests (by removing those that are guaranteed to fail).
> Cheers

Ryan
on **April 22, 2014 at 6:01 pm** said:

Hi Dr. Wheeler-

I appreciate the time you've taken in producing this (and the previous DESeq) tutorial. I had two questions regarding your python script, merge_tables.py for merging the count tables. (I employ HTSeq prior to running the merge_tables.py)

1) Prior to running your script (or before running DESeq), should I remove the rows without gene counts? That is, the rows that contain no_feature, ambigous, too_low_aQual, not_aligned, and alignment_not_unique?

2) Are there limitations on the number of count files that I can merge at one time? And if so, would it be appropriate to run the script twice, on half the count files and then merge them after the fact?

Cheers

> **Dave Wheeler**
> on **April 22, 2014 at 9:54 pm** said:
>
> Hi Ryan,
> 1) The script expects all the files to share the same row names (identifiers) so don't remove low/no count genes before you merge. You can do it after the merge (even using excel if you don't know scripting, just sort and chop the rows that are zero across ALL columns). By zero counts I would mean zero across all treatments, so you should only chop these once you see the results for all the columns otherwise its not sensible thing to do.
> 2) You can merge as many files as you like, just add the files and column headings to the guide file. DONT run the script twice, it will break. Just add all the

> files and corresponding column headings to the guide file and it will work.
> Cheers
> Dave

Quan
on **May 19, 2014 at 10:02 am** said:

Hi Dave
could you show me some scripts if I want to do multiple condition comparison?
So far I have 3 conditions and 3 samples per condition, I don't want to do three times like C1 vs C2, C2 vs C3, C1 vs C3, could you please tell me how to dow it just one time in DESeq2?
Thank you!
Quan

> **Dave Wheeler**
> on **May 19, 2014 at 7:53 pm** said:
>
> I'm pretty sure you have to do them pairwise with DESeq. From memory Cufflinks lets you do multiple comparisons at once, not sure.
>
> Quan
> on **May 20, 2014 at 10:54 am** said:
>
> And from your case PCA plot(in this blog), which is not total separate two classes(green and blue points), is this means that the difference expression between two conditions is not obvious?
> Furthermore , is it similiar to the edgeR's MDS plot?
>
> Quan
> on **May 20, 2014 at 10:56 am** said:
>
> PS: what is difference bwtweenMDS and PCA plot meaning?

> **Dave Wheeler**

on **May 20, 2014 at 7:57 pm** said:

Quan, correct, there is no grouping of treatments, but this was the result I was hoping for because it was a comparison between two controls (see the text above the figure). I;m not an expert like Google(-; the top hit for your other question looks pretty good: http://stats.stackexchange.com/questions/14002/whats-the-difference-between-principal-components-analysis-and-multidimensional

Quan
on **May 21, 2014 at 8:51 am** said:

Cheers!

Julia
on **May 25, 2014 at 6:33 pm** said:

Hi!
First of all, thank you for the great tutorial!
However, I have a problem now with ddsHTSeq<-
DESeqDataSetFromHTSeqCount(sampleTable=sampleTable, directory=directory, design=~condition)

At the console,
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
cannot open file 'C/Users/Julia/Desktop/I_375_Ago-IP.gene.hts': No such file or directory
appears.
Can you help me?

Best wishes
Julia

Dave Wheeler
on **May 25, 2014 at 9:44 pm** said:

Hi Julia, your file path looks wrong. Try:
directory<-file.path('C:','Users/Julia/Desktop')

Julia

on **May 26, 2014 at 6:04 am** said:

thanks a lot! Now it works!🙂

Rupesh
on **June 19, 2014 at 8:49 am** said:

Dear sir,

hello again

just few confusion need to resolve from your great experience.

Just wondering to compare DESeq2 and edgeR results of DE. But confusion is that…..In edgeR it is advisable to filter low tags by cpm counts before DE analysis while in DESeq2 its not. while The package DESeq2 consider independent fileting (via function results) step that is not available for edgeR ??

so my query is that
1) it is necessary to take filtration step of edgeR in edgeR (or its just optional)???
#####case 1 (followed default functions)
2) while comparing the results of DE gene of DESEq2 (without via edgeR of cpm just used default functions i.e. deseq and results) and in edgeR (with filtration via cpm), i got results almost similar and that is what, i was expecting….
####case 2 (applied filtration in both via cpm)
but while applying same filtering of edgeR via cpm in DESeq2 and then used the function deseq and results (with and without independent filtering by both; just for check), results are not similar et all…..very far DE in both (taking 0.05 padj as cutoff).

So please clear my doubt and suggest me what to do ???

> **Dave Wheeler**
> on **June 22, 2014 at 9:18 pm** said:
>
> IMHO the filtering is optional. It really is just there to try and reduce the effect of multiple testing. I would check the counts on your differentially expressed genes and make sure that its not all low count genes, as the calls on these as DEG might be artifacts and thus are sensitive to these filtering steps, but I'm not really sure?

Stefan
on **July 17, 2014 at 6:06 pm** said:

First at all I want to give you a BIG thank you! I already used your DESeq tutorial and both of them are very helpful.

When I was using you script, I stumbled accross two errors/problems:

First:

```
rownames(mat) <- colnames(mat) <- with(colData(dds), paste(condition, type, sep=" : "))
```

produced this error:

```
Error in eval(expr,envir, enclos) : object 'type' not found
```

Second:

```
plot(seq(along=which(showInPlot)), resFilt$pvalue[orderInPlot][showInPlot],
pch=".", xlab = expression(rank(p[i])), ylab=expression(p[i]))
```

produced this error:

```
Error in which(showInPlot) :
error in evaluating the argument 'x' in selecting a method for function 'which': Error: object 'showInPlot' not found
```

I used the dataset you provided. Could you help me to figure out, what the problem is?
I would appreciate your help!
Stefan

> **Dave Wheeler**
> on **July 19, 2014 at 2:24 am** said:
>
> Thanks for the kind words. Are you using the DESeqDataSetFromHTSeqCount function to import the table like is shown in the blog? I'm not sure about the second error (sorry), my guess is your using an newer version or R/DESeq2 and the function has changed (see my sessionInfo below). I'm a bit rusty on this (-:
>
> > Stefan
> > on **July 21, 2014 at 7:33 pm** said:
> >
> > Yes I am using the DESeqDataSetFromHTSeqCount function to import the table. I downloaded your files and used your script as shown in your blog.
> >
> > This would be my sessioninfo:
> >
> > ```
> > > sessionInfo()
> > R version 3.0.2 (2013-09-25)
> > Platform: x86_64-pc-linux-gnu (64-bit)
> > ```

```
locale:
[1] LC_CTYPE=en_USLC_NUMERIC=CLC_TIME=en_US
[4] LC_COLLATE=en_US LC_MONETARY=en_US
LC_MESSAGES=en_US
[7] LC_PAPER=en_US LC_NAME=C LC_ADDRESS=C
[10] LC_TELEPHONE=CLC_MEASUREMENT=en_US
LC_IDENTIFICATION=C

attached base packages:
[1] parallel stats graphics grDevices utils datasets methods
[8] base

other attached packages:
[1] gplots_2.14.1 RColorBrewer_1.0-5vsn_3.30.0
[4] Biobase_2.22.0 DESeq2_1.2.10RcppArmadillo_0.4.320.0
[7] Rcpp_0.11.2 GenomicRanges_1.14.4XVector_0.2.0
[10] IRanges_1.20.7BiocGenerics_0.8.0

loaded via a namespace (and not attached):
[1] affy_1.40.0 affyio_1.30.0 annotate_1.40.1
[4] AnnotationDbi_1.24.0BiocInstaller_1.12.1 bitops_1.0-6
[7] caTools_1.17 DBI_0.2-7 gdata_2.13.3
[10] genefilter_1.44.0grid_3.0.2 gtools_3.4.1
[13] KernSmooth_2.23-12lattice_0.20-29 limma_3.18.13
[16] locfit_1.5-9.1 preprocessCore_1.24.0 RSQLite_0.1.4
[19] splines_3.0.2 stats4_3.0.2 survival_2.37-7
[22] XML_3.98-1.1xtable_1.7-3 zlibbioc_1.8.0
```

Stefan
on **July 23, 2014 at 3:20 pm** said:

I had these two problems here, right?:

First:
rownames(mat) <- colnames(mat) <- with(colData(dds), paste(condition, type, sep=" : "))
produced this error:
Error in eval(expr, envir, enclos) : object 'type' not found

Second:
plot(seq(along=which(showInPlot)), resFilt$pvalue[orderInPlot][showInPlot],
pch=".", xlab = expression(rank(p[i])), ylab=expression(p[i]))
produced this error:

```
Error in which(showInPlot) :
error in evaluating the argument 'x' in selecting a method for function 'which': Error: object 'showInPlot' not found
```

I am pretty sure you must have gotten the same errorwhen you ran you script. For the first error:
paste(condition, type, sep=" : ") is not working, because at least the type object is not defined. Otherwise, you figure (https://dwheelerau.files.wordpress.com/2013/12/deseq2_heatmaps_samplebysample.png?w=584) would have labels seperated by " : " (like figure 5 in the orignial manual: http://www.bioconductor.org/packages/2.13/bioc/vignettes/DESeq2/inst/doc/DESeq2.pdf)

For the second error:
You forgot a few lines:

```
resFilt <- res[use & !is.na(res$pvalue),]
orderInPlot <- order(resFilt$pvalue)
showInPlot <- (resFilt$pvalue[orderInPlot] <= 0.08)
alpha <- 0.1
```

before you can run:

```
plot(seq(along=which(showInPlot)), resFilt$pvalue[orderInPlot][showInPlot],
pch=".", xlab = expression(rank(p[i])), ylab=expression(p[i]))
```

That is the reason, why showInPlot and orderInPlot are undefined…

Dave Wheeler
on **July 25, 2014 at 8:31 am** said:

Hi Stefan, sorry about the errors, thanks for posting some fixes! It really is a total pain in the arse moving between R and wordpress, adding figures, code, and rambling on (excuses excuses), so mistakes are bound to pop up. I try to do these about what I'm doing at work, and I'm not really doing much RNAseq at the moment (mostly teaching/paperwork). Hopefully I can find some time to update this next time I get the RNAseq bug (or feel guilty).

Dave Wheeler
on **July 25, 2014 at 10:25 pm** said:

I felt guilty and fixed this. Theres one slight issue with the histogram plot (yellow and blue), not quite sure what I've done there. Thanks for the feedback (and GO K-state wildcats).

Anil Kumar
on **August 20, 2014 at 2:03 pm** said:

Hi Sir,

Thank you very much for the descriptive tutorial. I have two doubts. Kindly help me.

1.In my case, there are 2 control replicates and 2 replicates for each treatment (cold stress 5 days -> treatment 1 &cold stress 10 days-> treatment 2). So can i take all the different treatment as just treated sample as below,

SampleCondition<-c("untreated","untreated","treated","treated","treated","treated")

2. In the exaple which you shown, sample conditions are specified by treatment first followed by untreatment. However, in the manual, it is instructed like this "control" or untreated" level as the first element ("base level"), so
that the log2 fold changes produced by default will be the expected comparison against the base level". Which means we should take control as first condition?

> **Dave Wheeler**
> on **August 20, 2014 at 8:46 pm** said:
>
> 1) I don’t think it would be correct because DESeq will treat your different treatments as replicates of one treatment rather than separate treatments. You would need to setup pairwise comparisons, cont v cold stress 5 days and cont V (cold stress 10 days). Cufflinks has a timeseries options if you wanted to do what you outlined.
>
> 2) Yep, I agree, but see this line “colData(ddsHTSeq)$condition<-factor(colData(ddsHTSeq)$condition, levels=c("untreated","treated"))" I specify untreated first so that the log values make sense, as you say I also mentioned this in the text "The levels in colData are important because they are used in the log calculations; it makes sense to set untreated or control first so that the direction of the logs fold changes doesn’t confuse everyone (typically we do comparisons to the control)! Now for the guts of the DEseq2 analysis.". I think I got it right?

Anil Kumar
on **August 22, 2014 at 2:09 am** said:

Hi Sir,
I wanted to withdraw my second doubt soon after posting it. As you mentioned, it was clearly indicated in the code as well as in text. Sorry for that..!!

As suggestion by SEQanswers members, Cufflinks and Cuffdiff consider replicate pair and will not give correct result if am interested in knowing the inter-replicate variation (actually i need to analyze inter-replicate variation also) .

As you suggested, i will proceed by splitting my samples as ctrl+cold 5 days and ctrl+cold 10 days separately. But still have a strong doubt whether such 'splitting up' processes is statistically significant when working on differential expression analysis.

I would like to know your suggestion once again..If that is ok, then i will convert my count matrix(generated by Featurecount) to count table (HTSeq) and proceed as your instructions with spitted data.
Thanks,
Anil.

David
on **September 5, 2014 at 3:38 pm** said:

Hi Dave,

First, like everyone else, I am very grateful for this tutorialIt has helped me tremendously with analyzing our RNAseq data.

I had a question concerning the heatmap functionIs there away to select for significantly expressed genes (say those with padj < 0.1) instead of normalized counts of the 30 highly expressed genes?I was having issues selecting counts for genes that were differentially expressed. Suggestions?

thank!

```
library("RColorBrewer")
library("gplots")
select <- order(rowMeans(counts(dds,normalized=TRUE)),decreasing=TRUE)[1:30]
hmcol <- colorRampPalette(brewer.pal(9, "GnBu"))(100)
heatmap.2(assay(vsd)[select,], col = hmcol,
Rowv = FALSE, Colv = FALSE, scale="none",
dendrogram="none", trace="none", margin=c(10, 6))
dev.copy(png,"DESeq2_heatmap3")
dev.off()
```

Dave Wheeler
on **September 5, 2014 at 11:21 pm** said:

Hi There and thanks. I remember i did something like this in DESeq[1]. Not really near a computer so I can’t play around with it but it might work?? Just change the select line and keep the rest the same.

```
select = order(res$padj,decreasing=FALSE)[1:20]
……
```

David
on **September 8, 2014 at 1:28 pm** said:

Hey Dave,

Thanks for the suggestion.I was definitely overthinking the problem.Order by res$padj works well for me.If any one else is interested, you will need to add normalization in your heatmap.2 script (scale = ‘row’).e.g.:

```
select.padj = order(res$padj,decreasing=FALSE)[1:20]
my_palette <- colorRampPalette(c("blue",'white','red'))(n=1000)
par(cex.main=0.8)
heatmap.2(assay(vsdMF)[select.padj,], col=my_palette,
scale="row", key=T, keysize=1,symkey=T,density.info="none",
trace="none",cexCol=0.6, labRow=F)
```

Gonzalo
on **January 29, 2015 at 8:12 pm** said:

thanks for this blog, is quite useful. I am new to DEseq2 and maybe this is silly question, but I do not understand why meanSdPlot fails…. any ideas to solve this problem?

```
library(“vsn”)
par(mfrow=c(1,3))
notAllZero 0)
meanSdPlot(log2(counts(dds,normalized=TRUE)[notAllZero,] + 1))
meanSdPlot(assay(rld[notAllZero,]))
meanSdPlot(assay(vsd[notAllZero,]))
```

Error: could not find function “meanSdPlot”

Thanks in advance

Dave Wheeler
on **January 30, 2015 at 3:09 am** said:

Sounds like the wrong library is being loaded (not sure).
I get the following when I type meanSdPlot:

```
>>meanSdPlot
standardGeneric for “meanSdPlot” defined from package “vsn”

function (x, ranks = TRUE, xlab = ifelse(ranks, “rank(mean)”,
“mean”), ylab = “sd”, pch = “.”, plot = TRUE, …)
standardGeneric(“meanSdPlot”)

Methods may be defined for arguments: x, ranks, xlab, ylab, pch, plot
Use showMethods(“meanSdPlot”)for currently available ones.
```

My session info from the above included this version of vsn and the dependencies:

```
>>sessionInfo()
…
Biobase_2.26.0
BiocGenerics_0.12.0
vsn_3.34.0

…
```

Gonzalo
on **January 30, 2015 at 1:48 pm** said:

Thanks. I wasnt loading the package “vsn”…

Anupam Sinha
on **April 15, 2015 at 12:43 pm** said:

Hey Dave,

Thanks for this blog, which is very informative and helpful. I have a query regarding drawing heatmap. Is it possible to draw a heatmap for about 50 genes that show greatest change in expression in both directions (i.eupregulation and downregulation)across samples ? Maybe using by using absolute values of log2Foldchange in descending order Thanks for any help…

> **Dave Wheeler**
> on **April 15, 2015 at 8:35 pm** said:
>
> Hi Sinha, the key line is:
> select <- order(rowMeans(counts(dds,normalized=TRUE)),decreasing=TRUE)[1:30]
>
> As you can see the heatmap is composed of the highest count genes as these give the most robust results. Changing to greatest fold change you would want to make sure these don't end up being lots of low count genes with distorted fold changes. I'm a bit rusty and don't have a dataset to test but play around with the "select" line until you get what you want, this might be a place to start:(?)
>
> select <- order(res$log2FoldChange)[1:30]
> or something like that!

Michael Love
on **April 22, 2015 at 2:39 pm** said:

hi David,

Great post. Thanks🙂

At some point last year we changed the distance matrix drawing code chunk to:

hc <- hclust(distsRL)
heatmap.2(mat, Rowv=as.dendrogram(hc), symm=TRUE, trace="none", col = rev(hmcol), margin=c(13, 13))

This uses the distances directly to order the rows and draw the dendrogram on top. Otherwise, heatmap.2 will calculate the distances of the distance matrix to order the rows. (This bug in the vignette was pointed out to me by Jesse Rowley)

> **Dave Wheeler**
> on **April 24, 2015 at 11:19 pm** said:
>
> Hey Michael, thanks for the update and I'm glad I haven't managed to terribly misrepresent your great work! And seriously thanks for DESeq and the great

> Documentation that goes with it!

Raheleh Sh
on **April 30, 2015 at 7:13 am** said:

Dear David,
Thanks for your post.
I am trying to make the heatmap only with the most 30 upregulated genes and I tried this command:

```
up 0)
select.padj <- order(up$padj,decreasing=FALSE)[1:30]
hmcol <- colorRampPalette(brewer.pal(9, "GnBu"))(100)
pdf("upreg9_heatmap3_vsd_IL22_wt_KO.pdf")
heatmap.2(assay(vsd)[select.padj,], col = hmcol,
Rowv = FALSE, Colv = FALSE, scale="column",
dendrogram="none", trace="none", margin=c(12, 1))
dev.off()
```

but, I am getting the heatmap from genes with highest count.
Would you please help me where is the problem?

Many thanks,
Rahel

> **Dave Wheeler**
> on **May 6, 2015 at 8:51 am** said:
>
> Sorry Rahel, been really busy This might help. This is based around res which is sorted by padj near the top of the code in the post. Using the same logic I think you could get res to be sorted by +log fold change then use it as follows.
>
> ```
> #Res is ordered by padj, get top 30
> top30padj<-rownames(res)[1:30]
> s<-which(rownames(dds) %in% top30padj)
> heatmap.2(assay(vsd)[s,], col = hmcol,
> Rowv = FALSE, Colv = FALSE, scale="none",
> dendrogram="none", trace="none", margin=c(10, 8))
> ```

shy
on **June 10, 2015 at 8:45 am** said:

Hi,I want o know if there are no replicates in my experiments,how can i do ? Here is my code for replicates. thank you very much!

```
cds<-DESeqDataSetFromMatrix(data.f3_mf,design=~condition,colData=colData);
cds<-estimateSizeFactors(cds);
cds<-estimateDispersions(cds);
cds<-nbinomWaldTest(cds);
cds<-DESeq(cds);
res <- results(cds)
resOrdered <- res[order(res$padj),]
plotMA(resOrdered);
addmargins( table( res_sig = resOrdered $padj < .1, res_sig = resOrdered $padj < .1 ) );
```

> **Dave Wheeler**
> on **June 10, 2015 at 8:17 pm** said:
>
> Never done it, but I'd say you just need to change this line:
>
> ```
> sampleCondition<-c("treated","treated","treated","untreated","untreated","untreated")
> ```
>
> to
>
> ```
> sampleCondition<-c("treated","untreated")
> ```
>
> and inport in only the two count files. Remember the value of biological RNAseq data without replicates is somewhat limited.

kabasokamfwa
on **June 27, 2015 at 2:52 pm** said:

Hi, how do I change the padj from the default 0.1 to 0.05

> **Dave Wheeler**
> on **June 28, 2015 at 1:51 am** said:
>
> Hi, just pass alpha to the function, most of the functions accept it and the default is 0.1 ie plotMA(dds,alpha=0.05).

> You can see how all the functions work bychecking the docs (http://www.bioconductor.org/packages/release/bioc/manuals/DESeq2/man/DESeq2.pdf)

jp
on **July 19, 2015 at 4:41 pm** said:

I cant follow your protocol because of this error:
> ddsHTSeq<-DESeqDataSetFromHTSeqCount(sampleTable=sampleTable, directory=directory, design = ~condition)
Error in DESeqDataSet(se, design = design, ignoreRank) :
some values in assay are not integers

any help ?

> **Dave Wheeler**
> on **July 19, 2015 at 8:56 pm** said:
>
> It sounds like some of the count values you are using are not integer numbers, they might be missing values or NaN or are decimals, or perhaps strings (from a header), I check the count files for anything funny and make sure they haven't already been normalised.
> Good luck.
> Cheers

Little_Jane
on **October 27, 2015 at 7:07 am** said:

hello, thanks for the good tutorial. I want to know which the method of p value adjusted is by default. And I also check on the website of DESeq2 and looked at the command results, which can be used to decide the method we want to use for adjusting. So, do you know about it?

> **Dave Wheeler**
> on **October 27, 2015 at 11:21 pm** said:

> Hi there,
> Its set to BH () for the FDR corrections. If you see the manual (https://bioconductor.org/packages/release/bioc/manuals/DESeq2/man/DESeq2.pdf) on pg 35 for the results function there is a setting pAdjustMethod = "BH".
> Based on the doc says you can change it to one of these:
>
> ```
> p.adjust.methods
> # c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY",
> # "fdr", "none")
> ```

Mayank Kaashyap
on **February 18, 2016 at 8:44 am** said:

Hi Dave,
I don't find your previous version on this. Here you have used amp, quot. Can I have the previous workflow on DESeq2.

> **Dave Wheeler**
> on **February 18, 2016 at 7:46 pm** said:
>
> Bugger, bloody wordpress – fixed.

yifangt
on **February 25, 2016 at 7:55 pm** said:

I am having problem with meanSdPlot() with this error:

```
> meanSdPlot(log2(counts(dds,normalized=TRUE)[notAllZero,] + 1), ylim = c(0,2.5))
Error: Unknown parameters: ylim
> meanSdPlot(assay(rld[notAllZero,]), ylim = c(0,2.5))
Error: Unknown parameters: ylim
> meanSdPlot(assay(vsd[notAllZero,]), ylim = c(0,2.5))
Error: Unknown parameters: ylim
>
```

And my sessionInfo() is:

```
> sessionInfo()
R version 3.2.3 (2015-12-10)
```

```
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 14.04.4 LTS

locale:
[1] LC_CTYPE=en_CA.UTF-8LC_NUMERIC=C
[3] LC_TIME=en_CA.UTF-8LC_COLLATE=en_CA.UTF-8
[5] LC_MONETARY=en_CA.UTF-8LC_MESSAGES=en_CA.UTF-8
[7] LC_PAPER=en_CA.UTF-8LC_NAME=C
[9] LC_ADDRESS=CLC_TELEPHONE=C
[11] LC_MEASUREMENT=en_CA.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] parallel stats4 stats graphics grDevices utils datasets
[8] methods base

other attached packages:
[1] gplots_2.17.0 RColorBrewer_1.1-2
[3] vsn_3.38.0DESeq2_1.10.1
[5] RcppArmadillo_0.6.500.4.0Rcpp_0.12.3
[7] SummarizedExperiment_1.0.2 Biobase_2.30.0
[9] GenomicRanges_1.22.4GenomeInfoDb_1.6.3
[11] IRanges_2.4.7S4Vectors_0.8.11
[13] BiocGenerics_0.16.1

loaded via a namespace (and not attached):
[1] genefilter_1.52.1gtools_3.5.0 locfit_1.5-9.1
[4] splines_3.2.3 lattice_0.20-33 colorspace_1.2-6
[7] survival_2.38-3XML_3.98-1.3foreign_0.8-66
[10] DBI_0.3.1 BiocParallel_1.4.3 affy_1.48.0
[13] lambda.r_1.1.7 affyio_1.40.0 plyr_1.8.3
[16] zlibbioc_1.16.0 munsell_0.4.3 gtable_0.1.2
[19] futile.logger_1.4.1caTools_1.17.1 labeling_0.3
[22] latticeExtra_0.6-28 geneplotter_1.48.0BiocInstaller_1.20.1
[25] AnnotationDbi_1.32.3preprocessCore_1.32.0 acepack_1.3-3.3
[28] KernSmooth_2.23-15xtable_1.8-2 scales_0.3.0
[31] limma_3.26.8 gdata_2.17.0 Hmisc_3.17-2
[34] annotate_1.48.0XVector_0.10.0 gridExtra_2.0.0
[37] ggplot2_2.0.0 digest_0.6.9 grid_3.2.3
[40] bitops_1.0-6 RSQLite_1.0.0 Formula_1.2-1
[43] cluster_2.0.3 futile.options_1.0.0 rpart_4.1-10
[46] nnet_7.3-12
```

Dave Wheeler
on **February 25, 2016 at 11:02 pm** said:

That is strange. Your command works on mymachine. I checked the manual for vsn and the meanSdPlot still has that paramater (http://bioconductor.org/packages/release/bioc/manuals/vsn/man/vsn.pdf).

> Perhaps there is a name clash with another libraryTry calling it spec using the package name ie
>
> vsn::meanSdPlot(log2(counts(dds,normalized=TRUE)[notAllZero,] + 1), ylim = c(0,2.5))

yifangt
on **February 26, 2016 at 3:15 pm** said:

Thanks Dave!
No, calling the function with package name still gave the same error!
However, it worked if the ylim parameter is removed. Do not know why?!

Pingback: RNAseq pipeline – alignment to DE analysis | Gene

FENG Lei
on **March 1, 2016 at 9:18 am** said:

Reblogged this on Gene and commented:
Good example.

Mayank Kaashyap
on **March 7, 2016 at 8:12 am** said:

Hi Dave,
Thanks for such a nice tutorial. May I please know what values are used to build up a heatmap. Do SIM means correlate to RPKM vlaues. Can I choose few genes of interest from the table to plot a vsd transformed heatmap?

> **Dave Wheeler**
> on **March 7, 2016 at 7:20 pm** said:

Hi, I’m not sure I understand your questions. But the heatmap is made from variance stabilised read counts. I guess the size factor normalised read counts will correlate with RPKM values, but they are different normalisation strategies and should not be used interchangeably and RPKM normalised values should **never** be used with DESeq2. If you want to do a heatmap with just a few genes, do the variance stabilisation as before ie vsd<-varianceStabilizingTransformation(dds, blind=TRUE), then you need to collect the 'index' (probably not R terminology) of the genes you are interested in and assign them to the value "select" in the tutorial as an array If you want to see all the index for the available genes use rownames(res), use the numbers for each of your genes in your "select" variable.

Mayank Kaashyap
on **March 8, 2016 at 1:37 am** said:

Thanks, Dave!!

Mayank
on **March 15, 2016 at 1:03 am** said:

Hi Dave,
Is there a need to calculate row z-scores?

Dave Wheeler
on **March 15, 2016 at 7:00 pm** said:

I think you are thinking microarrays? The is no need to calculate z-scores. DESeq normalises the counts and calculates ajusted p-values for the differential expression calls.
Cheers

Christopher Brand
on **May 23, 2016 at 3:02 pm** said:

Thank you so much, this was such a great help in learning

Lalit
on **September 15, 2016 at 8:15 am** said:

Hii, first of all its a nice tutorial. I was replicating your results taking test files provided by you. At one step I found one problem. When I used the commands;
par(mai=ifelse(1:4px ord ord <- ord[px[ord] ord last vstcol matplot(px[ord], cbind(assay(vsd)[, 1], log2(px))[ord,], type=l, lty=1, col=vstcol, xlab='n', ylab='f(n)')
I found this error
Error in str2vec(type) : object 'l' not found
Can you please help me regarding this?
Best,

> **Dave Wheeler**
> on **September 15, 2016 at 9:54 pm** said:
>
> Hi There, sorry about that.
> If these things happend best to check the help using the '?':
> ?matplot
> matplot(x, y, type = "p", lty = 1:5, lwd = 1, lend = par("lend"),
> pch = NULL,
> col = 1:6, cex = NULL, bg = NA,
> xlab = NULL, ylab = NULL, xlim = NULL, ylim = NULL,
> …, add = FALSE, verbose = getOption("verbose"))
>
> See how the type="p". So it looks like it is missing the quotes (sorry).
> The correct command should be:
>
> matplot(px[ord], cbind(assay(vsd)[, 1], log2(px))[ord,], type="l", lty=1, col=vstcol, xlab='n', ylab='f(n)')

Mayank Kaashyap
on **October 19, 2016 at 7:22 am** said:

Hi Dave,
When levels=c('untreated','treated') then why Data frame shows log2FoldChange (treated vs untreated)??

Dave Wheeler
on **October 20, 2016 at 1:23 am** said:

Hi, I think it is ok, but best check, just look at the normalised read counts (across the treatments and make sure they move in the same direction as the fold change calls (always good to do this as a sanity check anyway)