MIS-DESeq2

Sunit Jain

January 6, 2015

Contents

Dependencies

If you're unsure that you have all the pacakges required to run this workflow. Open the `Rmd` file in your favorite text editor (I used RStudio) and change the next line from `eval=FALSE` to `eval=TRUE`. Now, when you run this workflow, the dependencies should be installed first.

Generate a read count matrix using `htseq-count`

Sample command:

```
htseq-count -f bam -r name -t CDS -o scaffold.htseq.sam -i ID -q scaffold_sortedByName.bam
all_combined.gff
```

This command was run for each sample individually.

Merging duplicate genes

I performed a self blast and looked at results that had a percent identity greater than 98%, query coverage greater than 96% and a minimum alignment length of 500 bases. Once I had this subset, I screened out the hits to exons since we won't be considering them for this experiment anyway. I was left with the following two gene pairs:

- scaffold_344578___MIS_1109813.1 scaffold_133898___MIS_10093600.14
- scaffold_219988___MIS_10179608.12 scaffold_555373___MIS_1172265.1

that had high enough similarity based on the thresholds mentioned above that their count data needed to be merged. The perl script `mergeCounts.pl` was run on each htseq-count output individually in order to accomplish this. Here is a sample command used for one of the htseq-count outputs:

```
perl mergeCounts.pl -l realDuplicateGenes.list -tsv Day_1.htseqCount.tsv -o Day_1.htseqCount.merged.tsv
```

where, realDuplicateGenes.list contains the two gene pairs mentioned above.

Import Counts into DESeq2

Once we were satisfied with the genes and their counts. We imported the count data into DESeq2.

Reads per Sample

```
##   Day_1   Day_2   Day_3 Night_4 Night_5 Night_6
## 2739735  492104  691689 1105737 1587917  969992
```

Filtering the data

Get rid of genes which did not occur frequently enough. Here we say, lets get rid of genes with counts >=20 in at least 2 samples.

```
##   Day_1   Day_2   Day_3 Night_4 Night_5 Night_6
## 2667957  465453  663671 1079669 1501836  892686
```

This reduces the dataset from 1464832 tags to about 1317. For the filtered tags, there is very little power to detect differential expression, so little information is lost by filtering.

Exploring the Dataset

The `rlog` transformation

Many common statistical methods for exploratory analysis of multidimensional data, especially methods for clustering and ordination (e.g., principal-component analysis and the like), work best for (at least approximately) homoskedastic data; this means that the variance of an observed quantity (here, the expression strength of a gene) does not depend on the mean. In RNA-Seq data, however, variance grows with the mean. For example, if one performs PCA (principal components analysis) directly on a matrix of normalized read counts, the result typically depends only on the few most strongly expressed genes because they show the largest absolute differences between samples. A simple and often used strategy to avoid this is to take the logarithm of the normalized count values plus a small pseudocount; however, now the genes with low counts tend to dominate the results because, due to the strong Poisson noise inherent to small count values, they show the strongest relative differences between samples.

As a solution, DESeq2 offers the regularized-logarithm transformation, or rlog for short. For genes with high counts, the rlog transformation differs not much from an ordinary log2 transformation. For genes with lower counts, however, the values are shrunken towards the genes' averages across all samples. Using an empirical Bayesian prior on inter-sample differences in the form of a ridge penalty, this is done such that the rlog-transformed data are approximately homoskedastic.

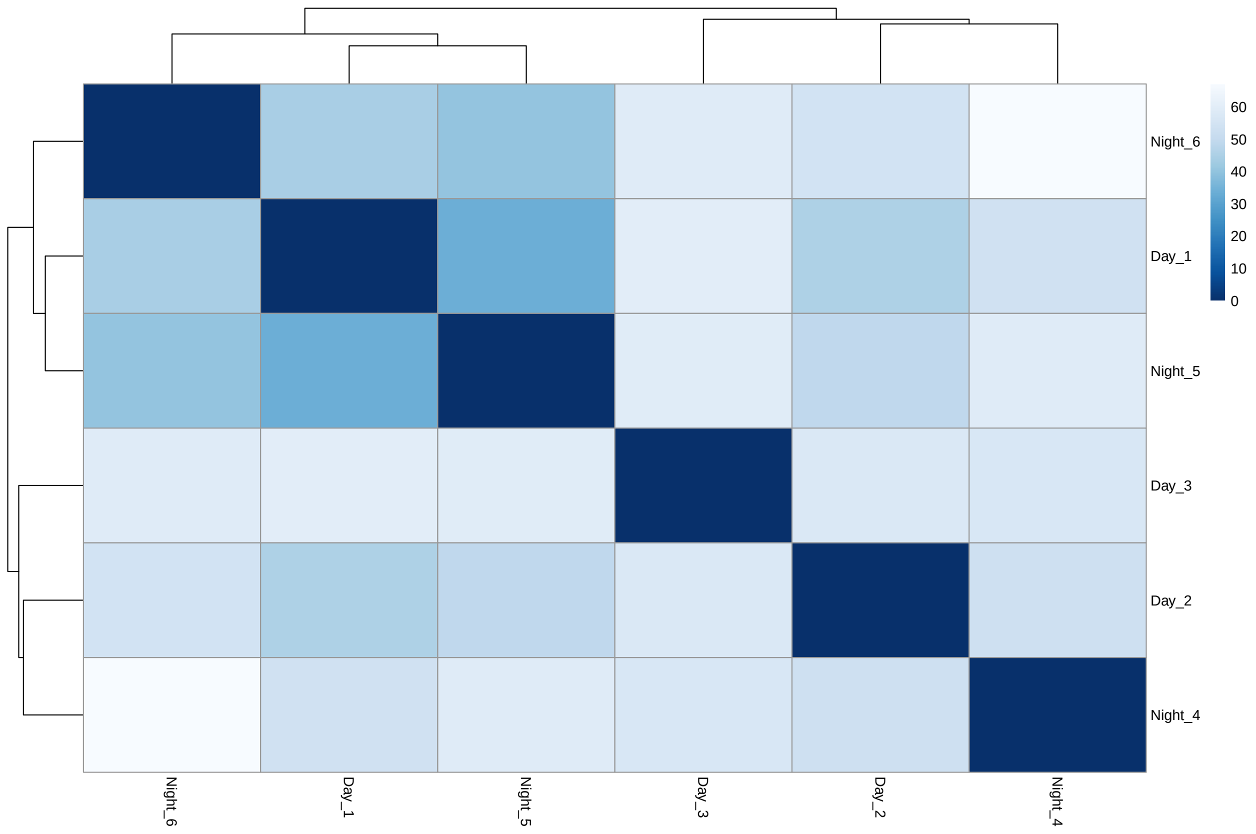
Note: the rlog transformation is provided for applications other than differential testing. For differential testing we recommend the DESeq function applied to raw counts, as described later in this workflow, which also takes into account the dependence of the variance of counts on the mean value during the dispersion estimation step.

Sample distances

A useful first step in an RNA-Seq analysis is often to assess overall similarity between samples: Which samples are similar to each other, which are different? Does this fit to the expectation from the experiment's design? We use the R function `dist` to calculate the Euclidean distance between samples. To avoid that the distance measure is dominated by a few highly variable genes, and have a roughly equal contribution from all genes, we use it on the rlog-transformed data:

```
##            Day_1    Day_2    Day_3  Night_4  Night_5
## Day_2   45.47282
## Day_3   60.18270 57.37411
## Night_4 53.93390 53.23661 56.68394
## Night_5 33.83684 49.17392 59.71269 58.99479
## Night_6 44.34909 54.61716 59.06997 67.11674 40.28136
```

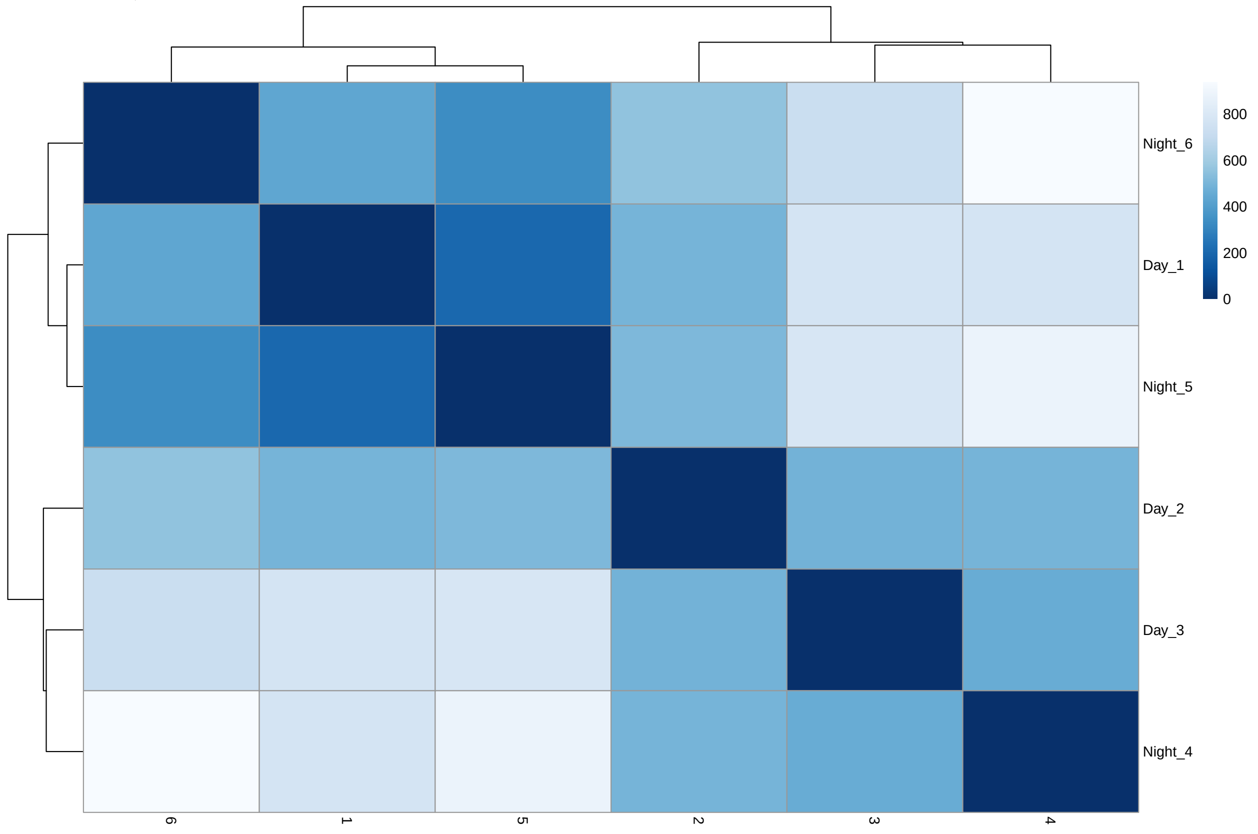
We visualize the distances in a heatmap:



60
50
40
30
20
10
0
Night_6
Day_1
Night_5
Day_3
Day_2
Night_4
Night_6
Day_1
Night_5
Day_3
Day_2
Night_4

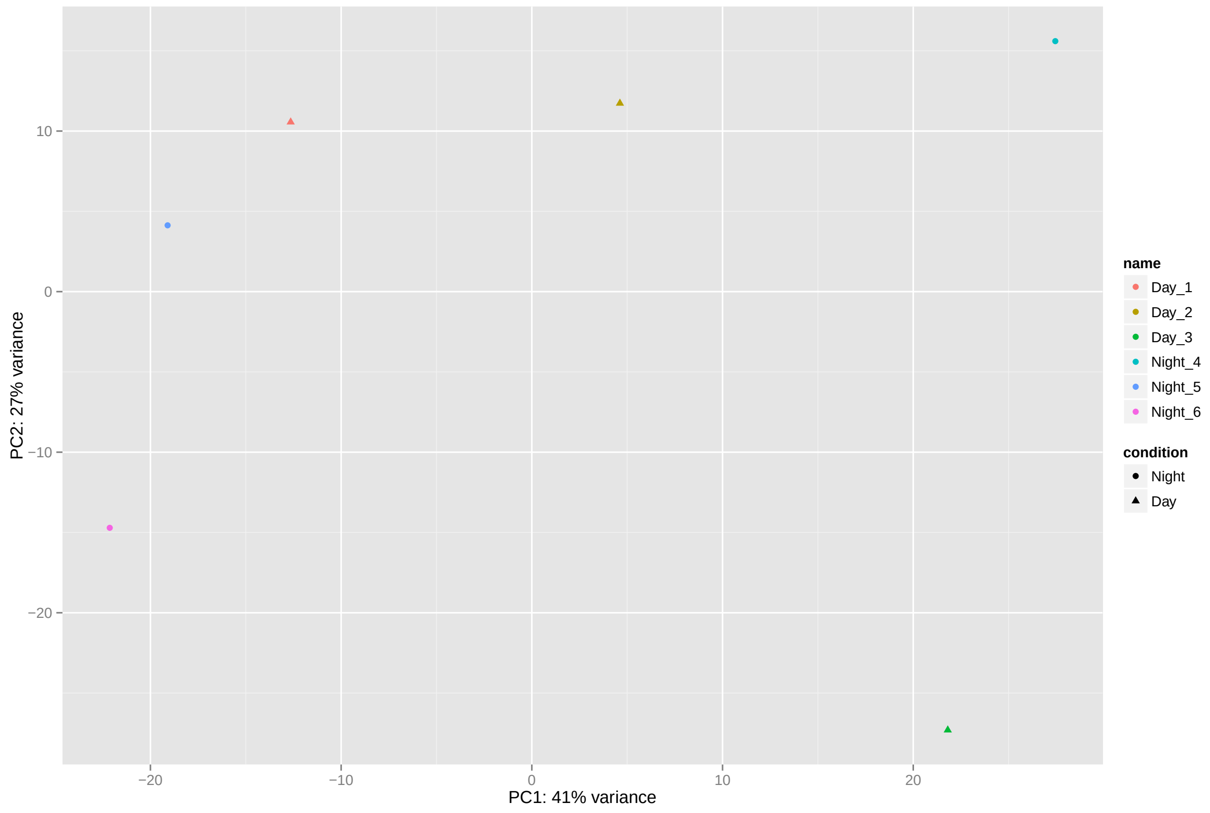
Poisson Distance

Another option for calculating sample distances is to use the Poisson Distance, implemented in the CRAN package PoiClaClu. Similar to the transformations offered in DESeq2, this measure of dissimilarity also takes the variance structure of counts into consideration when calculating the distances between samples. The PoissonDistance function takes the original count matrix (not normalized) with samples as rows instead of columns, so we need to transpose the counts in dds.



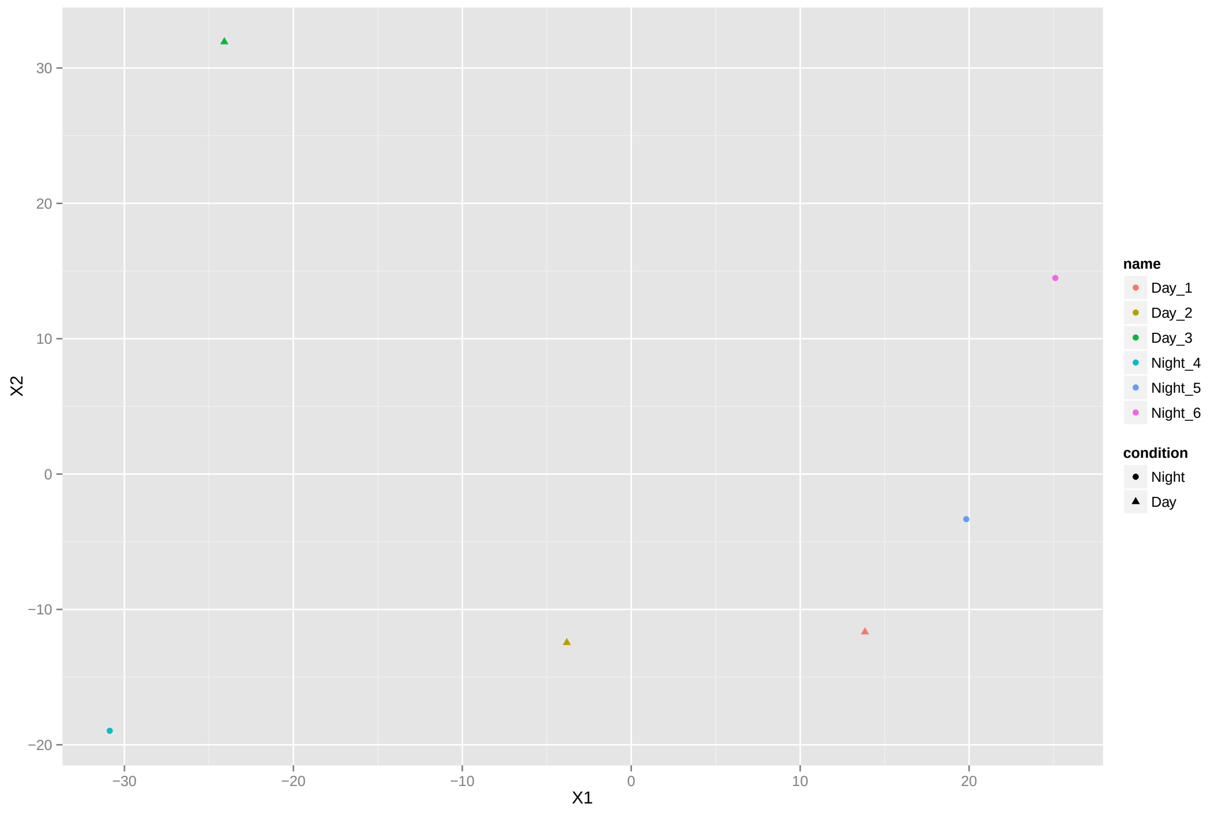
PCA plot

Another way to visualize sample-to-sample distances is a principal-components analysis (PCA). In this ordination method, the data points (i.e., here, the samples) are projected onto the 2D plane such that they spread out in the two directions which explain most of the differences in the data. The x-axis is the direction (or principal component) which separates the data points the most. The amount of the total variance which is contained in the direction is printed in the axis label.

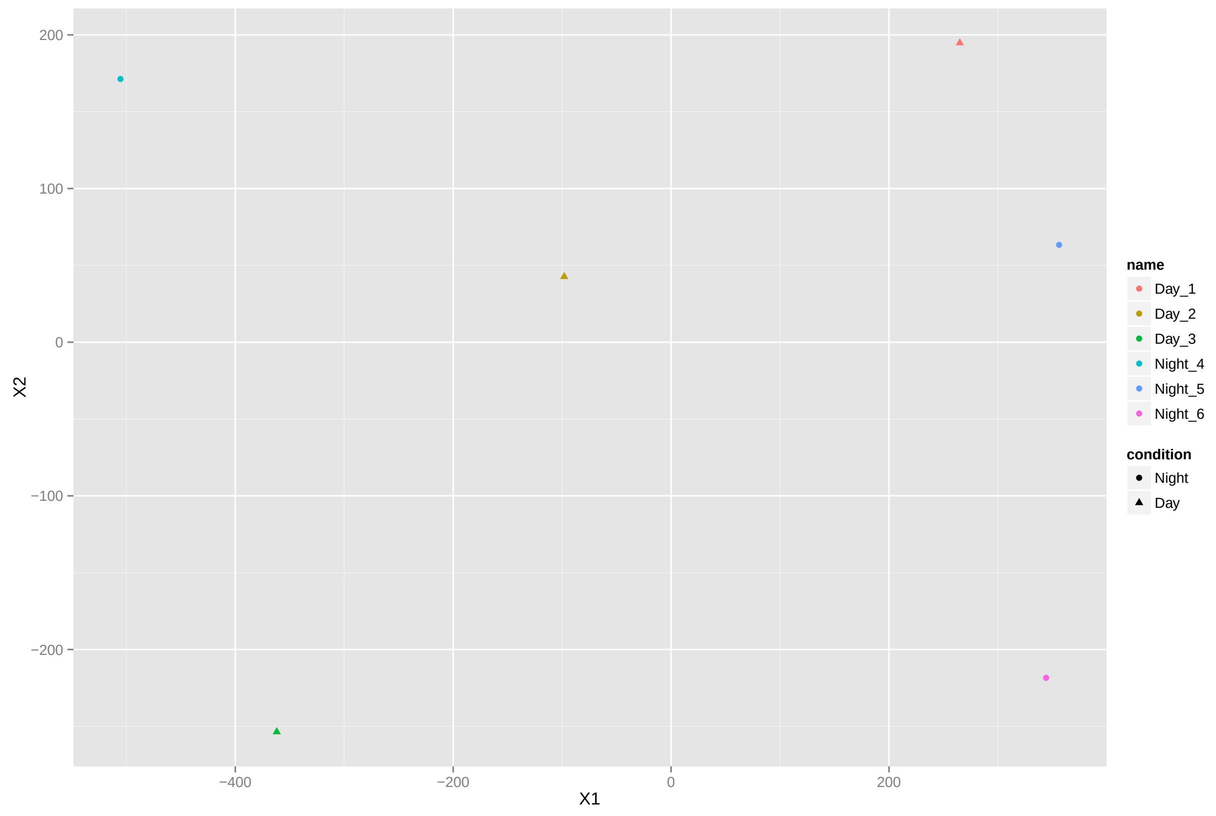


MDS plot

Another plot, very similar to the PCA plot, can be made using the multidimensional scaling (MDS) function in base R. This is useful when we don't have the original data, but only a matrix of distances. Here we have the MDS plot for the distances calculated from the rlog transformed counts:



And here from the PoissonDistance:



## Counts In order to normalise the raw counts we will start by determining the relative library sizes, or size factors for each library. For example, if the counts of the expressed genes in one sample are, on average, twice as high as in another, the size factor for the first sample should be twice as large as the one for the other sample. These size factors can be obtained with the function `estimateSizeFactors`:

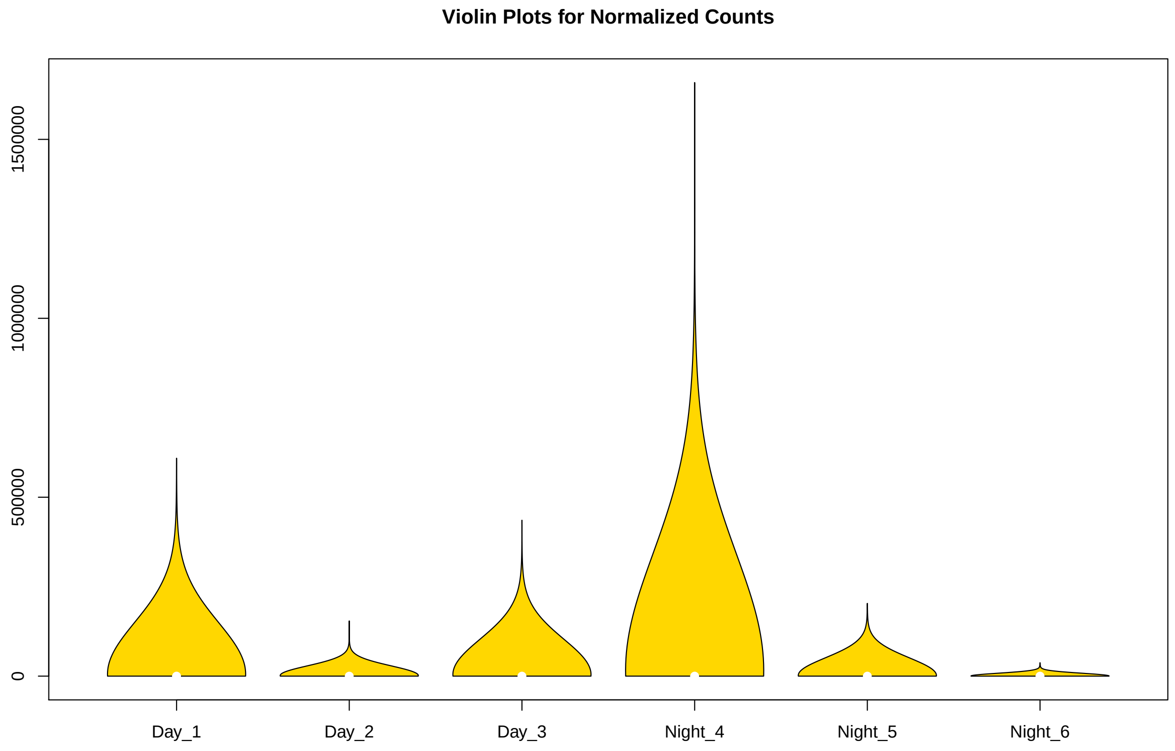
```
##     Day_1     Day_2     Day_3   Night_4   Night_5   Night_6
## 1.7426964 0.6564081 0.5451415 0.2674957 2.5752550 2.9477557
```

Raw vs Normalized Counts

Once we have this information, the normalised data is obtained by dividing each column of the count table by the corresponding size factor. We can perform this calculation by calling the function counts with a the `normalized` argument set as `TRUE`. Since we won't be normalizing this data, we'll set it as `FALSE`

Normalized

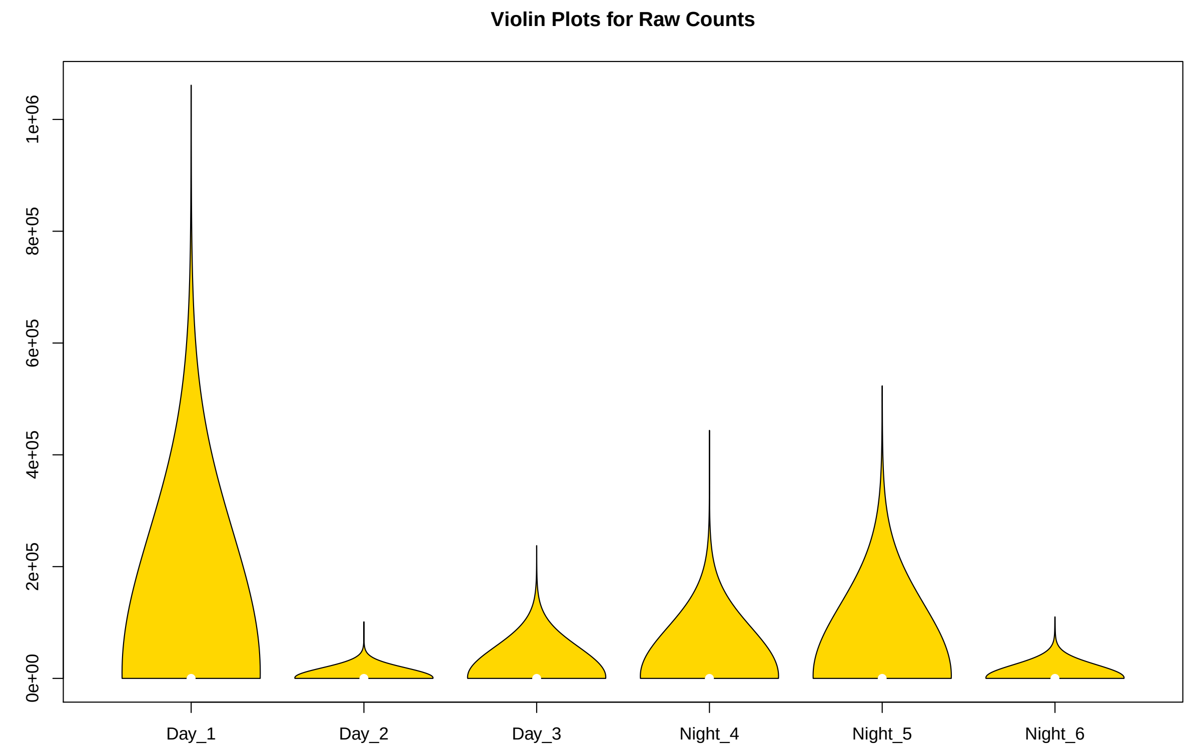
```
##                                       Day_1      Day_2       Day_3
## scaffold_10014__MIS_10010015.19 12.050292   7.617213    1.834386
## scaffold_10014__MIS_10010015.7  69.432634  97.500327   51.362816
## scaffold_100551__MIS_10060488.5 35.003229  35.039180    5.503159
## scaffold_101047__MIS_10060980.8  5.164411 108.164425 1373.955323
## scaffold_10109__MIS_10010110.17 12.624115   1.523443    1.834386
## scaffold_10109__MIS_10010110.20  9.754998  18.281311   12.840704
##                                    Night_4    Night_5    Night_6
## scaffold_10014__MIS_10010015.19   11.21513   9.319466  20.015227
## scaffold_10014__MIS_10010015.7   231.77943 100.960877  90.238143
## scaffold_100551__MIS_10060488.5   41.12216  12.037643   1.696206
## scaffold_101047__MIS_10060980.8 2138.35214  15.920754 439.995757
## scaffold_10109__MIS_10010110.17   52.33729   3.106489  10.516475
## scaffold_10109__MIS_10010110.20   22.43027  14.367509  34.941837
```



```
##      Day_1             Day_2              Day_3
##  Min.   :     0.0   Min.   :     0.00   Min.   :     0.0
##  1st Qu.:    11.5   1st Qu.:     9.14   1st Qu.:     7.3
##  Median :    19.5   Median :    19.80   Median :    20.2
##  Mean   :  1162.4   Mean   :   538.41   Mean   :   924.4
##  3rd Qu.:    46.5   3rd Qu.:    53.32   3rd Qu.:    58.7
##  Max.   :608960.3   Max.   :154251.61   Max.   :435661.2
##     Night_4            Night_5            Night_6
##  Min.   :      0.0   Min.   :     0.00   Min.   :    0.00
##  1st Qu.:      7.5   1st Qu.:    10.10   1st Qu.:    9.16
##  Median :     18.7   Median :    17.09   Median :   16.96
##  Mean   :   3064.7   Mean   :   442.81   Mean   :  229.94
##  3rd Qu.:     63.6   3rd Qu.:    41.16   3rd Qu.:   42.07
##  Max.   :1658819.2   Max.   :203224.54   Max.   :37304.99
```

Raw

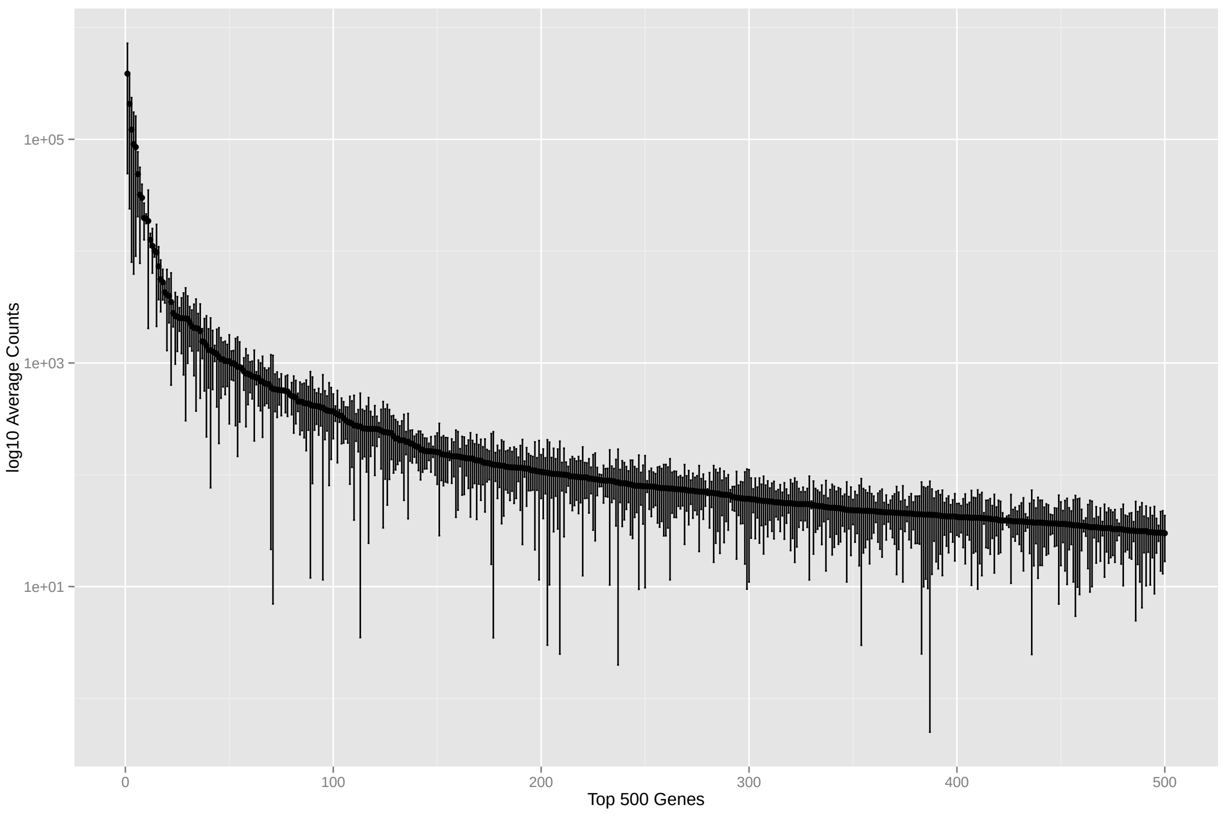
```
##                                    Day_1 Day_2 Day_3 Night_4 Night_5 Night_6
## scaffold_10014__MIS_10010015.19     21     5     1       3      24      59
## scaffold_10014__MIS_10010015.7     121    64    28      62     260     266
## scaffold_100551__MIS_10060488.5     61    23     3      11      31       5
## scaffold_101047__MIS_10060980.8      9    71   749     572      41    1297
## scaffold_10109__MIS_10010110.17     22     1     1      14       8      31
## scaffold_10109__MIS_10010110.20     17    12     7       6      37     103
```



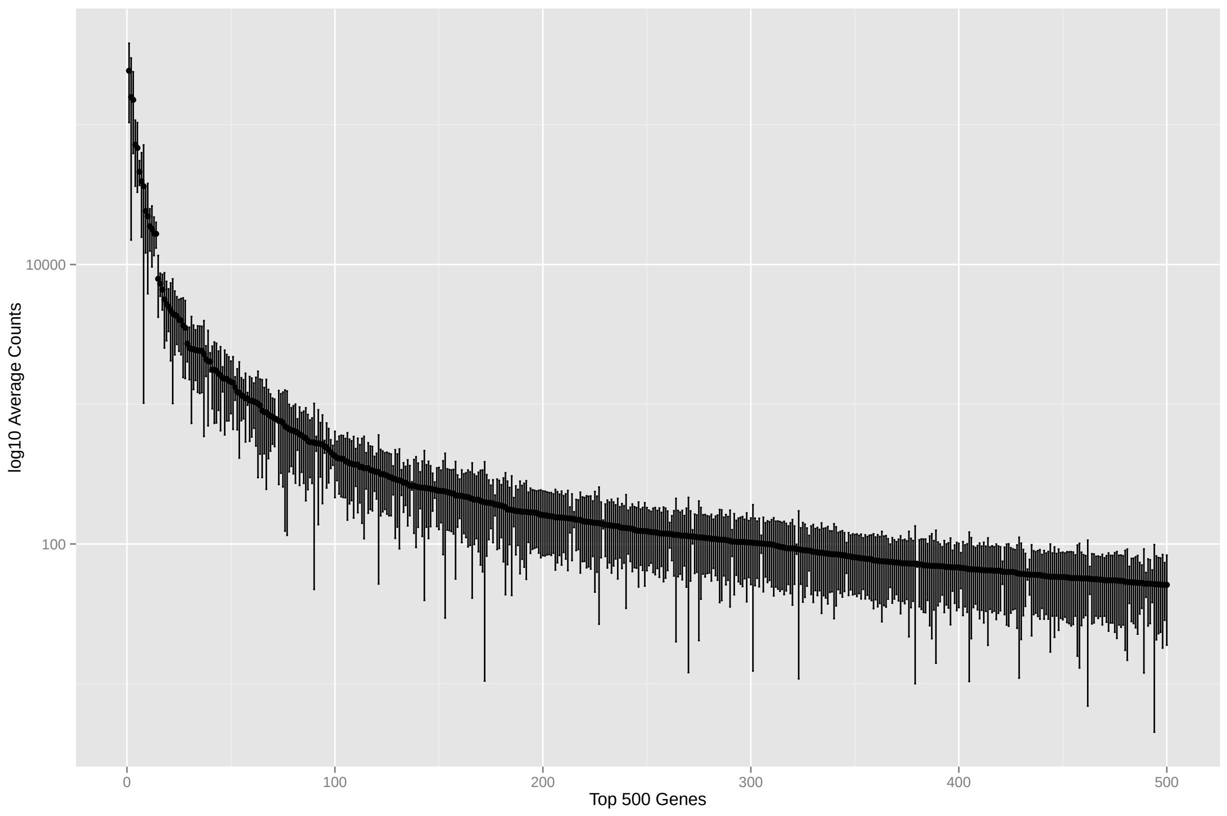
```
##      Day_1             Day_2              Day_3
##  Min.   :      0   Min.   :     0.0   Min.   :     0.0
##  1st Qu.:     20   1st Qu.:     6.0   1st Qu.:     4.0
##  Median :     34   Median :    13.0   Median :    11.0
##  Mean   :   2026   Mean   :   353.4   Mean   :   503.9
##  3rd Qu.:     81   3rd Qu.:    35.0   3rd Qu.:    32.0
##  Max.   :1061233   Max.   :101252.0   Max.   :237497.0
##     Night_4            Night_5           Night_6
##  Min.   :     0.0   Min.   :     0   Min.   :     0.0
##  1st Qu.:     2.0   1st Qu.:    26   1st Qu.:    27.0
##  Median :     5.0   Median :    44   Median :    50.0
##  Mean   :   819.8   Mean   :  1140   Mean   :   677.8
##  3rd Qu.:    17.0   3rd Qu.:   106   3rd Qu.:   124.0
##  Max.   :443727.0   Max.   :523355   Max.   :109966.0
```

Rank Abundance

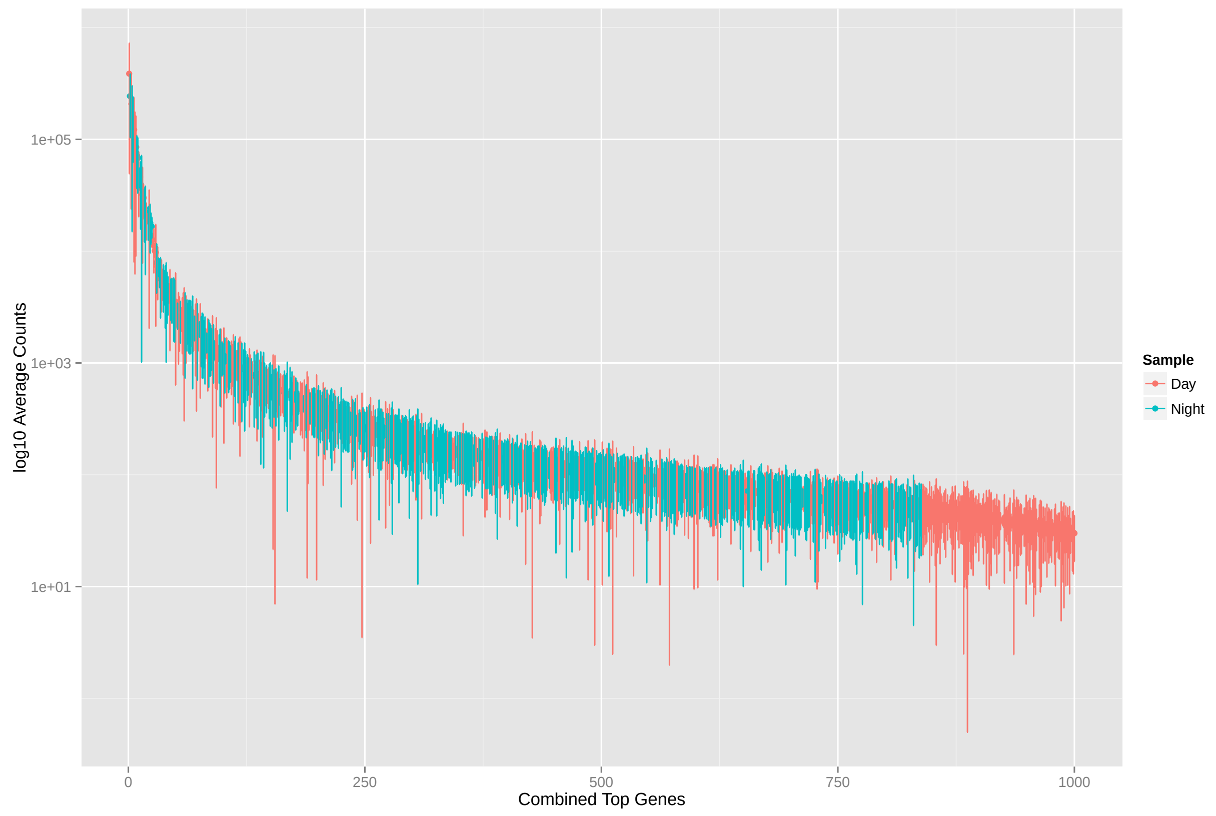
Day Counts



Night Counts



Day vs Night


1e+05
1e+03
1e+01
log10 Average Counts
0
250
500
750
1000
Combined Top Genes
Sample
Day
Night

Differential Expression

Differential expression was calculater using the DESeq2 wrapper function over 4 processors.

Results

As `res` is a DataFrame object, it carries metadata with information on the meaning of the columns:

```
## DataFrame with 6 rows and 2 columns
##                         type                                    description
##                  <character>                                    <character>
## baseMean        intermediate       mean of normalized counts for all samples
## log2FoldChange       results log2 fold change (MAP): condition Day vs Night
## lfcSE                results         standard error: condition Day vs Night
## stat                 results         Wald statistic: condition Day vs Night
## pvalue               results      Wald test p-value: condition Day vs Night
## padj                 results                        BH adjusted p-values
```

```
##
## out of 1317 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)     : 5, 0.38%
## LFC < 0 (down)   : 0, 0%
## outliers [1]     : 74, 5.6%
## low counts [2]   : 590, 45%
## (mean count < 19.4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Multiple testing

Novices in high-throughput biology often assume that thresholding these p values at a low value, say 0.05, as is often done in other settings, would be appropriate – but it is not. We briefly explain why: There are 72 genes with a p value below 0.05 among the 1243 genes, for which the test succeeded in reporting a p value.

Now, assume for a moment that the null hypothesis is true for all genes, i.e., no gene is affected by the treatment with dexamethasone. Then, by the definition of p value, we expect up to `5%` of the genes to have a p value below 0.05. This amounts to 62 genes. If we just considered the list of genes with a p value below 0.05 as differentially expressed, this list should therefore be expected to contain up to 62.15/72=0.8631944% false positives.

DESeq2 uses the Benjamini-Hochberg (BH) adjustment as described in the base R p.adjust function; in brief, this method calculates for each gene an adjusted p value which answers the following question: if one called significant all genes with a p value less than or equal to this gene's p value threshold, what would be the fraction of false positives (the false discovery rate, FDR) among them (in the sense of the calculation outlined above)? These values, called the BH-adjusted p values, are given in the column padj of the res object. Hence, if we consider a fraction of 10% false positives acceptable, we can consider all genes with an adjusted p value below 10% = 0.1 as significant. How many such genes are there?

```
## [1] 5
```

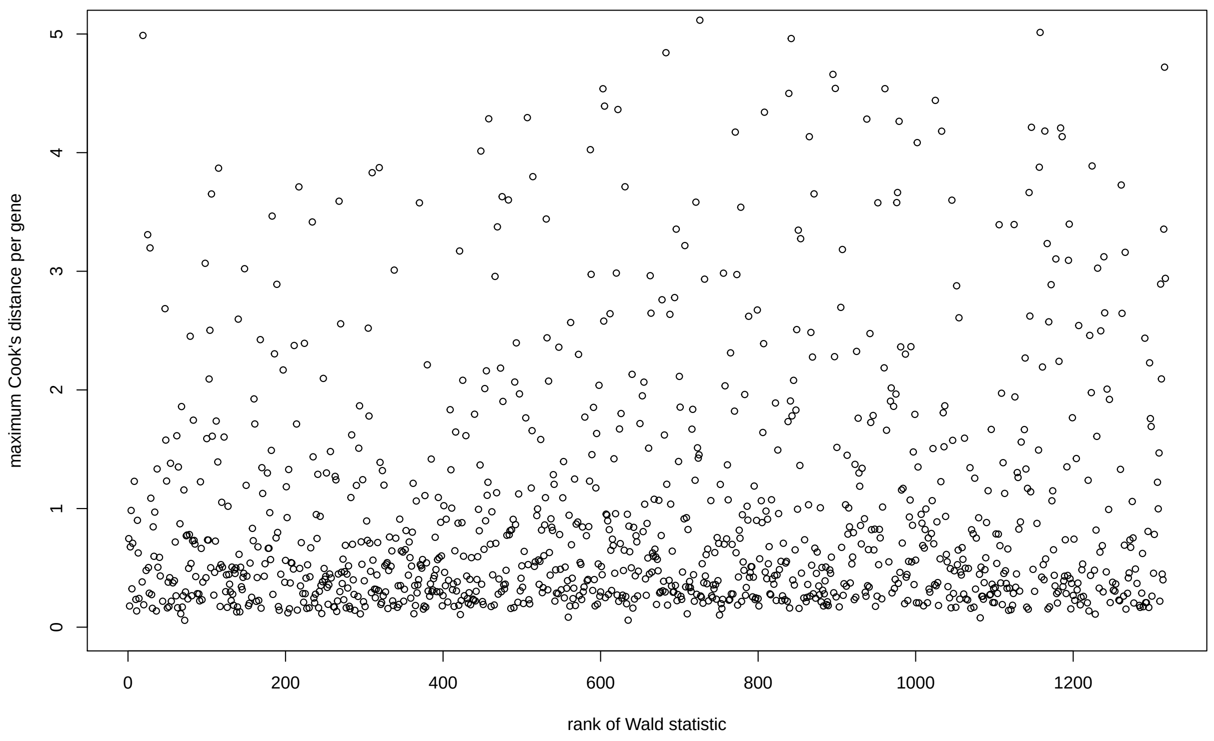
We subset the results table to these genes and then sort it by the log2 fold change estimate to get the significant genes with the strongest down-regulation.

```
## log2 fold change (MAP): condition Day vs Night
## Wald test p-value: condition Day vs Night
## DataFrame with 5 rows and 6 columns
##                                 baseMean log2FoldChange     lfcSE
##                                <numeric>      <numeric> <numeric>
## scaffold_160512__MIS_10120163.1  23.60387       2.202043 0.6293079
## scaffold_41583__MIS_10036250.27  68.87220       2.456451 0.7059744
## scaffold_12010__MIS_10012011.12 163.67507       2.485618 0.7365799
## scaffold_42417__MIS_10006030.1   34.09297       2.768271 0.7530712
## scaffold_12010__MIS_10012011.1   47.72121       2.926728 0.7275377
##                                     stat       pvalue       padj
##                                <numeric>    <numeric>  <numeric>
## scaffold_160512__MIS_10120163.1  3.499150 4.667442e-04 0.08200318
## scaffold_41583__MIS_10036250.27  3.479518 5.023166e-04 0.08200318
## scaffold_12010__MIS_10012011.12  3.374540 7.393928e-04 0.09656470
## scaffold_42417__MIS_10006030.1   3.675975 2.369426e-04 0.07736177
## scaffold_12010__MIS_10012011.1   4.022785 5.751396e-05 0.03755662
```

. . . and with the strongest upregulation.

```
## log2 fold change (MAP): condition Day vs Night
## Wald test p-value: condition Day vs Night
## DataFrame with 5 rows and 6 columns
##                                 baseMean log2FoldChange     lfcSE
##                                <numeric>      <numeric> <numeric>
## scaffold_12010__MIS_10012011.1   47.72121       2.926728 0.7275377
## scaffold_42417__MIS_10006030.1   34.09297       2.768271 0.7530712
## scaffold_12010__MIS_10012011.12 163.67507       2.485618 0.7365799
## scaffold_41583__MIS_10036250.27  68.87220       2.456451 0.7059744
## scaffold_160512__MIS_10120163.1  23.60387       2.202043 0.6293079
##                                     stat       pvalue       padj
##                                <numeric>    <numeric>  <numeric>
## scaffold_12010__MIS_10012011.1   4.022785 5.751396e-05 0.03755662
## scaffold_42417__MIS_10006030.1   3.675975 2.369426e-04 0.07736177
## scaffold_12010__MIS_10012011.12  3.374540 7.393928e-04 0.09656470
## scaffold_41583__MIS_10036250.27  3.479518 5.023166e-04 0.08200318
## scaffold_160512__MIS_10120163.1  3.499150 4.667442e-04 0.08200318
```

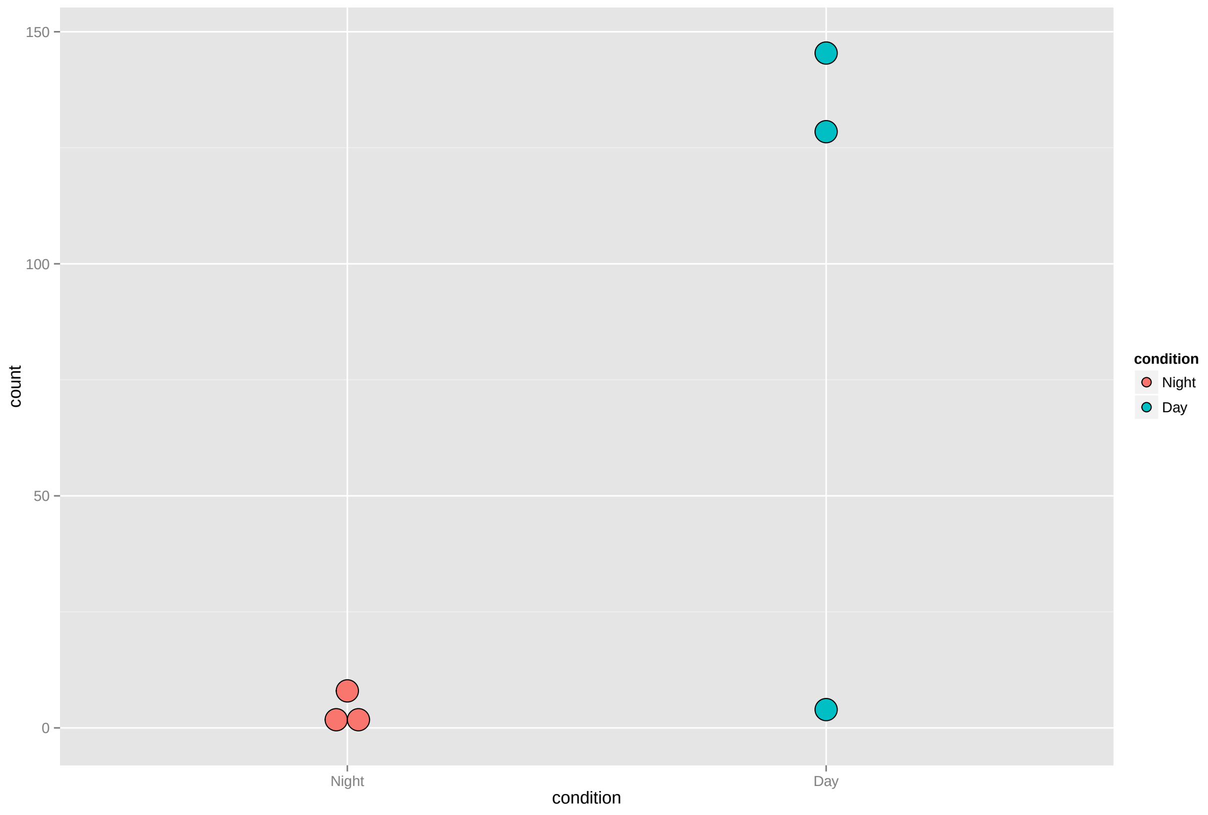
Cook's Distance per gene



Diagnostic Plots

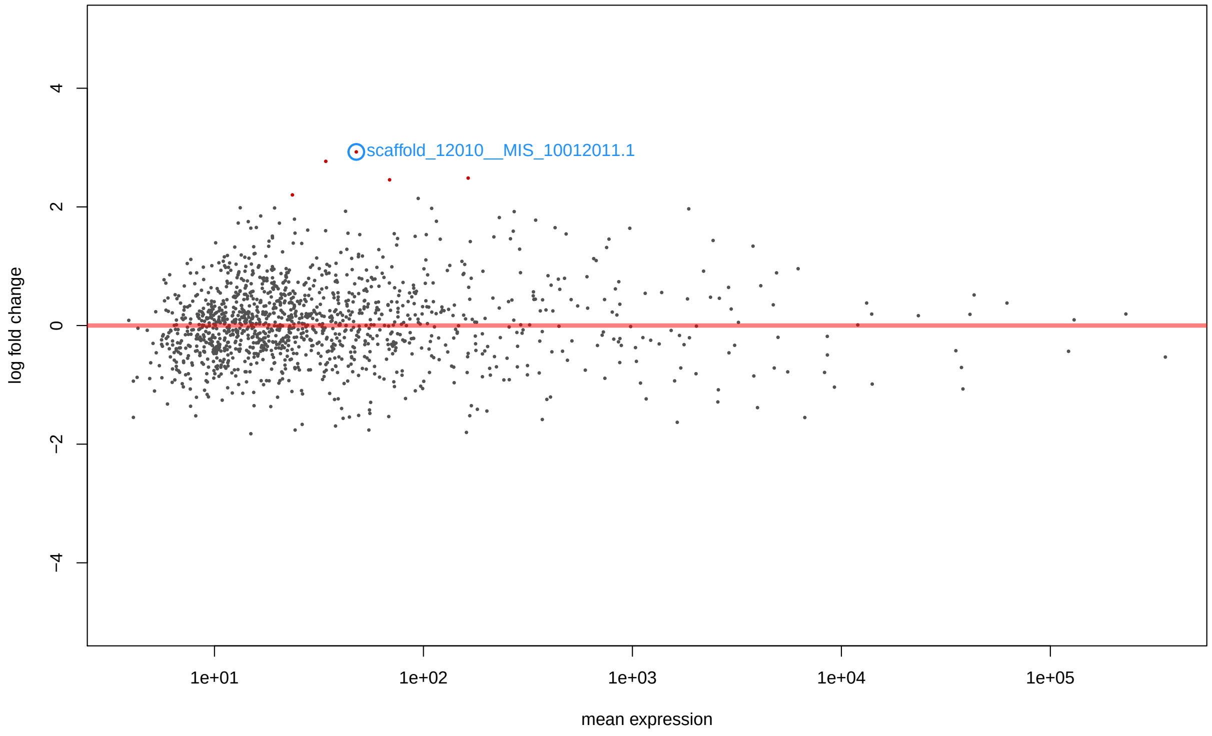
Plot Counts

A quick way to visualize the counts for a particular gene is to use the plotCounts function, which takes as arguments the DESeqDataSet, a gene name, and the group over which to plot the counts.



MA-Plots

An “MA-plot” provides a useful overview for an experiment with a two-group comparison. The log2 fold change for a particular comparison is plotted on the y-axis and the average of the counts normalized by size factor is shown on the x-axis (“M” for minus, because a log ratio is equal to log minus log, and “A” for average).

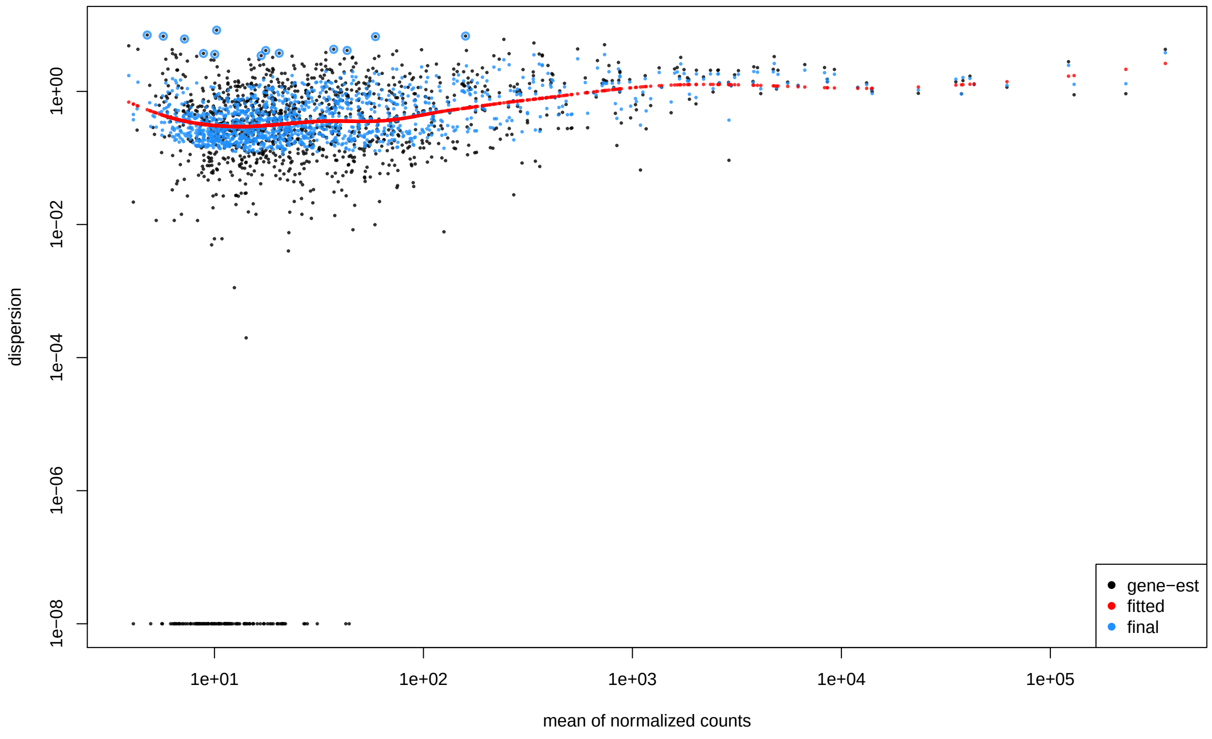


Each gene is represented with a dot. Genes with an adjusted p-value below a threshold (here 0.1, the default) are shown in red. The DESeq2 package incorporates a prior on log2 fold changes, resulting in moderated log2 fold changes from genes with low counts and highly variable counts, as can be seen by the narrowing of spread of points on the left side of the plot. This plot demonstrates that only genes with a large average normalized count contain sufficient information to yield a significant call.

Dispersion Estimataion

Whether a gene is called significant depends not only on its LFC but also on its within-group variability, which DESeq2 quantifies as the dispersion. For strongly expressed genes, the dispersion can be understood as a squared coefficient of variation: a dispersion value of 0.01 means that the gene's expression tends to differ by typically sqrt(0.01)=10% between samples of the same treatment group. For weak genes, the Poisson noise is an additional source of noise.

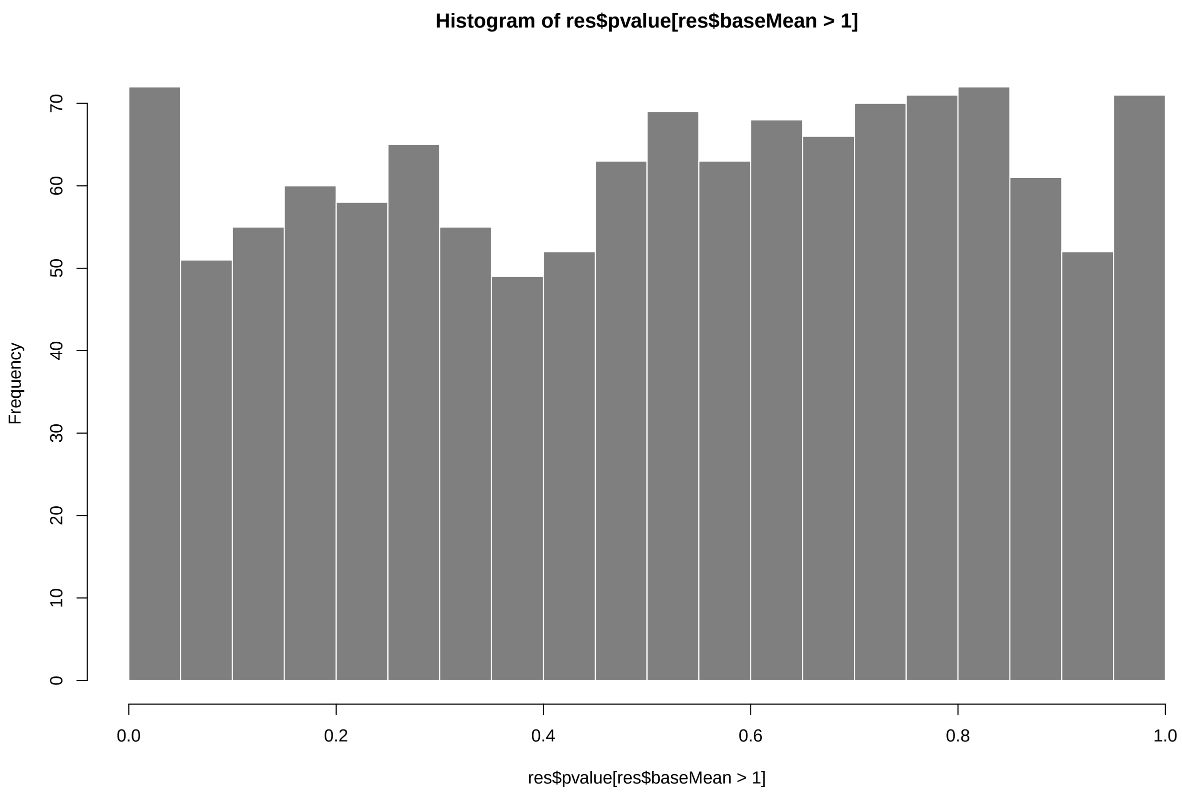
The function plotDispEsts visualizes DESeq2's dispersion estimates:



The black points are the dispersion estimates for each gene as obtained by considering the information from each gene separately. Unless one has many samples, these values fluctuate strongly around their true values. Therefore, we fit the red trend line, which shows the dispersions' dependence on the mean, and then shrink each gene's estimate towards the red line to obtain the final estimates (blue points) that are then used in the hypothesis test. The blue circles above the main "cloud" of points are genes which have high gene-wise dispersion estimates which are labelled as dispersion outliers. These estimates are therefore not shrunk toward the fitted trend line.

P-Value Histogram

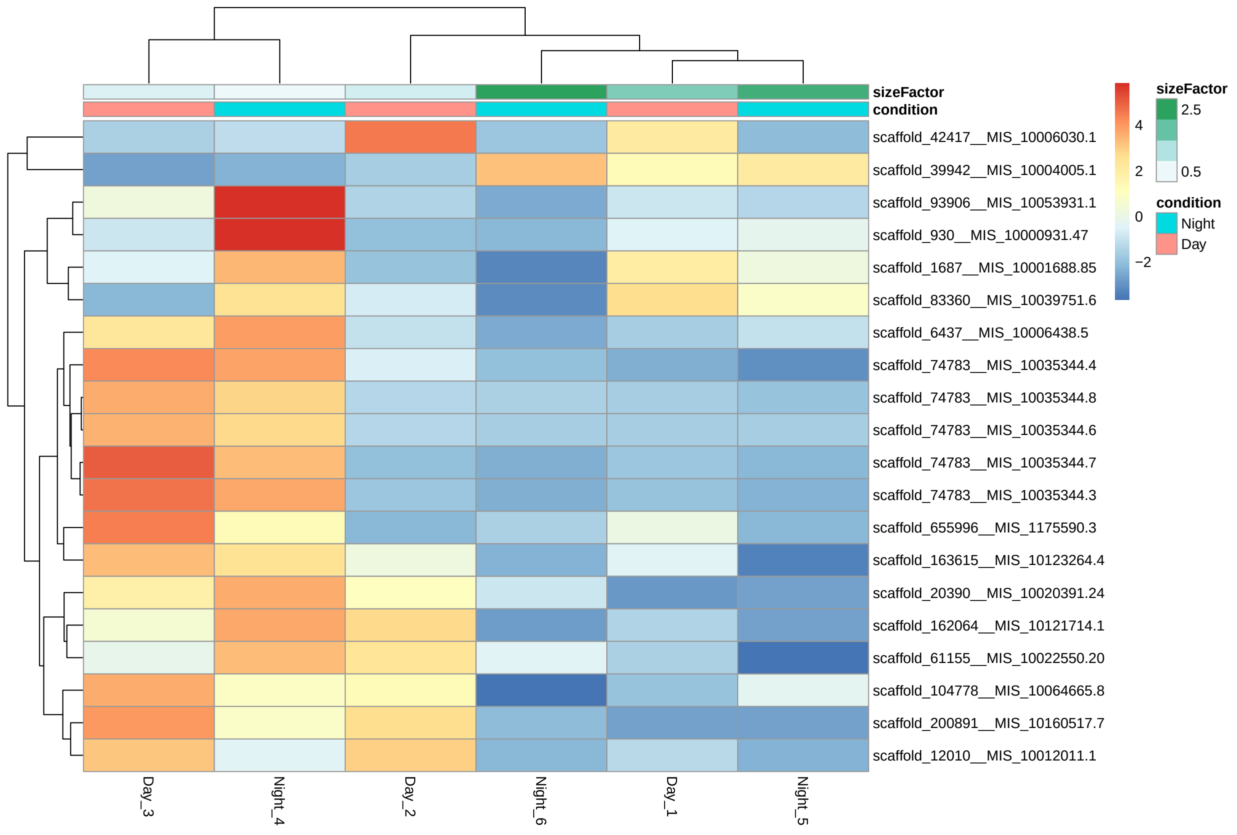
Another useful diagnostic plot is the histogram of the p values.



Gene clustering

In the sample distance heatmap made previously, the dendrogram at the side shows us a hierarchical clustering of the samples. Such a clustering can also be performed for the genes. Since the clustering is only relevant for genes that actually carry signal, one usually carries it out only for a subset of most highly variable genes. Here, for demonstration, let us select the 20 genes with the highest variance across samples. We will work with the rlog transformed counts:

The heatmap becomes more interesting if we do not look at absolute expression strength but rather at the amount by which each gene deviates in a specific sample from the gene's average across all samples. Hence, we center each genes' values across samples, and plot a heatmap. We provide the column side colors to help identify the treated samples (in blue) from the untreated samples (in grey).

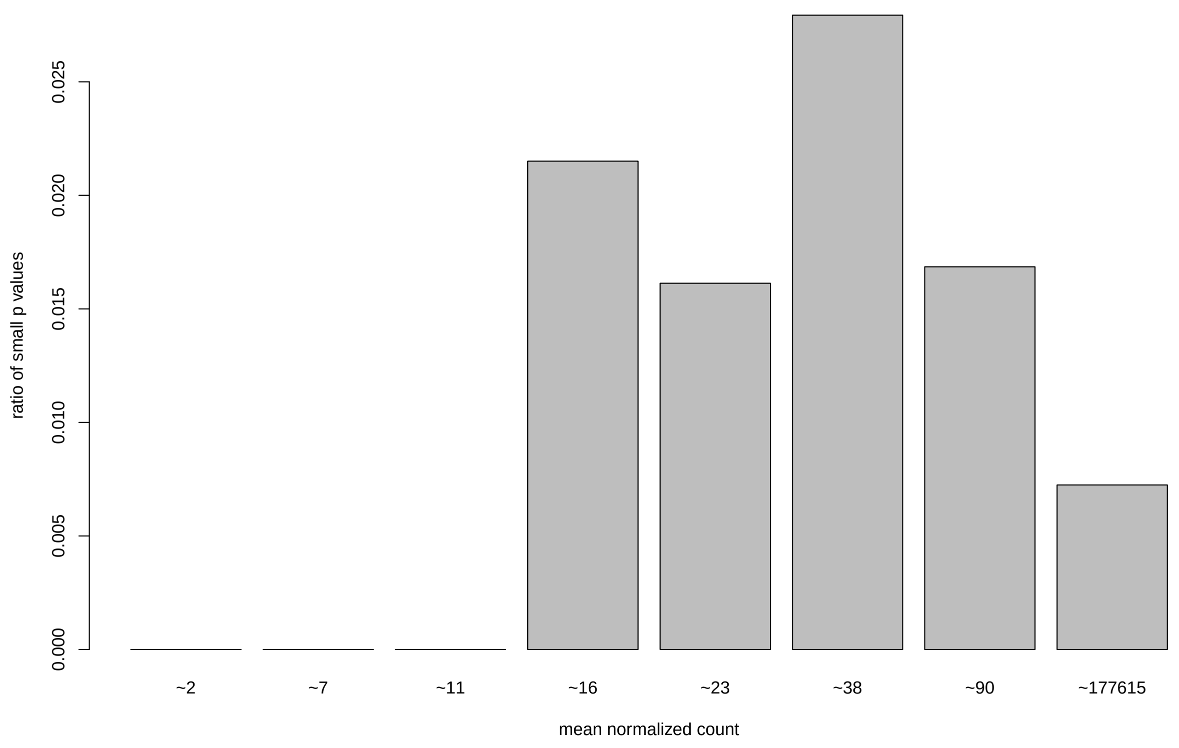


We can now see blocks of genes which covary across patients. Note that a set of genes at the top of the heatmap are separating the N061011 cell line from the others. At the bottom of the heatmap, we see a set of genes for which the treated samples have higher gene expression.

```
## log2 fold change (MAP): condition Day vs Night
## Wald test p-value: condition Day vs Night
## DataFrame with 6 rows and 7 columns
##                                   baseMean log2FoldChange     lfcSE
##                                  <numeric>      <numeric> <numeric>
## scaffold_12010__MIS_10012011.1    47.72121       2.926728 0.7275377
## scaffold_42417__MIS_10006030.1    34.09297       2.768271 0.7530712
## scaffold_160512__MIS_10120163.1   23.60387       2.202043 0.6293079
## scaffold_41583__MIS_10036250.27   68.87220       2.456451 0.7059744
## scaffold_12010__MIS_10012011.12 163.67507       2.485618 0.7365799
## scaffold_26660__MIS_10026661.5    41.26248      -1.564367 0.5022130
##                                         stat       pvalue       padj
##                                    <numeric>    <numeric>  <numeric>
## scaffold_12010__MIS_10012011.1   4.022785 5.751396e-05 0.03755662
## scaffold_42417__MIS_10006030.1   3.675975 2.369426e-04 0.07736177
## scaffold_160512__MIS_10120163.1  3.499150 4.667442e-04 0.08200318
## scaffold_41583__MIS_10036250.27  3.479518 5.023166e-04 0.08200318
## scaffold_12010__MIS_10012011.12  3.374540 7.393928e-04 0.09656470
## scaffold_26660__MIS_10026661.5  -3.114948 1.839774e-03 0.20022877
##                                                                 Name
##                                                          <character>
## scaffold_12010__MIS_10012011.1   scaffold_12010__MIS_10012011.1
## scaffold_42417__MIS_10006030.1   scaffold_42417__MIS_10006030.1
## scaffold_160512__MIS_10120163.1 scaffold_160512__MIS_10120163.1
## scaffold_41583__MIS_10036250.27 scaffold_41583__MIS_10036250.27
## scaffold_12010__MIS_10012011.12 scaffold_12010__MIS_10012011.12
## scaffold_26660__MIS_10026661.5   scaffold_26660__MIS_10026661.5
```

Independent Filtering

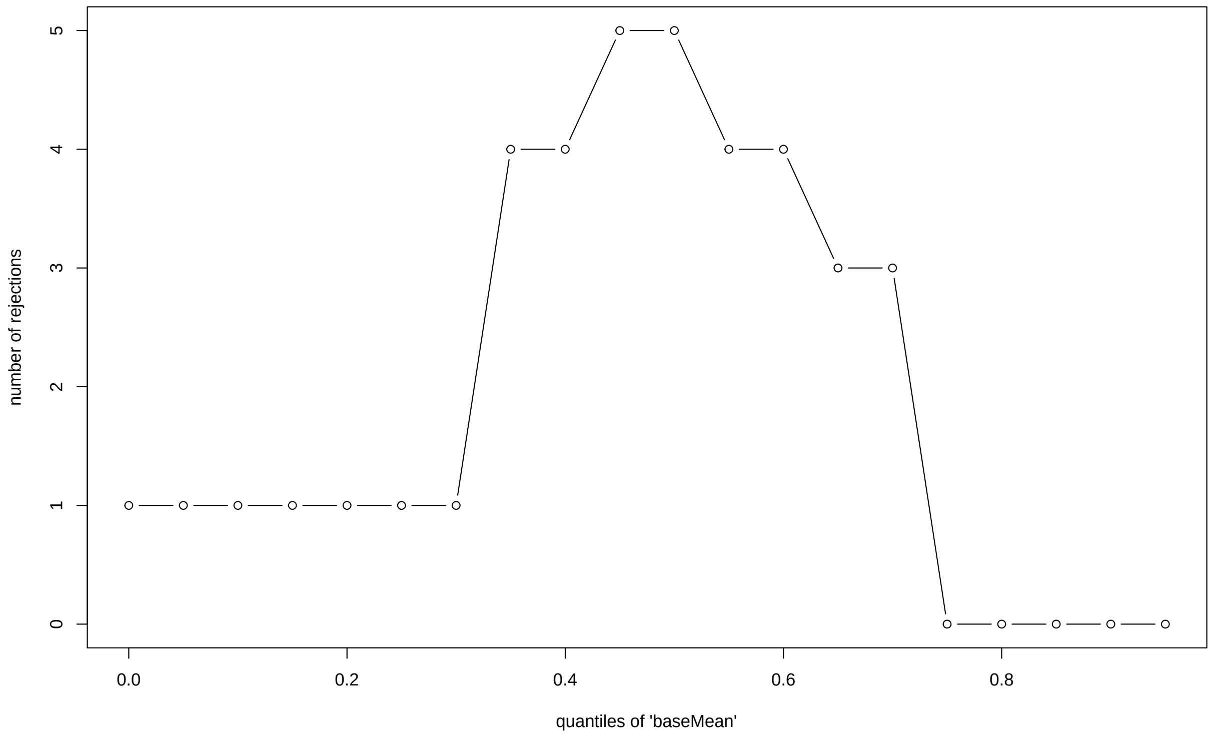
The MA plot highlights an important property of RNA-Seq data. For weakly expressed genes, we have no chance of seeing differential expression, because the low read counts suffer from so high Poisson noise that any biological effect is drowned in the uncertainties from the read counting. We can also show this by examining the ratio of small p values (say, less than, 0.01) for genes binned by mean normalized count:



At first sight, there may seem to be little benefit in filtering out these genes. After all, the test found them to be non-significant anyway. However, these genes have an influence on the multiple testing adjustment, whose performance improves if such genes are removed. By removing the weakly-expressed genes from the input to the FDR procedure, we can find more genes to be significant among those which we keep, and so improved the power of our test. This approach is known as independent filtering.

The DESeq2 software automatically performs independent filtering which maximizes the number of genes which will have adjusted p value less than a critical value (by default, alpha is set to 0.1). This automatic independent filtering is performed by, and can be controlled by, the results function. We can observe how the number of rejections changes for various cutoffs based on mean normalized count. The following optimal threshold and table of possible values is stored as an attribute of the results object.

```
##      45%
## 19.42829
```



The term independent highlights an important caveat. Such filtering is permissible only if the filter criterion is independent of the actual test statistic. Otherwise, the filtering would invalidate the test and consequently the assumptions of the BH procedure. This is why we filtered on the average over all samples: this filter is blind to the assignment of samples to the treatment and control group and hence independent. The independent filtering software used inside DESeq2 comes from the genefilter package, which contains a reference to a paper describing the statistical foundation for independent filtering.

Session Info

```
## R version 3.2.1 (2015-06-18)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.10.4 (Yosemite)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] genefilter_1.50.0         BiocParallel_1.2.9
##  [3] dplyr_0.4.2               tidyr_0.2.0
##  [5] vioplot_0.2               sm_2.2-5.4
##  [7] ggplot2_1.0.1             PoiClaClu_1.0.2
##  [9] pheatmap_1.0.7            RColorBrewer_1.1-2
## [11] DESeq2_1.8.1              RcppArmadillo_0.5.200.1.0
## [13] Rcpp_0.11.6               GenomicRanges_1.20.5
## [15] GenomeInfoDb_1.4.1        IRanges_2.2.5
## [17] S4Vectors_0.6.1           BiocGenerics_0.14.0
##
## loaded via a namespace (and not attached):
##  [1] locfit_1.5-9.1       reshape2_1.4.1       splines_3.2.1
##  [4] lattice_0.20-33      colorspace_1.2-6     htmltools_0.2.6
##  [7] yaml_2.1.13          survival_2.38-3      XML_3.98-1.3
## [10] foreign_0.8-65       DBI_0.3.1            lambda.r_1.1.7
## [13] plyr_1.8.3           stringr_1.0.0        munsell_0.4.2
## [16] gtable_0.1.2         futile.logger_1.4.1  evaluate_0.7
## [19] labeling_0.3         latticeExtra_0.6-26  Biobase_2.28.0
## [22] knitr_1.10.5         geneplotter_1.46.0   AnnotationDbi_1.30.1
## [25] proto_0.3-10         acepack_1.3-3.3      xtable_1.7-4
## [28] scales_0.2.5         formatR_1.2          Hmisc_3.16-0
## [31] annotate_1.46.1      XVector_0.8.0        gridExtra_2.0.0
## [34] digest_0.6.8         stringi_0.5-5        grid_3.2.1
## [37] tools_3.2.1          magrittr_1.5         lazyeval_0.1.10
## [40] RSQLite_1.0.0        Formula_1.2-1        cluster_2.0.2
## [43] futile.options_1.0.0 MASS_7.3-42          assertthat_0.1
## [46] rmarkdown_0.7        R6_2.1.0             rpart_4.1-10
## [49] nnet_7.3-10
```