`erccdashboard` Package Vignette

Sarah A. Munro

April 24, 2017

This vignette describes the use of the erccdashboard R package to analyze External RNA Controls Consortium (ERCC) spike-in control ratio mixtures in gene expression experiments. If you use this package for method validation of your gene expression experiments please cite our manuscript that describes this R package using citation("erccdashboard").

In this vignette we demonstrate analysis of two types of gene expression experiments from the SEQC project that used ERCC control ratio mixture spike-ins:

- Rat toxicogenomics methimazole-treated and control samples
- Human reference RNA samples from the MAQC I project, Universal Human Reference RNA (UHRR) and Human Brain Reference RNA (HBRR)

A subset of the large data set produced in the SEQC study are provided here as examples. The three sets of example data are:

1. Rat toxicogenomics RNA-Seq gene expression count data
2. UHRR/HBRR RNA-Seq gene expression count data
3. UHRR/HBRR Microarray gene expression fluorescent intensity data

1 Rat Toxicogenomics Example: MET (methimazole treatment) and CTL (control) Experiment

1.1 Load data and define input parameters

Load the package gene expression data.

```
> data(SEQC.Example)
```

The R workspace should now contain 5 objects Three of these objects are gene expression experiment expression measures:

- UHRR.HBRR.arrayDat - Fluorescent signal data from an Illumina beadarray microarray experiment with UHRR and HBRR in the SEQC interlaboratory study
- MET.CTL.countDat - RNA-Seq count data from a rat toxicogenomics experiment
- UHRR.HBRR.countDat - RNA-Seq count data from Lab 5 in the SEQC interlaboratory study with UHRR and HBRR

The other two objects are vectors of total reads for the 2 sequencing experiments

- MET.CTL.totalReads - total sequenced reads factors for each column in the corresponding rat experiment count table
- UHRR.HBRR.totalReads - total sequenced reads factors for each column in the corresponding UHRR/HBRR count table

1.2 Quick analysis: runDashboard

To run the default analysis function runDashboard on the MET-CTL rat toxicogenomics RNA-Seq experiment, the following input arguments are required:

```
> datType = "count" # "count" for RNA-Seq data, "array" for microarray data
> isNorm = FALSE # flag to indicate if input expression measures are already
>                # normalized, default is FALSE
> exTable = MET.CTL.countDat # the expression measure table
> filenameRoot = "RatTox" # user defined filename prefix for results files
> sample1Name = "MET" # name for sample 1 in the experiment
> sample2Name = "CTL" # name for sample 2 in the experiment
> erccmix = "RatioPair" # name of ERCC mixture design, "RatioPair" is default
> erccdilution = 1/100 # dilution factor used for Ambion spike-in mixtures
> spikeVol = 1 # volume (in microliters) of diluted spike-in mixture added to
>              #   total RNA mass
> totalRNAmass = 0.500 # mass (in micrograms) of total RNA
> choseFDR = 0.05 # user defined false discovery rate (FDR), default is 0.05
```

The first input argument, datType, indicates whether that data is integer count data from an RNA-Seq experiment ("count") or data from a microarray experiment ("array"). The isNorm argument indicates if the input expression measures are already normalized, the default value is FALSE. If you want to use normalized RNA-Seq or microarray data in your analysis, the isNorm argument must be set to TRUE. If isNorm is TRUE, then the software asks if the input data is length normalized. Type Y at the command line in the R console if the data is length normalized (e.g. FPKM or RPKM data) otherwise type N.

If the data is normalized, then limma will be used for array data differential expression (DE) testing, but for normalized RNA-Seq data, DE testing results must be generated outside of the erccdashboard pipeline and the DE testing results should be provided in the working directory in a file named 'filenameRoot ERCC Pvals.csv' (for details on how to do this see section **3.1 Flexibility in Differential Expression Testing**).

The third argument, exTable, is the expression measure table. Take a look at the expression measure table from the RatTox experiment, to see an example exTable argument:

```
> head(MET.CTL.countDat)
         Feature MET_1 MET_2 MET_3 CTL_1 CTL_2 CTL_3
16499 ERCC-00002 16629 18798 26568 36600 45436 25163
16500 ERCC-00003  1347  1565  1983  3048  3447  2195
16501 ERCC-00004  4569  5570  6755  1240  1484   902
16502 ERCC-00009   811   869  1123   909  1073   537
16503 ERCC-00012     0     0     0     0     0     0
16504 ERCC-00013     3     1     2     1     5     1
```

The first column of the expression measure table, Feature, contains unique names for all the transcripts that were quantified in this experiment. The remaining columns represent replicates of the pair of samples, in this expression measure table the control sample is labeled CTL and the treatment sample is labeled MET. An underscore is included to separate the sample names from the replicate numbers during analysis. This column name format Sample_Rep is required for the columns of any input expression measure table. Only one underscore (_) should be used in the column names.

The default differential expression testing of RNA-Seq experiments in the erccdashboard is done with the QuasiSeq package, which requires the use of integer count data. If you are trying to use either a different approach for DE testing of RNA-Seq count data (e.g. edgeR or DESeq) or RNA-Seq data that is not integer count data (e.g. FPKM data from Cufflinks) please see section **3.1 Flexibility in Differential Expression Testing**.

The erccdashboard default normalization for RNA-Seq count data is 75th percentile (also known as upper quartile) normalization. It is optional to provide a vector of per replicate normalization factors through the input argument repNormFactor, such as a vector of total reads for each replicate. The example total reads vectors we provide here were derived from the FASTQ files associated with each column in the RNA-Seq experiment count tables. Any repNormFactor vector will be used as a library size normalization factor for each column of exTable. This will be adjusted to be a per million reads factor.

For any experiment the sample spiked with ERCC Mix 1 is sample1Name and the sample spiked with ERCC Mix 2 is sample2Name. In this experiment sample1Name = MET and sample2Name = CTL. For a more robust experimental design the reverse spike-in design could be created using additional replicates of the treatment and control samples. ERCC Mix 2 would be spiked into MET samples and ERCC Mix 1 would be spiked into CTL control replicates.

The dilution factor of the pure Ambion ERCC mixes prior to spiking into total RNA samples is erccdilution. The amount of diluted ERCC mix spiked into the total RNA sample is spikeVol (units are μL). The mass of total RNA spiked with the diluted ERCC mix is totalRNAmass (units are μg).

The final required input parameter, choseFDR, is the False Discovery Rate (FDR) for differential expression testing. A typical choice would be 0.05 (5% FDR), so this is the default choseFDR value. For the rat data since most genes are not differentially expressed a less conservative FDR is chosen (FDR = 0.1) and for the UHRR and HBRR reference RNA samples FDR = 0.01 is chosen, because there is a large number of differentially expressed genes for this pair of samples.

The function runDashboard.R is provided for convenient default erccdashboard analysis. Execution of the runDashboard function calls the default functions for erccdashboard analysis and reports parameters and progress to the R console. The functions called within runDashboard.R are also available to the user (details provided in **Section 4**).

All data and analysis results are stored in the list object exDat. For convenience the main diagnostic figures are saved to a pdf file and the exDat object is saved to an .RData object named using the filenameRoot provided by the user.

Use the following command to run the default runDashboard script:

```
> exDat <- runDashboard(datType="count", isNorm = FALSE,
                        exTable=MET.CTL.countDat,
                        filenameRoot="RatTox", sample1Name="MET",
                        sample2Name="CTL", erccmix="RatioPair",
                        erccdilution=1/100, spikeVol=1,
                        totalRNAmass=0.500, choseFDR=0.1)
Initializing the exDat list structure...
choseFDR = 0.1
repNormFactor is NULL
Filename root is: RatTox.MET.CTL

Transcripts were removed with a mean count < 1 or more than 2
replicates with 0 counts.
Original data contained  16590 transcripts.
After filtering  11570 transcripts remain for  analysis.
A total of 29 out of 92
ERCC controls were filtered from the data set
The excluded ERCCs are:
ERCC-00012 ERCC-00014 ERCC-00016 ERCC-00017 ERCC-00024
ERCC-00041 ERCC-00048 ERCC-00057 ERCC-00061 ERCC-00073
ERCC-00075 ERCC-00081 ERCC-00083 ERCC-00086 ERCC-00097
ERCC-00098 ERCC-00104 ERCC-00117 ERCC-00120 ERCC-00123
ERCC-00126 ERCC-00134 ERCC-00137 ERCC-00138 ERCC-00142
ERCC-00147 ERCC-00150 ERCC-00156 ERCC-00164
```

```
repNormFactor is NULL,
 Using Default Upper Quartile Normalization Method  - 75th percentile

normVec:
438 517 473 397 546 389
Check for sample mRNA fraction differences(r_m)...

Number of ERCC Controls Used in r_m estimate
63

Outlier ERCCs for GLM r_m Estimate:
None

GLM log(r_m) estimate:
-0.07014034

GLM log(r_m) estimate weighted s.e.:
0.1494555

Number of ERCCs in Mix 1 dyn range:  63

Number of ERCCs in Mix 2 dyn range:  63
These ERCCs were not included in the signal-abundance plot,
because not enough non-zero replicate measurements of these
 controls were obtained for both samples:

ERCC-00058 ERCC-00067 ERCC-00077 ERCC-00168 ERCC-00028
ERCC-00033 ERCC-00040 ERCC-00109 ERCC-00154 ERCC-00158


Saving dynRangePlot to exDat

Starting differential expression tests

Show log.offset
6.082219 6.248043 6.159095 5.983936 6.302619 5.963579
Disp = 0.0625 , BCV = 0.25
Disp = 0.06249 , BCV = 0.25
[1] "Analyzing Gene # 2"
[1] "Analyzing Gene # 10"
[1] "Analyzing Gene # 100"
[1] "Analyzing Gene # 500"
[1] "Analyzing Gene # 1000"
[1] "Analyzing Gene # 2500"
[1] "Analyzing Gene # 5000"
[1] "Analyzing Gene # 10000"
[1] "Analyzing Gene # 2"
[1] "Analyzing Gene # 10"
[1] "Analyzing Gene # 100"
[1] "Analyzing Gene # 500"
```

```
[1] "Analyzing Gene # 1000"
[1] "Analyzing Gene # 2500"
[1] "Analyzing Gene # 5000"
[1] "Analyzing Gene # 10000"
[1] "Note: 'test.mat' not provided. Comparing each model \nfrom 'design.list' to first model in 'desig
[1] "Spline scaling factor: 0.936086493939411"
[1] "Spline scaling factor: 0.933744268776849"
[1] "Analyzing Gene # 2"
[1] "Analyzing Gene # 10"
[1] "Analyzing Gene # 2"
[1] "Analyzing Gene # 10"
[1] "Note: 'test.mat' not provided. Comparing each model \nfrom 'design.list' to first model in 'desig
[1] "Spline scaling factor: 0.933744268776849"
[1] "Finished DE testing"
[1] "Spline scaling factor: 0.933744268776849"

Finished examining dispersions

Threshold P-value
0.006797136

Generating ROC curve and AUC statistics...

Area Under the Curve (AUC) Results:
 Ratio   AUC Detected Spiked
   4:1 1.000       16     23
 1:1.5 0.950       16     23
   1:2 0.971       16     23

Estimating ERCC LODR
.............................................
 Ratio LODR Estimate 90% CI Lower Bound 90% CI Upper Bound
   4:1            27                 20                 34
 1:1.5           Inf               <NA>               <NA>
   1:2           310                150                450

LODR estimates are available to code ratio-abundance plot

Saving main dashboard plots to pdf file...

Saving exDat list to .RData file...
Analysis completed.
```

1.3 Results of dashboard analysis

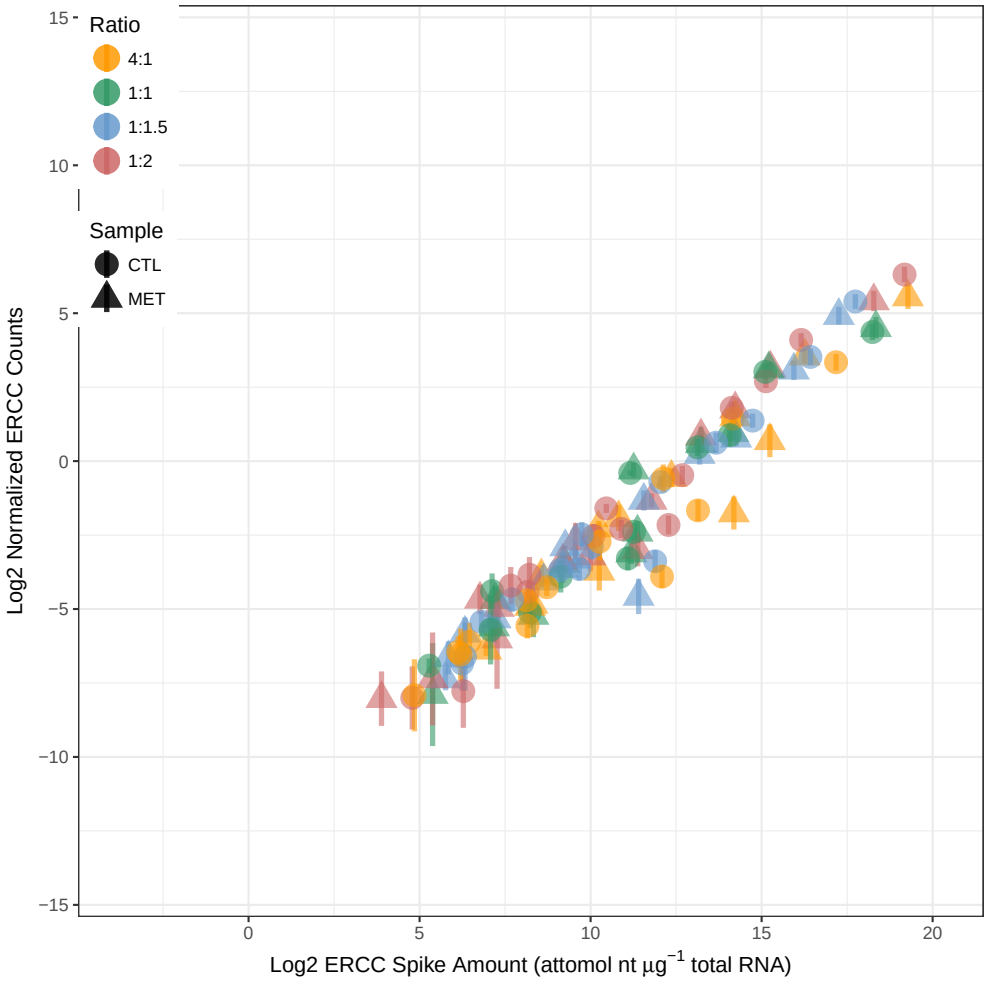
The summary function will give a top level view of the exDat list structure. The str function will give more detail. It is a good idea to set the max.level argument in the str function, because by the end of the analysis the exDat structure is quite large.

```
> summary(exDat)
              Length Class      Mode
sampleInfo    11     -none-     list
```

```
plotInfo       9     -none-     list
erccInfo       4     -none-     list
Transcripts    7     data.frame list
designMat      3     data.frame list
sampleNames    2     -none-     character
idCols         6     data.frame list
normERCCDat    7     data.frame list
normFactor     6     -none-     numeric
mnLibeFactor   1     -none-     numeric
spikeFraction  1     -none-     numeric
idColsAdj      6     data.frame list
Results       12     -none-     list
Figures        7     -none-     list
```

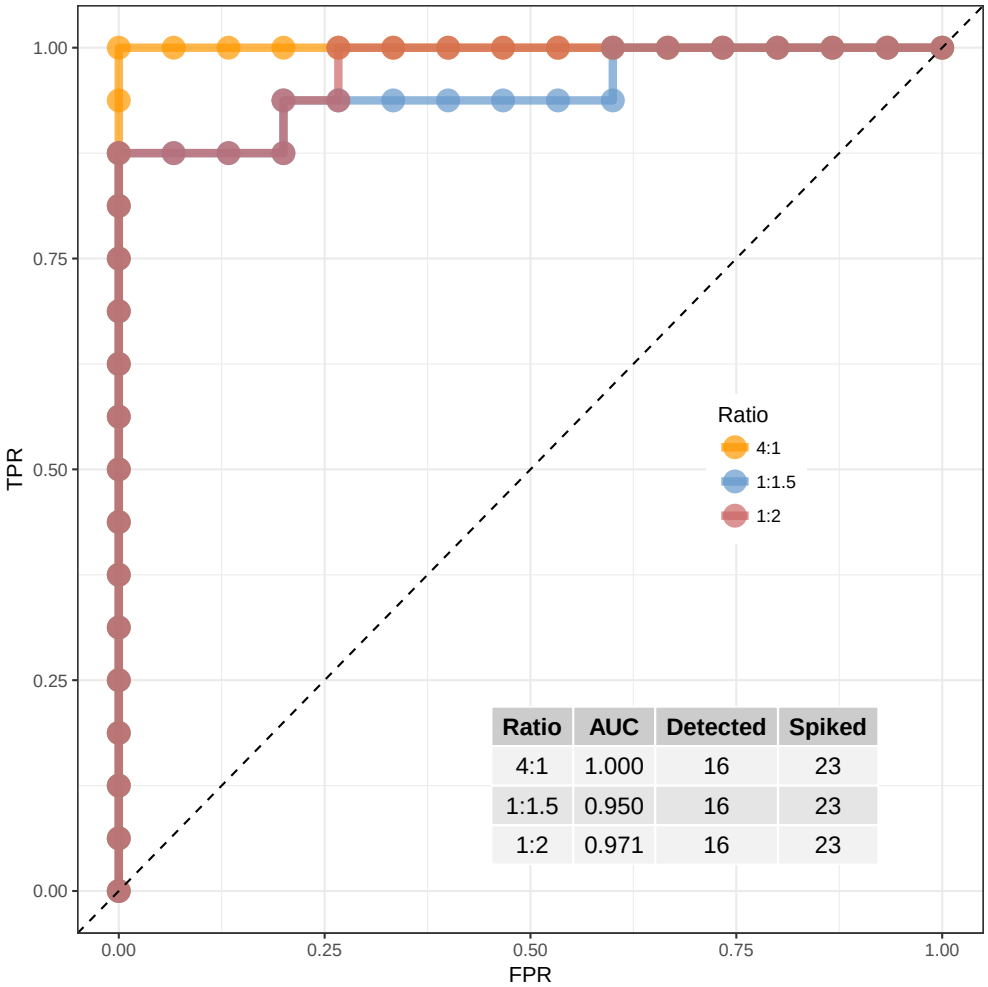
The figures from the analysis are stored in `exDat$Figures`. The four main diagnostic figures that are saved to a pdf file are the dynRangePlot, rocPlot, lodrERCCPlot, and maPlot.

```
> grid.arrange(exDat$Figures$dynRangePlot)
```



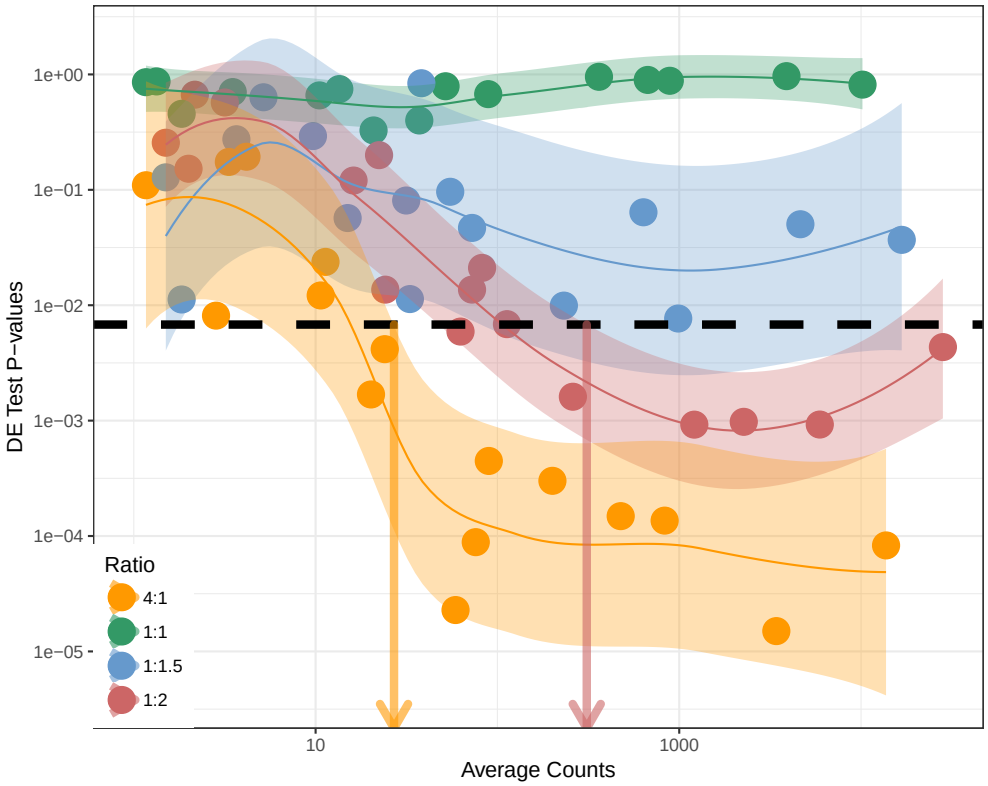
For this particular experiment the relationship between abundance and signal for the ERCC controls shows that the measurement results span a 2^{15} dynamic range. These ERCC mixtures were designed to span a 2^{20} dynamic range, but there was insufficient evidence to reliably quantify ERCC transcripts at low abundances.

```
> grid.arrange(exDat$Figures$rocPlot)
```



The receiver operator characteristic (ROC) curve and the Area Under the Curve (AUC) statistic provide evidence of the diagnostic power for detecting differential expression in this rat toxicogenomics experiment. As expected with increased fold change, diagnostic power increases. The AUC summary statistic for different experiments can be used to compare diagnostic performance.

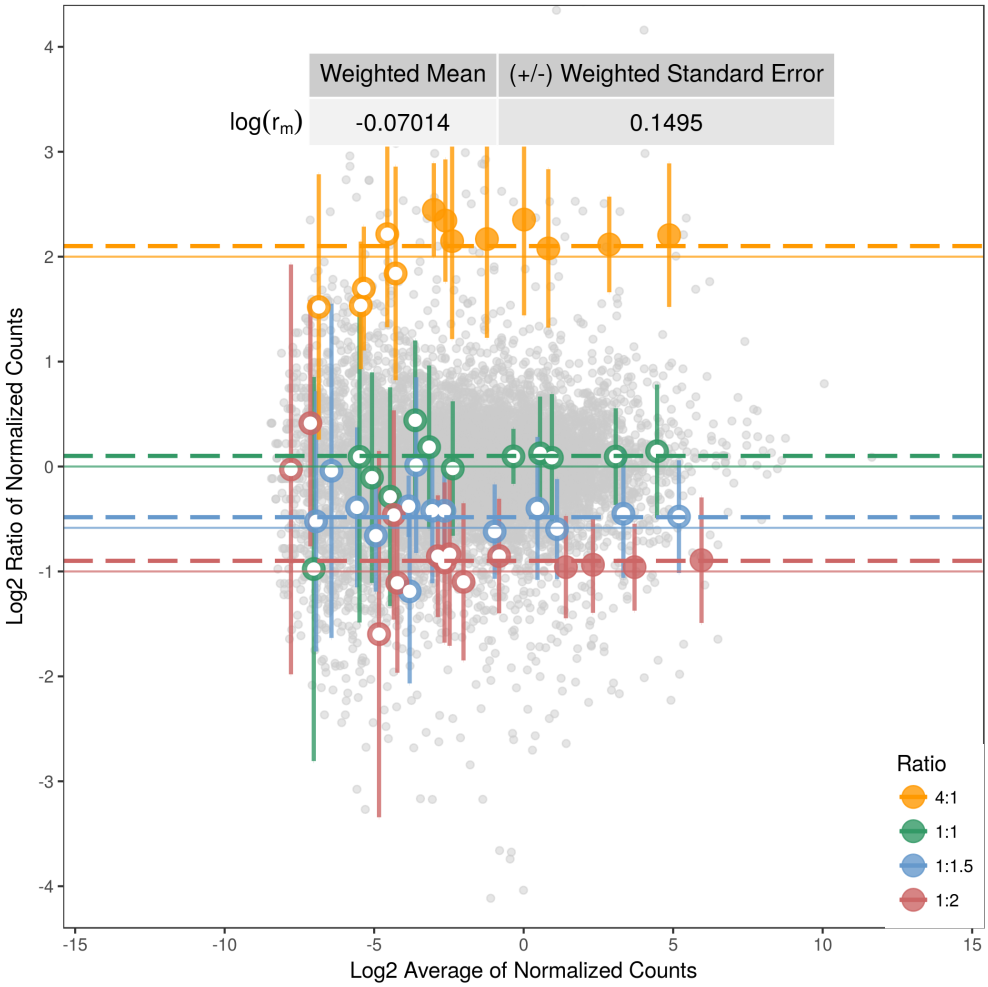
```
> grid.arrange(exDat$Figures$lodrERCCPlot)
```



Ratio	LODR Estimate	90% CI Lower Bound	90% CI Upper Bound
4:1	27	20	34
1:1.5	Inf	NA	NA
1:2	310	150	450

By modeling the relationship between average signal and p-values we can obtain Limit of Detection of Ratios (LODR) estimates for each differential fold change (or Ratio,indicated by color) and a threshold p-value, p.thresh, indicated by the dotted black line. LODR values can be compared between experiments to evaluate detection of differences between samples as a function of transcript abundance. If the input data used in erccdashboard analysis are unnormalized RNA-Seq counts, then the LODR estimates from the lodrERCCPlot are unnormalized counts. If normalized RNA-Seq data or microarray data are used for analysis then the LODR estimates will have corresponding units.

```
> grid.arrange(exDat$Figures$maPlot)
```



An MA plot (Ratio of Signals vs Average Signals) shows the ratio measurements of transcripts in the pair of samples as a function of abundance. The ERCC control ratios measurements are coded to indicate which controls are above a given LODR (solid circles) or below the LODR (open circles). This plot also shows the variability in ratio measurements as a function of dynamic range and the bias in control ratio measurements (r_m), which is influenced by the mRNA fraction difference between the pair of samples.

2 Comparison of Performance Between Experiments

The performance metrics provided here derived from measurements of ERCC ratios in gene expression experiments (AUC, LODR, r_m, and the standard deviations of the ERCC ratio measurements) can be used to assess performance between experiments within the same laboratory, or between different laboratories or technology platforms.

To illustrate the difference between technology platforms example data are also provided from the SEQC interlaboratory study. The interlaboratory experiments were performed with aliquots from single large batches of commercially available reference RNA samples, Universal Human Reference RNA (UHRR) and Human Brain Reference RNA (HBRR).To prepare these large batches of reference RNA samples, 50 μL of undiluted Ambion ERCC Mix 1 was spiked into 2500 μg of the UHRR total RNA and 50 μL of undiluted Ambion ERCC Mix 2 was spiked into 2500 μg the HBRR sample before these samples were aliquoted and shared amongst laboratories and across platforms.

2.1 Analysis of an UHRR vs. HBRR Microarray experiment

To analyze the example reference RNA experiment microarray data use the following command:

```
> exDat <- runDashboard(datType="array", isNorm = FALSE,
                        exTable=UHRR.HBRR.arrayDat,
                        filenameRoot = "Lab13.array",
                        sample1Name = "UHRR", sample2Name="HBRR",
                        erccmix = "RatioPair", erccdilution = 1,
                        spikeVol = 50, totalRNAmass = 2.5*10^(3), choseFDR=0.01)
```

Note that unnormalized fluorescent signals are expected for erccdashboard analysis of microarray data (with the argument isNorm = FALSE) and input data should not be log transformed.

As with RNA-Seq experiments for microarray experiments it is optional to provide a vector of per replicate normalization factors through the input argument repNormFactor.

Normalized microarray data may be analyzed with the erccdashboard if isNorm = TRUE.

2.2 Analysis of an UHRR vs. HBRR RNA-Seq experiment

To analyze the RNA-Seq reference RNA experiment data simply repeat the same command that you used in the previous section with the microarray data, but change the datType to "count", use UHRR.HBRR.countDat as your exTable, and change your filenameRoot to a different character string, so that your microarray data is not overwritten.

3 Analysis Details: Advanced Use of erccdashboard Functions

The analysis functions contained in the convenience wrapper function `runDashboard` can also be used directly by the user. Comments are provided above each analysis step included in `runDashboard` to describe the purpose and ordering constraints. View the runDashboard script to see comments describing the analysis functions and the ordering constraints:

```
> runDashboard
function (datType = NULL, isNorm = FALSE, exTable = NULL, repNormFactor = NULL,
    filenameRoot = NULL, sample1Name = NULL, sample2Name = NULL,
    erccmix = "RatioPair", erccdilution = 1, spikeVol = 1, totalRNAmass = 1,
    choseFDR = 0.05, ratioLim = c(-4, 4), signalLim = c(-14,
        14), userMixFile = NULL)
{
```

```
    exDat <- initDat(datType = datType, isNorm = isNorm, exTable = exTable,
        repNormFactor = repNormFactor, filenameRoot = filenameRoot,
        sample1Name = sample1Name, sample2Name = sample2Name,
        erccmix = erccmix, erccdilution = erccdilution, spikeVol = spikeVol,
        totalRNAmass = totalRNAmass, choseFDR = choseFDR, ratioLim = ratioLim,
        signalLim = signalLim, userMixFile = userMixFile)
    exDat <- est_r_m(exDat)
    exDat <- dynRangePlot(exDat)
    exDat <- geneExprTest(exDat)
    exDat <- erccROC(exDat)
    exDat <- estLODR(exDat, kind = "ERCC", prob = 0.9)
    exDat <- annotLODR(exDat)
    saveERCCPlots(exDat, saveas = "pdf")
    cat("\nSaving exDat list to .RData file...")
    nam <- paste(exDat$sampleInfo$filenameRoot, "exDat", sep = ".")
    assign(nam, exDat)
    to.save <- ls()
    saveName <- paste0(exDat$sampleInfo$filenameRoot, ".RData")
    save(list = to.save[grepl(pattern = nam, x = to.save)], file = saveName)
    cat("\nAnalysis completed.")
    return(exDat)
    dev.off()
}
<bytecode: 0xa6ea180>
<environment: namespace:erccdashboard>
```

3.1 Flexibility in Differential Expression Testing

The `geneExprTest` function has 2 major actions:

1. differential expression testing
2. selecting a p-value threshold based on DE testing

For differential expression testing, `geneExprTest` first looks for a correctly formatted filenameRoot.All.Pvals.csv file in the working directory. If this file is not found then `geneExprTest` proceeds with QuasiSeq differential expression testing package for datType = "count" or uses the limma package for differential expression testing if datType = "array".

To use results from another DE testing tool instead of QuasiSeq (such as edgeR or DESeq for count data or Cuffdiff for FPKM data from Cufflinks) the working directory must contain a csv file with the name "filenameRoot.All.Pvals.csv" and correct column headers.

A correctly formatted "filenameRoot.All.Pvals.csv" file will have column names "Feature", "MnSignal", "Pval", and "Fold". "Feature" is a column of transcript names including ERCC controls and endogenous transcripts, "MnSignal" is the mean signal across sample replicates (e.g. counts or FPKM, etc.), "Pval" are the unadjusted P-values from differential expression testing (Q-values will be automatically estimated for the P-values, and take into account multiple hypothesis testing), and "Fold" are the numeric fold changes for the ERCC controls (0.5, 4, 1, and 0.667) and for endogenous transcripts the "Fold" is NA.

The `geneExprTest` function must locate the "filenameRoot.All.Pvals.csv" file in the working directory to select a threshold p-value for use in the `estLODR` function for LODR estimation.

3.2 Options for LODR Estimation

The default behavior of runDashboard is to use the `estLODR` function to obtain an LODR estimate using empirical data from the ERCCs and a model-based simulation using the endogenous genes in the sample. The type of LODR estimation is selected using the argument kind in the `estLODR` function. The other parameter that may be adjusted is the probability for the LODR estimate, in the default analysis prob = 0.9 is selected.

3.3 Options for Printing Plots to File

The function `saveERCCPlots` will save selected figures to a pdf file. The default is to print the 4 main erccdashboard figures to a single page (plotsPerPg = "main"). If plotsPerPg = "single" then each plot is placed on an individual page in one pdf file. If plotlist is not defined (plotlist = NULL) then all plots in `exDat$Figures` are printed to the pdf file.

3.4 Analysis of Alternative Spike-in Designs

By default the package is configured to analyze the ERCC ratio mixtures produced by Ambion (ERCC ExFold RNA Spike-In Mixes, Catalog Number 4456739). This pair of control ratio mixtures were designed to have 1:1, 4:1, 1:1.5, and 1:2 ratios of 92 distinct RNA transcripts (23 different RNA control sequences are in each of these four ratio subpools). Alternative ERCC RNA control ratio mixture designs can be produced using the NIST DNA Plasmid Library for External Spike-in Controls (NIST Standard Reference Material 2374, https://www-s.nist.gov/srmors/certificates/2374.pdf). For example, a pair of RNA control mixtures could be created with a ternary ratio design, three subpools of RNA controls with either no change (1:1) or 2-fold increased (2:1) and 2-fold decreased (1:2) relative abundances between the pair of mixtures (Mix 1/Mix 2). To use alternative spike-in mixture designs with the dashboard a csv file must be provided to the package with the argument userMixFile for the initDat function.

If all samples from both conditions were only spiked with a single ERCC mixture (e.g. Ambion Catalog Number 4456740, ERCC RNA Spike-In Mix) a limited subset of the package functions can be used (`initDat`, `est_r_m`, and `dynRangePlot`. For `initDat` use ERCCMixes="Single" and `est_r_m` and `dynRangePlot` functions can then be used to examine the mRNA fraction differences for the pair of samples and evaluate the dynamic range of the experiment.

4 Notes on R version and session information

The results shown in this R vignette are the same as the results shown in our manuscript and were obtained with the following R session information.

```
> sessionInfo()
R version 3.4.0 (2017-04-21)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.2 LTS

Matrix products: default
BLAS: /home/biocbuild/bbs-3.5-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.5-bioc/R/lib/libRlapack.so

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
 [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] stats     graphics  grDevices utils     datasets
[6] methods   base

other attached packages:
[1] erccdashboard_1.10.0 gridExtra_2.2.1
[3] ggplot2_2.2.1

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.10       magrittr_1.5
 [3] edgeR_3.18.0       splines_3.4.0
 [5] MASS_7.3-47        munsell_0.4.3
 [7] colorspace_1.3-2   lattice_0.20-35
 [9] quadprog_1.5-5     QuasiSeq_1.0-8
[11] stringr_1.2.0      ROCR_1.0-7
[13] plyr_1.8.4         caTools_1.17.1
[15] tools_3.4.0        grid_3.4.0
[17] nlme_3.1-131       gtable_0.2.0
[19] mgcv_1.8-17        KernSmooth_2.23-15
[21] gtools_3.5.0       digest_0.6.12
[23] lazyeval_0.2.0     tibble_1.3.0
[25] Matrix_1.2-9       reshape2_1.4.2
[27] bitops_1.0-6       qvalue_2.8.0
[29] labeling_0.3       pracma_2.0.4
[31] limma_3.32.0       gdata_2.17.0
[33] stringi_1.1.5      compiler_3.4.0
[35] gplots_3.0.1       scales_0.4.1
[37] locfit_1.5-9.1
```