`testerccdashboard` Package Example

Sarah A. Munro

November 8, 2013

`erccdashboard` Package Vignette

This vignette describes the use of the erccdashboard R package to analyze External RNA Control Consortium (ERCC) spike-in control ratio mixtures in gene expression experiments. Two types of data from the SEQC/MAQC III project were analyzed.

1. Rat toxicogenomics treatment and control samples for different drug treatments
2. Human reference RNA samples from the MAQC I project, Universal Human Reference RNA (UHRR) and Human Brain Reference RNA (HBRR)

1 Rat Toxicogenomics Example: MET treatment

1.1 Define input data and parameters

Load the testerccdashboard package.

```
> library( "testerccdashboard" )
```

Load the Rat Toxicogenomics Data set.

```
> load(file = system.file("data/SEQC.RatTox.Example.RData",
                          package = "testerccdashboard"))
```

The R workspace should now contain 5 count tables and for each count table a corresponding total reads vector. Take a look at the data for the MET experiment.

```
> head(COH.RatTox.ILM.MET.CTL.countTable)
     Feature MET_1 MET_2 MET_3 CTL_1 CTL_2 CTL_3
1 ERCC-00002 16629 18798 26568 36600 45436 25163
2 ERCC-00003  1347  1565  1983  3048  3447  2195
3 ERCC-00004  4569  5570  6755  1240  1484   902
4 ERCC-00009   811   869  1123   909  1073   537
5 ERCC-00013     3     1     2     1     5     1
6 ERCC-00019    24    32    43     5    13     4
> COH.RatTox.ILM.MET.CTL.totalReads
[1] 41423502 46016148 44320280 38400362 47511484 33910098
```

The first column of the count 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 count 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 naming convention Sample_Rep is needed for the columns of any input count table.

The total reads vectors will be used for library size normalization of the count tables. Total reads can either represent the total number of reads in FASTQ files or total mapped reads. In the examples provided with this package FASTQ file total reads are used.

For our analysis of the MET-CTL experiment start by assigning the MET-CTL data to the input data variables countTable and totalReads.

```
> countTable <- COH.RatTox.ILM.MET.CTL.countTable
> totalReads <- COH.RatTox.ILM.MET.CTL.totalReads
```

In addition to countTable and totalReads, there are 7 additional variables that must be defined by the user. First the filename prefix for results files, filenameRoot, needs to be defined. Here we choose to use the lab abbreviation COH and the platform abbreviation ILM as our identifiers, but this is flexible for the user.

```
> filenameRoot = "COH.ILM"
```

Next, 5 parameters associated with the ERCC control ratio mixtures need to be defined, sample1Name, sample2Name, ERCCdilution, spikeVol, and totalRNAmass.

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 also be produced 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 is ERCCdilution. If no dilution was performed then ERCCdilution should be 1. 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)

```
> sample1Name = "MET"
> sample2Name = "CTL"
> ERCCdilution = 1/100
> spikeVol = 1
> totalRNAmass = 0.500
```

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), for the rat data a more liberal FDR was used, choseFDR = 0.1.

```
> choseFDR = 0.1
```

In addition to the required input variables the user can also choose whether to print the results directly to a PDF file (the default is TRUE) with the variable printPDF.

1.2 Use initDat function to create expDat

The expDat list is created with the initDat function:

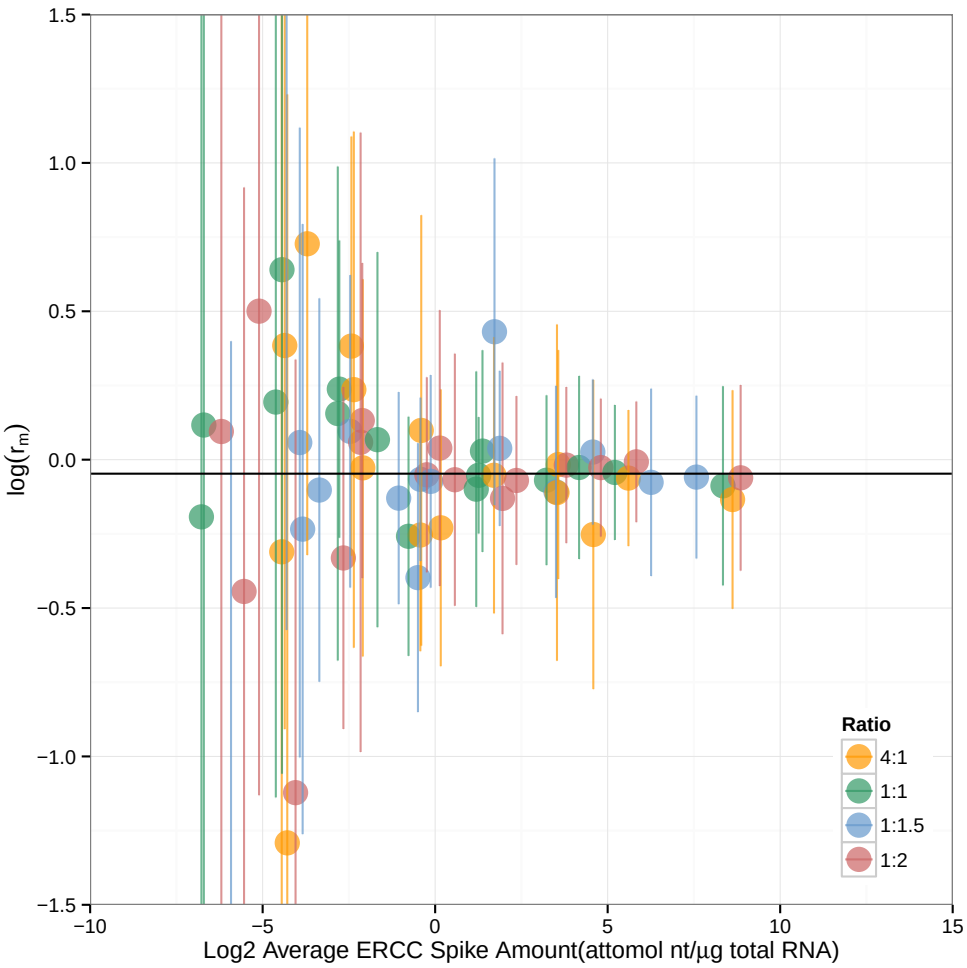
```
> expDat <- initDat(countTable, totalReads, filenameRoot, sample1Name,
                    sample2Name, ERCCdilution, spikeVol, totalRNAmass, choseFDR,
                    printPDF = F)
[1] "COH.ILM.MET.CTL"
[1] "Library sizes:"
[1] 41.42350 46.01615 44.32028 38.40036 47.51148 33.91010
[1] "Using total sequencing reads mean library size = "
[1] 41.93031
```

Look at the structure of expDat

```
> summary(expDat)
                Length Class      Mode
sampleInfo      18     -none-     list
totalReads       6     -none-     numeric
Transcripts      7     data.frame list
designMat        3     data.frame list
sampleNames      2     -none-     character
idCols           6     data.frame list
totalReads       6     -none-     numeric
expressDat       7     data.frame list
libeSize         6     -none-     numeric
ERCCxlabelIndiv  1     -none-     expression
ERCCxlabelAve    1     -none-     expression
spikeFraction    1     -none-     numeric
mnLibeFactor     1     -none-     numeric
sampleLibeSums   6     -none-     numeric
```

Estimate r_m for the sample pair using a negative binomial glm

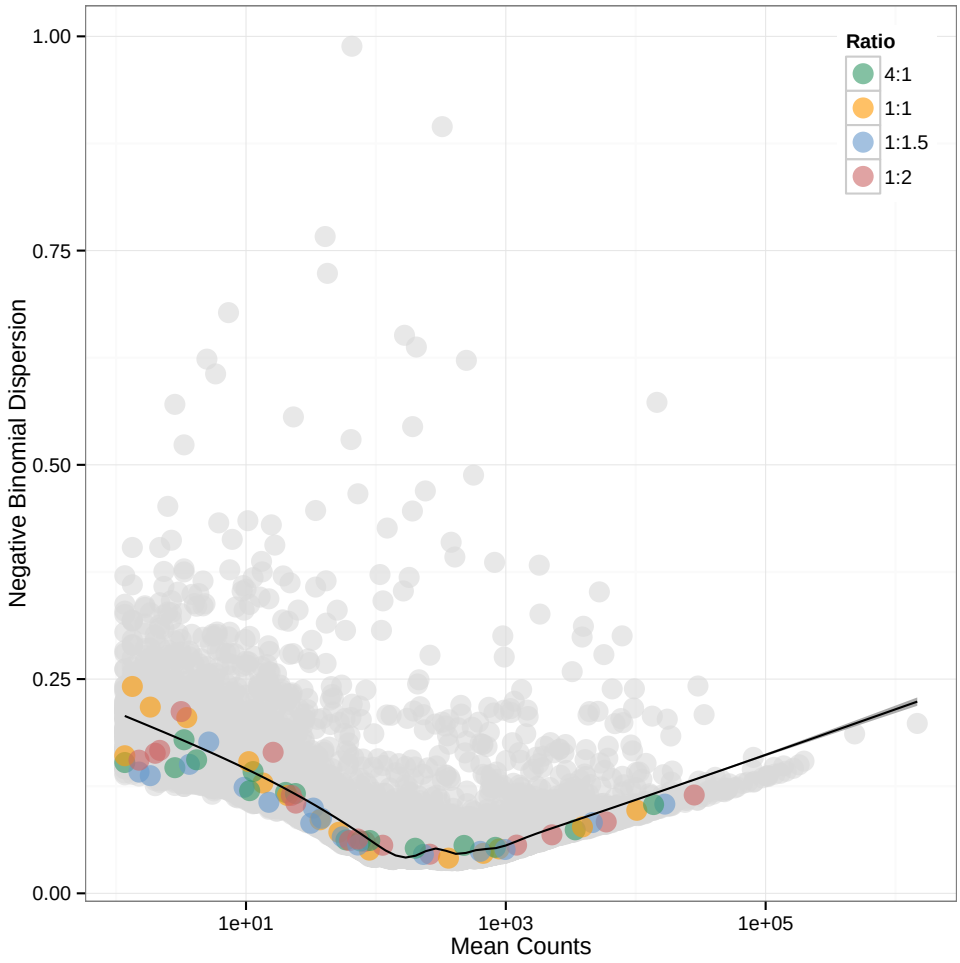
```
> expDat <- est_r_m(expDat, cnt = expDat$Transcripts, printPlot = T)
[1] "Check for sample mRNA fraction differences(r_m)..."
[1] "log.offset"
[1] 17.53936 17.64450 17.60695 17.46358 17.67648 17.33922
[1] "Number of ERCC Controls Used in r_m estimate"
[1] 63
[1] "Outlier ERCCs for GLM r_m Estimate:"
character(0)
[1] " GLM log(r_m) estimate:"
[1] -0.0472291
[1] "GLM log(r_m) estimate standard deviation: "
[1] 0.02061546
[1] 63
[1] " GLM r_m estimate:"
[1] 0.9538688
[1] "upper limit"
[1] 0.9599503
[1] "lower limit"
[1] 0.9478259
```



Test for differential expression with geneExprTest

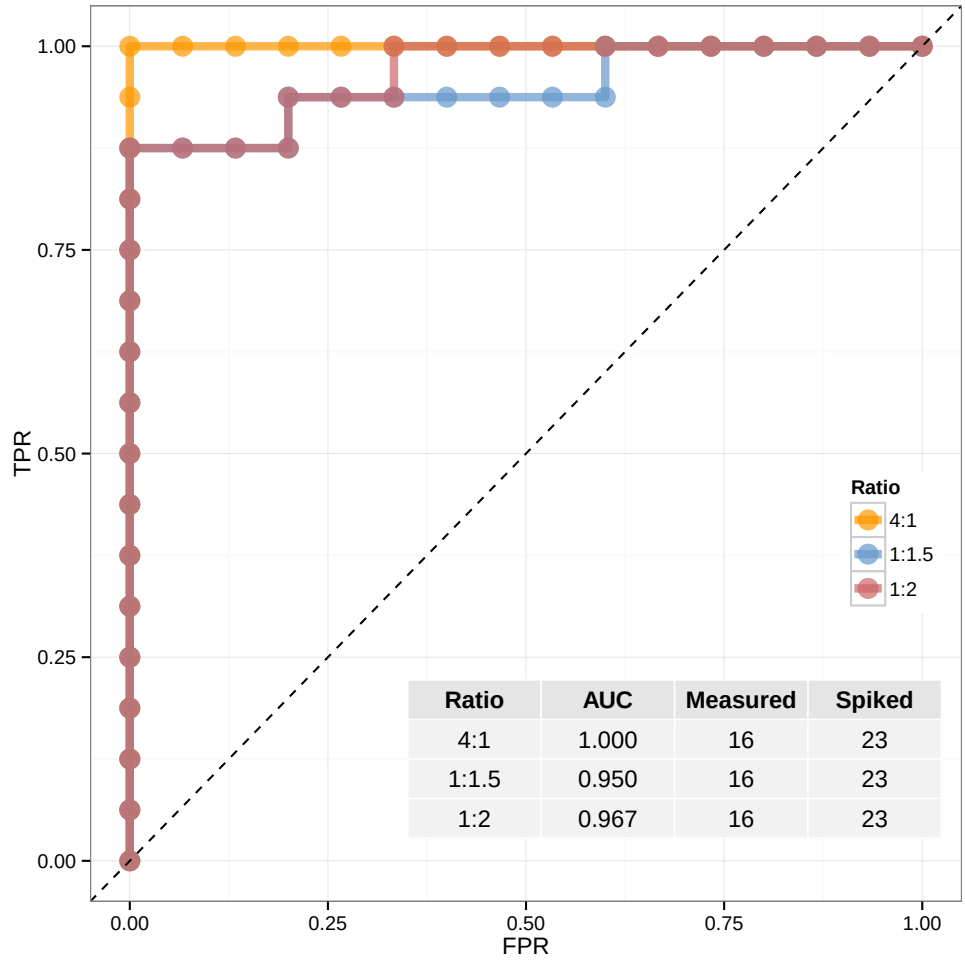
```
> expDat <- geneExprTest(expDat, cnt = expDat$Transcripts,
                         designMat = expDat$designMat )
'data.frame':        11583 obs. of  7 variables:
 $ Feature: chr  "ERCC-00002" "ERCC-00003" "ERCC-00004" "ERCC-00009" ...
 $ MET_1  : int  16629 1347 4569 811 3 24 162 42 3 1 ...
 $ MET_2  : int  18798 1565 5570 869 1 32 184 54 7 1 ...
 $ MET_3  : int  26568 1983 6755 1123 2 43 227 74 6 8 ...
 $ CTL_1  : int  36600 3048 1240 909 1 5 323 50 1 4 ...
 $ CTL_2  : int  45436 3447 1484 1073 5 13 446 62 0 4 ...
 $ CTL_3  : int  25163 2195 902 537 1 4 218 29 0 3 ...
NULL
[1] "Using Total Reads"
[1] 41423502 46016148 44320280 38400362 47511484 33910098
[1] 41423502 46016148 44320280 38400362 47511484 33910098
Disp = 0.06277 , BCV = 0.2505
Disp = 0.06266 , BCV = 0.2503
[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] "Comparing each model from design.list to the full model in design.list (which must be the full mo
[1] "Spline scaling factor: 0.965416753210292"
[1] "Spline scaling factor: 0.962292238986595"
[1] "Analyzing Gene # 2"
[1] "Analyzing Gene # 10"
[1] "Analyzing Gene # 2"
[1] "Analyzing Gene # 10"
[1] "Comparing each model from design.list to the full model in design.list (which must be the full mo
[1] "Spline scaling factor: 0.962292238986595"
[1] "Finished DE testing"
[1] "Spline scaling factor: 0.962292238986595"
[1] "Threshold P-value"
[1] 0.006663508
```



Generate ROC curves for the selected differential ratios

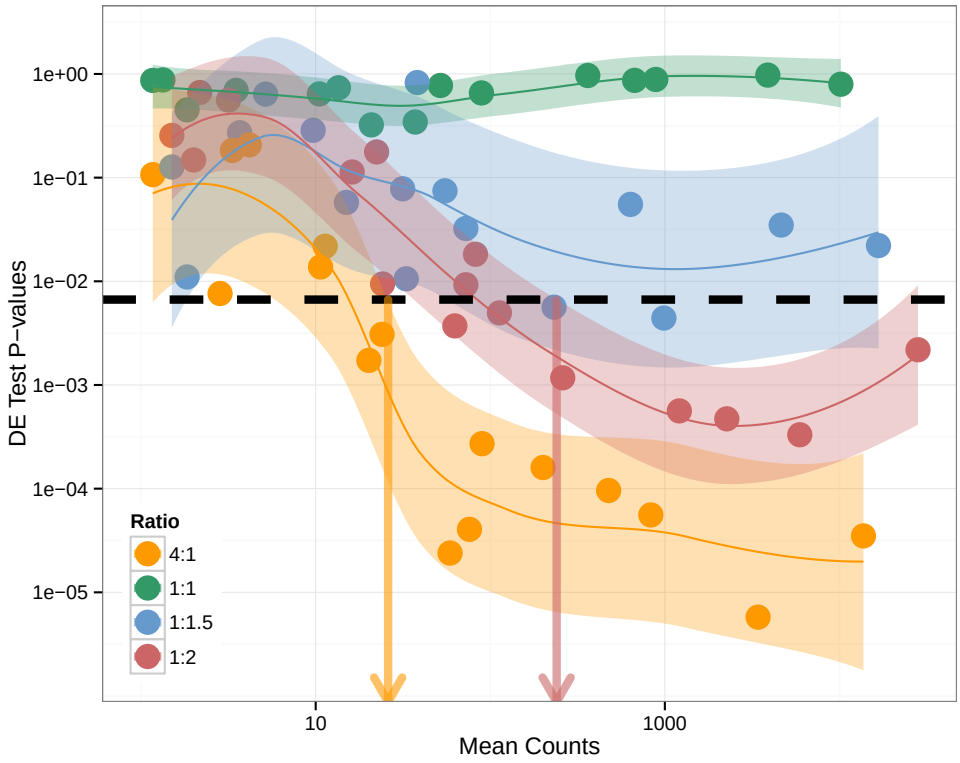
```
> erccROC.res = erccROC(expDat)
  Ratio   AUC Measured Spiked
1   4:1 1.000       16     23
2 1:1.5 0.950       16     23
3   1:2 0.967       16     23
```



Ratio	AUC	Measured	Spiked
4:1	1.000	16	23
1:1.5	0.950	16	23
1:2	0.967	16	23

Find LODR estimates using the ERCC data p-values.

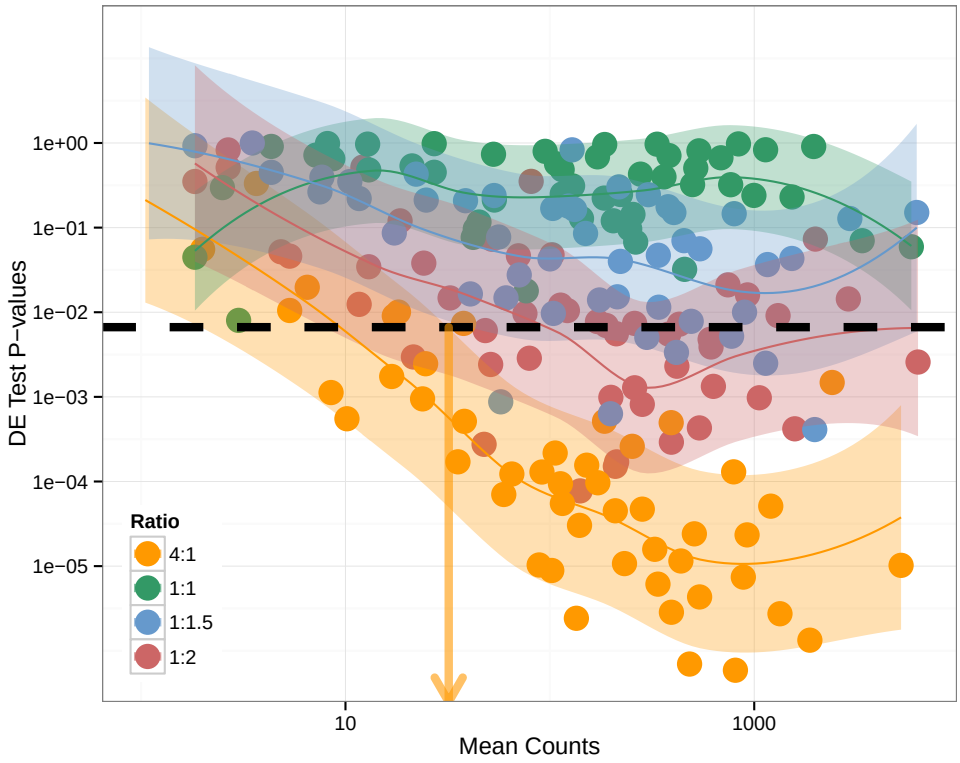
```
> lodr.ERCC = estLODR(expDat,kind = "ERCC", prob=0.9)
  Ratio LODR Estimate 90% CI Lower Bound 90% CI Upper Bound
1   4:1            26                 19                 31
3 1:1.5           Inf               <NA>               <NA>
4   1:2           240                120                340
```



Ratio	LODR Estimate	90% CI Lower Bound	90% CI Upper Bound
4:1	26	19	31
1:1.5	Inf	NA	NA
1:2	240	120	340

One can also obtain LODR estimates using p-values simulated from endogenous transcripts

```
> lodr.Sim = estLODR(expDat, kind = "Sim", prob = 0.9)
  Ratio LODR Estimate 90% CI Lower Bound 90% CI Upper Bound
1   4:1            32                 21                 41
3 1:1.5           Inf               <NA>               <NA>
4   1:2           Inf               <NA>               <NA>
```

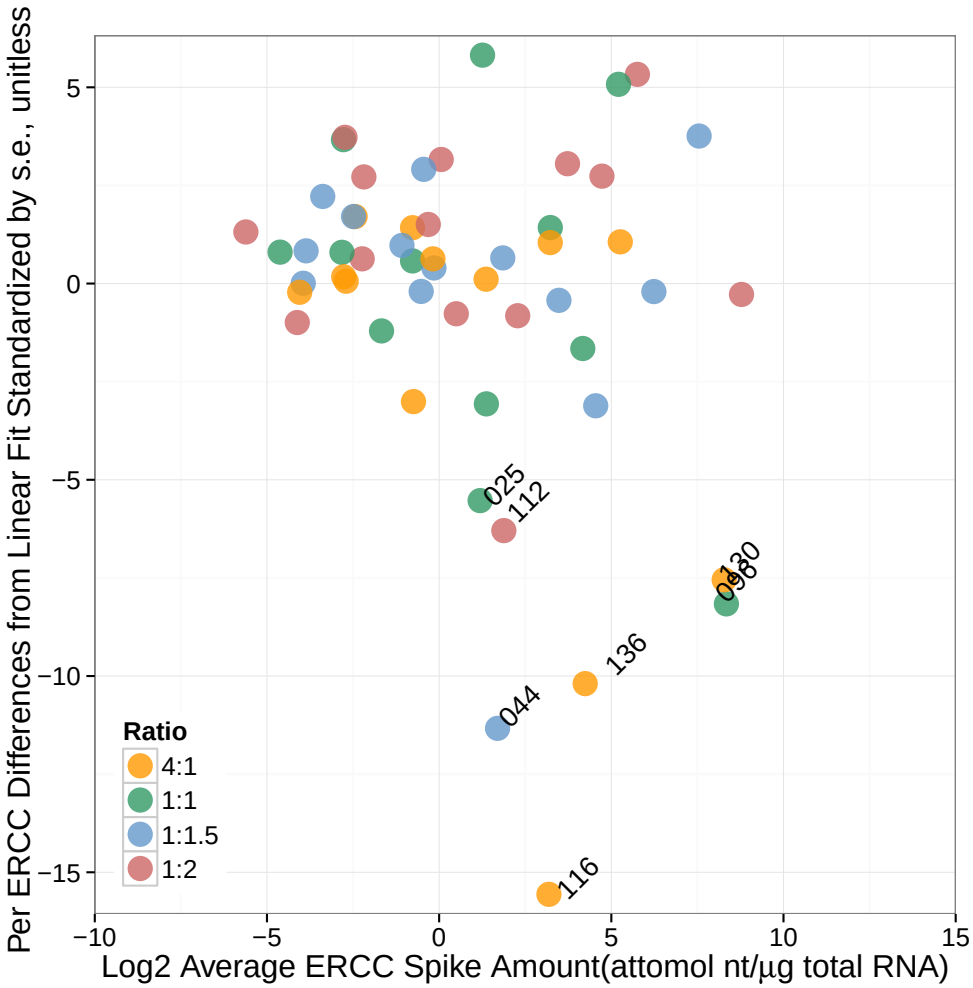


Ratio	LODR Estimate	90% CI Lower Bound	90% CI Upper Bound
4:1	32	21	41
1:1.5	Inf	NA	NA
1:2	Inf	NA	NA

1.3 Use dynRangePlot function to evaluate dynamic range data

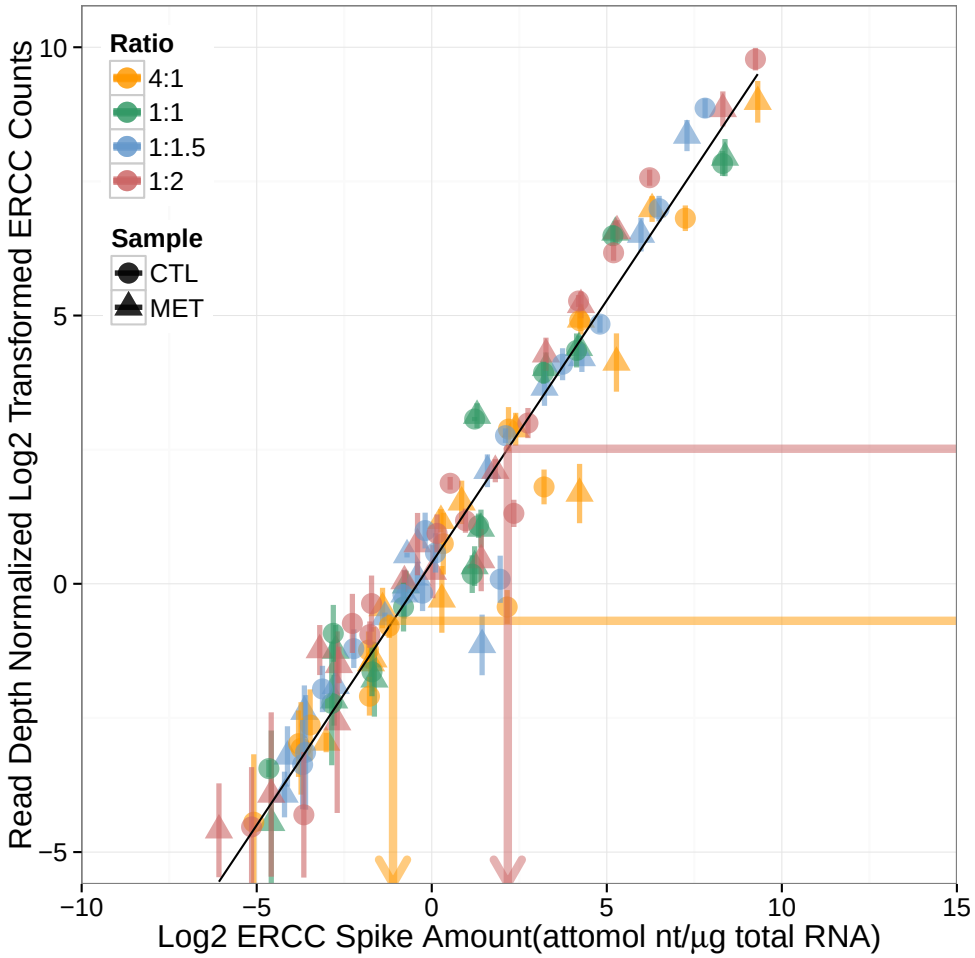
Evaluate the dynamic range of the experiment using the ERCC controls.

```
> dynRangeDat = dynRangePlot(expDat, expressDat = expDat$expressDat,
                             designMat = expDat$designMat, noErrorBars = F)
[1] "Number of ERCCs in Mix 1 dyn range:"
[1] 63
[1] "Number of ERCCs in Mix 2 dyn range:"
[1] 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:
 [1] "ERCC-00058" "ERCC-00067" "ERCC-00077" "ERCC-00168"
 [5] "ERCC-00028" "ERCC-00033" "ERCC-00040" "ERCC-00109"
 [9] "ERCC-00154" "ERCC-00158"
```



Get LODR annotations for adding to plots and then annotate the dynamic range plot with LODR estimate information.

```
> LODR.annot.ERCC <- printLODRres(expDat, dynRangeDat,
                                  lodr.res = lodr.ERCC)
   Fold Ratio Count Log2Count_normalized  Log2Conc
1   4:1   4:1    26            -0.689482 -1.106341
2   1:1   1:1    NA                   NA        NA
3 1:1.5 1:1.5   Inf                  Inf       Inf
4   1:2   1:2   240             2.516969  2.172251
> dynRangePlotLODR(dynRangeRes = dynRangeDat$dynRangePlotRes,
                   LODR.annot.ERCC = LODR.annot.ERCC)
```



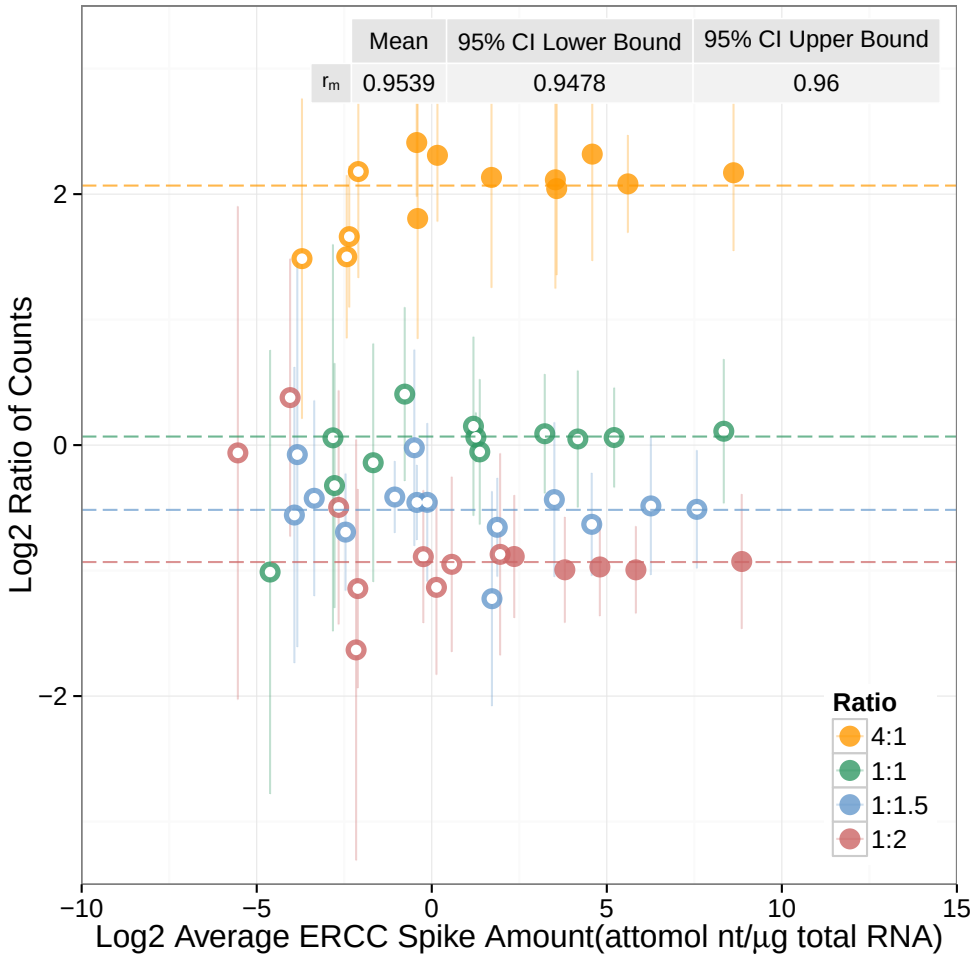
Generate MA plots of erccs coded by concentrations from LODR

```
> maPlotAB = maConcPlot(expDat, LODR.annot.ERCC, alphaPoint = 0.8, r_mAdjust = T,
                        replicate = T)
List of 18
 $ sampleInfo     :List of 18
  ..$ sample1Name  : chr "MET"
  ..$ sample2Name  : chr "CTL"
  ..$ choseFDR     : num 0.1
  ..$ ERCCdilution : num 0.01
  ..$ spikeVol     : num 1
  ..$ totalRNAmass : num 0.5
  ..$ printPDF     : logi FALSE
  ..$ DEtest       : logi TRUE
  ..$ totalSeqReads: logi TRUE
  ..$ libeSizeNorm : logi TRUE
  ..$ myYLimMA     : num [1:2] -3.5 3.5
  ..$ myXLim       : num [1:2] -10 15
  ..$ myYLim       : NULL
  ..$ filenameRoot : chr "COH.ILM.MET.CTL"
```

```
 ..$ idColsSRM    :'data.frame':        96 obs. of  6 variables:
 .. ..$ Feature: Factor w/ 96 levels "ERCC-00002","ERCC-00003",..: 1 2 3 4 5 6 7 8 9 10 ...
 .. ..$ Length : int [1:96] 1061 1023 523 1135 984 994 808 1957 844 1136 ...
 .. ..$ GC     : num [1:96] 0.51 0.33 0.34 0.46 0.47 0.51 0.43 0.44 0.48 0.51 ...
 .. ..$ Ratio  : Factor w/ 4 levels "a","b","c","d": 4 4 1 NA 2 3 4 4 3 1 ...
 .. ..$ Conc1  : num [1:96] 15000 938 7500 0 938 ...
 .. ..$ Conc2  : num [1:96] 30000 1875 1875 0 938 ...
 ..$ MixDef       :'data.frame':        96 obs. of  4 variables:
 .. ..$ Feature              : Factor w/ 96 levels "ERCC-00002","ERCC-00003",..: 1 2 3 4 5 6 7 8 9 10
 .. ..$ Ratio                : Factor w/ 4 levels "a","b","c","d": 4 4 1 NA 2 3 4 4 3 1 ...
 .. ..$ Mix1Conc.Attomoles_ul: num [1:96] 15000 938 7500 0 938 ...
 .. ..$ Mix2Conc.Attomoles_ul: num [1:96] 30000 1875 1875 0 938 ...
 ..$ FCcode       :'data.frame':        4 obs. of  2 variables:
 .. ..$ Ratio: Factor w/ 4 levels "a","b","c","d": 1 2 3 4
 .. ..$ FC   : num [1:4] 4 1 0.667 0.5
 ..$ legendLabels : chr [1:4] "4:1" "1:1" "1:1.5" "1:2"
$ totalReads     : int [1:6] 41423502 46016148 44320280 38400362 47511484 33910098
$ Transcripts    :'data.frame':        11583 obs. of  7 variables:
 ..$ Feature: chr [1:11583] "ERCC-00002" "ERCC-00003" "ERCC-00004" "ERCC-00009" ...
 ..$ MET_1  : int [1:11583] 16629 1347 4569 811 3 24 162 42 3 1 ...
 ..$ MET_2  : int [1:11583] 18798 1565 5570 869 1 32 184 54 7 1 ...
 ..$ MET_3  : int [1:11583] 26568 1983 6755 1123 2 43 227 74 6 8 ...
 ..$ CTL_1  : int [1:11583] 36600 3048 1240 909 1 5 323 50 1 4 ...
 ..$ CTL_2  : int [1:11583] 45436 3447 1484 1073 5 13 446 62 0 4 ...
 ..$ CTL_3  : int [1:11583] 25163 2195 902 537 1 4 218 29 0 3 ...
$ designMat      :'data.frame':        6 obs. of  3 variables:
 ..$ countSet: Factor w/ 6 levels "CTL_1","CTL_2",..: 4 5 6 1 2 3
 ..$ Sample  : Factor w/ 2 levels "CTL","MET": 2 2 2 1 1 1
 ..$ Rep     : Factor w/ 3 levels "1","2","3": 1 2 3 1 2 3
$ sampleNames    : chr [1:2] "MET" "CTL"
$ idCols         :'data.frame':        63 obs. of  6 variables:
 ..$ Feature: Factor w/ 96 levels "ERCC-00002","ERCC-00003",..: 1 2 3 5 7 12 13 16 17 18 ...
 ..$ Length : int [1:63] 1061 1023 523 984 808 644 751 1994 1130 1138 ...
 ..$ GC     : num [1:63] 0.51 0.33 0.34 0.47 0.43 0.49 0.47 0.5 0.51 0.48 ...
 ..$ Ratio  : Factor w/ 4 levels "a","b","c","d": 4 4 1 2 4 1 4 2 1 2 ...
 ..$ Conc1  : num [1:63] 318.3 19.1812 78.45 18.45 0.0148 ...
 ..$ Conc2  : num [1:63] 636.6 38.3625 19.6125 18.45 0.0296 ...
$ totalReads     : int [1:6] 41423502 46016148 44320280 38400362 47511484 33910098
$ expressDat     :'data.frame':        63 obs. of  7 variables:
 ..$ Feature: chr [1:63] "ERCC-00002" "ERCC-00003" "ERCC-00004" "ERCC-00009" ...
 ..$ MET_1  : num [1:63] 401.4388 32.5178 110.2997 19.5783 0.0724 ...
 ..$ MET_2  : num [1:63] 408.5088 34.0098 121.0445 18.8847 0.0217 ...
 ..$ MET_3  : num [1:63] 599.4547 44.7425 152.4133 25.3383 0.0451 ...
 ..$ CTL_1  : num [1:63] 953.116 79.374 32.291 23.672 0.026 ...
 ..$ CTL_2  : num [1:63] 956.316 72.551 31.235 22.584 0.105 ...
 ..$ CTL_3  : num [1:63] 742.0503 64.73 26.5997 15.836 0.0295 ...
$ libeSize       : num [1:6] 41.4 46 44.3 38.4 47.5 ...
$ ERCCxlabelIndiv:  expression(paste("Log2 ERCC Spike Amount(attomol nt/",      mu, "g total RNA)", s
$ ERCCxlabelAve  :  expression(paste("Log2 Average ERCC Spike Amount(attomol nt/",      mu, "g total
$ spikeFraction  : num 0.02
$ mnLibeFactor   : num 41.9
```

```
 $ sampleLibeSums : int [1:6] 41423502 46016148 44320280 38400362 47511484 33910098
 $ idColsAdj      :'data.frame':        63 obs. of  6 variables:
  ..$ Feature: Factor w/ 96 levels "ERCC-00002","ERCC-00003",..: 1 2 3 5 7 12 13 16 17 18 ...
  ..$ Length : int [1:63] 1061 1023 523 984 808 644 751 1994 1130 1138 ...
  ..$ GC     : num [1:63] 0.51 0.33 0.34 0.47 0.43 0.49 0.47 0.5 0.51 0.48 ...
  ..$ Ratio  : Factor w/ 4 levels "a","b","c","d": 4 4 1 2 4 1 4 2 1 2 ...
  ..$ Conc1  : num [1:63] 318.3 19.1812 78.45 18.45 0.0148 ...
  ..$ Conc2  : num [1:63] 607.2329 36.5928 18.7078 17.5989 0.0282 ...
 $ r_m.res        :List of 3
  ..$ r_m.mn   : num -0.0472
  ..$ r_m.upper: num -0.0409
  ..$ r_m.lower: num -0.0536
 $ p.thresh       : num 0.00666
 $ expressDat_l   :'data.frame':        378 obs. of  5 variables:
  ..$ Feature   : Factor w/ 63 levels "ERCC-00002","ERCC-00003",..: 1 2 3 4 5 6 7 8 9 10 ...
  ..$ Ratio     : Factor w/ 4 levels "a","b","c","d": 4 4 1 2 4 1 4 2 1 2 ...
  ..$ NormCounts: num [1:378] 4.01e-04 3.25e-05 1.10e-04 1.96e-05 7.24e-08 ...
  ..$ Sample    : Factor w/ 2 levels "CTL","MET": 2 2 2 2 2 2 2 2 2 2 ...
  ..$ Rep       : Factor w/ 3 levels "1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
NULL
[1] "LODR estimates:"
[1] -1.106341       Inf  2.172251
     Feature Ratio         M         A Replicate
1 ERCC-00002     d -1.247472 -10.52797         1
2 ERCC-00003     d -1.287443 -14.12561         1
3 ERCC-00004     a  1.772209 -13.77583         1
4 ERCC-00009     b -0.273908 -15.49694         1
5 ERCC-00013     d  1.475633 -24.27583         1
6 ERCC-00019     a  2.153705 -21.42651         1
[1] "These ERCCs were not included in the ratio-abundance plot, because not enough non-zero replicate
 [1] "ERCC-00028" "ERCC-00033" "ERCC-00040" "ERCC-00058"
 [5] "ERCC-00067" "ERCC-00077" "ERCC-00109" "ERCC-00154"
 [9] "ERCC-00158" "ERCC-00168"
[1] "Global Ratio SD for this sample pair is:"
[1] 0.7785041
[1] "Using default parameters"
$Mo
[1] 0.1

$lambda
[1] 2

[1] 0.4787852
[1] "Try changing initial guess for parameters"
$Mo
[1] 0.7785041

$lambda
[1] 0.2

             Estimate Std. Error   t value     Pr(>|t|)
```

```
(Intercept) 0.4787852 0.03264151 14.667986 4.049430e-20
Mo          1.6195817 0.20936506  7.735683 3.770026e-10
lambda      0.4123437 0.05767326  7.149652 3.158191e-09
                     Estimate Std. Error   t value
Minimum SD Estimate 0.4787852 0.03264151 14.667986
Maximum SD Estimate 1.6195817 0.20936506  7.735683
Lambda              0.4123437 0.05767326  7.149652
                        Pr(>|t|)
Minimum SD Estimate 4.049430e-20
Maximum SD Estimate 3.770026e-10
Lambda              3.158191e-09
[1] "Printing MA plot with LODR coding"
```



If you wish, save your results to an Rdata file that can be reused. saveResults(expDat, erccROC.res, maPlotAB, lodr.ERCC)