Anaquin - Vignette

Ted Wong (t.wong@garvan.org.au)

October 17, 2016

Citation

[1] S.A Hardwick. Spliced synthetic genes as internal controls in RNA sequencing experiments. *Nature Methods*, 2016.

Need help?

Visit our website to learn more about sequins: www.sequin.xyz.

We have an active Slack channel, sequins.slack.com, where you can chat directly with our team. Please email us at anaquin@garvan.org.au for an invitation. You can also post your questions on support.bioconductor.org, www.biostars.org and www.seqanswers.com, we will answer your questions as we actively monitor the sites.

Overview

In this document, we show how to conduct statistical analysis that models the performance of sequin controls in next-generation-sequencing (NGS) experiment. The controls can be used for RNA-Seq analysis, e.g., normalization, gene expression, differential analysis, and other applications. We call the sequins **RnaQuin** for *RNA-Seq sequins*, and the statistical framework **Anaquin**.

This vignette is written for R-usage. However, Anaquin is a framework covering the entire RNA-Seq workflow; reads alignment, transcriptome assembly, gene expression and differential analysis, etc. Consequently, the R-package (and it's documentation) is a subset of the overall Anaquin framework. We also distribute a workflow guide with this vignette on how Anaquin can be integrated with a bioinformatics workflow. The guide should be distributed with this vignette and is also available on our website.

It is important to note Anaquin is both command-line tool and R-package. Our workflow guide has details on how the command-line tool can be used with the R-package.

Sequins

Next-generation sequencing (NGS) enables rapid, cheap and high-throughput determination of sequences within a user's sample. NGS methods have been applied widely, and have fuelled major advances in the life sciences and clinical health care over the past decade. However, NGS typically generates a large amount of sequencing data that must be first analyzed and interpreted with bioinformatics tools. There is no standard way to perform an analysis of NGS data; different tools provide different advantages in different situations. The complexity and variation of sequences further compound this problem, and there is little reference by which compare next-generation sequencing and analysis.

To address this problem, we have developed a suite of synthetic nucleic-acid sequins (sequencing spike-ins). Sequins are fractionally added to the extracted nucleic-acid sample prior to library preparation, so they are sequenced along with your sample of interest. We can use the sequins as an internal quantitative and qualitative control to assess any stage of the next-generation sequencing workflow.

Mixture

Sequins are combined together across a range of concentrations to formulate a mixture. Mixture file (CSV) is a text file that specifies the concentration of each sequin within a mixture. Mixture files are often required as input to enable Anaquin to perform quantitative analysis. Mixture file can be downloaded from our website:

www.sequin.xyz

Let's demonstrate RnaQuin mixture A with a simple example. Load the mixture file (you can also download the file directly from our website):

```
library('Anaquin')
data(mixtureA)
head(mixtureA)
```

```
##         Length         MXA
## R2_38_1    829 0.003933907
## R2_38_2    760 0.007867813
## R2_76_1    908 0.008429800
## R1_91_1    605 0.014305115
## R2_38_3    872 0.015735626
## R2_72_1   1639 0.015735626
```

Each row represents a sequin. *Column ID* gives the sequin names, *Length* is the length of the sequins in nucleotide bases, *MXA* gives the concentration level in attoml/ul.

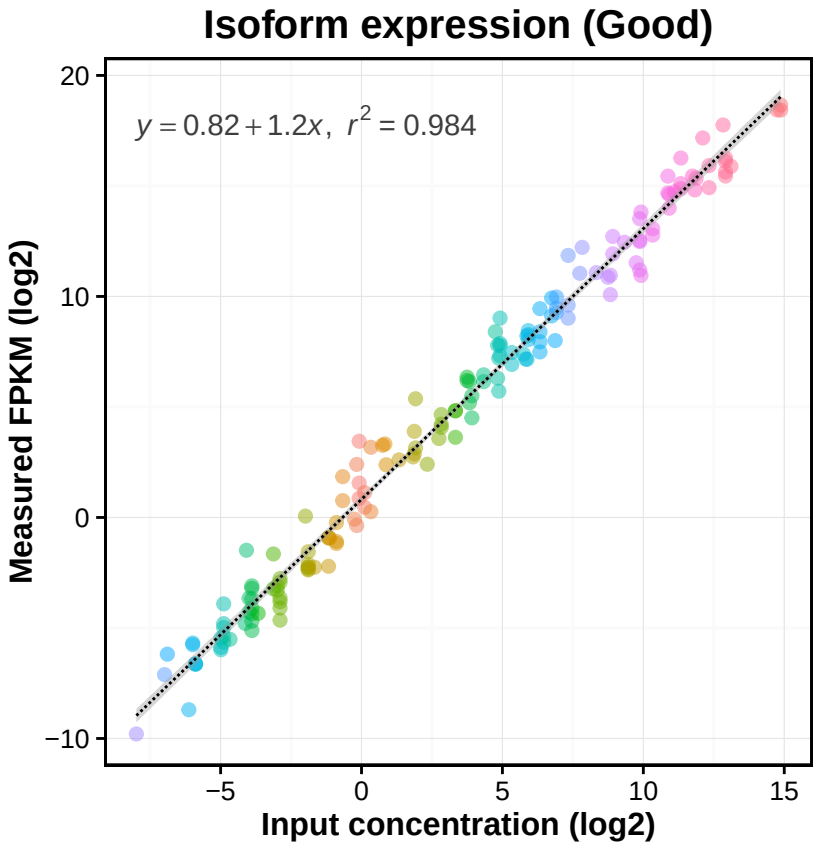
Imagine we have two RNA-Seq experiments; a well-designed experiment and a poorly-designed experiment We would like to quantify their isoform expression.

Let's simulate the experiments:

```
set.seed(1234)
sim1 <- 1.0 + 1.2*log2(mixtureA$MXA) + rnorm(nrow(mixtureA),0,1)
sim2 <- c(1.0 + rnorm(100,1,3), 1.0 +
          1.2*log2(tail(mixtureA,64)$MXA) +
          rnorm(64,0,1))
```

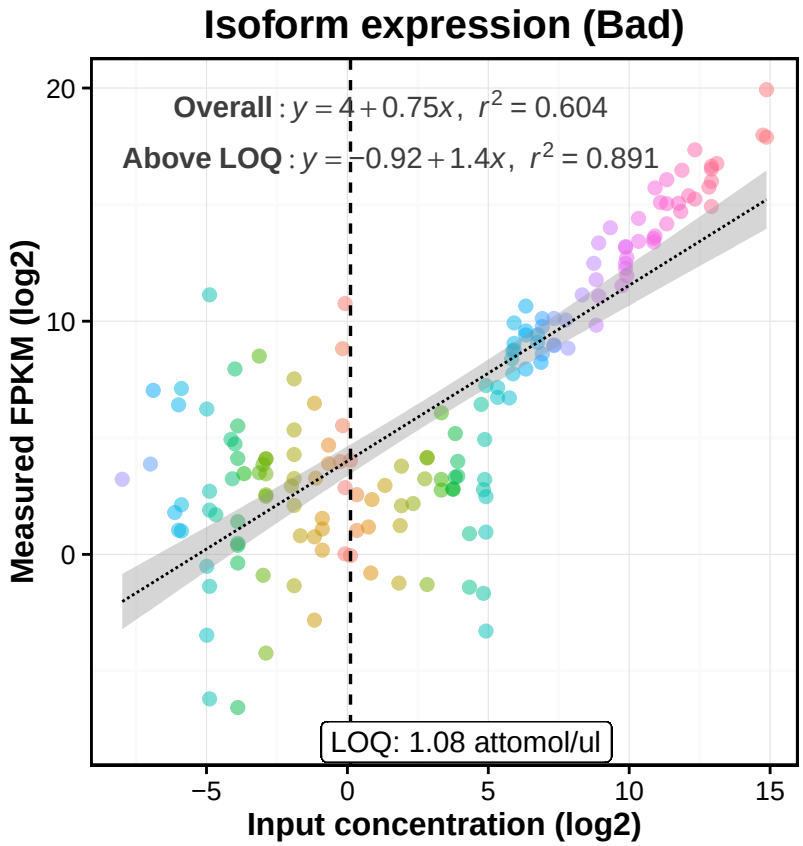
In the first experiment, sequins are expected to correlate linearly with the measured FPKM. Indeed, the variables are strongly correlated:

```
names <- row.names(mixtureA)
input <- log2(mixtureA$MXA)

anaquin <- AnaquinData(analysis='PlotLinear',
                             seqs=names,
                            input=input,
                         measured=sim1)

title <- 'Isoform expression (Good)'
xlab  <- 'Input concentration (log2)'
ylab  <- 'Measured FPKM (log2)'

plotLinear(anaquin, title=title, xlab=xlab, ylab=ylab)
```



In our second experiment, the weakly expressed isoforms exhibit stochastic behavior and are clearly not linear with the input concentration. Furthermore, there is a limit of quantification (LOQ); below which accuracy of the experiment becomes questionable.

```
names <- row.names(mixtureA)
input <- log2(mixtureA$MXA)

anaquin <- AnaquinData(analysis='PlotLinear',
                            seqs=names,
                           input=input,
                        measured=sim2)

title <- 'Isoform expression (Bad)'
xlab  <- 'Input concentration (log2)'
ylab  <- 'Measured FPKM (log2)'

plotLinear(anaquin, title=title, xlab=xlab, ylab=ylab)
```



Quantifying transcriptome assembly

To quantify RNA-Seq transcriptome assembly, we need to run a transcriptome assembler; i.e., a software that can assembles transcripts and estimates their abundances. Our workflow guide has the details.

Here, we use a data set generated by Cufflinks, described in **Section 5.4.5.1** of the user guide:

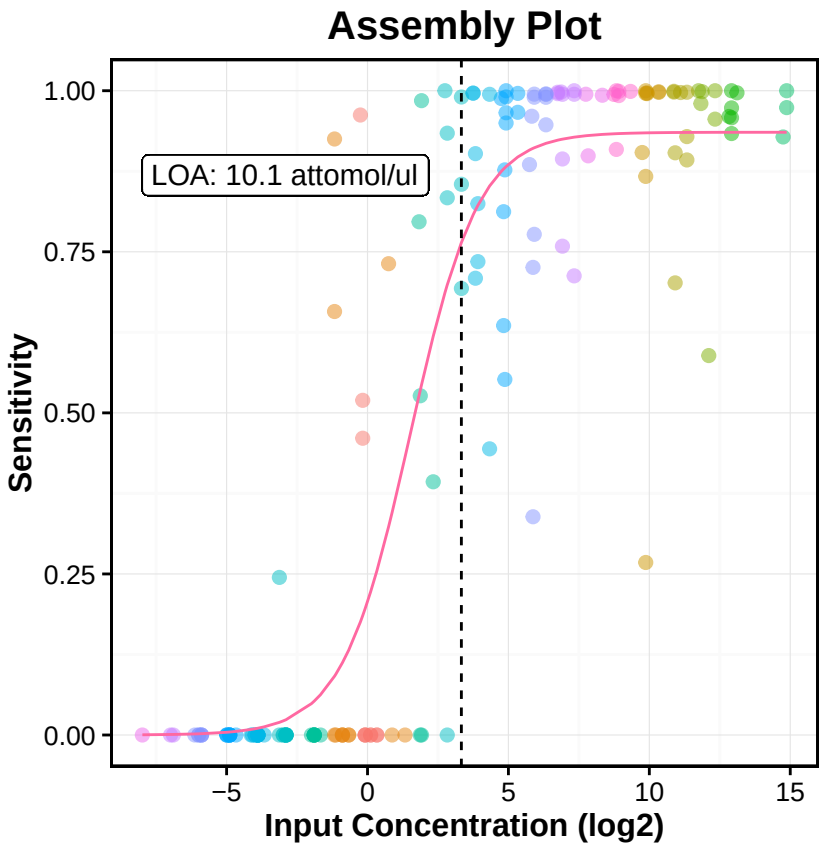
```
data(UserGuideData_5.4.5.1)
head(UserGuideData_5.4.5.1)
```

```
##          InputConcent       Sn
## R1_101_1      10.0708 0.990264
## R1_101_2       5.0354 0.393023
## R1_102_1       0.8886 0.519463
## R1_102_2      14.2176 0.902349
## R1_103_1     107.4220 0.995439
## R1_103_2     859.3750 0.904095
```

The first column gives the input concentration for each sequin in attomol/ul. The second column is the measured sensitivity. Run the following R-code to generate a sensitivity plot.

```
title <- 'Assembly Plot'
xlab  <- 'Input Concentration (log2)'
ylab  <- 'Sensitivity'
```

```
# Sequin names
names <- row.names(UserGuideData_5.4.5.1)

# Input concentration
input <- log2(UserGuideData_5.4.5.1$InputConcent)

# Measured sensitivity
measured <- UserGuideData_5.4.5.1$Sn

anaquin <- AnaquinData(analysis='PlotLogistic',
                               seqs=names,
                              input=input,
                           measured=measured)

plotLogistic(anaquin, title=title, xlab=xlab, ylab=ylab, showLOA=TRUE)
```



The fitted logistic curve reveals clear relationship between input concentration and sensitivity. Unsurprisingly, the assembler is more sensitive to highly expressed isoforms. The limit-of-assembly (LOA) is defined as the intersection of the curve to sensitivity of 0.70.

Quantifying gene expression

Quantifying gene/isoform expression involves building a linear model between input concentration and measured FPKM. In this section, we consider a dataset generated by Cufflinks, described in details in **Section 5.4.5.1** of the user guide.

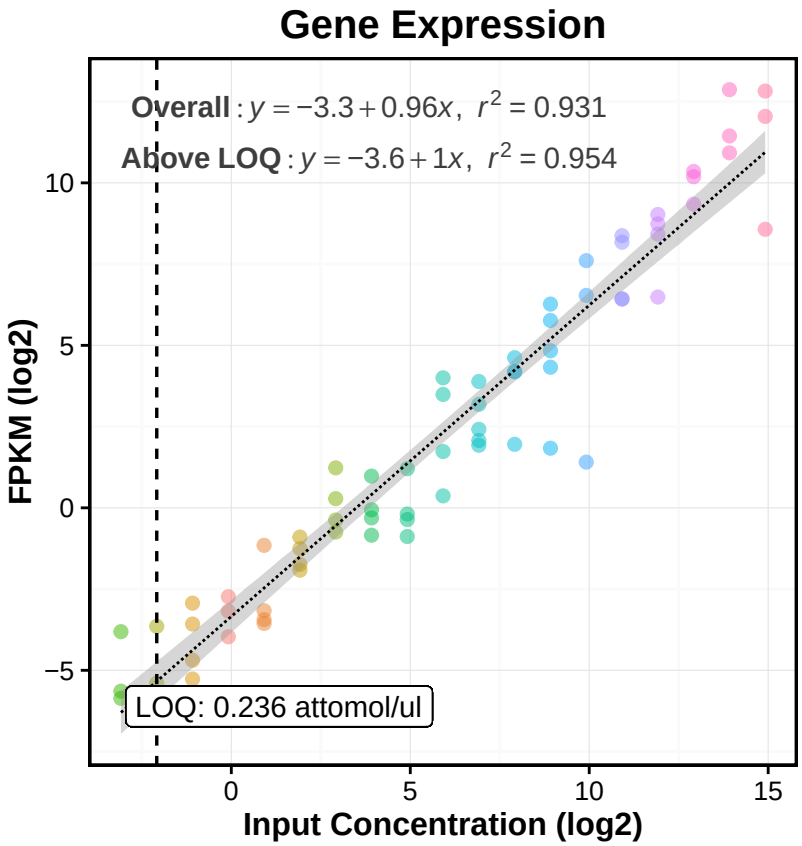
Load the data set:

```
data(UserGuideData_5.4.6.3)
head(UserGuideData_5.4.6.3)
```

```
##         InputConcent   Observed1   Observed2   Observed3
## R1_101       15.1062    0.958838    1.456650    0.960190
## R1_102       15.1062    0.806596    0.604539    0.652783
## R1_103      966.7970    2.650470    2.890570    3.211090
## R1_11       241.6990    3.876010    3.919950    4.246390
## R1_12        30.2124    0.779118    0.898644    0.733175
## R1_13      7734.3800 1305.710000 1328.950000 1358.970000
```

The first column gives input concentration for each sequin in attomol/ul. The other columns are the FPKM values for each replicate (three replicates in total). The following code will quantify the first replicate:

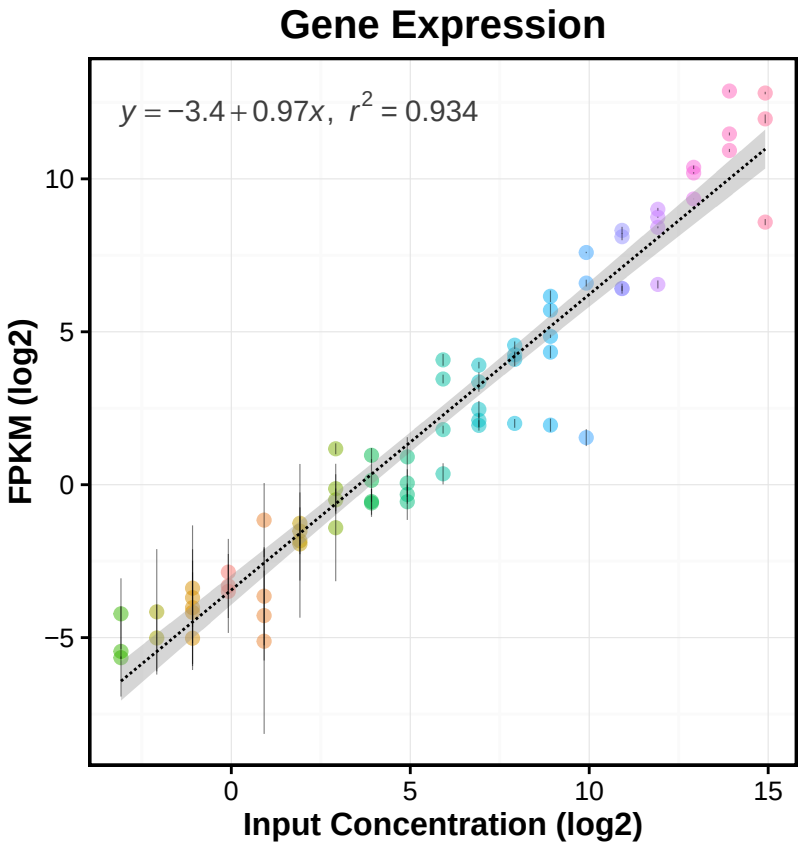
```
title <- 'Gene Expression'
xlab  <- 'Input Concentration (log2)'
ylab  <- 'FPKM (log2)'

# Sequin names
names <- row.names(UserGuideData_5.4.6.3)

# Input concentration
input <- log2(UserGuideData_5.4.6.3$InputConcent)

# Measured FPKM
measured <- log2(UserGuideData_5.4.6.3$Observed1)

anaquin <- AnaquinData(analysis='PlotLinear',
                            seqs=names,
                           input=input,
                        measured=measured)

plotLinear(anaquin, title=title, xlab=xlab, ylab=ylab, showLOQ=TRUE)
```



Coefficient of determination is over 0.90; over 90% of the variation (eg: technical bias) can be explained by the model. LOQ is 3.78 attomol/ul, this is the estimated emphirical detection limit.

We can also quantify multiple replicates:

```
title <- 'Gene Expression'
xlab  <- 'Input Concentration (log2)'
ylab  <- 'FPKM (log2)'

# Sequin names
names <- row.names(UserGuideData_5.4.6.3)

# Input concentration
input <- log2(UserGuideData_5.4.6.3$InputConcent)

# Measured FPKM
measured <- log2(UserGuideData_5.4.6.3[,2:4])

anaquin <- AnaquinData(analysis='PlotLinear',
                           seqs=names,
                         input=input,
                      measured=measured)

plotLinear(anaquin, title=title, xlab=xlab, ylab=ylab, showLOQ=TRUE)
```



Differential analysis

In this section, we show how to quantify differential expression analysis between expected fold-change and measured fold-change. We apply our method to a data set described in details in **Section 5.6.3** of the user guide.

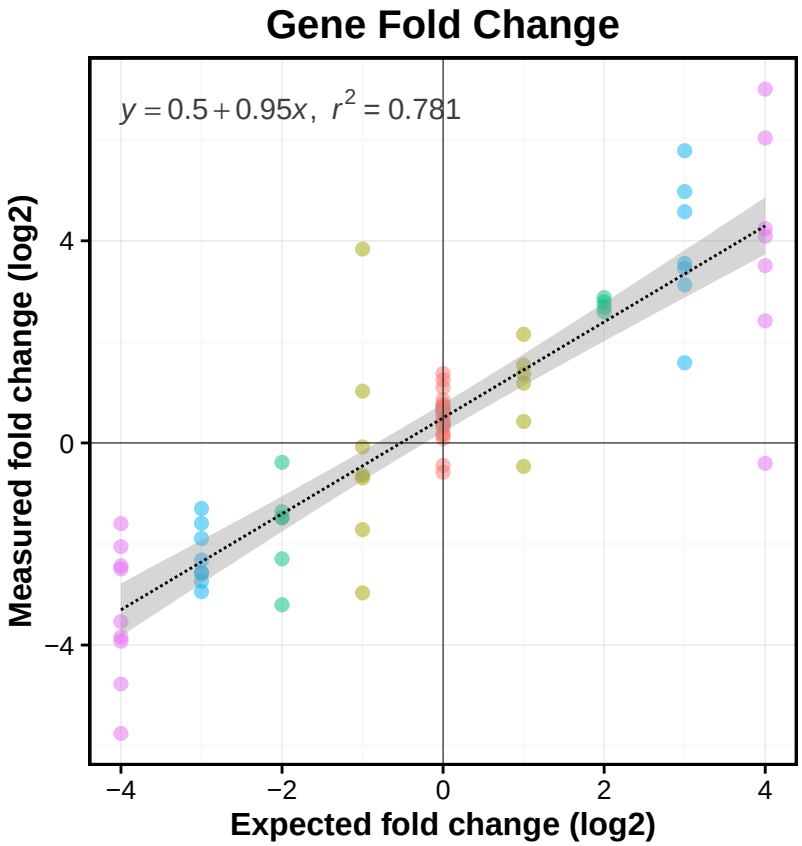
```
data(UserGuideData_5.6.3)
head(UserGuideData_5.6.3)
```

```
##         ExpLFC    ObsLFC       SD         Pval         Qval         Mean
## R1_101      -3 -1.890122 0.701723 7.069675e-03 2.056337e-02     9.953556
## R1_102      -4 -2.051777 0.546374 1.731616e-04 7.646243e-04    17.285262
## R1_103      -1  3.837784 0.377602 2.883289e-24 6.534028e-23  1221.301532
## R1_11       -4 -2.431582 0.591352 3.924117e-05 1.974336e-04    47.174250
## R1_12        1  1.542757 0.425562 2.887104e-04 1.214989e-03    73.008720
## R1_13        0  0.717701 0.242493 3.079564e-03 1.000416e-02 44053.259914
##         Label
## R1_101     TP
## R1_102     TP
## R1_103     TP
## R1_11      TP
## R1_12      TP
## R1_13      FP
```

For each of the sequin gene, we have expected log-fold change, measured log-fold change, standard deviation, p-value, q-value and mean. The estimation was done by DESeq2.

Run the following code to construct a folding plot:

```
title <- 'Gene Fold Change'
xlab  <- 'Expected fold change (log2)'
ylab  <- 'Measured fold change (log2)'

# Sequin names
names <- row.names(UserGuideData_5.6.3)

# Expected log-fold
input <- UserGuideData_5.6.3$ExpLFC

# Measured log-fold
measured <- UserGuideData_5.6.3$ObsLFC

anaquin <- AnaquinData(analysis='PlotLinear',
                            seqs=names,
                           input=input,
                        measured=measured)

plotLinear(anaquin, title=title, xlab=xlab, ylab=ylab, showAxis=TRUE,
           showLOQ=FALSE)
```



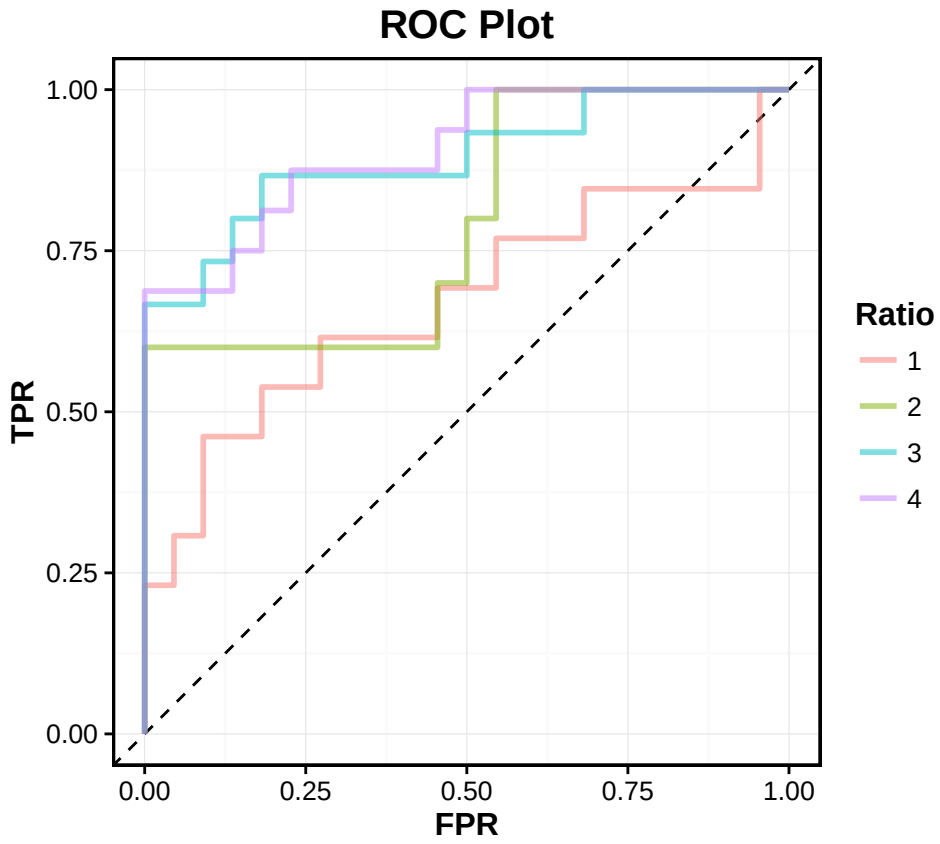
Outliers are obvious throughout the reference scale. Overall, DESeq2 is able to account for 78% of the

variation.

We can also construct a ROC plot. [1] has details on how the true-positives and false-positives are defined.

```
title <- 'ROC Plot'

# Sequin names
names <- row.names(UserGuideData_5.6.3)

# Expected ratio
ratio <- UserGuideData_5.6.3$ExpLFC

# How the ROC points are ranked (scoring function)
score <- 1-UserGuideData_5.6.3$Pval

# Classified labels (TP/FP)
label <- UserGuideData_5.6.3$Label

anaquin <- AnaquinData(analysis='PlotROC',
                           seqs=names,
                         ratio=ratio,
                         score=score,
                         label=label)

plotROC(anaquin, title=title, refRats=0)
```

```
##
##
## Ratio       AUC
## ------  -------
## 1        0.6713
## 2        0.7955
## 3        0.8939
## 4        0.9062
```



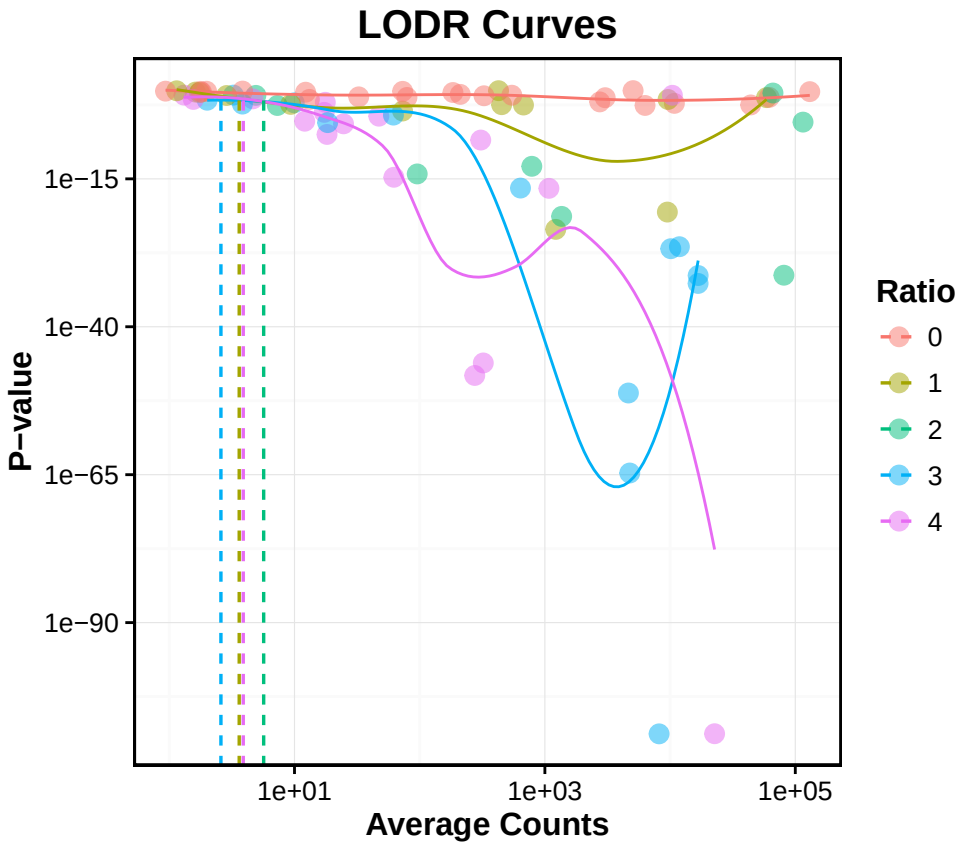
AUC statistics for LFC 3 and 4 are higher than LFC 1 and 2. Overall, all LFC ratios can be correctly classified relative to LFC 0.

Furthermore, we can construct limit of detection ratio (LODR) curves:

```
xlab  <- 'Average Counts'
ylab  <- 'P-value'
title <- 'LODR Curves'

# Sequin names
names <- row.names(UserGuideData_5.6.3)

# Measured mean
measured <- UserGuideData_5.6.3$Mean

# Expected log-fold
ratio <- UserGuideData_5.6.3$ExpLFC

# P-value
pval <- UserGuideData_5.6.3$Pval

# Q-value
qval <- UserGuideData_5.6.3$Qval

anaquin <- AnaquinData(analysis='PlotLODR',
                              seqs=names,
                          measured=measured,
```

```
                          ratio=ratio,
                           pval=pval,
                           qval=qval)

plotLODR(anaquin, xlab=xlab, ylab=ylab, title=title, FDR=0.1)
```

```
## [1] "Threshold: 0.0330366"
## [1] "Estmating LODR for 0"
## [1] "Estmating LODR for 1"
## [1] "Estmating LODR for 2"
## [1] "Estmating LODR for 3"
## [1] "Estmating LODR for 4"
##
##
##       Ratio       LODR
## ---  ------  ---------
## 2         1   3.617511
## 3         2   5.673070
## 4         3   2.584553
## 5         4   3.896317
```



Unsurprisingly, p-value is inverse quadratically related with average counts. All the LFC ratios systematically outperform LFC 0. The function also estimates the empirical detection limits, [1] has the details.