Chapter 18

Experimental Design and Power Calculation for RNA-seq Experiments

Zhijin Wu and Hao Wu

Abstract

Power calculation is a critical component of RNA-seq experimental design. The flexibility of RNA-seq experiment and the wide dynamic range of transcription it measures make it an attractive technology for whole transcriptome analysis. These features, in addition to the high dimensionality of RNA-seq data, bring complexity in experimental design, making an analytical power calculation no longer realistic. In this chapter we review the major factors that influence the statistical power of detecting differential expression, and give examples of power assessment using the R package *PROPER*.

Key words RNA-Seq, Gene expression, Sample size, Statistical power, Experimental design

1 Introduction

RNA-sequencing (RNA-seq) has become a routine technique in transcriptome analysis, where identifying differential expression (DE) remains a major task. As expression is a stochastic process in nature, the technological improvement in RNA-seq cannot bypass the presence of biological variability. It has been well recognized that replication is still necessary in making reliable statistical inference of DE [1]. The determination, or choice, of the number of replicates becomes a natural question in experimental design. In this section, we review the major factors affecting statistical power calculation in DE detection using RNA-seq. Specific examples of power assessment using the R package *PROPER*, released by Bioconductor [2], will be given in Subheading 3.

Classical power calculation that deals with a single hypothesis takes a few simple assumptions. These include (1) the effect size, representing the minimum difference that is scientifically meaningful between groups in comparison; (2) within-group variation, representing natural variation in observations regardless of between-group difference; (3) an acceptable type I error rate, usually in the form of p-value; and (4) the sample size. With these

Ewy Mathé and Sean Davis (eds.), *Statistical Genomics: Methods and Protocols*, Methods in Molecular Biology, vol. 1418, DOI 10.1007/978-1-4939-3578-9_18,

values set, one can calculate statistical power, the probability of rejecting the null hypothesis when the effect is as large as assumed. If a certain power level is desired, one can also do a reverse calculation to determine the minimum sample size to achieve the desired power while controlling the type I error rate.

In DE analysis for RNA-seq experiments, we consider similar factors with more complexity since it is a high throughput experiment querying all transcripts simultaneously, and these transcripts are not exchangeable. Below we discuss the main factors affecting the power calculation in RNA-seq:

1. **Average sequencing count levels**. Sequencing can be considered as a counting process. For a gene, the total variation we observe in its read counts from biological replicates reflects both the fluctuation of gene expression (biological variation) and the counting error from sequencing process (technical variation). The technical variation can be well approximated by a Poisson distribution. When counts are low, the variation due to Poisson counting error can easily shadow true DE we are interested in detecting. The average count level for a gene depends on its expression level, the sequencing efficiency (GC content and gene length [3] affect the number of reads yielded from a gene, for example), and the sequencing depth for the entire sample. Sequencing depth can be a factor of choice, at least to some extent. The distribution of expression levels includes which genes are being transcribed as well as the quantity of their transcription. This depends on the transcriptome of study and should not be arbitrarily specified.
2. **Natural variation of gene expression**. Even housekeeping genes do not have constant expression levels, so there is a natural variation of expression level between biological replicates. Gene expression varies differently for each gene, thus the ability to detect DE at a certain fold change varies between genes. In RNA-seq data this is often parameterized as dispersion parameter in a negative binomial model, which has close relationship to the variance parameter in log transformed expression [3].
3. **Type I error control**. The number of true positives identified is closely related to the amount of false positive (type I error) one can tolerate. In high throughput experiments like RNA-seq, the most widely used choice of type I error is false discovery rate (FDR), representing the expected proportion of false discoveries among all declared discoveries (positives). Controlling FDR takes into account of multiple testing, and yet is not overly conservative as controlling the family-wise error rate (the probability of making *any* false discovery). However, we emphasize the difference between nominal FDR and actual

FDR. Most DE detection methods provide only an estimate of FDR, which does not always provide adequate control of the FDR.

4. **Distribution of DE**. Abundant and strong signals are easier to detect than sparse and weak signals. What fraction of genes have DE between the conditions we plan to compare and contrast? For those with DE, how much do they differ? The magnitude of DE is often different between genes, with some genes undergoing dramatic changes and others subtle differences. Thus the distribution of DE size in addition to the fraction of genes with DE affects statistical power.
5. **Goal in DE detection**. The probability of detecting all genes with DE is often very low, especially considering that some DE genes experience only subtle differences. Are we interested in all genes that have any degree of DE, or are we only interested in DE beyond certain magnitude? Are we interested in power in terms of the proportion of true DE genes identified, or the actual number of true DE genes identified? When a large number of genes have DE, even a small fraction is a considerable number.

Making explicit assumptions on all of these factors is challenging, if not unreasonable. One more issue that complicates the sample size determination is that most DE detection methods, adequately, take advantage of the high throughput nature of RNA-seq data and apply some empirical Bayes techniques [3–6], creating dependency among the genes. On the other hand, to make the computation tractable, many sample size determination methods rely on strong and overly simplified assumptions. For example, some simplify by considering single gene expression [7, 8], some simplify by using Poisson model [9], or by setting the effect size, dispersion, and average count the same for all genes [10]. We argue, however, these overly simplified assumptions do not faithfully reflect the complexity of RNA-seq data. We advocate power evaluation in a comprehensive manner, under various scenarios of sample size and sequencing depth, rather than fixing too many high-dimensional factors and producing a single power curve.

2 Materials

As discussed in Subheading 1, multiple factors influence the statistical power in an RNA-seq experiment, including the transcriptome baseline profile (the number of genes, baseline expression levels, natural biological variation within group), target signal (proportion of genes with DE, the magnitude of DE), and technical choice (the number of samples, sequencing depth). Rather than making

specific strong assumptions on all of these, it makes much more sense to relate the overall profile, such as the distributions of baseline expression and dispersion, in the planned experiment to an existing real data set. Thus we encourage the use of actual RNA-seq data sets as the basis for simulation. The *PROPER* package provides information extracted from several RNA-seq data sets, including

- Cheung data: lymphoblastoid cell lines from 41 CEU individuals in International HapMap Project. The individuals represent a random sample from a population, thus the expressions show relatively large biological variations.
- Gilad data: human liver sample comparisons between male and female. The biological variations are smaller than those from Cheung data.
- Bottomly data: samples from two strains of inbred mice. Since the data are from inbred animal models, the biological variations among replicates are much smaller.
- MAQC data: benchmark data sets generated for the quality control of the sequencing technology. The replicates are technical replicates so there is no biological variation. These samples represent the lower bound of dispersion observed in RNA-seq data.

These data sets are chosen to represent different levels of dispersion distributions, which DE detection is sensitive to. Most of the real RNA-seq data are expected to have dispersions in between these. Another useful source of RNA-seq data summarized as count tables is ReCount [11]. In addition, pilot data set with count tables generated by investigators, if available, are probably the best to be used for establishing simulation scenarios.

3 Method

In this section we provide detailed examples of using the *PROPER* package to perform in silico experiments and evaluate power in realistic settings.

3.1 Obtaining PROPER

PROPER is a free and open-source R package released via the Bioconductor project. To install *PROPER*, start R and use the installation script provided by Bioconductor by entering

```
source("http://bioconductor.org/biocLite.R")
biocLite("PROPER")
```

The installation only needs to be run once. Packages that PROPER depends on will be automatically installed as well. Once installed, load the package with

```
library(PROPER)
```

Below we give examples for the most common situation of two group comparison. To give the users more flexibility, every function has options beyond we can include in the examples below. We encourage the readers to explore those with the function helps distributed with the package.

3.2 Setting Up Simulation Scenario

3.2.1 Using PROPER Provided Simulation Parameters

- **Simple Setting** Suppose we are considering a transcriptome with 20,000 genes (this depends on the species) and we think the baseline expression level (*lBaselineExpr*) and gene expression variation (*lOD*) are similar to those seen in the Cheung data (a random population of individuals). For signal we expect 5 % of all genes are DE, but we are not sure about the fold changes of each DE gene, thus we leave it at the default setting (normal with mean 0 and standard deviation 1.5). The following commands set up simulation option with these assumptions (note that we omit the magnitude parameter *lfc* since we choose the default setting).

```
sim.opts1 = RNAseq.SimOptions.2grp(ngenes = 20000, p.DE=0.05,
  lOD="cheung", lBaselineExpr="cheung")
```

One does not have to assume that the baseline expression and the variation of expression are based on the same source. For example, if our experiment involves more homogeneous population, we may expect the variation to be closer to that observed in inbred animals. We can simply change the *lOD* option alone to "*bottomly*" by

```
sim.opts2 = RNAseq.SimOptions.2grp(ngenes = 20000, p.DE=0.05,
  lOD="bottomly", lBaselineExpr="cheung")
```

- **Change the DE distribution** As mentioned above, the default setting for the magnitude of DE is a random variable from normal distribution with mean 0 and standard deviation 1.5. A normal random variable with mean 0 is a common choice in simulating the amount of differential expression. This means that among genes that do have differential expression, most of the effect sizes are near zero. Depending on the biological system under study, the user may choose other distributions of effect sizes that are more reasonable for their situation. This is done by passing an argument of *lfc*. For details see *?RNAseq.SimOptions.2grp*. Here we provide two examples below.

  To assume that all DE genes have differential expression of twofold change, while assuming the baseline expression and dispersion resemble those in the *Bottomly* data, we can set

```
sim.opts3 = RNAseq.SimOptions.2grp(ngenes = 20000, p.DE=0.05,
  lOD="bottomly", lBaselineExpr="bottomly", lfc=log(2))
```

A user can also specify a distribution for the magnitude of log fold change (*lfc*). This is done by passing a user-specified function as the *lfc* option. For example, to assume that the magnitude of log fold change (*lfc*) is centered around 1 with standard deviation 0.2, and that the up- and down-regulation is balanced, one may define the following function *mylfc*.

```
mylfc=function(n){
  rnorm(n,1,.2)*sign(rbinom(n,1,.5)-0.5)
}
# to visualize what data the function mylfc generates you may try
# hist(mylfc(1000))
sim.opts4 = RNAseq.SimOptions.2grp(ngenes = 20000, p.DE=0.05,
  lOD="bottomly", lBaselineExpr="bottomly", lfc=mylfc)
```

3.2.2 Using User Identified Data Source to Set Up Simulation Basis

If we want to use an independent source as pilot data to establish the simulation basis instead of using one of the data sources provided in PROPER, it can be estimated by the *estParam* function. The data input can simply be a matrix of sequencing counts or an *ExpressionSet*.[1] For the pilot data object *myCountMatrix*, do

```
my.param = estParam(myCountMatrix)
sim.opts5 = RNAseq.SimOptions.2grp(ngenes = 20000, p.DE=0.05,
  lOD=my.param$lOD, lBaselineExpr=my.param$lmean)
```

3.3 Running Simulation and DE Detection

Once the simulation scenario is prepared, we are ready to perform experiments in silico and evaluate how well the signals can be detected. This is done with the *runSims* function, which generates RNA-seq count data sets at the user-specified sample sizes and simulation settings, analyzes the data sets, and reports with a DE detection method the user chooses. The process will be repeated for a number of (user-specified) times. Since many RNA-seq data sets will be analyzed, this is computationally the most intensive part. However this only needs to be run once under a particular setting, and then different types of power-related quantities can be calculated. *PROPER* currently offers options to use *edgeR*, *DSS*, and *DESeq* as DE detection method. To install *DSS*, run

```
source("http://bioconductor.org/biocLite.R")
biocLite("DSS")
```

In the following example, we simulate data sets with the number of replicates in each group varying at 3, 5, 7, or 10, with the first simulation option we created in Subheading 3.2.

```
simres1 = runSims(Nreps = c(3, 5, 7, 10), sim.opts=sim.opts1,
  DEmethod="edgeR", nsims=20)
```

[1] ExpressionSet is a basic class of object used in Bioconductor. See http://www.bioconductor.org/packages/release/bioc/vignettes/Biobase/inst/doc/ExpressionSetIntroduction.pdf for more details

As a quick example, the above command used only 20 simulations (*nsims=20*). This is the minimum recommended number, which can be used for exploratory analysis or fast comparison between several *simOptions* settings. For a final determined simulation scenario, we recommend *nsims* 100 or above.

```
simres1b = runSims(Nreps = c(3, 5, 7, 10), sim.opts=sim.opts1,
  DEmethod="edgeR", nsims=100)
```

The rest of the chapter will use results from simulation output in *simres1b* for illustration.

3.4 Power Assessment and Comparison

Now that the simulation is completed and we are ready to evaluate power: the ability to identify true DE genes of interest. As in any analysis, there is some probability to produce false positives—genes that are not DE but falsely identified as DE genes, i.e. the type I errors. In order to evaluate the ability to detect true positives, we have to specify the level of false positives we are willing to put up with. The user decides both the type (FDR or raw *p*-value, by setting *alpha.type="fdr"* or *alpha.type="pval"*) and level of type I error control (*alpha.nominal*). Since we are dealing with high throughput experiments that query thousands of genes simultaneously, and in RNA-seq data the power is not uniform for every gene, there are often true but trivial DE that we do not find relevant. So the user also defines the DE signal of interest by setting the minimum magnitude of differential expression (via the option *delta*). The power to detect these signals of interest is referred to as the "*targeted power*."

The main function for power evaluation is *comparePower*, with default settings *alpha.type="fdr", alpha.nominal=0.1,delta=0.5*. For example, the statistical power assessment using the previous simulation result *simres1b* is obtained by

```
power1b = comparePower(simres1b)
summaryPower(power1b)
   Sample size Nominal FDR Actual FDR Marginal power Avg # of TD Avg # of FD FDC
[1,]          3        0.1         0.40           0.27          43          29 0.67
[2,]          5        0.1         0.22           0.41          68          20 0.29
[3,]          7        0.1         0.15           0.50          83          15 0.18
[4,]         10        0.1         0.10           0.58          98          12 0.12
```

The summary results above give marginal values of errors and sensitivity.

Interpretation of the Marginal Power Summary Table We use the above table as example to illustrate the interpretation of marginal power and error rates. With replicate number 3, we see that FDR is not well controlled by the edgeR method in this simulation. Though we aim to control FDR at 10 % level (nominal FDR), the actual FDR is as high as 40 %. Even with this high level of false discovery, only 27 % of true DE genes with log fold change

over 0.5 will be detected on average. We emphasize though, due to the large number total genes (20,000 in this example), even a small proportion may be a considerable number of genes. On average we can identify about 43 true DE genes with the cost of 29 false positives, make the false discovery cost (FDC) 0.67 for each true discovery. With the increase of samples size, as expected, power improves and cost of false discovery drops to 0.12 at 10 replicates each group.

The above example only shows power computed at default settings. In practice, the user may choose different nominal error rate and/or definition for DE of interest. For example, if one wants to re-evaluate power using raw *p*-values at 0.00001 level (this is lower than classical alpha level 0.05 since we have massive multiple testing), and target DE with at least twofold change, one can simply do

```
power1b.update = comparePower(simres1b,alpha.nominal=1e-5, alpha.type="pval",
  delta=log(2))
summaryPower(power1b.update)
Sample size    Nominal    Actual Marginal  Avg # of TD Avg # of FD   FDC
      type I error type I error   power
[1,]    3      1e-05      0.00150   0.12         16       6.00 0.3700
[2,]    5      1e-05      0.00062   0.26         37       2.60 0.0680
[3,]    7      1e-05      0.00036   0.36         53       1.50 0.0280
[4,]   10      1e-05      0.00012   0.47         69       0.52 0.0075
```

Stratified Targeted Power The marginal power table is a concise summary, but we urge the users to visualize the stratified power. This is important for RNA-seq experiments because the coverage on genes varies a great deal, and coverage is a crucial factor affecting statistical power. Even if one is not interested in genes with very low copies of transcripts, the variation in sequencing efficiency means that low coverage is not necessarily a sign for low expression. To visualize stratified power for the analysis output *power1b*, simply do

```
plotPower(power1b)
```

The *plotPower* function shows stratified power for each strata of expression level, as shown in Fig. 1a. Not surprisingly, the power is low in the first stratum, when average counts are below 10. When counts are as low as such, meaningful biological variation is hard to distinguish from Poisson counting error. However, we see that when counts are above 10, the power is much more acceptable.

Depending on the purpose of the RNA-seq experiment, we may hope to identify most of the true DE genes of interest (thus aiming for a high percentage, i.e., average power), or we may be interested in identifying a number of leads in a hypothesis generating project. In the latter case, the crude number of DE genes rather

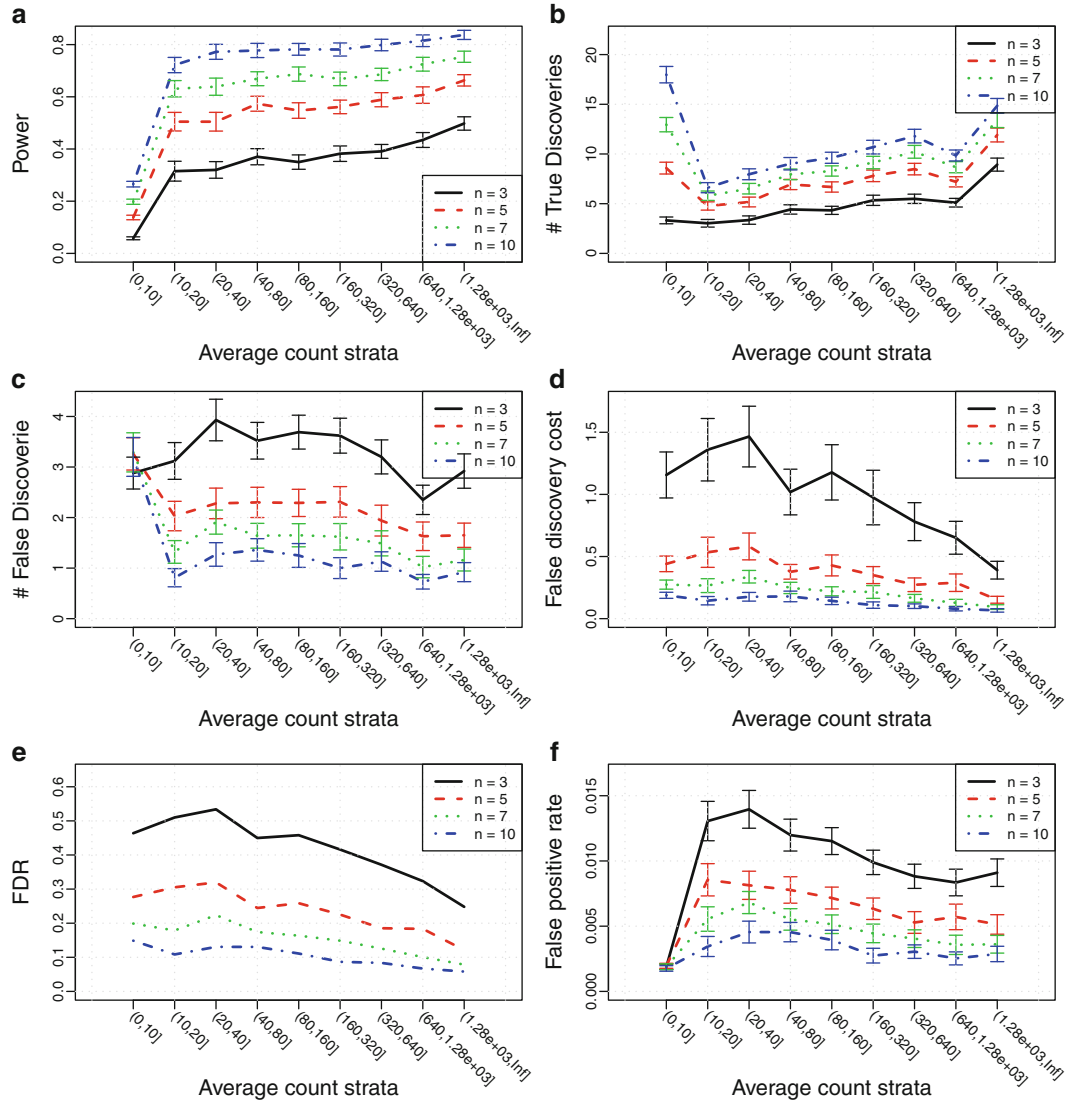

Fig. 1 Comprehensive visualization of stratified power, generated by the function plotAll

than the proportion is of interest. We can visualize this with the *plotPowerTD* function, which is shown in Fig. 1b.

```
plotPowerTD(power1b)
```

Based on the average power in Fig. 1a, we may consider filtering out genes with counts lower than 10 since the statistical power is very low. However, based on Fig. 1b, we realize that though the proportion of true DE genes identified in the first stratum is low, the actual number of DE genes is often higher than other strata, because a large fraction of genes fall in this range. A natural question arises: is it worthwhile to keep this stratum? The function *plotPowerFD* shows the number of false discoveries and *plotFDcost*

puts the true and false discoveries together in terms of FDC, which represents the number of false discoveries accompanying each true discovery at the current cutoff.

```
plotFDcost(power1b)
```

For a complete set of figures including the average power, power as crude number of true discoveries, false discoveries, FDC, actual FDR, and actual type I error rates for each strata, one can simply call the function

```
plotAll(power1b)
```

This produces the combined figure as shown in Fig. 1.

3.5 More Samples or Deeper Sequencing

One striking observation in the power curves is that genes with low count numbers are associated with low statistical power even for as many as 10 replicates. To get around this and increase power, we may consider using more samples or increasing the sequencing depth—but which choice provides more benefit? We provide a table that allows the user to compare both types of choices in a simple matrix form. To see power changes associated with various sequencing depths and sample sizes, we use the function *power.seqDepth* and provide two inputs: the simulation result and the initial power assessment output.

```
power.seqDepth(simres1b,power1b)
  3 reps 5 reps 7 reps 10 reps
0.2  0.21  0.34  0.42   0.50
0.5  0.24  0.38  0.46   0.55
1    0.27  0.41  0.50   0.58
2    0.30  0.44  0.53   0.61
5    0.34  0.48  0.57   0.65
10   0.36  0.51  0.60   0.68
```

Each row of the table shows the results if the experiment was done at a sequencing depth relative to the one used in initial power assessment. For example, the third row (with relative depth 1) represents power at the same sequencing depth used in the original simulation setting, and we have 41 % marginal power using 5 replicates in each group. If we use half the sequencing depth (relative depth 0.5) but double the sample size (10 reps), we can get an increased power at 55 % for the same amount of sequencing. Whether to increase the sample size or sequencing depth depends on the availability of biological samples and the sequencing cost, which changes rather rapidly, thus we leave the decision to the user. The table enables the user to make informed decision taking statistical power into account.

3.6 To Filter or Not to Filter

As we realize the particular challenge in detecting true DE genes from noise among genes with very low counts, we often face the difficult choice of whether to filter out these genes. Filtering decreases the number of total tests and thus reduces the burden for multiple testing adjustment, a necessary component in high throughput data analysis. Though we forego the possibility of detecting DE in those genes, we do not generate false discoveries from these either. This increases marginal power, and may reduce overall FDC. The decision to filter should be made prior to actual data analysis, as repeated analysis of the same data set (trying several filters, for example) is another source of increased error. Using the simulation result we can assess whether filtering is likely to help in data generated in similar settings, thus make an informed decision before analyzing the actual experimental data.

To compare how power would have been if we filtered out the first strata (*strata.filtered=1*) in the expression level (*filter.by="expr"*), we use the following

```
powers1b.update = comparePower(simres1b, alpha.type="fdr", alpha.nominal=0.1,
 strata = c(0, 10, 2^(1:7)*10, Inf), filter.by="expr",strata.filtered=1, delta=0.5)
summaryPower(powers1b.update)
Sample size Nominal FDR Actual FDR Marginal power Avg # of TD  Avg # of FD  FDC
[1,]       3         0.1       0.42          0.43          44          31 0.72
[2,]       5         0.1       0.24          0.61          62          20 0.32
[3,]       7         0.1       0.16          0.71          72          14 0.20
[4,]      10         0.1       0.11          0.81          82          11 0.13
```

The option *strata* defines how the average counts are stratified. The values for *strata* define eight strata for average gene counts: $(0, 10]$, $(10, 20]$, $(20, 40]$, $(40, 80]$, $(80, 160]$, $(160, 320]$, $(320, 640]$, $(640, 1280]$, and $(1280, \infty)$. One can choose to filter out more than one strata by changing *strata.filtered*. Compared with the earlier results without filtering, the marginal powers are greatly improved. For example, with 5 replicates in each group, the marginal power increased from 0.41 to 0.61. However, the average number of true discovery also reduced from 68 to 62. Whether filtering is beneficial depends on the balance between the reduction of false positive and the reduction of true positives, which varies from data set to data set, and depends on the definition of target size.

PROPER provides flexible ways for gene filtering. For example, if we only want to filter out genes with counts up to 8, we can re-define the *strata* as shown below:

```
power1b.update = comparePower(simres1b, alpha.type="fdr", alpha.nominal=0.1,
 strata = c(0, 8, 2^(1:7)*10, Inf), filter.by="expr",
 strata.filtered=1, stratify.by="expr", delta=0.5)
summaryPower(power1b.update)# the output is omitted below
```

4 Notes

Results from the above examples are generated for the purpose of illustrating the use of *PROPER*. We do not argue that these necessarily reflect the typical situation in RNA-seq experiments that a user will run into. The users are encouraged to choose simulation scenarios that are closest to their study. One may consider the population heterogeneity of samples (ecological samples from random populations, or controlled experiments on cell lines), tissues types, and species in selecting a data source as simulation basis. One may consult reported differential expression in published results, especially those validated in other platforms, as expected DE distribution.

The DE detection considered here is at the gene or transcript level. Examples of methods using read counts at this level include edgeR [5], DESeq [4], and DSS [3]. Another class of methods such as Cufflinks [12] considers alternative splicing and different isoforms of the same transcript. These first estimate isoform expressions and then perform DE analysis on the estimates. PROPER does no apply to these methods.

The results shown in the examples are generated using R version 3.2.0 and Bioconductor version 3.1.0, PROPER version 1.2.0.

References

1. Hansen KD, Wu Z, Irizarry RA, Leek JT (2011) Sequencing technology does not eliminate biological variability. Nat Biotechnol 29 (7):572–573
2. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J et al (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5(10):R80
3. Wu H, Wang C, Wu Z (2013) A new shrinkage estimator for dispersion improves differential expression detection in RNA-seq data. Biostatistics 14(2):232–243
4. Anders S, Huber W (2010) Differential expression analysis for sequence count data. Genome Biol 11(10):R106
5. Robinson MD, McCarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26 (1):139–140
6. McCarthy DJ, Chen Y, Smyth GK (2012) Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. Nucleic Acids Res 40 (10):4288–4297
7. Fang Z, Cui X (2011) Design and validation issues in RNA-seq experiments. Brief Bioinform 12(3):280–287
8. Hart SN, Therneau TM, Zhang Y, Poland GA, Kocher J-P (2013) Calculating sample size estimates for RNA sequencing data. J Comput Biol 20(12):970–978
9. Li C-I, Su P-F, Shyr Y (2013) Sample size calculation based on exact test for assessing differential expression analysis in RNA-seq data. BMC Bioinformatics 14(1):357
10. Li C-I, Su P-F, Guo Y, Shyr Y (2013) Sample size calculation for differential expression analysis of RNA-seq data under Poisson distribution. Int J Comput Biol Drug Des 6 (4):358–375
11. Frazee AC, Langmead B, Leek JT (2011) ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets. BMC Bioinformatics 12(1):449
12. Trapnell C, Roberts A, Goff L, Pertea G, Kim D, Kelley DR, Pimentel H, Salzberg SL, Rinn JL, Pachter L (2012) Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. Nat Protoc 7(3):562–578