PROPER: PROspective Power Evaluation for RNAseq

Hao Wu

Department of Biostatistics and Bioinformatics
Emory University
Atlanta, GA 303022

hao.wu@emory.edu

October 14, 2014

Contents

Abstract

This vignette introduces the use of PROPER (**PRO**spective **P**ower **E**valuation for **R**NAseq), which is designed to provide power–sample size assessment in differential expression detection from RNA-seq data.

1 Introduction

RNA-seq has become a routine technique in quantifying and comparing gene expressions, and is widely used in differential expression (DE) detection. The determination of sample sizes is an important question in RNA-seq experimental design, and required in many grant applications. Due to the complex nature of RNA-seq data and the analysis techniques, traditional methods for evaluating power vs. sample size is not applicable. The main difficulty in the problem lie in following aspects:

- Multiple-testing problem. Since the DE test is performed simultaneously for many genes, the type I error is often controlled by false discovery rate (FDR), which poses difficulty in traditional power calculation.
- Coverage depth. Unlike gene expression microarrays, the statistical power of the DE test from RNA-seq data is greatly affected by the expression level (or the sequence counts). Genes with higher counts (such as longer genes) will be easier to detect even when the effect sizes are the same as the low-count genes (shorter genes).
- DE detection procedure. Many existing RNA-seq DE methods implement different type of "shrinkage" procedures to stabilize the estimation of within group variations. This makes the power calculation an intractable problem.

Due to these difficulties, we argue that the power calculation for RNA-seq can not be solve analytically. We develop R package PROPER (stands for **PRO**spective **P**ower **E**valuation for **R**NAseq), which provides simulation-based method to compute power-sample size relationship. The software package contain following three modules:

1. Data generation. The RNA-seq count data are simulated from a negative binomial model based on user specified model parameters.
2. DE detection. The DE detection is performed using existing software package.
3. Power assessment. The DE detection results are assessed to generate power–sample size relationship. Various power-related quantities are computed, including marginal and stratified power, FDR, number of true/false discoveries, false discovery cost, type I error. These quantities will help investigators to determine desirable sample sizes.

2 Using `PROPER`

To assess the power-sample size relationship using `PROPER`, one needs to follow the simple steps below.

1. Set up a simulation scenario and select a method for detecting DE.
2. Run simulation and DE detection.
3. Evaluate the power–sample size relationship.

2.1 Set up a simulation scenario

To set up a simulation scenario, call the `RNAseq.SimOptions.2grp` function. Users have the option to set up number of genes, percentage of genes with DE, sequencing depth (total number of read counts), baseline expression level, biological variations, and magnitude of DE (effect sizes). These factors all affect the power of detecting DE, yet making assumptions on these is not a trivial task. We provide options to make use of parameters estimated from real RNAseq data sets:

- Cheung data: expressions of lymphoblastoid cell lines from 41 CEU individuals in International HapMap Project. The data are from unrelated individuals, so the expressions show large biological variations overall.
- Gilad data: human liver sample comparisons between male and female. The biological variations are smaller than those from Cheung data.
- Bottomly data: include 21 samples from two stains of inbreed mice. Since the data are from inbreed animal models, the biological variations among replicates are much smaller.
- MAQC data: these are the benchmark datasets generate by FDA to test the sequencing technology. The replicates are technical replicates so there's essentially no dispersion.

These data sets are chosen for they represent different levels of dispersion distributions, which DE detection results can be sensitive to. Most of the real RNA-seq data should have dispersions in between of these.

The following command sets up simulation options with 20,000 genes, 5% genes being DE, baseline expression and dispersion based on Cheung data:

```
>   library(PROPER)
>   sim.opts.Cheung = RNAseq.SimOptions.2grp(ngenes = 20000, p.DE=0.05,
+     lOD="Cheung", lBaselineExpr="Cheung")
```

To make simulation options based on Bottomly data, do:

```
>   sim.opts.Bottomly = RNAseq.SimOptions.2grp(ngenes = 20000, p.DE=0.05,
+     lOD="Bottomly", lBaselineExpr="Bottomly")
```

To set up meaningful effect size between two groups is non-trivial. The package allows for both parametric setting for the log fold change as well as resampling based simulation if the user wishes to use some pilot data. The default is a mixture of 0 for non-DE genes and $N(0, 1.5^2)$ for DE genes.

2.2 Run simulation and DE detection

With the simulation settings ready, one can call `runSims` function to generate count data and perform DE detection. Since the main goal of the function is to evaluate power and sample size relationship, users specify the sample sizes (number of replicates in each group) for which they want to evaluate power. By default, we evaluate power when there are 3, 5, 7, and 10 samples in each treatment group. We found through preliminary results that 10 samples in each group provides adequate power when the dispersions are small

(similar to those from Bottomly data). If the dispersions are large, we suggest users increase sample sizes, but it will take more computational time.

This is the most computational intensive part of the software. The computational performance depends on the DE detection software and the scale of the simulation. For a simulation with 50,000 genes, four different sample sizes, and using `edgeR` for DE detection, each simulation takes around 10 seconds on a Macbook Pro laptop with 2.7Ghz i7 CPU and 16G RAM, which translates to 17 minutes for 100 simulations.

The number of simulations need to be specified. We recommend to run at least 20 simulations in order to get stable results. The simulation results should be saved for the next step power evaluation.

The built-in DE detection method inlcude `edgeR`, `DESeq` and `DSS`. These Bioconductor packages need to be installed in order to use them.

To run simulation and DE detection using edgeR, do:

```
>   simres = runSims(Nreps = c(3, 5, 7, 10), sim.opts=sim.opts.Cheung,
+     DEmethod="edgeR", nsims=20)
```

2.3 Evaluate the powers

With the simulation results from `runSims`, one can use `comparePower` function to evaluate powers in different ways. `comparePower` is a rather comprehensive function with many options. Users have the ability to specify:

- Method to control type I error (by raw p-values or FDR), and the desired type I error threshold for calling DE.
- Method to stratify genes (by baseline expression level or dispersion) and the strata, since the function will provide stratified power-related quantities.
- Method to define "biologically interesting genes". There could be many true DE genes with small changes, which can be difficult to detect. Including these gene in the "true DE" set will hurt the power. We advocate to estimate the "targeted" power, which is defined as the power to detect biologically interesting genes, or genes with at least moderate expression changes. We argue that the targered power are of more interests for investigators. This function provide options for defining biologically interesting genes, by either log fold change or standardize log fold change (log fold change divided by square root of dispersion).

It is important to point out that this power evaluation step is independent of the previous simulation/DE dection step (using `runSims`). With the same results from `runSims`, one can evaluate the powers using different criteria in `comparePower`.

The following command evaluate the powers with following criteria: (1) genes with FDR<0.1 are deemed DE; (2) stratification is done by baseline expression level; (3) biologically interesting genes are defined as those with log fold change greater than 0.5:

```
>   powers = comparePower(simres, alpha.type="fdr", alpha.nominal=0.1,
+     stratify.by="expr", delta=0.5)
```

Users can summarize or visualize the results from `comparePower` function. For example, the result can be nicely summarized by calling `summaryPower` function, which genreates following table for marginal power-related quantities.

```
> summaryPower(powers)
```

Sample size	Nominal FDR	Actual FDR	Marginal power	Avg # of TD	Avg # of FD	FDC
3	0.10	0.44	0.27	106.42	86.00	0.81
5	0.10	0.27	0.41	165.79	62.53	0.38
7	0.10	0.19	0.49	205.13	49.21	0.24
10	0.10	0.14	0.58	245.15	38.64	0.16

The table summarizes marginal power-related results under each sample size, including marginal power, true discovery (TD), false discovery (FD), and false discovery cost (FDC, defined as the number of FD divided by the number of TD).

To visualize the power results, there are a list of functions generating different figures for visualization. Specifically:

- To plot the stratified power, do

  ```
  > plotPower(powers)
  ```

  Each line in the figure represents the stratified power under a certain sample size. In the figure below, the powers are calculated stratitifed by the average counts of genes. It shows that when the counts are very low (between 0 and 10), the powers are very low. For genes with more than 10 reads, powers increase significantly. Moreover, it shows when there are 10 samples in each group, the powers are close to the canonically desired level of 0.8 for genes with reasonably large counts.

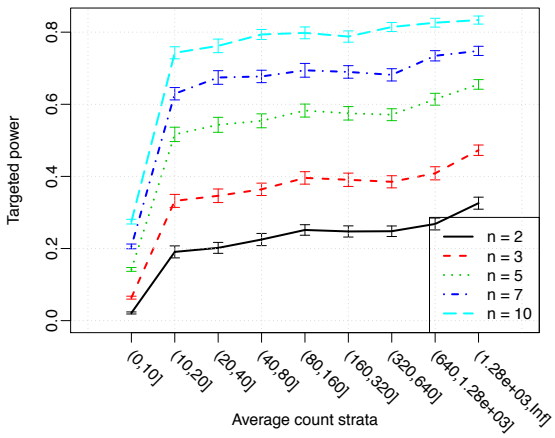

Figure 1: Stratified target power

- To plot the number of true discoveries, do

  ```
  > plotPowerTD(powers)
  ```

  Again, each line represents results from one sample size. The numbers of true discoveries from each stratum are plotted.

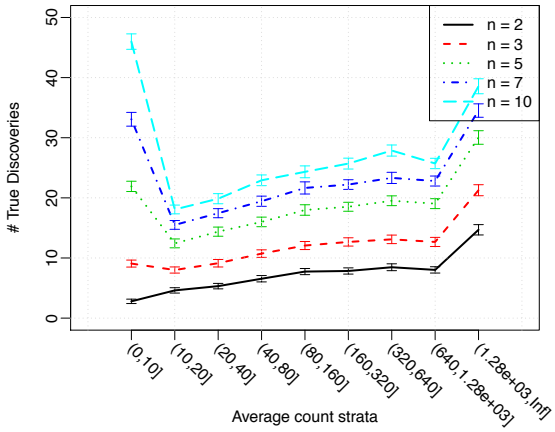

Figure 2: Stratified number of true discoveries

- To plot the stratified false discovery cost, do

```
> plotFDcost(powers)
```

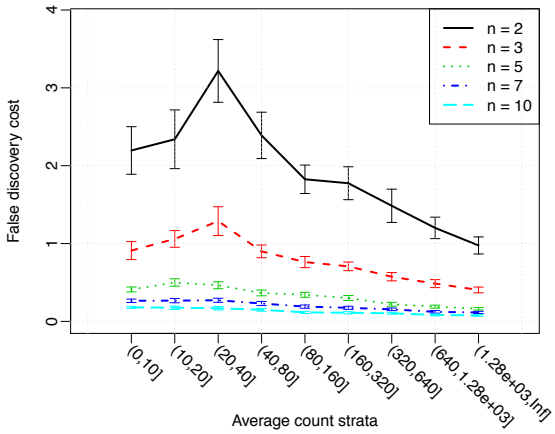

Figure 3: Stratified false discovery cost

- Or one can call `plotAll` to generate all plots in a single figure:

```
> plotAll(powers)
```

2.4 Additional functionalities

2.4.1 Power and sequencing depth

In RNAseq experiments, the design stage involves decisions beyond sample size. One may face a choice between fewer samples sequenced deeper versus more samples sequenced shallower. The sequencing depth and sample size relationship can be assessed using `power.seqDepth` function:

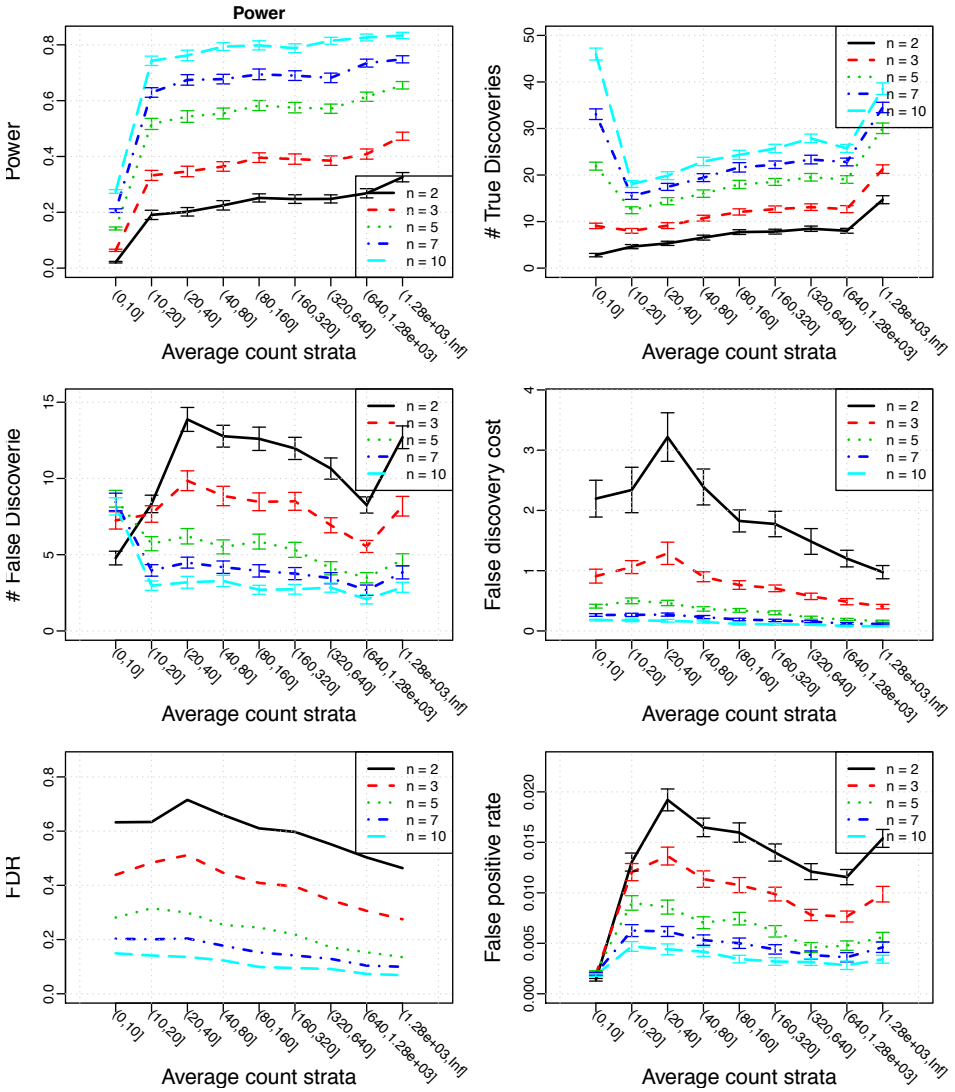

Figure 4: Visualization of stratified power-related quantities

```
> power.seqDepth(simres, powers)
```

This generates the following table.

	3 reps	5 reps	7 reps	10 reps
0.2	0.21	0.33	0.41	0.50
0.5	0.24	0.37	0.46	0.55
1	0.27	0.41	0.49	0.58
2	0.30	0.44	0.53	0.62
5	0.34	0.48	0.57	0.66
10	0.36	0.51	0.60	0.69

This table reads as the following: Using 5 replicates in each group with average coverage similar to that from the Cheung data (or whatever you used in your simulation set up), the marginal power is 0.41. Doubling the depth would increase the power to 0.44, yet doubling the sample size at the same depth would increase the power to 0.58. Thus for this example, using more replicates is more helpful than sequencing deeper.

2.5 To filter or not to filter

Differential expression is particularly difficult to detect in genes with very low counts. This is because the Poisson counting error is comparable or even greater than the biological variation. Thus it is difficult to identify true DE from noise in these genes and it is often desirable to filter out some genes before performing DE test. Filtering decreases the number of total tests and thus reduced the burden for multiple testing adjustment. Though we forego the possibility of detecting DE for the genes filtered out, we do not pay the price in false positives either. This may lead to better power, but the tradeoff is not apparent and the cutoff for filtering is not obvious.

For the example described in Section 2, if we want to evalute power when we filter out genes within the first strata of expression (average counts in 0-10), we specify the options `filter.by` and `strata.filtered`:

```
> powers = comparePower(simres, alpha.type="fdr", alpha.nominal=0.1,
+     strata = c(0, 10, 2^(1:7)*10, Inf), filter.by="expr",
+     strata.filtered=1, stratify.by="expr", delta=0.5)
```

2.6 Interpret the result

The major application of `PROPER` is to help choose proper sample size, especially for grant applications. The results from PROPER can be interpreted and written as the following in a grant application (with proper modification):

"Assume the transcriptome mean and variation profiles are similar to those from mouse striatum cells in previous studies, and the magnitude of true differential expression is similar to the level observed between two strains of mice. If we expect to identify 80% of DE genes whose log fold change is beyond 0.5, when the sequencing depth is at 5-million and FDR is controlled at 0.2, we need to have at least 5 samples in each treatment group."

3 Advanced options

3.1 setting the effect size by resampling from pilot or public data

Sometimes the user may not want to make a parametric assumption for the effect size, but can feel comfortable expecting the overall DE is similar to that observed in another experiment. In this case we may use resampling based simulation. We include an example data frame of mean expression measures (in log scale) in PBMC samples under various conditions. The data can be access by `data(pbmc)`.

3.2 setting the target of interest by standardized effect size

There is no general agreement on what makes an interesting or relevant amount of differential expression. Many researchers use fold change as a unit of effect size, as this is a unit less measure relative to reference expression level. Other argue that the relevant effect size may depend on a gene's natural biological coefficient of variation (BCV). If the user wishes to define targets of interest as genes with DE relative to BCV, one may change set the option `target.by` to `effectsize` (the default is `lfc` for log fold change).

The following command evaluate the powers with following criteria: (1) genes with estimated fdr<0.1 are declared DE; (2) stratification is done by dispersion level; (3) biologically interesting genes are defined as those with standardized log fold change greater than 1:

```
>     powers = comparePower(simres, alpha.type="fdr", alpha.nominal=0.1,
+       stratify.by="dispersion", target.by="effectsize", delta=1)
```

3.3 Choosing raw p-value instead of FDR

As one inspects the DE detection in simulation, a user may notice that the actual false discovery proportion differs from the nominal FDR reported from the DE detection method. One may decide to use unadjusted raw p-values to define DE gene in such situations (though typically much lower than 0.05 to account for multiple testing).

The following command evaluate the powers with following criteria: (1) genes with pvalue<0.001 are deemed DE; (2) stratification is done by dispersion level; (3) biologically interesting genes are defined as those with standardized log fold change greater than 1:

```
>     powers = comparePower(simres, alpha.type="pval", alpha.nominal=0.001,
+       stratify.by="dispersion", target.by="effectsize", delta=1)
```

4 Conclusion

`PROPER` provides methods to assess the power–sample size relationship for DE detection from RNA-seq data. A comprehensive evaluation of statistical power, as well as actual type I error, over a range of sample sizes, are obtained based on simulation studies.

5 Session Info

```
> sessionInfo()
```

```
R version 3.1.0 (2014-04-10)
Platform: x86_64-apple-darwin10.8.0 (64-bit)

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
[1] tools_3.1.0
```