RnaSeqSampleSize: Sample size estimation based on real RNA-seq data

Shilin Zhao

October 14, 2014

Abstract

In this vignette, we demonstrated the application of RnaSeqSampleSize as an sample size estimation tool for RNA-seq data.

RnaSeqSampleSize package provides the following features:

- Estimation of sample size or power by single read count and dispersion;
- Estimation of sample size or power by prior data;
- Power curve visualization;
- Optimization by power or sample size matrix;

Contents

1 Introduction

Sample size estimation is the most important issue in the design of RNA sequencing experiments.However, thousands of genes are quantified and tested for differential expression simultaneously in RNA-seq experiments. The false discovery rate for statistic tests should be controlled. At the same time, the thousands of genes have widely distributed read counts and dispersions, which were often estimated by experience or set at the most conservative values in previous methods. As a result, the estimated sample size will be inaccurate or over-estimated.

To solve these issues, we developed a sample size estimation method based on the distributions of gene read counts and dispersions from real data. Datasets from the user's preliminary experiments or the Cancer Genome Atlas (TCGA) can be used as reference. The read counts and their related dispersions will be selected randomly from the reference based on their distributions, and from that, the power and sample size will be estimated and summarized.

2 User friendly web interface

A user friendly web interface for RnaSeqSampleSize package is provided at http://cqs.mc.vanderbilt.edu/shiny/RnaSeqSampleSize/. Most of the functions in Examples section can be performed in this website.

3 Examples

3.1 Estimation of sample size or power by single read count and dispersion

3.1.1 Power estimation

For example, if we are estimating the power of finding significant genes for RNA-seq data with specified sample size, and we have the following parameters:

- Samples in each group: 63;
- Minimal fold change between two groups: 2;
- Minimal average read counts: 5;
- Maximal dispersion: 0.5;
- False discovery rate (FDR): 0.01;

As a result, the estimated power is 0.8 by *est_power* function.

```
example(est_power)

##
## est_pw> n<-63;rho<-2;lambda0<-5;phi0<-0.5;f<-0.01
##
```

```
## est_pw> est_power(n=n, rho=rho, lambda0=lambda0, phi0=phi0,f=f)
## [1] 0.8
```

3.1.2 Sample size estimation

For example, if we are estimating the sample size for RNA-seq data to achieve desired power of finding significant genes, and we have the following parameters:

- Desired power of finding significant genes: 0.8;
- Minimal fold change between two groups: 2;
- Minimal average read counts: 5;
- Maximal dispersion: 0.5;
- False discovery rate (FDR): 0.01;

As a result, the estimated sample size is 63 by *sample_size* function.

```
example(sample_size)

##
## smpl_s> power<-0.8;rho<-2;lambda0<-5;phi0<-0.5;f<-0.01
##
## smpl_s> sample_size(power=power, f=f,rho=rho, lambda0=lambda0,
   phi0=phi0)
## [1] 63
```

3.2 Estimation of sample size or power by prior data

3.2.1 Power estimation

For example, if we are estimating the power of finding significant genes for RNA-seq data with specified sample size, and we have the following parameters:

- Samples in each group: 65;
- Minimal fold change between two groups: 2;
- Prior data: TCGA READ data;
- False discovery rate (FDR): 0.01;

```
example(est_power_distribtuion)
```

```
##
## est_p_> #Please note here the parameter repNumber was very small
   (5) to make the example code faster.
## est_p_> #We suggest repNumber should be at least set as 100 in
   real analysis.
## est_p_> est_power_distribtuion(n=65,f=0.01,rho=2,
   distributionObject=”TCGA_READ”,repNumber=5)
## [1] 0.8096763
##
## est_p_> #Power estimation based on some interested genes. We use
   storeProcess=TRUE to return the details for all selected genes.
## est_p_> selectedGenes<-names(TCGA_READ$pseudo.counts.mean)[c
   (1,3,5,7,9,12:30)]
##
## est_p_> powerDistribution<-est_power_distribtuion(n=65,f=0.01,rho
   =2,distributionObject=”TCGA_READ”,selectedGenes=selectedGenes,
   minAveCount=1,storeProcess=TRUE)
##
## est_p_> str(powerDistribution)
## List of 3
##  $ power      : num [1:24] 0.6753 0.6847 0.687 0.0272 0.9946 ...
##  $ count      : Named num [1:24] 55 27 2000 249 1365 ...
##   ..- attr(*, ”names”)= chr [1:24] ”A1BG” ”A2BP1” ”A2M” ”A4GALT”
   ...
##  $ dispersion: num [1:24] 0.7 2.7 0.3 0.5 0.2 1.1 6.4 0.6 0.2 0.2
   ...
##
## est_p_> mean(powerDistribution$power)
## [1] 0.774561
##
## est_p_> #Power estimation based on genes in interested pathway
## est_p_> powerDistribution<-est_power_distribtuion(n=65,f=0.01,rho
   =2,distributionObject=”TCGA_READ”,pathway=”00010”,minAveCount=1,
   storeProcess=TRUE)
##
## est_p_> mean(powerDistribution$power)
## [1] 0.774056
```

Here we demonstrated the power estimation by prior data in three different situations.

- If we are intesested in all genes, we can use repNumber parameter to specify random number of genes to perform power estimation;

- If we are only intesested in a list of genes, we can use selectedGenes parameter to specify the list of genes to perform power estimation;

- If we are only intesested a specified pathway, we can use pathway and species parameters to specify the genes in a pathway to perform power estimation.

As a result, we use *est_power_distribtuion* function and find the estimated power is 0.81 for random genes, 0.77 for specified gene list, and 0.77 for genes in Glycolysis and Gluconeogenesis (pathway 00010) pathway.

3.2.2 Sample size estimation

For example, if we are estimating the power of finding significant genes for RNA-seq data with specified sample size, and we have the following parameters:

- Desired power of finding significant genes: 0.8;
- Minimal fold change between two groups: 2;
- Prior data: TCGA READ data;
- False discovery rate (FDR): 0.01;

As a result, we use *sample_size_distribution* function and find the estimated sample size is 65 for random genes.

```
example(sample_size_distribution)

##
## smpl__> #Please note here the parameter repNumber was very small
   (5) to make the example code faster.
## smpl__> #We suggest repNumber should be at least set as 100 in
   real analysis.
## smpl__> sample_size_distribution(power=0.8,f=0.01,
   distributionObject=”TCGA_READ”,repNumber=5,showMessage=TRUE)
## [1] ”x= 1   f(x)= −0.8”
## [1] ”x= 33   f(x)= −0.272193328875705”
## [1] ”x= 65   f(x)= 0.00149392491701938”
## [1] ”x= 49   f(x)= −0.120384771506399”
## [1] ”x= 57   f(x)= −0.0544219672787534”
## [1] ”x= 61   f(x)= −0.0251251289997526”
## [1] ”x= 63   f(x)= −0.0114954578587679”
## [1] ”x= 64   f(x)= −0.00490859274826694”
## [1] 65
```

3.3 Power curve visualization

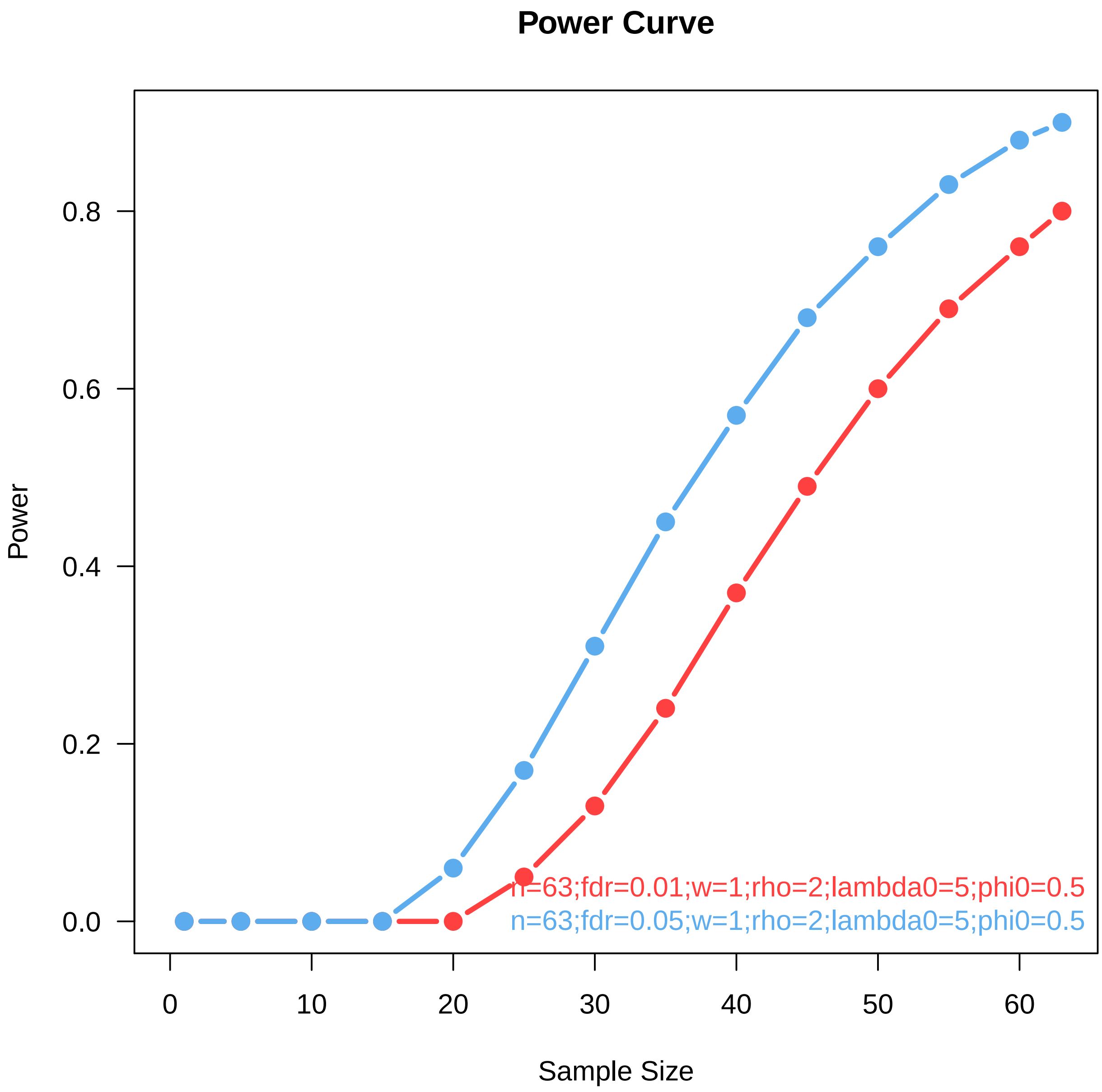
For example, if we are going to compare the power of finding significant genes for different false discovery rate, and we have the following parameters:

- Samples in each group: 63;
- Minimal fold change between two groups: 2;
- Minimal average read counts: 5;

- Maximal dispersion: 0.5;
- False discovery rate (FDR): 0.01 and 0.05;

```
example(est_power_curve)

##
## est_p_> result1<-est_power_curve(n=63, f=0.01, rho=2, lambda0=5,
   phi0=0.5)
##
## est_p_> result2<-est_power_curve(n=63, f=0.05, rho=2, lambda0=5,
   phi0=0.5)
##
## est_p_> plot_power_curve(list(result1,result2))
```



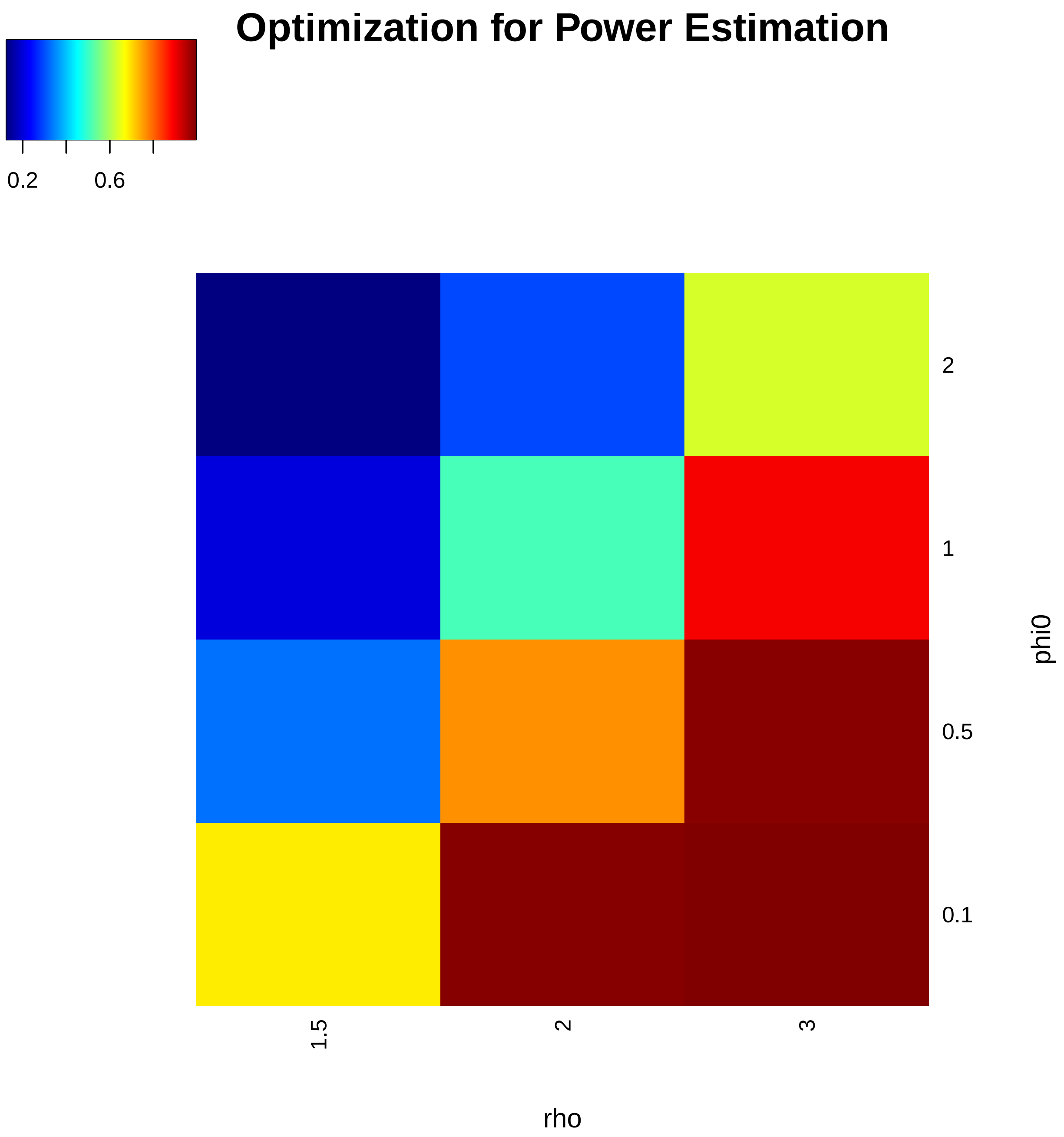
As a result, the relation between power and sample size can be estimated by *est_power_curve* function and the power curves can be generated by *plot_power_curve* function.

3.4 Optimization by power or sample size matrix

For example, if we are going to compare the power of finding significant genes for different false discovery rates, and we have the following parameters:

```
example(optimize_parameter)

##
## optmz_> #Optimization for power estimation
## optmz_> result<-optimize_parameter(opt1=”rho”,opt2=”phi0”,
   opt1Value=c(1.5,2,3),opt2Value=c(0.1,0.5,1,2),n=20)
```



```
##
## optmz_> #Optimization for sample size estimation
## optmz_> ## Not run:
## optmz_> ##D result<-optimize_parameter(f=sample_size,opt1=”lambda0
   ”,opt2=”phi0”,opt1Value=c(1,3,5),opt2Value=c(1.5,2,3),power=0.8)
## optmz_> ## End(Not run)
## optmz_>
## optmz_>
## optmz_>
```

4 More details