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

Shilin Zhao[1]

June 12, 2024

Abstract

In this vignette, we demonstrated the application of *RnaSeqSampleSize* as an sample size estimation tool for RNA-seq data. A user friendly web interface is also provided at https://cqs.app.vumc.org/shiny/RnaSeqSampleSize/ for researchers not familiar with R.

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 real data;
- Visualization of sample size and power by power curves;
- Optimization by power or sample size matrix;

[1]zhaoshilin@gmail.com

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 sample size estimation 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 https://cqs.app.vumc.org/shiny/RnaSeqSampleSize/. Most of the functions in Examples section can be performed in this website.

3 Examples

First we will load the *RnaSeqSampleSize* package.

```
library(RnaSeqSampleSize)
```

```
## Loading required package:  ggplot2
```

```
## Loading required package:  RnaSeqSampleSizeData
```

```
## Loading required package:  edgeR
```

```
## Loading required package:  limma
```

```
## Setting options('download.file.method.GEOquery'='auto')
```

```
## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

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:

- Number of samples in each group: 63;
- Minimal fold change between two groups: 2;
- Minimal average read counts: 5;
- Maximal dispersion: 0.5;
- False discovery rate: 0.01;

As a result, the estimated power is 0.8 by `est_power` function. It means that we have 80% probability to find the significant genes with 63 samples in each group.

```
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: 0.01;

As a result, the estimated sample size is 63 by `sample_size` function. It means that if we want to have 80% probability to find the significant genes, we need 63 samples in each group.

```
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 reference data

3.2.1 Power estimation with datasets in RnaSeqSampleSizeData package

RnaSeqSampleSizeData package contains the read counts and dispersion distribution from some real datasets and can be used as prior data for sample size or power estimation. They can be called with following names:

```
##  [1] "TCGA_BLCA" "TCGA_BRCA" "TCGA_CESC" "TCGA_COAD" "TCGA_HNSC" "TCGA_KIRC"
##  [7] "TCGA_LGG"  "TCGA_LUAD" "TCGA_LUSC" "TCGA_PRAD" "TCGA_READ" "TCGA_THCA"
## [13] "TCGA_UCEC"
```

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:

- Number of samples in each group: 65;
- Minimal fold change between two groups: 2;
- Prior data: TCGA READ data, stored in *RnaSeqSampleSizeData* package, can be used with name TCGA_READ;
- False discovery rate: 0.01;

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;

```
est_power_distribution(n=65,f=0.01,rho=2,
                       distributionObject="TCGA_READ",repNumber=5)

## [1] 0.7924236
```

Please note here the parameter repNumber was very small (5) to make the example code faster. We suggest repNumber should be at least set as 100 in real analysis.

- 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;

```
#Power estimation based on some interested genes.
#We use storeProcess=TRUE to return the details for all selected genes.
selectedGenes<-c("A1BG","A2BP1","A2M","A4GALT","AAAS")
powerDistribution<-est_power_distribution(n=65,f=0.01,rho=2,
                        distributionObject="TCGA_READ",
                        selectedGenes=selectedGenes,
                        storeProcess=TRUE)
str(powerDistribution)

## List of 3
##  $ power     : num [1:5] 0.844 0.069 0.983 0.937 0.996
##  $ count     : Named num [1:5] 43 36 2000 260 1950
##   ..- attr(*, "names")= chr [1:5] "A1BG" "A2BP1" "A2M" "A4GALT" ...
##  $ dispersion: num [1:5] 0.6 2.3 0.4 0.5 0.1

mean(powerDistribution$power)

## [1] 0.765763
```

- 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.

```
powerDistribution<-est_power_distribution(n=65,f=0.01,rho=2,
                        distributionObject="TCGA_READ",pathway="00010",
                        minAveCount=1,storeProcess=TRUE)
mean(powerDistribution$power)
```

As a result, we use `est_power_distribution` function and find the estimated power is 0.91 for random genes, 0.81 for specified gene list, and 0.77 for genes in Glycolysis and Gluconeogenesis (pathway 00010) pathway.

3.2.2 Sample size estimation with datasets in RnaSeqSampleSizeData package

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;
- Prior data: TCGA READ data, stored in *RnaSeqSampleSizeData* package, can be used with name TCGA_READ;
- False discovery rate (FDR): 0.01;

As a result, we use *sample_size_distribution* function and find the estimated sample size is 41 for random genes.

```
sample_size_distribution(power=0.8,f=0.01,distributionObject="TCGA_READ",
                         repNumber=5,showMessage=TRUE)
```

```
## x= 1  f(x)= -0.8

## x= 33  f(x)= -0.0269628438759061

## x= 65  f(x)= 0.0328681464357486

## x= 49  f(x)= 0.0121007435579614

## x= 41  f(x)= -0.00146528587900185

## x= 45  f(x)= 0.00607217407312277

## x= 43  f(x)= 0.00255169381422016

## x= 42  f(x)= 0.000641660485024786

## [1] 42
```

Please note here the parameter repNumber was very small (5) to make the example code faster. We suggest repNumber should be at least set as 100 in real analysis.

3.2.3 Sample size or power estimation with user's prior dataset

For example, if the user has a RNA-seq data with 10000 genes and 10 samples as prior dataset:

```
# Generate a 10000*10 RNA-seq data as prior dataset
set.seed(123)
dataMatrix <- matrix(sample(0:3000, 1e+05, replace = TRUE), nrow = 10000, ncol = 10)
colnames(dataMatrix) <- c(paste0("Control", 1:5), paste0("Treatment", 1:5))
row.names(dataMatrix) <- paste0("gene", 1:10000)
head(dataMatrix)
```

```
##       Control1 Control2 Control3 Control4 Control5 Treatment1 Treatment2
## gene1     2462       56      421      219     2344       1636       1486
## gene2     2510     1353     1606      591      251        538       2230
```

```
## gene3     2226     2680      269     2668      170       2976       625
## gene4      525      461     1948     1589     1315       1580      1915
## gene5      194     2048      259     1761     2564       1862      2498
## gene6     2985     2678     2364     2111      704       2629      1852
##       Treatment3 Treatment4 Treatment5
## gene1       1961       1924       1412
## gene2        382       2764       1491
## gene3       1685       2100       1248
## gene4       1280        228        441
## gene5       2947       1093       1761
## gene6        353        783       1097
```

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

- Number of samples in each group: 65;
- Minimal fold change between two groups: 2;
- Prior data: User's prior dataset with 10000 genes and 10 samples;
- False discovery rate: 0.01;

We will use `est_count_dispersion` to estitamete the gene read count and dispersion distribution of user's prior dataset. And then `est_power_distribution` function will be used to estimate power.

```
#Estitamete the gene read count and dispersion distribution
dataMatrixDistribution<-est_count_dispersion(dataMatrix,
                        group=c(rep(0,5),rep(1,5)))

## Disp = 0.603 , BCV = 0.7765

#Power estimation by read count and dispersion distribution
est_power_distribution(n=65,f=0.01,rho=2,
                        distributionObject=dataMatrixDistribution,repNumber=5)

## [1] 0.7925597
```

As a result, we can find the estimated power is 0.91. Please note here the parameter repNumber was very small (5) to make the example code faster. We suggest repNumber should be at least set as 100 in real analysis.

3.3 Analyze public Rna-Seq data to find best parameters

If we are not sure about the parameters (such as library size, number of differential genes, or fold change cutoff), we can try to find these numbers from public Rna-Seq data with similar samples and design.

3.3.1 Library size and number of expression genes

For example, if we are planning a project with K562 cells and shRNA trasfection, we can download a data set with similar design (SRP009615) from recount database.

```
library(recount)
studyId="SRP009615"
url <- download_study(studyId)
load(file.path(studyId, 'rse_gene.Rdata'))

#show percent of mapped reads
plot_mappedReads_percent(rse_gene)
#show propotion of gene counts in different range
plot_gene_counts_range(rse_gene,targetSize = 4e+07)
```

As a result, we know about 90

3.3.2 Number of differential genes and fold changes between two groups

For example, if we are planning a project to compare lung tissue between COVID-19 patients and healthy controls, we can download a data set with similar design (E-ENAD-46) from Expression Atlas database.

```
library(ExpressionAtlas)



projectId="E-ENAD-46"
allExps <- getAtlasData(projectId)
ExpressionAtlasObj <- allExps[[ projectId ]]$rnaseq

#only keeping "g2" (COVID-19) and "g4" (normal) samples
expObj=ExpressionAtlasObj[,which(colData(ExpressionAtlasObj)$AtlasAssayGroup %in% c("g2","g
expObjGroups= 2-as.integer(as.factor(colData(expObj)$AtlasAssayGroup)) #0 for normal and 1
```

```
#only keeping genes with at least 10 counts
minAveCount=10
averageCountsGene=rowSums(assay(expObj))/ncol(expObj)
expObjFilter=expObj[which(averageCountsGene>=minAveCount),]

result=analyze_dataset(expObjFilter,expObjGroups=expObjGroups)
```

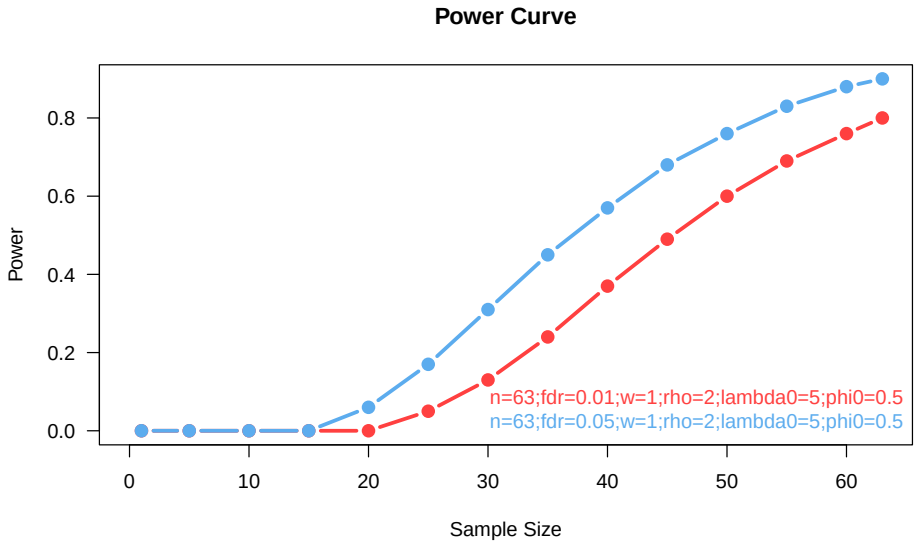
As a result, we used healthy samples as negative control, and find that there are few differential genes if testing between healthy samples and the fold change is also small. We also compared the COVID-19 samples and healthy samples in this dataset and identified 4k+ differential genes. The third figure indicated the similar dispersion pattern when using control samples only or disease and control samples. It means we can use healthy samples only to estimate dispersion distribution, if we don't have COVID-19 samples.

3.4 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:

- Number of samples in each group: 63;
- Minimal fold change between two groups: 2;
- Minimal average read counts: 5;
- Maximal dispersion: 0.5;
- False discovery rate: 0.01 and 0.05;

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

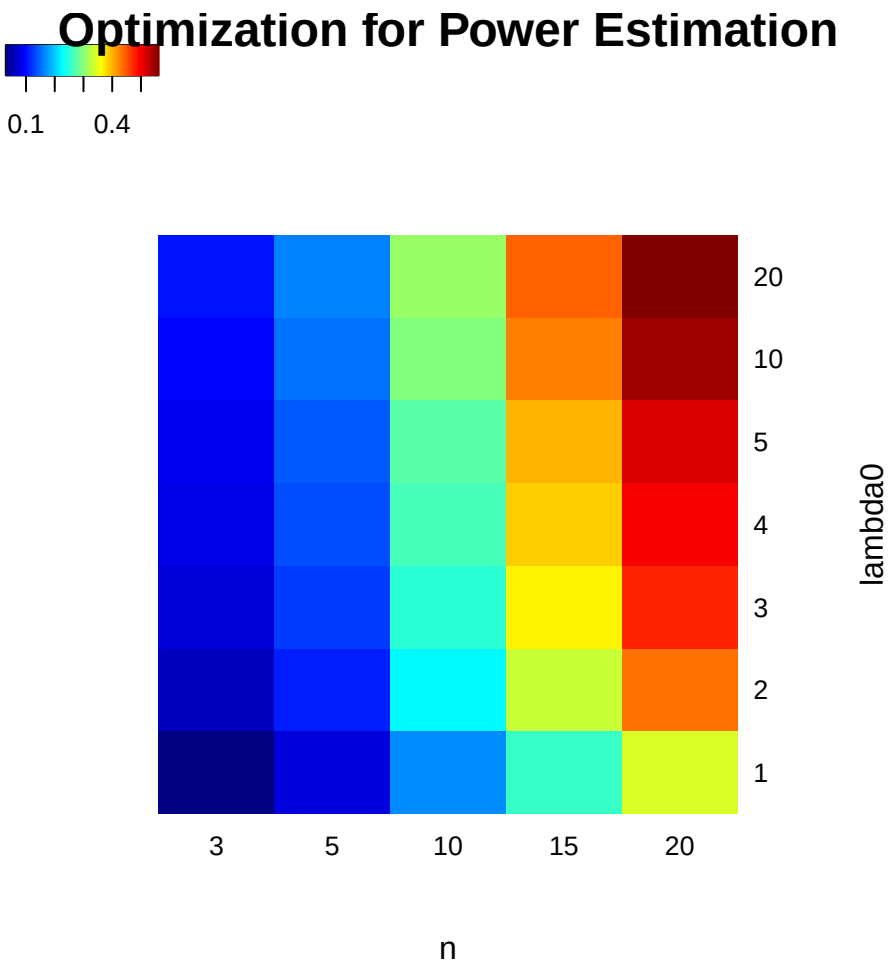


As a result, the relation between power and sample size can be estimated by `est_power_curve` function and the power curves can be generated by `plot_power_curve` function.

3.5 Optimization by power or sample size matrix

For example, if the budget is limited, we need to balance the number of replications and sequence depth. We can use the `optimize_parameter` function to find the relation between sample size, read counts, and estimated power. And then the optimized parameters can be determined.

```
result<-optimize_parameter(fun=est_power,opt1="n",
                           opt2="lambda0",opt1Value=c(3,5,10,15,20),
                           opt2Value=c(1:5,10,20))
```



As a result, the estimated power distribution indicates that the number of replications plays a more significant role in determining the power than the number of read counts.