Sample Size Calculation for RNA-Seq Experimental Design – the *ssizeRNA* package

Ran Bi, Peng Liu

Department of Statistics, Iowa State Univeresity

January 5, 2017

Abstract

Sample size calculation is a crucial issue when designing an RNA-seq experiment. This vignette explains the use of the package *ssizeRNA*, which is designed to provide an estimation of sample size while controlling false discovery rate (FDR) for RNA-seq experimental design.

ssizeRNA **version:** 1.2.9

Contents

1 Introduction

RNA-seq technologies have been popularly applied in transcriptomic studies. In the statistical analysis of RNA-seq data, identifying differentially expressed (DE) genes across treatments or conditions is a major step or main focus. Many statistical methods have been proposed for the detection of DE genes with RNA-seq data, such as *edgeR* Robinson et al. (2010), *DESeq* Anders and Huber (2010), *DESeq2* Love et al. (2014) and *QuasiSeq* Lund et al. (2012).

Due to the genetic complexity, RNA-seq experiments are rather costly. Many experiments only employ a small number of replicates, which may lead to unreliable statistical inference. Thus, one of the principal questions in designing an RNA-seq experiment is: how large of the sample size do we need?

Many of the current sample size calculation methods are simulation based, which are quite time-consuming. We propose a much less computationally intensive method and R package *ssizeRNA* for sample size calculation in designing RNA-seq experiments Bi and Liu (2016).

2 Using `ssizeRNA`

We first load the *ssizeRNA* package.

```
library(ssizeRNA)
```

To determine the sample size for an RNA-seq experiment, users need to specify the following parameters:

- G: total number of genes for testing;

- $pi0$: proportion of non-DE genes;

- fdr: FDR level to control;

- *power*: desired average power to achieve;

- *mu*: average read count for each gene in control group (without loss of generality, we assume that the normalization factors are equal to 1 for all samples);

- *disp*: dispersion parameter for each gene;

- *fc*: fold change for each gene.

We will give several examples of using *ssizeRNA* sample size estimation as follows.

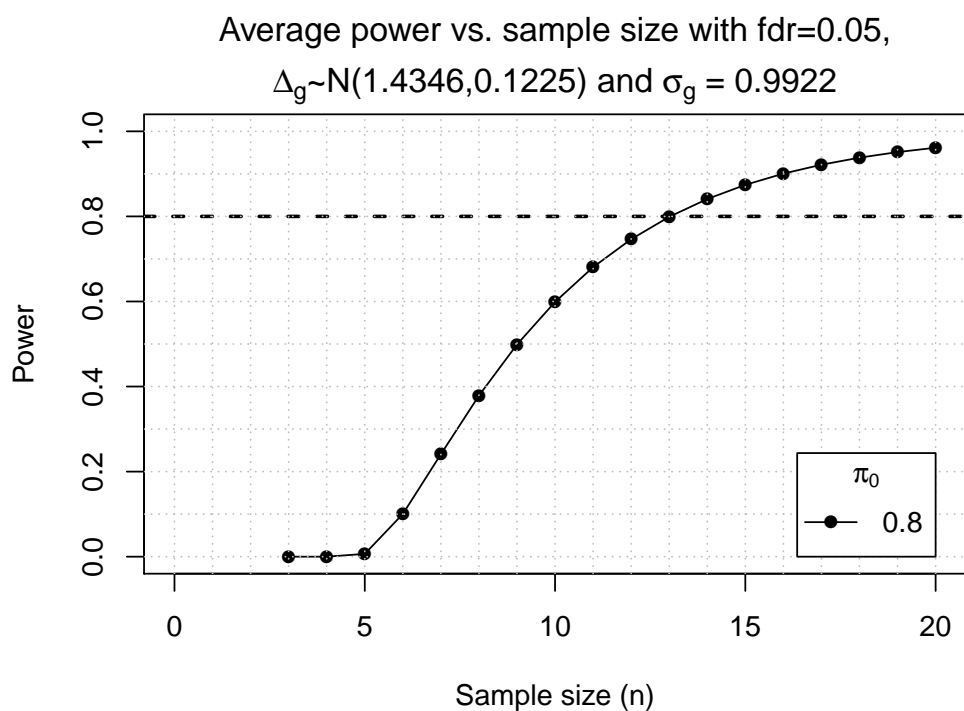## 2.1 Sample size calculation for a single set of parameter

Here we consider the situation of single set of parameter, i.e. all genes share the same set of average read count in control group, dispersion parameter, and fold change.

For example, if we are estimating the sample size for an RNA-seq experiment with

- Total number of genes: $G = 10000$;

- Proportion of non-DE genes: $pi0 = 0.8$;

- FDR level to control: $fdr = 0.05$;

- Desired average power to achieve: $power = 0.8$;

- Average read count for each gene in control group: $mu = 10$;

- Dispersion parameter for each gene: $disp = 0.1$;

- Fold change for each gene: $fc = 2$.

4

The estimated sample size is 14 with anticipated power 0.84 by *ssiz-eRNA_single* function. The function also gives the power vs. sample size curve estimated by our method.

```
set.seed(2016)
size1 <- ssizeRNA_single(nGenes = 10000, pi0 = 0.8, m = 200, mu = 10,
                         disp = 0.1, fc = 2, fdr = 0.05,
                         power = 0.8, maxN = 20)
```


Average power vs. sample size with fdr=0.05, Δ_g~N(1.4346,0.1225) and σ_g = 0.9922

```
size1$ssize

##      pi0 ssize    power
## [1,] 0.8    14 0.840891
```

To check whether desired power would be achieved at the calculated sample size 14 for voom and limma pipeline Law et al. (2014); Smyth (2004), we could use the *check.power* function, which gives the observed power and true FDR by Benjamini and Hochberg's method Benjamini and Hochberg (1995) and Storey's q-value procedure Storey et al. (2004) respectively. The results below are based on 10 simulations, indicating that desired power is achieved and FDR is controlled successfully.

```
check.power(m = 14, mu = 10, disp = 0.1, fc = 2, sims = 10)

## $pow_bh_ave
## [1] 0.86235
##
## $fdr_bh_ave
## [1] 0.03496079
##
## $pow_qvalue_ave
## [1] 0.88335
##
## $fdr_qvalue_ave
## [1] 0.04628789
```

2.2   Sample size calculation for gene-specific mean and dispersion with fixed fold change

Now we will give an example of sample size calculation for gene-specific mean and dispersion. Here we use the real RNA-seq dataset from Hammer, P. et al., 2010 Hammer et al. (2010) to generate gene-specific mean and dispersion parameters.

```
data(hammer.eset)
counts <- exprs(hammer.eset)[, phenoData(hammer.eset)$Time == "2 weeks"]
counts <- counts[rowSums(counts) > 0,]   ## filter zero count genes
trt <- hammer.eset$protocol[which(hammer.eset$Time == "2 weeks")]

## average read count in control group for each gene
mu <- apply(counts[, trt == "control"], 1, mean)

## dispersion for each gene
d <- DGEList(counts)
d <- calcNormFactors(d)
d <- estimateCommonDisp(d)
d <- estimateTagwiseDisp(d)
disp <- d$tagwise.dispersion
```
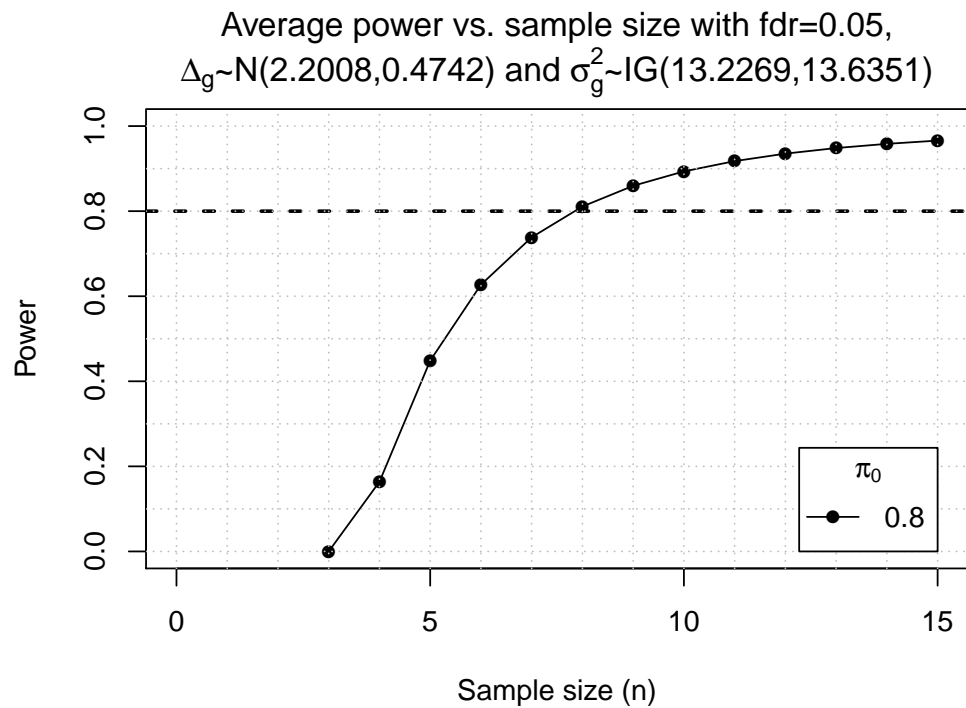
If we would like to estimate the sample size for the above RNA-seq experiment with

- Total number of genes: $G = 10000$;

- Proportion of non-DE genes: $pi0 = 0.8$;

- FDR level to control: $fdr = 0.05$;

- Desired average power to achieve: $power = 0.8$;

- Fold change for each gene: $fc = 2$.

The estimated sample size is 8 with anticipated power 0.81 by *ssiz-eRNA_vary* function. The function also gives the power vs. sample size curve estimated by our method.

```
set.seed(2016)
size2 <- ssizeRNA_vary(nGenes = 10000, pi0 = 0.8, m = 200, mu = mu,
                       disp = disp, fc = 2, fdr = 0.05,
                       power = 0.8, maxN = 15, replace = FALSE)
```

Average power vs. sample size with fdr=0.05, $\Delta_g \sim N(2.2008, 0.4742)$ and $\sigma_g^2 \sim IG(13.2269, 13.6351)$


```
size2$ssize
```

```
##      pi0 ssize     power
## [1,] 0.8     8 0.8108615
```

The observed power and true FDR by Benjamini and Hochberg's method and Storey's q-value procedure could also be checked by the *check.power* function.
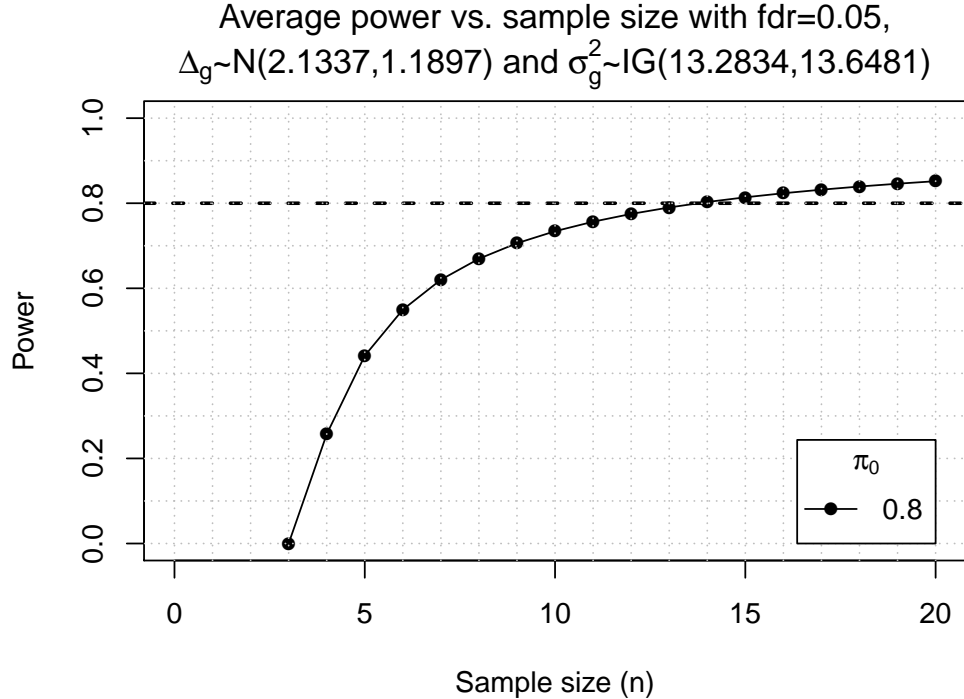
8

2.3 Sample size calculation for gene-specific mean and dispersion with different fold change

If not all genes share the same fold change, for example, if fold change comes from a log-normal distribution,

$$fc \sim log - Normal(log(2), 0.5 * log(2))$$

other parameters remain the same as in subsection 2.2, then the estimated sample size is 14 with anticipated power 0.80 by *ssizeRNA_vary* function.

```
set.seed(2016)
fc <- function(x){exp(rnorm(x, log(2), 0.5*log(2)))}
size3 <- ssizeRNA_vary(nGenes = 10000, pi0 = 0.8, m = 200, mu = mu,
                       disp = disp, fc = fc, fdr = 0.05,
                       power = 0.8, maxN = 20, replace = FALSE)
```

Average power vs. sample size with fdr=0.05,
Δ_g~N(2.1337,1.1897) and σ_g^2~IG(13.2834,13.6481)

```
size3$ssize
```

```
##      pi0 ssize     power
## [1,] 0.8    14 0.8029062
```

By the following command, we verified that the desired power 0.8 is achieved at the calculated sample size 14 for voom and limma pipeline.

```
check.power(m = 14, mu = mu, disp = disp, fc = fc, sims = 10,
            replace = FALSE)
```

```
## $pow_bh_ave
## [1] 0.8253
##
## $fdr_bh_ave
## [1] 0.036631
##
## $pow_qvalue_ave
## [1] 0.8329
##
## $fdr_qvalue_ave
## [1] 0.04546319
```

3   Conclusion

ssizeRNA provides a quick calculation for sample size, and an accurate estimate of power. Examples in section 2 demonstrate that our proposed method offers a reliable approach for sample size calculation for RNA-seq experiments.

4   Session Info

```
sessionInfo()

## R version 3.3.2 (2016-10-31)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X El Capitan 10.11.6
##
## locale:
## [1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] ssizeRNA_1.2.9      edgeR_3.16.5        limma_3.30.7
## [4] Biobase_2.34.0      BiocGenerics_0.20.0 knitr_1.15.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.8      magrittr_1.5     splines_3.3.2    MASS_7.3-45
##  [5] munsell_0.4.3    colorspace_1.3-2 lattice_0.20-34  highr_0.6
##  [9] stringr_1.1.0    plyr_1.8.4       tools_3.3.2      grid_3.3.2
## [13] gtable_0.2.0     lazyeval_0.2.0   assertthat_0.1   tibble_1.2
## [17] ssize.fdr_1.2    reshape2_1.4.2   ggplot2_2.2.1    qvalue_2.6.0
## [21] evaluate_0.10    stringi_1.1.2    scales_0.4.1     locfit_1.5-9.1
```

References

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome biol*, 11(10):R106.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate:

a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300.

Bi, R. and Liu, P. (2016). Sample size calculation while controlling false discovery rate for differential expression analysis with rna-sequencing experiments. *BMC bioinformatics*, 17:146.

Hammer, P., Banck, M. S., Amberg, R., Wang, C., Petznick, G., Luo, S., Khrebtukova, I., Schroth, G. P., Beyerlein, P., and Beutler, A. S. (2010). mrna-seq with agnostic splice site discovery for nervous system transcriptomics tested in chronic pain. *Genome research*, 20(6):847–860.

Law, C. W., Chen, Y., Shi, W., and Smyth, G. K. (2014). Voom: precision weights unlock linear model analysis tools for rna-seq read counts. *Genome Biol*, 15(2):R29.

Love, M., Anders, S., and Huber, W. (2014). Differential analysis of count data–the deseq2 package. *Genome Biology*, 15:550.

Lund, S., Nettleton, D., McCarthy, D., Smyth, G., et al. (2012). Detecting differential expression in rna-sequence data using quasi-likelihood with shrunken dispersion estimates. *Statistical applications in genetics and molecular biology*, 11(5):8.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140.

Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical applications in genetics and molecular biology*, 3(1).

Storey, J. D., Taylor, J. E., and Siegmund, D. (2004). Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(1):187–205.