# powsimR: Power Analysis and Sample Size Estimation for Bulk and Single Cell RNA-Seq Experiments

Beate Vieth *

October 27, 2017

Report issues on https://github.com/bvieth/powsimR/issues

*vieth@bio.lmu.de

# Contents

# 1 Installation Guide

PowsimR has a number of dependencies that need to be installed before hand (see also the README file on github). I recommend to increase the maximum number of DLLs that can be loaded to at least 500. The environmental variable R_MAX_NUM_DLLS can be set in R_HOME/etc/Renviron prior to starting R (for further details see `Startup`). For that locate the Renviron file and add the following line: R_MAX_NUM_DLLS=xy where xy is the number of DLLs. On my Ubuntu machine, the Renviron file is in /usr/lib/R/etc/ and I can set it to 500. Apparently this is not possible on MACs where the maximum that worked was 153. This is enough for powsimR, however.

```
ipak <- function(pkg, repository = c("CRAN", "Bioconductor",
    "github")) {
    new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
    if (length(new.pkg)) {
        if (repository == "CRAN") {
            install.packages(new.pkg, dependencies = TRUE)
        }
        if (repository == "Bioconductor") {
            source("https://bioconductor.org/biocLite.R")
            biocLite(new.pkg, dependencies = TRUE, ask = FALSE)
        }
        if (repository == "github") {
            devtools::install_github(new.pkg, build_vignettes = FALSE,
                dependencies = TRUE)
        }
    }
}

# CRAN PACKAGES
cranpackages <- c("methods", "stats", "matrixStats", "Rtsne",
    "moments", "minpack.lm", "glmnet", "cluster", "mclust", "MASS",
    "gtools", "doParallel", "parallel", "snow", "reshape2", "plyr",
    "dplyr", "tidyr", "tibble", "data.table", "ggplot2", "grid",
    "ggthemes", "ggExtra", "cowplot", "scales", "cobs", "msir",
    "drc", "DrImpute", "VGAM", "NBPSeq")
ipak(cranpackages, repository = "CRAN")

# BIOCONDUCTOR
biocpackages <- c("S4Vectors", "DEDS", "AnnotationDbi", "Biobase",
    "BiocGenerics", "SummarizedExperiment", "BiocParallel", "RUVSeq",
    "scran", "scater", "Linnorm", "edgeR", "limma", "DESeq2",
    "baySeq", "NOISeq", "EBSeq", "MAST", "scde", "scDD", "ROTS",
    "monocle", "IHW", "qvalue")
ipak(biocpackages, repository = "Bioconductor")
```

```
# GITHUB
githubpackages <- c("nghiavtr/BPSC", "rhondabacher/SCnorm", "catavallejos/BASiCS")
ipak(githubpackages, repository = "github")
```

After installing the dependencies, *powsimR* can be installed by using devtools as well.

```
devtools::install_github("bvieth/powsimR", build_vignettes = TRUE,
    dependencies = FALSE)

library(powsimR)
```

Some users have experienced issues installing powsimR due to Tex compilation errors. If that is the case, you can leave out building the vignette.

# 2 Introduction

In this vignette, we illustrate the features of *powsimR* by assessing the power to detect differential expression between two groups of embryonic stem cells cultured in standard 2i medium (E-MTAB-2600) [1].

# 3 powsimR Workflow

The basic workflow of *powsimR* is illustrated in figure 1: A) The mean-dispersion relationship is estimated from RNA-seq data, which can be either single cell or bulk data. The users can provide their own count tables, download from a database (see below) or one of our example data sets (powsimRData). B) These distribution parameters are then used to set up the simulations. For better comparability, the parameters for the simulation of differential expression are set separately. C) Finally, the TPR and FDR are calculated. Both can be either returned as marginal estimates per sample configuration, or stratified according to the estimates of mean expression, dispersion or dropout rate.

# 4 Parameter Estimation

## 4.1 Gene Expression

The parameters of the (zero-inflated) negative binomial distribution, i.e. mean and dispersion are estimated by the function `estimateParam`. In addition, the dropout probability, i.e. the fraction of zero counts per gene, is calculated. The user can choose between two estimation frameworks:

**NB** Negative binomial distribution.
**ZINB** Zero-inflated negative binomial distribution.

In both cases matching moments estimation of mean and dispersion are based on normalized counts.

The user can choose between multiple normalisation methods (see details section of `estimateParam`). Furthermore, a number of methods are group sensitive, e.g. batch labels, namely scran SCnorm, Census, BASiCS, RUV.

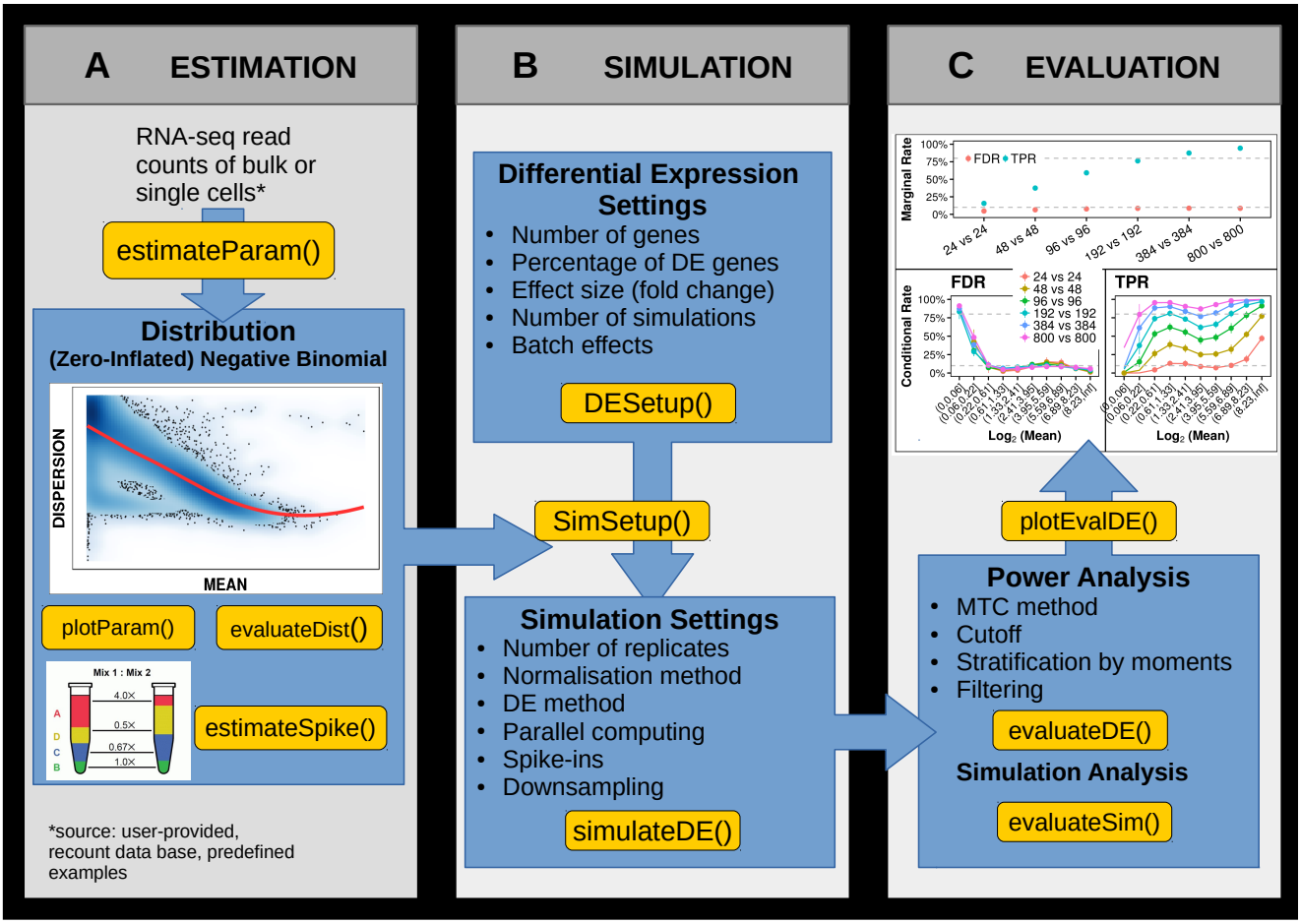


Figure 1: **PowsimR schematic overview.** (A) Estimation: (B) Simulation: (C) Evaluation. Functions are given in orange.

The estimates, sequencing depth and normalisation factors are plotted with `plotParam`.

With the following command, we estimate and plot the parameters for the embryonic stem cells cultured in standard 2i+lif medium (Kolodziejczyk) (figure 2). As expected for single cell RNA-seq, the variability (i.e. dispersion) and dropout rates are high. Furthermore, the dispersion strongly depends on the mean and does not level off with higher mean values.

```
# download count table
githubURL <- "https://github.com/bvieth/powsimRData/raw/master/data-raw/kolodziejczk_cnts.rda"
download.file(url = githubURL, destfile = "kolodziejczk_cnts.rda",
    method = "wget")
load("kolodziejczk_cnts.rda")
kolodziejczk_cnts <- kolodziejczk_cnts[, grep("standard", colnames(kolodziejczk_cnts))]
colnames(kolodziejczk_cnts) <- paste0(colnames(kolodziejczk_cnts),
    "_", 1:ncol(kolodziejczk_cnts))
TwoiLIF.params <- estimateParam(countData = kolodziejczk_cnts,
    batchData = NULL, spikeData = NULL, spikeInfo = NULL, Lengths = NULL,
    MeanFragLengths = NULL, Distribution = "ZINB", RNAseq = "singlecell",
```

```
    normalisation = "scran", sigma = 1.96, NCores = NULL)
plotParam(TwoiLIF.params)
```

We have implemented a read count simulation framework assuming either a negative binomial distribution or a zero-inflated negative binomial distribution. To predict the dispersion given a random draw of mean expression value observed, we apply a locally weighted polynomial regression fit. To capture the variability of dispersion estimates observed, a local variability prediction band is applied. For bulk RNA-seq experiments, dropouts are less probable but can still occur. To include this phenomenon we sample from the observed dropout rates for genes that have a mean expression value below 5% dropout probability determined by a decrease constrained B-splines regresssion of dropout rate against mean expression (`cobs` in *cobs*). The resulting read count matrix has similar distributional characteristics as the original Kolodziejczyk data set (figure 3). For the zero-inflated negative binomial distribution, the mean-dispersion relation is similarly estimated, but based on positive read counts. Furthermore, the dropouts are also predicted based on a locally weighted polynomial regression fit between mean and dropouts. Of note, this fit is done separately for amplified and non-amplified transcripts separately and similar proportions of genes as observed are also generated in the simulations [2]. We have found that the negative binomial distribution is particularly suited for UMI-methods (e.g. SCRB-Seq, Drop-Seq, 10XGenomics) [3].

### 4.2 Spike-ins

Some normalisation methods can use spike-ins as part of their normalisation (SCnorm, RUV, Census). To use spike-in information in the simulations, their distributional characteristics need to be estimated. We follow the estimation and simulation framework presented in [4] where the variance is decomposed into shot noise and mRNA loss due to capture and sequencing efficiency. In short, the parameters for a Beta-distribution describes the RNA molecule capture efficiency and the parameters of a Gamma distribution describes the sequencing efficiency, which we can then use to simulate in silico spike-ins given a mean expression value. We assume that biological variance does not contribute to spike-in expression. The user needs to define the spike-in expression table and the spike-in information table (IDs, molecules, lengths per spike-in) in the function `estimateSpike`. The following formula can be used for calculating the number of molecules of spike-ins:

$$Y_j = c_j * V * 10^{-3} * D^{-1} * 10^{-18} * Avogadro, \quad j = 1, \ldots, 92$$

The number of molecules $Y_j$ for each ERCC spike-in species is the product of the molar concentration $c_j$ (attomoles per microlitre), the dilution factor $1/D$, the volume $V$ (nanolitre), Avogadros' constant ($6.02214129 * 10^{23}$) and conversion factors between unit scales.

## 5 Simulations

For simulating differential expression between two groups, the number of genes, number of simulations, percentage of differential expression and effect size are set up with the function `DESetup`. The effect size is here defined as the log2 fold change which can be a constant, sampled from a vector or function. The uniform, normal and gamma distributions are possible options and illustrated in figure 4. Depending on the settings, these distribution can be broader or narrower.

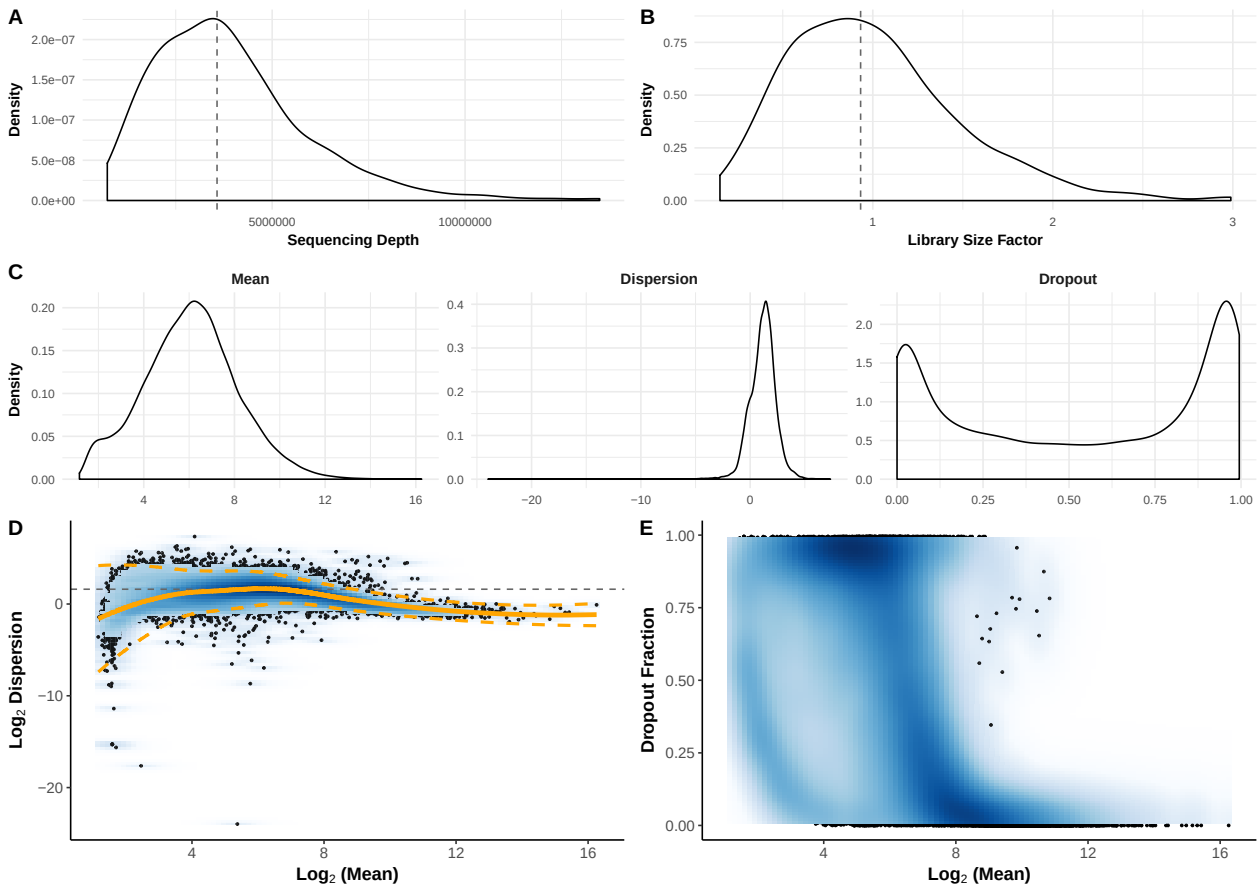


Figure 2: **Estimated parameters for Kolodziejczyk data set.** A) Sequencing depth per sample with median sequencing depth (grey dashed line). B) Library size normalisation factor per sample with median size factor (grey dashed line). C) Marginal Distribution of log2(mean), log2(dispersion) and dropout. D) Local polynomial regression fit between log2(mean) and log2(dispersion) estimates with variability band (yellow) per gene. Common dispersion estimate (grey dashed line). E) Fraction of dropouts versus estimated mean expression per gene.

If using this option, we recommend to choose a distribution that closely resembles previously observed or expected fold changes.

The distribution estimates and these settings are then combined to one object with `SimSetup`. This allows the user to assess power of multiple groupwise comparisons and different differential testing methods. The following command sets up simulations with 10,000 genes, 20% genes being DE, log fold change sample from a narrow gamma distribution and parameter estimates based on Kolodziejczyk data:

```
lfc.gamma = function(x) sample(c(-1, 1), size = x, replace = T) *
    rgamma(x, 3, 3)
de.opts = DESetup(ngenes = 10000, nsims = 25, p.DE = 0.2, pLFC = lfc.gamma,
    p.B = NULL, bLFC = NULL, sim.seed = 58673)
sim.opts = SimSetup(desetup = de.opts, params = TwoiLIF.params,
    size.factors = "given", downsample = FALSE, geneset = FALSE)
```

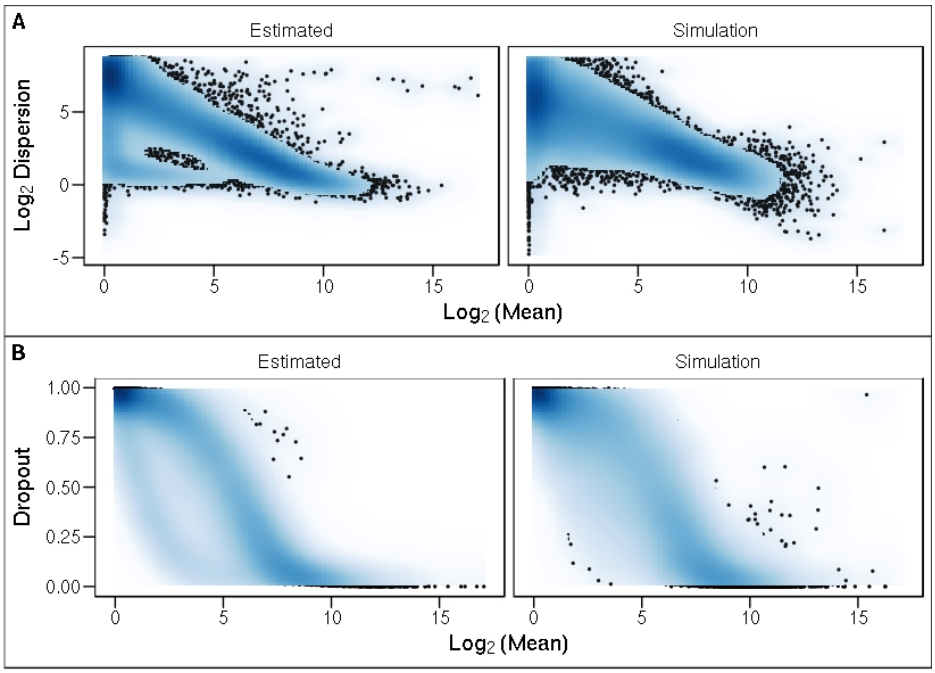


Figure 3: **Comparison of estimated and simulated read counts** (A) Dispersion versus Mean. (B) Dropout versus Mean.

With the setup defined, the differential expression simulation is run with `simulateDE`. For this, the user needs to set the following options:

**Replicates** The number of sample replicates per group (n1 and n2). These can be unbalanced.

**DEmethod** The differential testing method. The user can choose between 11 methods in total. 7 developed for bulk, 4 developed for single cells (see the detail section of `simulateDE`).

**normalisation** The normalisation method. The user can choose between 11 methods in total. 6 developed for bulk, 5 developed for single cells (see the detail section of `estimateParam`).

```
simDE = simulateDE(n1 = c(24, 48, 96, 192, 384, 800), n2 = c(24,
    48, 96, 192, 384, 800), sim.settings = sim.opts, DEmethod = "MAST",
    normalisation = "scran", Preclust = FALSE, Preprocess = NULL,
    spikeIns = FALSE, NCores = 10, verbose = TRUE)
```

There also additional options: Whether to apply a preprocessing step prior to normalisation; whether spike-in information should be used (if available).

## 6 Evaluation

The results of differential expression simulation are evaluated with `evaluateDE`. We have separated the evaluation from DE detection to allow the user to evaluate power in a comprehensive way as advocated by [5]. In this function, the proportions and error rates are estimated. The rates can be stratified by mean, dispersion dropout or log fold change. Furthermore, the user can choose between different multiple testing correction methods (see `p.adjust.methods`, `ihw`

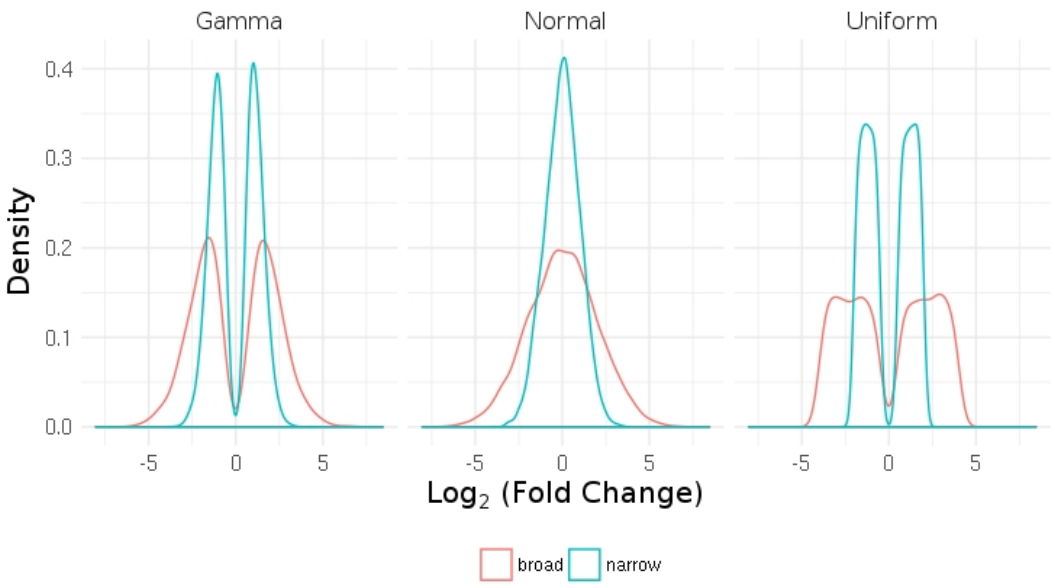


Figure 4: Log2 fold change examples for gamma, uniform and normal distribution

in *IHW* and qvalue in *qvalue*). Also, the genes can be filtered by mean, dispersion or dropout. To define biologically interesting genes, a cutoff for the log2 fold change with delta can be set.

With the following command we evaluate the marginal TPR and FDR conditional on the mean expression for the simulation based on Kolodziejczyk data.

```
evalDE = evaluateDE(simRes = simDE, alpha.type = "adjusted",
    MTC = "BH", alpha.nominal = 0.1, stratify.by = "mean", filter.by = "none",
    strata.filtered = 1, target.by = "lfc", delta = 0)
```

The results of the evaluation can be plotted with `plotEvalDE`.

**rate** Marginal or conditional error rates calculations. The conditional error rates are determined and calculated with `evaluateDE`. The number of genes per stratum are also summarised.

**quick** If this is set to *R*TRUE then only the TPR and FDR will be plotted.

With the following commands, the quick marginal and conditional power assessments for the Kolodziejczyk data are plotted.

```
plotEvalDE(evalRes = evalDE, rate = "marginal", quick = TRUE,
    annot = TRUE)

plotEvalDE(evalRes = evalDE, rate = "stratified", quick = TRUE,
    annot = TRUE)
```

# 7 Additional Functionalities

## 7.1 Evaluate Simulation Framework

It is important to validate the appropiateness of the chosen simulation framework. The function `evaluateDist` compares the theoretical fit of the Poisson, negative binomial, zero-inflated Poisson and zero-inflated negative binomial and beta-Poisson distribution to the empirical RNA-seq read counts ([6], [7], [8]). The evaluation is then plotted with the function `plotEvalDist` which summarizes the best fitting distribution per gene based on goodness-of-fit statistics (Chi-square test), Akaike Information Criterium, comparing observed dropouts with zero count prediction of the models and comparing the model fitness with Likelihood Ratio Test and Vuong Test. Bulk RNA-seq experiments are usually conducted with a small number of samples. We therefore recommend to rely on the goodness-of-fit validation by [9]. To use this approach in `evaluateDist`, the user should allow for permutation simulations by setting the value of nsims to at least 100. If available, the computation can be run on multiple cores by setting the number of cores (ncores).

With the following command, we estimate and plot the parameters for the embryonic stem cells cultured in standard 2i lif medium (Kolodziejczyk).

```
TwoiLIF.distfit = evaluateDist(cnts = kolodziejczk_cnts, RNAseq = "singlecell",
    ncores = 1, nsims = 1, frac.genes = 1, min.meancount = 1,
    min.libsize = 1000)

plotEvalDist(evalDist = TwoiLIF.distfit, annot = F)
```

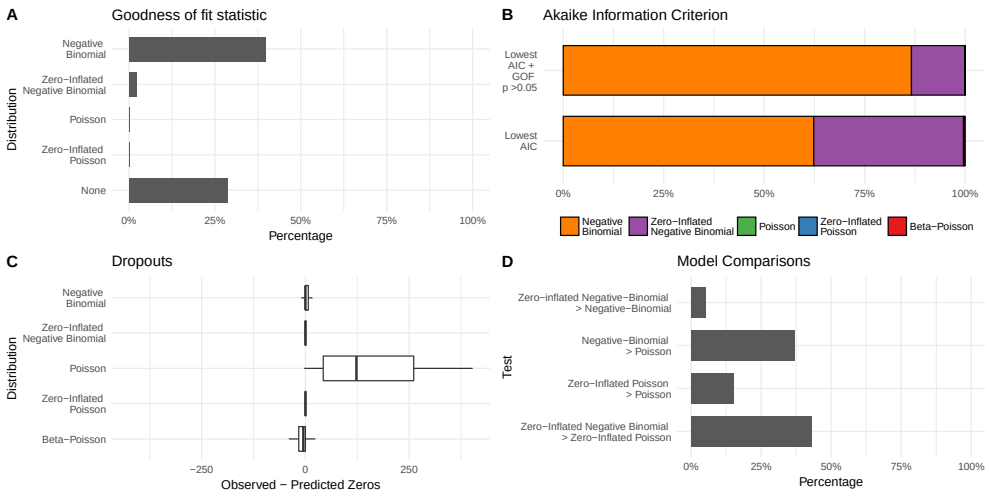


Figure 5: A) Goodness of fit of the model per gene assessed with a Chi-square test based on residual deviance and degrees of freedom. B) The fraction of genes for which the respective distribution has the lowest AIC and additionally the distribution with the lowest AIC as well as not rejected by the goodness of fit statistic. C) Observed versus predicted dropouts per distributional model and gene. D) Model assessment per gene based on Likelihood Ratio Test for nested models and Vung Test for non-nested models.

## 7.2 Negative Binomial Parameters

### 7.2.1 in silico Parameter Definition

We have also implemented the option to approximate the read count matrix simulation based on random distribution functions in *R*. The user then has to define the mean, dispersion, dropout and library size in `insilicoNBParam`. In the absence of suitable pilot studies, a typical single cell RNA-seq experiment could be approximated with:

- mean: `function(x) rgamma(x, 4, 2)` where x is the number of genes
- dispersion: `function(x) 2 + 100/x` where x is the mean
- library size: `function(x) 2**rnorm(n=x, mean=0, sd=0.25)` where x is the number of samples

The same functionality can also be used for bulk RNA-seq.

### 7.2.2 Count matrices of single cell RNA-seq experiments

We have uploaded read count matrices of 5 single cell RNA-seq experiments on github. The user can calculate the negative binomial parameters with `estimateParam`, view these estimates with `plotParam` and use it as an input for `SimSetup`.

### 7.2.3 Access to raw read counts stored in online data base

We have provided a number of exemplatory single cell RNA-seq data sets for parameter estimation. Nevertheless, you might not find a data set that matches your own experimental setup. In those cases, we recommend to check online repositories for a suitable data set. Below you can find an example script to get count tables from recount (https://jhubiostatistics.shinyapps.io/recount/) [10]. For a single cell RNA-seq data base, see conquer and its tutorial (http://imlspenticton.uzh.ch:3838/conquer/). As before, the user can then calculate the negative binomial parameters with `estimateParam`, view these estimates with `plotParam` and use it as an input for `SimSetup`.

```
# Install and load the R package
source("http://bioconductor.org/biocLite.R")
biocLite("recount")
library("recount")

# Download the data set
url <- download_study("SRP060416")

# Load the data
load(file.path("SRP060416", "rse_gene.Rdata"))

# count table
cnts <- assay(rse_gene)
# sample annotation
sample.info <- data.frame(colData(rse_gene)@listData, stringsAsFactors = F)
# gene annotation
gene.info <- data.frame(GeneID = rowData(rse_gene)@listData$gene_id,
    GeneLength = rowData(rse_gene)@listData$bp_length, stringsAsFactors = F)
```

### 7.3 Simulation settings

By default, there is no difference in library sizes between the samples. If the user wishes for a more realistic, i.e. more variable distribution of read counts across samples, the library sizes can be sampled from observed, vector or function.

## 8 Session info

Here is the output of `sessionInfo` on the system on which this document was compiled:

- R version 3.4.2 (2017-09-28), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_GB.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=de_DE.UTF-8`, `LC_COLLATE=en_GB.UTF-8`, `LC_MONETARY=de_DE.UTF-8`, `LC_MESSAGES=en_GB.UTF-8`, `LC_PAPER=de_DE.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=de_DE.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 14.04.5 LTS`
- Matrix products: default
- BLAS: `/usr/lib/atlas-base/atlas/libblas.so.3.0`
- LAPACK: `/usr/lib/lapack/liblapack.so.3.0`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: knitr 1.17
- Loaded via a namespace (and not attached): backports 1.1.1, BiocStyle 2.4.1, compiler 3.4.2, digest 0.6.12, evaluate 0.10.1, formatR 1.5, highr 0.6, htmltools 0.3.6, magrittr 1.5, Rcpp 0.12.13, rmarkdown 1.6, rprojroot 1.2, stringi 1.1.5, stringr 1.2.0, tools 3.4.2, yaml 2.1.14

## References

[1] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Jason C H Tsang, Tomislav Ilicic, Johan Henriksson, Kedar N Natarajan, Alex C Tuck, Xuefei Gao, Marc Bühler, Pentao Liu, John C Marioni, and Sarah A Teichmann. Single cell RNA-Sequencing of pluripotent states unlocks modular transcriptional variation. *Cell Stem Cell*, 17(4):471–485, 1 October 2015. doi:10.1016/j.stem.2015.09.011.

[2] Christoph Ziegenhain, Beate Vieth, Swati Parekh, Björn Reinius, Amy Guillaumet-Adkins, Martha Smets, Heinrich Leonhardt, Holger Heyn, Ines Hellmann, and Wolfgang Enard. Comparative analysis of Single-Cell RNA sequencing methods. *Mol. Cell*, 65(4):631–643.e4, 16 February 2017. doi:10.1016/j.molcel.2017.01.023.

[3] Beate Vieth, Christoph Ziegenhain, Swati Parekh, Wolfgang Enard, and Ines Hellmann. powsimr: Power analysis for bulk and single cell RNA-seq experiments. *Bioinformatics*, 11 July 2017. doi:10.1093/bioinformatics/btx435.

[4] Jong Kyoung Kim, Aleksandra A Kolodziejczyk, Tomislav Illicic, Sarah A Teichmann, and John C Marioni. Characterizing noise structure in single-cell RNA-seq distinguishes genuine from technical stochastic allelic expression. *Nat. Commun.*, 6:8687, 22 October 2015. doi:10.1038/ncomms9687.

[5] Hao Wu, Chi Wang, and Zhijin Wu. PROPER: comprehensive power evaluation for differential expression using RNA-seq. *Bioinformatics*, 31(2):233–241, 15 January 2015. doi:10.1093/bioinformatics/btu640.

[6] A Colin Cameron and Pravin K Trivedi. *Regression Analysis of Count Data (Econometric Society Monographs)*. Cambridge University Press, 2 edition edition, 27 May 2013.

[7] Jong Kyoung Kim and John C Marioni. Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. *Genome Biol.*, 14(1):R7, 28 January 2013. doi:10.1186/gb-2013-14-1-r7.

[8] Mihails Delmans and Martin Hemberg. Discrete distributional differential expression (D3E)–a tool for gene expression analysis of single-cell RNA-seq data. *BMC Bioinformatics*, 17:110, 29 February 2016. doi:10.1186/s12859-016-0944-6.

[9] Gu Mi, Yanming Di, and Daniel W Schafer. Goodness-of-fit tests and model diagnostics for negative binomial regression of RNA sequencing data. *PLoS One*, 10(3):e0119254, 18 March 2015. doi:10.1371/journal.pone.0119254.

[10] Leonardo Collado-Torres, Abhinav Nellore, Kai Kammers, Shannon E Ellis, Margaret A Taub, Kasper D Hansen, Andrew E Jaffe, Ben Langmead, and Jeffrey T Leek. Reproducible RNA-seq analysis using recount2. *Nat. Biotechnol.*, 35(4):319–321, 11 April 2017. doi:10.1038/nbt.3838.