

# Package ‘SCRIP’

January 20, 2025

**Type** Package

**Title** An Accurate Simulator for Single-Cell RNA Sequencing Data

**Version** 1.0.0

**Date** 2021-11-15

**Description** We provide a comprehensive scheme that is capable of simulating Single Cell RNA Sequencing data for various parameters of Biological Coefficient of Variation, bursting kinetics, differential expression (DE), cell or sample groups, cell trajectory, batch effect and other experimental designs.

'SCRIP' proposed and compared two frameworks with Gamma-Poisson and Beta-Gamma-Poisson models for simulating Single Cell RNA Sequencing data.

Other reference is available in Zappia et al. (2017) <<https://genomebiology.biomedcentral.com/articles/10.1186/s13059-017-1305-0>>.

**License** GPL-3

**LazyData** TRUE

**Depends** R (>= 4.0)

**Imports** splatter(>= 1.16.1), S4Vectors(>= 0.30.0), SummarizedExperiment(>= 1.22.0), SingleCellExperiment(>= 1.14.1), edgeR(>= 3.34.0), methods, stats, mgcv, knitr, BiocManager, BiocGenerics, Seurat, crayon, fitdistrplus, checkmate (>= 2.0.0)

**URL** <https://github.com/thecailab/SCRIP>

**RoxygenNote** 7.1.1

**VignetteBuilder** knitr

**Encoding** UTF-8

**Language** en-GB

**Suggests** rmarkdown, testthat (>= 3.0.0)

**NeedsCompilation** no

**Author** Fei Qin [aut, cre, cph]

**Maintainer** Fei Qin <[fqin@email.sc.edu](mailto:fqin@email.sc.edu)>

**Repository** CRAN

**Date/Publication** 2021-11-19 07:50:02 UTC

## Contents

acinar.data . . . . .	2
bridge . . . . .	3
bringItemsForward . . . . .	3
getLNormFactors . . . . .	4
getPathOrder . . . . .	4
logistic . . . . .	5
params_acinar . . . . .	5
SCRIPsimBatchCellMeans . . . . .	6
SCRIPsimBatchEffects . . . . .	6
SCRIPsimBCVMeans . . . . .	7
SCRIPsimDropout . . . . .	7
SCRIPsimGeneMeans . . . . .	8
SCRIPsimGroupCellMeans . . . . .	8
SCRIPsimGroupDE . . . . .	9
SCRIPsimLibSizes . . . . .	9
SCRIPsimPathCellMeans . . . . .	10
SCRIPsimPathDE . . . . .	10
SCRIPsimSingleCellMeans . . . . .	11
SCRIPsimTrueCounts . . . . .	11
SCRIPsimu . . . . .	12
simu.VEGs . . . . .	13
simu_cluster . . . . .	14
simu_DE . . . . .	14
<b>Index</b>	<b>16</b>

---

acinar.data	<i>parameter files estimated from acinar.data using splatEstimate</i>
-------------	---

---

### Description

parameter files estimated from acinar.data using splatEstimate

### Usage

acinar.data

### Format

parameters estimated using splatEstimate

---

bridge	<i>Brownian bridge</i>
--------	------------------------

---

**Description**

Calculate a smoothed Brownian bridge between two points. A Brownian bridge is a random walk with fixed end points.

**Usage**

```
bridge(x = 0, y = 0, N = 5, n = 100, sigma.fac = 0.8)
```

**Arguments**

x	starting value.
y	end value.
N	number of steps in random walk.
n	number of points in smoothed bridge.
sigma.fac	multiplier specifying how extreme each step can be.

**Value**

Vector of length n following a path from x to y.

---

bringItemsForward	<i>Bring items forward</i>
-------------------	----------------------------

---

**Description**

Move selected items to the start of a list.

**Usage**

```
bringItemsForward(ll, items)
```

**Arguments**

ll	list to adjust item order.
items	vector of items to bring to the front. Any not in the list will be ignored.

**Value**

list with selected items first

---

getLNormFactors      *Get log-normal factors*

---

**Description**

Randomly generate multiplication factors from a log-normal distribution.

**Usage**

```
getLNormFactors(n.facs, sel.prob, neg.prob, fac.loc, fac.scale)
```

**Arguments**

n.facs	Number of factors to generate.
sel.prob	Probability that a factor will be selected to be different from 1.
neg.prob	Probability that a selected factor is less than one.
fac.loc	Location parameter for the log-normal distribution.
fac.scale	Scale factor for the log-normal distribution.

**Value**

Vector containing generated factors.

---

getPathOrder      *Get path order*

---

**Description**

Identify the correct order to process paths so that preceding paths have already been simulated.

**Usage**

```
getPathOrder(path.from)
```

**Arguments**

path.from	vector giving the path endpoints that each path originates from.
-----------	--

**Value**

Vector giving the order to process paths in.

---

logistic	<i>Logistic function</i>
----------	--------------------------

---

**Description**

Implementation of the logistic function

**Usage**

```
logistic(x, x0, k)
```

**Arguments**

x	value to apply the function to.
x0	midpoint parameter. Gives the centre of the function.
k	shape parameter. Gives the slope of the function.

**Value**

Value of logistic function with given parameters

---

params_acinar	<i>A data frame with 1000 genes and 80 cells</i>
---------------	--

---

**Description**

A data frame with 1000 genes and 80 cells

**Usage**

```
params_acinar
```

**Format**

A data frame with 1000 genes and 80 cells

SCRIPsimBatchCellMeans

*Simulate batch means*

---

**Description**

Simulate a mean for each gene in each cell incorporating batch effect factors.

**Usage**

```
SCRIPsimBatchCellMeans(sim, params)
```

**Arguments**

sim                    SingleCellExperiment to add batch means to.  
params                SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with simulated batch means.

---

SCRIPsimBatchEffects    *Simulate batch effects*

---

**Description**

Simulate batch effects. Batch effect factors for each batch are produced using [getLNormFactors](#) and these are added along with updated means for each batch.

**Usage**

```
SCRIPsimBatchEffects(sim, params)
```

**Arguments**

sim                    SingleCellExperiment to add batch effects to.  
params                SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with simulated batch effects.

---

SCRIPsimBCVMeans	<i>Simulate BCV means</i>
------------------	---------------------------

---

**Description**

Simulate means for each gene in each cell that are adjusted to follow a mean-variance trend using Biological Coefficient of Variation taken from and inverse gamma distribution.

**Usage**

```
SCRIPsimBCVMeans(data, sim, params)
```

**Arguments**

data	data are used to fit the mean-BCV trend for simulation
sim	SingleCellExperiment to add BCV means to.
params	SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with simulated BCV means.

---

SCRIPsimDropout	<i>Simulate dropout</i>
-----------------	-------------------------

---

**Description**

A logistic function is used to form a relationship between the expression level of a gene and the probability of dropout, giving a probability for each gene in each cell. These probabilities are used in a Bernoulli distribution to decide which counts should be dropped.

**Usage**

```
SCRIPsimDropout(sim, params)
```

**Arguments**

sim	SingleCellExperiment to add dropout to.
params	SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with simulated dropout and observed counts.

---

SCRIPsimGeneMeans      *Simulate gene means*

---

**Description**

Simulate gene means from a gamma distribution. Also simulates outlier expression factors. Genes with an outlier factor not equal to 1 are replaced with the median mean expression multiplied by the outlier factor.

**Usage**

```
SCRIPsimGeneMeans(data, sim, params)
```

**Arguments**

data	raw dataset.
sim	SingleCellExperiment to add gene means to.
params	SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with simulated gene means.

---

SCRIPsimGroupCellMeans  
*Simulate Group CellMeans*

---

**Description**

Simulate group cell means

**Usage**

```
SCRIPsimGroupCellMeans(sim, params)
```

**Arguments**

sim	SingleCellExperiment to add cell means to.
params	SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with added cell means.



---

SCRIPsimGroupDE	<i>Simulate group differential expression</i>
-----------------	---

---

**Description**

Simulate differential expression. Differential expression factors for each group are produced using [getLNormFactors](#) and these are added along with updated means for each group. For paths care is taken to make sure they are simulated in the correct order.

**Usage**

```
SCRIPsimGroupDE(sim, params)
```

**Arguments**

sim	SingleCellExperiment to add differential expression to.
params	splatParams object with simulation parameters.

**Value**

SingleCellExperiment with simulated differential expression.

---

SCRIPsimLibSizes	<i>Simulate library sizes</i>
------------------	-------------------------------

---

**Description**

Simulate expected library sizes. Typically a log-normal distribution is used but there is also the option to use a normal distribution. In this case any negative values are set to half the minimum non-zero value.

**Usage**

```
SCRIPsimLibSizes(sim, params, libsize)
```

**Arguments**

sim	SingleCellExperiment to add library size to.
params	SplatParams object with simulation parameters.
libsize	Provide the library size directly instead of using parameters to estimate

**Value**

SingleCellExperiment with simulated library sizes.

SCRIPsimPathCellMeans *sim PathCellMeans*

---

**Description**

simulate cell means for path

**Usage**

```
SCRIPsimPathCellMeans(sim, params)
```

**Arguments**

sim	SingleCellExperiment to add dropout to.
params	SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with cell means for path simulation.

---

SCRIPsimPathDE *Sim PathDE*

---

**Description**

simulate DE factors for path

**Usage**

```
SCRIPsimPathDE(sim, params)
```

**Arguments**

sim	SingleCellExperiment to add dropout to.
params	SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with DE for path simulation.

---

SCRIPsimSingleCellMeans  
*Simulate cell means*

---

**Description**

Simulate a gene by cell matrix giving the mean expression for each gene in each cell. Cells start with the mean expression for the group they belong to (when simulating groups) or cells are assigned the mean expression from a random position on the appropriate path (when simulating paths). The selected means are adjusted for each cell's expected library size.

**Usage**

```
SCRIPsimSingleCellMeans(sim, params)
```

**Arguments**

sim	SingleCellExperiment to add cell means to.
params	SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with added cell means.

---

SCRIPsimTrueCounts      *Simulate true counts*

---

**Description**

Simulate a true counts matrix. Counts are simulated from a poisson distribution where Each gene in each cell has it's own mean based on the group (or path position), expected library size and BCV.

**Usage**

```
SCRIPsimTrueCounts(sim, params)
```

**Arguments**

sim	SingleCellExperiment to add true counts to.
params	SplatParams object with simulation parameters.

**Value**

SingleCellExperiment with simulated true counts.

SCRIPsimu

*SCRIP simulation***Description**

Simulate count data for single cell RNA-sequencing using SCIRP method

**Usage**

```
SCRIPsimu(
  data,
  params,
  method = "single",
  base_allcellmeans_SC = NULL,
  pre.bcv.df = NULL,
  libsize = NULL,
  bcv.shrink = 1,
  Dropout_rate = NULL,
  mode = "GP-trendedBCV",
  de.prob = NULL,
  de.downProb = NULL,
  de.facLoc = NULL,
  de.facScale = NULL,
  path.skew = NULL,
  batch.facLoc = NULL,
  batch.facScale = NULL,
  path.nSteps = NULL,
  ...
)
```

**Arguments**

<code>data</code>	data matrix required to fit the mean-BCV trend for simulation
<code>params</code>	SplatParams object containing parameters for the simulation
<code>method</code>	"single", "groups" or "paths"
<code>base_allcellmeans_SC</code>	base mean vector provided to help setting DE analysis
<code>pre.bcv.df</code>	BCV.df enables us to change the variation of BCV values
<code>libsize</code>	library size can be provided directly
<code>bcv.shrink</code>	factor to control the BCV levels
<code>Dropout_rate</code>	factor to control the dropout rate directly
<code>mode</code>	"GP-commonBCV", "BP-commonBCV", "BP", "BGP-commonBCV" and "BGP-trendedBCV"
<code>de.prob</code>	the proportion of DE genes

de.downProb	the proportion of down-regulated DE genes
de.facLoc	DE location factor
de.facScale	DE scale factor
path.skew	Controls how likely cells are from the start or end point of the path
batch.facLoc	DE location factor in batch
batch.facScale	DE scale factor in batch
path.nSteps	number of steps between the start point and end point for each path
...	Other parameters

**Value**

SingleCellExperiment file

**Examples**

```
data(params_acinar)
data(acinar.data)
sim_trend = SCRIPsimu(data=acinar.data, params=params_acinar, mode="GP-trendedBCV")
```

---

simu.VEGs	<i>SCRIP simulation for clustering analysis</i>
-----------	---

---

**Description**

Simulate count data for clustering analysis by preserving variably expressed genes

**Usage**

```
simu.VEGs(
  counts.matrix,
  params = params,
  base_allcellmeans,
  mode = "GP-trendedBCV",
  nCells,
  nfeatures = 1000
)
```

**Arguments**

counts.matrix	data matrix required for simulation
params	SplatParams object containing parameters for the simulation
base_allcellmeans	base cell means specified directly for simulating counts
mode	"GP-commonBCV", "BP-commonBCV", "BP", "BGP-commonBCV" and "BGP-trendedBCV"
nCells	number of cells simulated
nfeatures	parameter required for FinalVariable function in Seurat package

**Value**

simulated read counts data

---

simu_cluster	<i>SCRIP simulation for clustering analysis with multiple cell types</i>
--------------	--

---

**Description**

Simulate count data for clustering analysis by preserving variably expressed genes with multiple cell types

**Usage**

```
simu_cluster(expre_data, pheno_data, CTlist, mode, nfeatures, seed = 2021)
```

**Arguments**

expre_data	data matrix required for simulation
pheno_data	phenotype data information
CTlist	cell types used for simulation
mode	"GP-commonBCV", "BP-commonBCV", "BP", "BGP-commonBCV" and "BGP-trendedBCV"
nfeatures	parameter required for FinalVariable function in Seurat package
seed	seed used for simulation

**Value**

simulated read counts data with cell type information

---

simu_DE	<i>SCRIP simulation for differential expression</i>
---------	---

---

**Description**

Simulate count data for differential expression analysis using SCRIP

**Usage**

```
simu_DE(  
  expre_data,  
  params,  
  nGenes = NULL,  
  nDE,  
  ncells = NULL,  
  FC,  
  Dropout_rate = NULL,  
  libsize = NULL,  
  pre.bcv.df = NULL,  
  bcv.shrink = 1,  
  seed = 2021  
)
```

**Arguments**

expre_data	data matrix required for simulation
params	SplatParams object containing parameters for the simulation
nGenes	number of genes simulated
nDE	number of differentially expressed genes simulated
ncells	number of cells simulated
FC	fold change rate simulated between two groups
Dropout_rate	factor to control the dropout rate directly
libsize	library size used for simulation
pre.bcv.df	BCV.df enables us to change the variation of BCV values
bcv.shrink	factor to control the BCV levels
seed	seed for simulation

**Value**

SummarizedExperiment files from both groups for DE analysis and DE genes index

# Index

## \* datasets

acinar.data, 2  
params\_acinar, 5

acinar.data, 2

bridge, 3  
bringItemsForward, 3

getLNormFactors, 4, 6, 9  
getPathOrder, 4

logistic, 5

params\_acinar, 5

SCRIPsimBatchCellMeans, 6  
SCRIPsimBatchEffects, 6  
SCRIPsimBCVMeans, 7  
SCRIPsimDropout, 7  
SCRIPsimGeneMeans, 8  
SCRIPsimGroupCellMeans, 8  
SCRIPsimGroupDE, 9  
SCRIPsimLibSizes, 9  
SCRIPsimPathCellMeans, 10  
SCRIPsimPathDE, 10  
SCRIPsimSingleCellMeans, 11  
SCRIPsimTrueCounts, 11  
SCRIPsimu, 12  
simu.VEGs, 13  
simu\_cluster, 14  
simu\_DE, 14