Package 'simPIC'

May 29, 2024

Version 1.0.0

Date 2023-02-02

Type Package

Title simPIC: flexible simulation of paired-insertion counts for single-cell ATAC-sequencing data

Depends R (>= 4.4.0), SingleCellExperiment

Imports BiocGenerics, checkmate (>= 2.0.0), fitdistrplus, matrixStats, Matrix, stats, SummarizedExperiment, actuar, rlang, S4Vectors, methods, scales, scuttle

Description simPIC is a package for simulating single-cell ATAC-seq count data. It provides a user-friendly, well documented interface for data simulation. Functions are provided for parameter estimation, realistic scATAC-seq data simulation, and comparing real and simulated datasets.

biocViews SingleCell, ATACSeq, Software, Sequencing, ImmunoOncology, DataImport

License GPL-3

Encoding UTF-8

Suggests ggplot2 (>= 3.4.0), knitr, rmarkdown, BiocStyle, testthat (>= 3.0.0)

VignetteBuilder knitr

RoxygenNote 7.3.1

Config/testthat/edition 3

URL https://github.com/sagrikachugh/simPIC

BugReports https://github.com/sagrikachugh/simPIC/issues

git_url https://git.bioconductor.org/packages/simPIC

git_branch RELEASE_3_19

git_last_commit 51db9ff

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-28

Author Sagrika Chugh [aut, cre] (<https://orcid.org/0000-0002-8050-5214>),
Davis McCarthy [aut],
Heejung Shim [aut]

Maintainer Sagrika Chugh <sagrika.chugh@gmail.com>

Contents

`simPIC-package` *simPIC: A uniform quantification based method for simulating Paired-Insertion count matrices for single-cell ATAC sequencing data*

Description

simPIC is a package for simulating single-cell ATAC-seq count data. It provides a user-friendly, well documented interface for data simulation. Functions are provided for parameter estimation, realistic scATAC-seq data simulation, and comparing real and simulated datasets.

- count class (`newsimPICcount`)
- estimate (`simPICestimate`)
- simulate (`simPICsimulate`)
- plots (`simPICcompare`)

Author(s)

Maintainer: Sagrika Chugh <sagrika.chugh@gmail.com> (ORCID)

Authors:

- Davis McCarthy
- Heejung Shim

See Also

Useful links:

- https://github.com/sagrikachugh/simPIC
- Report bugs at https://github.com/sagrikachugh/simPIC/issues

---

`addFeatureStats` *Add feature statistics*

Description

Add additional feature statistics to a SingleCellExperiment object

Usage

```
addFeatureStats(
  sce,
  value = "counts",
  log = FALSE,
  offset = 1,
  no.zeros = FALSE
)
```

Arguments

`sce`	SingleCellExperiment to add feature statistics to.
`value`	the count value to calculate statistics.
`log`	logical. Whether to take log2 before calculating statistics.
`offset`	offset to add to avoid taking log of zero.
`no.zeros`	logical. Whether to remove all zeros from each feature before calculating statistics.

Details

Currently adds the following statistics: mean and variance. Statistics are added to the `rowData` slot and are named `Stat[Log]Value[No0]` where `Log` and `No0` are added if those arguments are true.

Value

SingleCellExperiment with additional feature statistics

convert_to_SCE *Convert Sparse Matrix to SingleCellExperiment object*

Description

This function converts a dgc/sparse matrix into a SingleCellExperiment(SCE) object.

Usage

```
convert_to_SCE(sparse_data)
```

Arguments

sparse_data	A sparse matrix containing count data, where rows are peaks and columns represent cells.

Value

A SingleCellExperiment(SCE) object with the sparse matrix stored in the "counts" assay.

getCounts *Get counts from Single Cell Experiment object*

Description

Get counts matrix from a SingleCellExperiment object. If counts is missing a warning is issued and the first assay is returned.

Usage

```
getCounts(sce)
```

Arguments

sce	SingleCellExperiment object

Value

counts matrix

`global` *simPIC: Simulate single-cell ATAC-seq data*

Description

simPIC: Simulate single-cell ATAC-seq data

Value

globalvariables

`newsimPICcount` *newsimPICcount*

Description

Create a newsimPICcount object to store parameters.

Usage

```
newsimPICcount(...)
```

Arguments

...	Variables to set newsimPICcount object parameters.

Details

This function creates the object variable which is passed in all functions.

Value

new object from class simPICcount.

Examples

```
object <- newsimPICcount()
```

plot_theme *Custom theme for ggplot2*

Description

This function defines a custom theme for ggplot2 to ensure consistent visual appearance across multiple plots.

Usage

```
plot_theme()
```

Value

A ggplot2 theme object with predefined settings.

rbindMatched *Bind rows (matched)*

Description

Bind the rows of two data frames, keeping only the columns that are common to both.

Usage

```
rbindMatched(df1, df2)
```

Arguments

df1	first data.frame to bind.
df2	second data.frame to bind.

Value

data.frame containing rows from df1 and df2 but only common columns.

selectFit *Select fit*

Description

Trying two fitting methods and selecting the best one.

Usage

```
selectFit(data, distr, verbose = TRUE)
```

Arguments

data	The data to fit.
distr	Name of the distribution to fit.
verbose	logical. To print messages or not.

Details

The distribution is fitted to the data using each of the `fitdist` fitting methods. The fit with the smallest Cramer-von Mises statistic is selected.

Value

The selected fit object

setsimPICparameters *Set simPIC parameters*

Description

Set input parameters of the simPICcount object.

Usage

```
setsimPICparameters(object, update = NULL, ...)
```

Arguments

object	input simPICcount object.
update	new parameters.
...	set new parameters for simPICcount object.

Value

simPICcount object with updated parameters.

Examples

```
object <- newsimPICcount()
object <- setsimPICparameters(object, nCells = 200, nPeaks = 500)
```

`simPICcompare` *Compare SingleCellExperiment objects*

Description

Combine data from several SingleCellExperiment objects and produce some basic plots comparing them.

Usage

```
simPICcompare(
  sces,
  point.size = 0.2,
  point.alpha = 0.1,
  fits = TRUE,
  colours = NULL
)
```

Arguments

`sces`	named list of SingleCellExperiment objects to combine and compare.
`point.size`	size of points in scatter plots.
`point.alpha`	opacity of points in scatter plots.
`fits`	whether to include fits in scatter plots.
`colours`	vector of colours to use for each dataset.

Details

The returned list has three items:

`RowData` Combined row data from the provided SingleCellExperiments.

`ColData` Combined column data from the provided SingleCellExperiments.

`Plots` Comparison plots

- `Means` Boxplot of mean distribution.
- `Variances` Boxplot of variance distribution.
- `MeanVar` Scatter plot with fitted lines showing the mean-variance relationship.
- `LibrarySizes` Boxplot of the library size distribution.
- `ZerosPeak` Boxplot of the percentage of each peak that is zero.
- `ZerosCell` Boxplot of the percentage of each cell that is zero.

`MeanZeros` Scatter plot with fitted lines showing the mean-zeros relationship.

The plots returned by this function are created using `ggplot` and are only a sample of the kind of plots you might like to consider. The data used to create these plots is also returned and should be in the correct format to allow you to create further plots using `ggplot`.

Value

List containing the combined datasets and plots.

Examples

```
sim1 <- simPICsimulate(
    nPeaks = 1000, nCells = 500,
    pm.distr = "weibull", seed = 7856
)
sim2 <- simPICsimulate(
    nPeaks = 1000, nCells = 500,
    pm.distr = "gamma", seed = 4234
)
comparison <- simPICcompare(list(weibull = sim1, gamma = sim2))
names(comparison)
names(comparison$Plots)
```

`simPICcount` *The simPICcount class*

Description

S4 class that holds parameters for simPIC simulation.

Value

a simPIC class object. The parameters not shown in brackets can be estimated from real data using `simPICestimate`. For details of the simPIC simulation see `simPICsimulate`. The default parameters are based on PBMC10k dataset and can be reproduced using test data and script provided in inst/script

Parameters

simPIC simulation parameters:

`nPeaks` The number of peaks to simulate.

`nCells` The number of cells to simulate.

`[seed]` Seed to use for generating random numbers.

`[default]` The logical variable whether to use default parameters (TRUE) or learn from data (FALSE)

Library size parameters `lib.size.meanlog` meanlog (location) parameter for the library size log-normal distribution.

`lib.size.sdlog` sdlog (scale) parameter for the library size log-normal distribution.

Peak mean parameters `mean.scale` scale parameter for the mean weibull distribution.

`mean.shape` shape parameter for the mean weibull distribution.

Cell sparsity parameters `sparsity` probability of openness to be multiplied to the input of poisson distribution to generate final simulated matrix.

`simPICestimate` *Estimate simPIC simulation parameters*

Description

Estimate simulation parameters for library size, peak means, and sparsity for simPIC simulation from a real peak by cell input matrix

Usage

```
simPICestimate(
  counts,
  object = newsimPICcount(),
  pm.distr = c("gamma", "weibull", "pareto", "lngamma"),
  verbose = TRUE
)

## S3 method for class 'SingleCellExperiment'
simPICestimate(
  counts,
  object = newsimPICcount(),
  pm.distr = "weibull",
  verbose = TRUE
)

## S3 method for class 'dgCMatrix'
simPICestimate(
  counts,
  object = newsimPICcount(),
  pm.distr = "weibull",
  verbose = TRUE
)
```

Arguments

`counts`	either a sparse peak by cell count matrix, or a SingleCellExperiment object containing count data to estimate parameters.
`object`	simPICcount object to store estimated parameters and counts.

pm.distr	statistical distribution for estimating peak mean parameters. Available distributions: gamma, weibull, lngamma, pareto. Default is weibull.
verbose	logical variable. Prints the simulation progress if TRUE.

Value

simPICcount object containing all estimated parameters.

Examples

```
counts <- readRDS(system.file("extdata", "test.rds", package = "simPIC"))
est <- newsimPICcount()
est <- simPICestimate(counts, pm.distr = "weibull")
```

---

simPICestimateLibSize *Estimate simPIC library size parameters.*

Description

Estimate the library size parameters for simPIC simulation.

Usage

```
simPICestimateLibSize(counts, object, verbose)
```

Arguments

counts	count matrix.
object	simPICcount object to store estimated values.
verbose	logical. To print messages or not.

Details

Parameters for the lognormal distribution are estimated by fitting the library sizes using fitdist. All the fitting methods are tried and the fit with the best Cramer-von Mises statistic is selected.

Value

simPICcount object with estimated library size parameters.

`simPICestimatePeakMean` *Estimate simPIC peak means*

Description

Estimate peak mean parameters for simPIC simulation

Usage

```
simPICestimatePeakMean(norm.counts, object, pm.distr, verbose)
```

Arguments

`norm.counts`	library size normalised counts matrix.
`object`	simPICcount object to store estimated values.
`pm.distr`	distribution parameter for peak means.
`verbose`	logical. To print progress messages or not.

Details

Parameters for gamma distribution are estimated by fitting the mean normalised counts using `fitdist`. All the fitting methods are tried and the fit with the best Cramer-von Mises statistic is selected.

Value

simPICcount object containing all estimated parameters

`simPICestimateSparsity` *Estimate simPIC peak sparsity.*

Description

Extract the accessibility proportion (sparsity) of each cell among all peaksvfrom the input count matrix.

Usage

```
simPICestimateSparsity(norm.counts, object, verbose)
```

Arguments

`norm.counts`	A sparse count matrix to estimate parameters from.
`object`	simPICcount object to store estimated parameters.
`verbose`	logical. To print messages or not.

Details

Vector of non-zero cell proportions of peaks is calculated by dividing the number of non-zero entries over the number of all cells for each peak.

Value

simPICcount object with updated non-zero cell proportion parameter.

---

simPICget *Get a single simPICcount parameter*

Description

Get the value of a single variable from input simPICcount object.

Usage

```
simPICget(object, name)
```

Arguments

object	input simPICcount object.
name	name of the parameter.

Value

Value of the input parameter.

Examples

```
object <- newsimPICcount()
nPeaks <- simPICget(object, "nPeaks")
```

---

simPICgetparameters *Get parameters*

Description

Get multiple parameter values from a simPIC object.

Usage

```
simPICgetparameters(object, names)
```

Arguments

object	input object to get values from.
names	vector of names of the parameters to get.

Value

List with the values of the selected parameters.

Examples

```
object <- newsimPICcount()
simPICgetparameters(object, c("nPeaks", "nCells", "peak.mean.shape"))
```

---

simPICsimulate *simPIC simulation*

Description

Simulate peak by cell count matrix from a sparse single-cell ATAC-seq peak by cell input using simPIC methods.

Usage

```
simPICsimulate(
  object = newsimPICcount(),
  verbose = TRUE,
  pm.distr = "weibull",
  ...
)
```

Arguments

object	simPICcount object with simulation parameters. See simPICcount for details.
verbose	logical variable. Prints the simulation progress if TRUE.
pm.distr	distribution parameter for peak means. Available distributions: gamma, weibull, lngamma, pareto. Default is weibull.
...	Any additional parameter settings to override what is provided in simPICcount object.

Details

simPIC provides the option to manually adjust each of the simPICcount object parameters by calling setsimPICparameters.

The simulation involves following steps:

1. Set up simulation parameters

2. Set up SingleCellExperiment object
3. Simulate library sizes
4. Simulate sparsity
5. Simulate peak means
6. Create final synthetic counts

The final output is a SingleCellExperiment object that contains the simulated count matrix. The parameters are stored in the colData (for cell specific information), rowData (for peak specific information) or assays (for peak by cell matrix) slots. This additional information includes:

Value

SingleCellExperiment object containing the simulated counts.

Examples

```
# default simulation
sim <- simPICsimulate(pm.distr = "weibull")
```

---

simPICsimulateLibSize *Simulate simPIC library sizes*

Description

Generate library sizes for cells in simPIC simulation based on the estimated values of mus and sigmas.

Usage

```
simPICsimulateLibSize(object, sim, verbose)
```

Arguments

object	simPICcount object with simulation parameters.
sim	SingleCellExperiment object containing simulation parameters.
verbose	logical. To print progress messages.

Value

SingleCellExperiment object with simulated library sizes.

`simPICsimulatePeakMean` *Simulate simPIC peak means.*

Description

Generate peak means for cells in simPIC simulation based on the estimated values of shape and rate parameters.

Usage

```
simPICsimulatePeakMean(object, sim, pm.distr, verbose)
```

Arguments

`object`	simPICcount object with simulation parameters.
`sim`	SingleCellExperiment object containing simulation parameters.
`pm.distr`	distribution parameter for peak means. Available distributions: gamma, weibull, lngamma, pareto. Default is weibull.
`verbose`	logical. Whether to print progress messages.

Value

SingleCellExperiment object with simulated peak means.

`simPICsimulateTrueCounts` *Simulate true counts.*

Description

Counts are simulated from a poisson distribution where each peak has a mean, expected library size and proportion of accessible chromatin.

Usage

```
simPICsimulateTrueCounts(object, sim)
```

Arguments

`object`	simPICcount object with simulation parameters.
`sim`	SingleCellExperiment object containing simulation parameters.

Value

SingleCellExperiment object with simulated true counts.

Index