Package 'nucleoSim'

November 12, 2026

Version 1.40.0

Date 2015-07-24

Title Generate synthetic nucleosome maps

Description This package can generate a synthetic map with reads covering
the nucleosome regions as well as a synthetic map with
forward and reverse reads emulating next-generation sequencing.
The synthetic hybridization data of "Tiling Arrays" can also be generated.
The user has choice between three different
distributions for the read positioning: Normal, Student and Uniform.
In addition, a visualization tool is provided to explore the synthetic
nucleosome maps.

Encoding UTF-8

Imports stats, IRanges, S4Vectors, graphics, methods

Suggests BiocStyle, BiocGenerics, knitr, rmarkdown, testthat

License Artistic-2.0

URL https://github.com/arnauddroitlab/nucleoSim

BugReports https://github.com/arnauddroitlab/nucleoSim/issues

VignetteBuilder knitr

NeedsCompilation no

biocViews Genetics, Sequencing, Software, StatisticalMethod, Alignment

RoxygenNote 7.1.2

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

git_branch RELEASE_3_23

git_last_commit d807a7a

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Rawane Samb [aut],
Astrid Deschênes [cre, aut] (ORCID:
<https://orcid.org/0000-0001-7846-6749>),
Pascal Belleau [aut] (ORCID: <https://orcid.org/0000-0002-0802-1071>),
Arnaud Droit [aut]

Maintainer Astrid Deschênes <adeschen@hotmail.com>

Contents

nucleoSim-package 2
createNucReadsFromNucMap 3
plot.syntheticNucMap 4
plot.syntheticNucReads 5
syntheticNucMapFromDist 6
syntheticNucMapFromDistValidation 8
syntheticNucReadsFromDist 10
syntheticNucReadsFromMap 12
syntheticNucReadsValidation 13

Index **15**

`nucleoSim-package` *nucleoSim: Generate synthetic nucleosome maps*

Description

This package can generate a synthetic map with reads covering the nucleosome regions as well as a synthetic map with forward and reverse reads emulating next-generation sequencing. The user has choice between three different distributions for the read positioning: Normal, Student and Uniform.

Author(s)

Rawane Samb, Astrid Deschenes, Pascal Belleau and Arnaud Droit

Maintainer: Astrid Deschenes <adeschen@hotmail.com>

See Also

- `syntheticNucMapFromDist` to generate a synthetic nucleosome map

createNucReadsFromNucMap *Create a synthetic nucleosome reads from a synthetic nucleosome map*

Description

Generate a synthetic nucleosome map using a synthetic nucleosome map.

Usage

```
createNucReadsFromNucMap(map, read.len, offset, call)
```

Arguments

map	a list of class "syntheticNucMap"
read.len	the length of each of the paired-end reads. Default = 40.
offset	the number of bases used to offset all nucleosomes and reads. This is done to ensure that all nucleosome positions and read alignment are of positive values.
call	the function call.

Value

an list of class "syntheticNucReads" containing the following elements:

- call the matched call.
- dataIP a data.frame with the chromosome name, the starting and ending positions and the direction of all forward and reverse reads for all well-positioned and fuzzy nucleosomes.
- wp a data.frame with the positions of all the well-positioned nucleosomes, as well as the number of paired-reads associated to each one.
- fuz a data.frame with the positions of all the fuzzy nucleosomes, as well as the number of paired-reads associated to each one.
- paired a data.frame with the starting and ending positions of the reads used to generate the paired-end reads.

Author(s)

Astrid Deschenes

Examples

```
## Generate a synthetic map with 30 well-positioned nucleosomes, 5 fuzzy
## nucleosomes and 6 deleted nucleosomes using a Student distribution
## with a variance of 10 for the well-positioned nucleosomes,
## a variance of 15 for the fuzzy nucleosomes and a seed of 1335
map_call <- call("syntheticNucMapFromDist", wp.num = 30, wp.del = 6,
wp.var = 10, fuz.num = 5, fuz.var = 15, rnd.seed = 1335,
```

```
distr = "Student")
syntheticMap <- eval(map_call)

syntheticReads <- nucleoSim:::createNucReadsFromNucMap(syntheticMap,
read.len = 40, offset = 1000, call = map_call)
```

---

plot.syntheticNucMap *Generate a graph of a synthetic nucleosome map*

Description

Generate a graph for a list marked as an syntheticNucMap class

Usage

```
## S3 method for class 'syntheticNucMap'
plot(x, ...)
```

Arguments

x	a list marked as an syntheticNucMap class
...	... extra arguments passed to the plot function

Value

a graph of a synthetic nucleosome map

Author(s)

Astrid Deschenes, Rawane Samb

Examples

```
## Generate a synthetic map with 20 well-positioned nucleosomes, 5 fuzzy
## nucleosomes and 10 deleted nucleosomes using a Student distribution
## with a variance of 10 for the well-positioned nucleosomes,
## a variance of 20 for the fuzzy nucleosomes and a seed of 15
syntheticMap <- syntheticNucMapFromDist(wp.num = 30, wp.del = 10,
wp.var = 10, fuz.num = 5, fuz.var = 20, rnd.seed = 15,
distr = "Student")

## Create graph using the synthetic map
plot(syntheticMap, xlab="Position", ylab="Coverage")
```

`plot.syntheticNucReads` *Generate a graph of a synthetic nucleosome map containing forward and reverse reads*

Description

Generate a graph for a list marked as an `syntheticNucReads` class

Usage

```
## S3 method for class 'syntheticNucReads'
plot(x, ...)
```

Arguments

`x`	a list marked as a `syntheticNucReads` class
`...`	`...` extra arguments passed to the `plot` function

Value

a graph of a synthetic nucleosome map containing forward and reverse reads

Author(s)

Astrid Deschenes, Rawane Samb

Examples

```
## Generate a synthetic map with 20 well-positioned nucleosomes, 5 fuzzy
## nucleosomes and 10 deleted nucleosomes using a Student distribution
## with a variance of 10 for the well-positioned nucleosomes,
## a variance of 20 for the fuzzy nucleosomes and a seed of 15
syntheticNucSample <- syntheticNucReadsFromDist(wp.num = 30, wp.del = 10,
wp.var = 10, fuz.num = 5, fuz.var = 20, rnd.seed = 15,
distr = "Student", offset = 1000)

## Create graph using the synthetic map
plot(syntheticNucSample, xlab="Position", ylab="Coverage")
```

`syntheticNucMapFromDist` *Generate a synthetic nucleosome map containing complete sequences*

Description

Generate a synthetic nucleosome map, a map with complete sequences covering the nucleosome regions, using the distribution selected by the user. The distribution is used to assign the start position to the sequences associated with the nucleosomes. The user has choice between three different distributions: Normal, Student and Uniform.

The synthetic nucleosome map creation is separated into 3 steps :

1. Adding well-positioned nucleosomes following specified parameters. The nucleosomes are all positioned at equidistance. Assigning sequences of variable length to each nucleosome using a normal distribution and specified variance.

2. Deleting some well-positioned nucleosomes following specified parameters. Each nucleosome has an equal probability to be selected.

3. Adding fuzzy nucleosomes following an uniform distribution ad specified parameters. Assigning sequences of variable length to each nucleosome using the specified distribution and parameters. The sequence length is always following a normal distribution.

This function is a modified version of the syntheticNucMap() function from Bioconductor nucleR package (Flores and Orozco, 2011).

Usage

```
syntheticNucMapFromDist(
  wp.num,
  wp.del,
  wp.var,
  fuz.num,
  fuz.var,
  max.cover = 100,
  nuc.len = 147,
  len.var = 10,
  lin.len = 20,
  rnd.seed = NULL,
  as.ratio = FALSE,
  distr = c("Uniform", "Normal", "Student")
)
```

Arguments

`wp.num`	a non-negative `integer`, the number of well-positioned (non-overlapping) nucleosomes.
`wp.del`	a non-negative `integer`, the number of well-positioned nucleosomes to remove to create uncovered regions.

wp.var	a non-negative integer, the variance associated with the distribution used to assign the start position to the sequences of the well-positioned nucleosomes. This parameter introduces some variation in the starting positions.
fuz.num	a non-negative integer, the number of fuzzy nucleosomes. Those nucleosomes are distributed accordingly to an uniform distribution all over the region. Those nucleosomes can overlap other well-positioned or fuzzy nucleosomes.
fuz.var	a non-negative integer, the variance associated with the distribution used to assign the start position to the sequences of the fuzzy nucleosomes. This parameter introduces some variation in the starting positions.
max.cover	a positive integer, the maximum coverage for one nucleosome. The final coverage can have a higher value than max.cover since sequences from different nucleosomes can be overlapping. Default = 100.
nuc.len	a non-negative numeric, the nucleosome length. Default = 147.
len.var	a non-negative integer, the variance associated to the normal distribution used to add some variance to the length of each sequence. Default = 10.
lin.len	a non-negative integer, the length of the DNA linker. Default = 20.
rnd.seed	a single value, interpreted as an integer, or NULL. If a integer is given, the value is used to set the seed of the random number generator. By fixing the seed, the generated results can be reproduced. Default = NULL.
as.ratio	a logical, if TRUE, a synthetic naked DNA control map is created and the ratio between it and the nucleosome coverage are calculated. It can be used to simulate hybridization ratio data, like the one in Tiling Arrays. Both control map and obtained ratio are returned. Default = FALSE.
distr	the name of the distribution used to generate the nucleosome map. The choices are : "Uniform", "Normal" and "Student". Default = "Uniform".

Value

an list of class "syntheticNucMap" containing the following elements:

- call the matched call.
- wp.starts a vector of integer, the start positions of all well-positioned nucleosome regions. The central position of the nucleosome is calculated as wp.starts + round(nuc.len/2).
- wp.nreads a vector of integer, the number of sequences associated to each well-positioned nucleosome.
- wp.reads a IRanges containing the well-positioned nucleosome sequences.
- fuz.starts a vector of integer, the start position of all the fuzzy nucleosomes.
- fuz.nreads a vector of integer, the number of sequences associated to each fuzzy nucleosome.
- fuz.reads a IRanges containing the fuzzy nucleosome sequences.
- syn.reads a IRanges containing all the synthetic nucleosome sequences (from both fuzzy and well-positioned nucleosomes).
- nuc.len a numeric the nucleosome length.

The following elements will be only returned if `as.ratio=TRUE`:

- `ctr.reads` a `IRanges` containing the naked DNA (control) sequences.
- `syn.ratio` a `Rle` containing the calculated ratio between the nucleosome coverage and the control coverage.

Author(s)

Rawane Samb, Astrid Deschenes

Examples

```
## Generate a synthetic map with 20 well-positioned nucleosomes and 10 fuzzy
## nucleosomes using a Normal distribution with a variance of 30 for the
## well-positioned nucleosomes, a variance of 40 for the fuzzy nucleosomes
## and a seed of 15.
syntheticNucMapFromDist(wp.num = 20, wp.del = 0, wp.var = 30,
    fuz.num = 10, fuz.var = 40, rnd.seed = 15,
    distr = "Normal")

## Same output but with ratio
syntheticNucMapFromDist(wp.num = 20, wp.del = 0, wp.var = 30,
    fuz.num = 10, fuz.var = 40,
    rnd.seed = 15, as.ratio = TRUE, distr = "Normal")
```

---

syntheticNucMapFromDistValidation

Parameter validation for the `syntheticNucMapFromDist` *function*

Description

Validate that all values passed to the function are formated for the `syntheticNucMapFromDist` function.

Usage

```
syntheticNucMapFromDistValidation(
  wp.num,
  wp.del,
  wp.var,
  fuz.num,
  fuz.var,
  max.cover,
  nuc.len,
  len.var,
  lin.len,
```

```
  rnd.seed,
  as.ratio
)
```

Arguments

wp.num	a non-negative integer, the number of well-positioned (non overlapping) nucleosomes.
wp.del	a non-negative integer, the number of well-positioned nucleosomes to remove to create uncovered regions.
wp.var	a non-negative integer, the variance of the starting position of the sequences associated with well-positioned nucleosomes. This parameter introduces some variation in the starting position of the sequences describing a nucleosome.
fuz.num	a non-negative integer, the number of fuzzy nucleosomes. Those nucleosomes can overlap other well-positioned or fuzzy nucleosomes.
fuz.var	a non-negative integer, the variance of the fuzzy nucleosomes. This variance can be different than the one used for the well-positioned nucleosome reads.
max.cover	a positive integer, the maximum coverage for one nucleosome. The final coverage can have a higher value than max.cover since reads from differents nucleosomes can be overlapping.
nuc.len	a non-negative integer, the nucleosome length.
len.var	a non-negative integer, the variance of the distance between a forward read and its paired reverse read.
lin.len	a non-negative integer, the length of the DNA linker DNA.
rnd.seed	a single value, interpreted as an integer, or NULL. If a integer is given, the value is used to set the seed of the random number generator. By fixing the seed, the generated results can be reproduced.
as.ratio	a logical, if TRUE, a synthetic naked DNA control map is created and the ratio between it and the nucleosome coverage are calculated. It can be used to simulate hybridization ratio data, like the one in Tiling Arrays. Both control map and calculated ratio are returned.

Value

0 indicating that all parameters validations have been successful.

Author(s)

Astrid Deschenes

Examples

```
## The function returns 0 when all paramaters are valid
nucleoSim:::syntheticNucMapFromDistValidation(wp.num = 20, wp.del = 2,
wp.var = 3, fuz.num = 10, fuz.var = 5, max.cover = 100, nuc.len = 147,
len.var = 4, lin.len = 40, rnd.seed = 201, as.ratio = FALSE)
```

```
## The function raises an error when at least one paramater is not valid
## Not run: nucleoSim:::syntheticNucMapFromDistValidation(wp.num = -1,
wp.del = 2, wp.var = 3, fuz.num = 10, fuz.var = 5, max.cover = 100,
nuc.len = 147, len.var = 4, lin.len = 40, rnd.seed = 201, as.ratio = FALSE)
## End(Not run)
## Not run: nucleoSim:::syntheticNucMapFromDistValidation(wp.num = 20,
wp.del = 2, wp.var = -3, fuz.num = 10, fuz.var = 5, max.cover = 100,
nuc.len = 147, len.var = 4, lin.len = 40, rnd.seed = 201, as.ratio = FALSE)
## End(Not run)
```

`syntheticNucReadsFromDist` *Generate a synthetic nucleosome map containing forward and reverse reads (paired-end reads)*

Description

Generate a synthetic nucleosome map, a map with forward and reverses reads (paired-end reads) covering the nucleosome regions, using the distribution selected by the user. The distribution is used to assign the start position to the forward reads associated with the nucleosomes. The user has choice between three different distributions: Normal, Student and Uniform. The final map is composed of paired-end reads.

#' The synthetic nucleosome map creation is separated into 3 steps :

1. Adding well-positioned nucleosomes following specified parameters. The nucleosomes are all positioned at equidistance. Assigning the starting positions of forward reads using the specified distribution and parameters. The distance between starting positions of paired-end reads is assigned using a normal distribution and specified variance.

2. Deleting some well-positioned nucleosomes following specified parameters. Each nucleosome has an equal probability to be selected.

3. Adding fuzzy nucleosomes following an uniform distribution and specified parameters. Assigning the starting positions of forward reads using the specified distribution and parameters. The distance between starting positions of paired-end reads is assigned using a normal distribution and specified variance.

This function has been largely inspired by the Generating synthetic maps section of the nucleR package (Flores et Orozco, 2011).

Usage

```
syntheticNucReadsFromDist(
  wp.num,
  wp.del,
  wp.var,
  fuz.num,
  fuz.var,
  max.cover = 100,
```

```
  nuc.len = 147,
  len.var = 10,
  lin.len = 20,
  read.len = 40,
  rnd.seed = NULL,
  distr = c("Uniform", "Normal", "Student"),
  offset
)
```

Arguments

wp.num	a non-negative integer, the number of well-positioned (non-overlapping) nucleosomes.
wp.del	a non-negative integer, the number of well-positioned nucleosomes to remove to create uncovered regions.
wp.var	a non-negative integer, the variance associated with the distribution used to assign the start position to the forward reads of the well-positioned nucleosomes. This parameter introduces some variation in the starting positions.
fuz.num	a non-negative numeric, the number of fuzzy nucleosomes. Those nucleosomes are distributed accordingly to an uniform distribution all over the region. Those nucleosomes can overlap other well-positioned or fuzzy nucleosomes.
fuz.var	a non-negative numeric, the maximum variance of the fuzzy nucleosomes. This variance can be different than the one used for the well-positioned nucleosome reads.
max.cover	a positive numeric, the maximum coverage for one nucleosome. The final coverage can have a higher value than max.cover since reads from different nucleosomes can be overlapping. Default = 100.
nuc.len	a positive integer, the nucleosome length. Default = 147.
len.var	a positive numeric, the variance of the distance between a forward read and its paired reverse read. Default = 10.
lin.len	a non-negative integer, the length of the DNA linker. Default = 20.
read.len	a positive integer, the length of each of the paired-end reads. Default = 40.
rnd.seed	a single value, interpreted as an integer, or NULL. If an integer is given, the value is used to set the seed of the random number generator. By fixing the seed, the generated results can be reproduced. Default = NULL.
distr	the name of the distribution used to generate the nucleosome map. The choices are : "Uniform", "Normal" and "Student". Default = "Uniform".
offset	a non-negative integer, the number of bases used to offset all nucleosomes and reads. This is done to ensure that all nucleosome positions and read alignment are of positive values.

Value

an list of class "syntheticNucReads" containing the following elements:

- call the matched call.

- dataIP a data.frame with the chromosome name, the starting and ending positions and the direction of all forward and reverse reads for all well-positioned and fuzzy nucleosomes. Paired-end reads are identified with an unique id.
- wp a data.frame with the positions of all the well-positioned nucleosomes, as well as the number of paired-reads associated to each one.
- fuz a data.frame with the positions of all the fuzzy nucleosomes, as well as the number of paired-reads associated to each one.
- paired a data.frame with the starting and ending positions of the reads used to generate the paired-end reads. Paired-end reads are identified with an unique id.

Author(s)

Pascal Belleau, Rawane Samb, Astrid Deschenes

Examples

```
## Generate a synthetic map with 20 well-positioned + 10 fuzzy nucleosomes
## using a Normal distribution with a variance of 30 for the well-positioned
## nucleosomes, a variance of 40 for the fuzzy nucleosomes and a seed of 15.
## Because of the fixed seed, each time is going to be run, the results
## are going to be the seed.
res <- syntheticNucReadsFromDist(wp.num = 20, wp.del = 0, wp.var = 30,
fuz.num = 10, fuz.var = 40, rnd.seed = 15, distr = "Normal",
offset = 1000)
```

---

syntheticNucReadsFromMap

Generate a synthetic nucleosome map containing forward and reverse reads

---

Description

Generate a synthetic nucleosome map using a synthetic nucleosome map.

This function is using a modified version of the syntheticNucMap() function from Bioconductor nucleR package (Flores and Orozco, 2011).

Usage

```
syntheticNucReadsFromMap(syntheticNucMap, read.len = 40, offset)
```

Arguments

syntheticNucMap	a list of class "syntheticNucMap"
read.len	a positive integer, the length of each of the paired-end reads. Default = 40.
offset	a non-negatvie integer,the number of bases used to offset all nucleosomes and reads. This is done to ensure that all nucleosome positions and read alignments are of positive values.

Value

a `list` of `class` "syntheticNucReads" containing the following elements:

- `call` the matched call.
- `dataIP` a `data.frame` with the chromosome name, the starting and ending positions and the direction of all forward and reverse reads for all well-positioned and fuzzy nucleosomes. Paired-end reads are identified with an unique id.
- `wp` a `data.frame` with the positions of all the well-positioned nucleosomes, as well as the number of paired-reads associated to each one.
- `fuz` a `data.frame` with the positions of all the fuzzy nucleosomes, as well as the number of paired-reads associated to each one.
- `paired` a `data.frame` with the starting and ending positions of the reads used to generate the paired-end reads. Paired-end reads are identified with an unique id.

Author(s)

Pascal Belleau, Rawane Samb, Astrid Deschenes

Examples

```
## Generate a synthetic map with 20 well-positioned + 10 fuzzy nucleosomes
## using a Normal distribution with a variance of 30 for the well-positioned
## nucleosomes, a variance of 40 for the fuzzy nucleosomes and a seed of 15
## Because of the fixed seed, each time is going to be run, the results
## are going to be the seed
syntheticMap <- syntheticNucMapFromDist(wp.num = 20, wp.del = 0,
    wp.var = 30, fuz.num = 10, fuz.var = 40,
    rnd.seed = 335, as.ratio = FALSE, distr = "Uniform")

res <- nucleoSim:::syntheticNucReadsFromMap(syntheticMap, read.len = 45,
    offset = 1000)
```

---

syntheticNucReadsValidation

Subsection of parameter validation for identical parameters between `syntheticNucReadsFromMap` *and* `syntheticNucReadsFromDist` *functions.*

Description

Validate that identical values passed to both `syntheticNucReadsFromMap` and `syntheticNucReadsFromDist` functions are correctly formatted.

Usage

```
syntheticNucReadsValidation(read.len, offset)
```

Arguments

`read.len`	a positive `integer`, the length of each of the paired-end reads.
`offset`	a non-negative `integer`, the number of bases used to offset all nucleosomes and reads. This is done to ensure that all nucleosome positions and read alignment are of positive values.

Value

`0` indicating that all parameters validations have been successful.

Author(s)

Astrid Deschenes

Examples

```
## The function returns 0 when all paramaters are valid
nucleoSim:::syntheticNucReadsValidation(read.len = 40, offset = 100)

## The function raises an error when at least one paramater is not valid
## Not run: nucleoSim:::syntheticNucReadsValidation(read.len = 0, offset = 100)
## Not run: nucleoSim:::syntheticNucReadsValidation(read.len = 30, offset = -1)
```

Index

∗ **internal**
createNucReadsFromNucMap, 3
syntheticNucMapFromDistValidation, 8
syntheticNucReadsValidation, 13
∗ **package**
nucleoSim-package, 2

createNucReadsFromNucMap, 3

nucleoSim *(*nucleoSim-package*)*, 2
nucleoSim-package, 2

plot.syntheticNucMap, 4
plot.syntheticNucReads, 5

syntheticNucMapFromDist, *2*, 6, *8*
syntheticNucMapFromDistValidation, 8
syntheticNucReadsFromDist, 10
syntheticNucReadsFromMap, 12
syntheticNucReadsValidation, 13