Package ‘MOSim’

March 28, 2024

Title Multi-Omics Simulation (MOSim)

Version 1.16.0

Description MOSim package simulates multi-omic experiments that mimic regulatory mechanisms within the cell, allowing flexible experimental design including time course and multiple groups.

Encoding UTF-8

Depends R (>= 4.2.0)

License GPL-3

LazyData false

biocViews Software, TimeCourse, ExperimentalDesign, RNASeq

BugReports https://github.com/ConesaLab/MOSim/issues

URL https://github.com/ConesaLab/MOSim

Imports HiddenMarkov, zoo, methods, matrixStats, dplyr, stringi, lazyeval, rlang, scales, stringr, tibble, tidyr, ggplot2, Biobase, IRanges, S4Vectors

Suggests testthat, knitr, rmarkdown, BiocStyle

Collate 'AllClass.R' 'AllGeneric.R' 'Simulator.R' 'SimulatorRegion.R' 'ChIP-seq.R' 'DNase-seq.R' 'functions.R' 'Simulation.R' 'MOSim.R' 'RNA-seq.R' 'simulate_WGBS_functions.R' 'methyl-seq.R' 'miRNA-seq.R' 'zzz.R'

RoxygenNote 7.2.3

VignetteBuilder knitr

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

git_branch RELEASE_3_18

git_last_commit 4c2e0b9

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-03-27
Author Carlos Martínez [aut],
   Carolina Monzó [aut],
   Sonia Tarazona [cre, aut]
Maintainer Sonia Tarazona <sotacam@gmail.com>

R topics documented:

MOSim-package .......................................................... 2
discretize ................................................................. 2
experimentalDesign ....................................................... 3
is.declared ............................................................... 4
mosim ................................................................. 4
MOSimulation-class ...................................................... 6
MOSimulator-class ....................................................... 7
MOSimulatorRegion-class .............................................. 8
omicData ............................................................. 9
omicResults ........................................................... 10
omicSettings .......................................................... 11
omicSim .............................................................. 12
plotProfile ............................................................ 13
sampleData .......................................................... 14
Index 15

MOSim-package MOSim

Description

Multiomics simulation package.

discretize Discretize ChIP-Seq counts to simulate a binary dataset

Description

Discretize ChIP-Seq counts to simulate a binary dataset

Usage

discretize(df, omic)

Arguments

df A MOSimulated object
omic Character string of the omic to transform into binary data
**Value**

A regulator dataframe of 0 and 1

**Examples**

```r
omic_list <- c("RNA-seq", "ChIP-seq")
rnaseq_simulation <- mosim(omics = omic_list, omicsOptions = c(omicSim("ChIP-seq", totalFeatures = 2500)))
rnaseq_simulated <- omicResults(rnaseq_simulation, omic_list)
discrete_ChIP <- discretize(rnaseq_simulated, "ChIP-seq")
```

---

**experimentalDesign**

Retrieves the experimental design

**Description**

Retrieves the experimental design

**Usage**

```r
experimentalDesign(simulation)
```

**Arguments**

- `simulation` A MOSimulation object

**Value**

A data frame containing the experimental design used to simulate the data.

**Examples**

```r
omic_list <- c("RNA-seq")
rnaseq_simulation <- mosim(omics = omic_list)
# This will be a data frame with RNA-seq counts

design_matrix <- experimentalDesign(rnaseq_simulation)
```
is.declared  

Description  
Check if a variable is declared.

Usage  
is.declared(object, key = NULL)

Arguments  
object  Variable name to check  
key  Optional key to check inside object.

Value  
TRUE or FALSE indicating if the variable is initialized & non-empty.

mosim  

Description  
Performs a multiomic simulation by chaining two actions: 1) Creating the "MOSimulation" class with the provided params. 2) Calling "simulate" method on the initialized object.

Usage  
mosim(  
  omics,  
  omicsOptions,  
  diffGenes,  
  numberReps,  
  numberGroups,  
  times,  
  depth,  
  profileProbs,  
  minMaxFC,  
  TFtoGene  
)
Arguments

**omics** Character vector containing the names of the omics to simulate, which can be "RNA-seq", "miRNA-seq", "DNase-seq", "ChIP-seq" or "Methyl-seq" (e.g. c("RNA-seq", "miRNA-seq")). It can also be a list with the omic names as names and their options as values, but we recommend to use the argument omicSim to provide the options to simulated each omic.

**omicsOptions** List containing the options to simulate each omic. We recommend to apply the helper method `omicSim` to create this list in a friendly way, and the function `omicData` to provide custom data (see the related sections for more information). Each omic may have different configuration parameters, but the common ones are:

- **simuData/idToGene** Seed sample and association tables for regulatory omics. The helper function `omicData` should be used to provide this information (see the following section).
- **regulatorEffect** For regulatory omics. List containing the percentage of effect types (repressor, activator or no effect) over the total number of regulators. See vignette for more information.
- **totalFeatures** Number of features to simulate. By default, the total number of features in the seed dataset.
- **depth** Sequencing depth in millions of reads. If not provided, it takes the global parameter passed to `mosim` function.
- **replicateParams** List with parameters $a$ and $b$ for adjusting the variability in the generation of replicates using the negative binomial. See vignette for more information.
- **diffGenes** Number of differentially expressed genes to simulate, given in percentage (0 - 1) or in absolute number (> 1). By default 0.15
- **numberReps** Number of replicates per experimental condition (and time point, if time series are to be generated). By default 3.
- **numberGroups** Number of experimental groups or conditions to simulate.
- **times** Vector of time points to consider in the experimental design.
- **depth** Sequencing depth in millions of reads.
- **profileProbs** Numeric vector with the probabilities to assign each of the patterns. Defaults to 0.2 for each.
- **minMaxFC** Numeric vector of length 2 with minimum and maximum fold-change for differentially expressed features, respectively.
- **TFtoGene** A logical value indicating if default transcription factors data should be used (TRUE) or not (FALSE), or a 3 column data frame containing custom associations. By default FALSE.

Value

Instance of class "MOSimulation" containing the multiomic simulation data.
Examples

```r
mosimulation <- mosim(
  omics = c("RNA-seq"),
  numberReps = 3,
  times = c(0, 2, 6, 12, 24)
)

# Retrieve simulated count matrix for RNA-seq
dataRNAseq <- omicResults(mosimulation, "RNA-seq")
```

MOSimulation-class

This class manages the global simulation process, like associating genes with gene classes, regulatory programs and other settings. Finally it will initialize the simulators with their options that will use the previously generated settings to simulate the data.

Description

This class manages the global simulation process, like associating genes with gene classes, regulatory programs and other settings. Finally it will initialize the simulators with their options that will use the previously generated settings to simulate the data.

Slots

- `simulators`: Vector containing either S4 initialized classes of simulators or a list with the class name as keys, and its options as value, see example.
- `totalGenes`: A number with the total number of genes including not expressed. Overwritten if a genome reference is provided. Currently not used as we force to provide real data.
- `diffGenes`: A number with the total number of differential genes (if value > 1) or % or total genes (if value < 1).
- `numberReps`: Number of replicates of the experiment.
- `numberGroups`: Number of samples considered on the experiment.
- `times`: Numeric vector containing the measured times. If numberGroups < 2, the number of times must be at least 2.
- `geneNames`: Read only. List containing the IDs of the genes. Overwrited by the genome reference if provided. Currently not used as we force to provide real data.
- `simSettings`: List of settings that overrides initializing the configuration of the simulation by passing a previously generated list. This could be used to tweak by hand the assigned profiles, genes, regulatory programs, etc.
- `noiseFunction`: Noise function to apply when simulating counts. Must accept the parameter `n` and return a vector of the same length. Defaults to `rnorm`.
- `profiles`: Named list containing the patterns with their coefficients.
MOSimulator-class

profileProbs Numeric vector with the probabilities to assign each of the patterns. Defaults to 0.2 for each.
noiseParams Default noise parameters to be used with noise function.
depth Default depth to simulate.
TFtoGene Boolean (for default data) or 3 column data frame containing Symbol-TFGene-LinkedGene
minMaxQuantile Numeric vector of length 2 indicating the quantiles to use in order to retrieve the absolute minimum and maximum value that a differentially expressed feature can have.
minMaxFC Numeric vector of length 2 indicating the minimum and maximum fold-change that a differentially expressed feature can have.

MOSimulator-class Virtual class containing common methods and slots for child classes.

Description

Virtual class containing common methods and slots for child classes.

Slots

name Name of the simulator to be used in messages.
data Data frame containing the initial sample to be used, with the features IDs as rownames and only one column named "Counts".
regulator Boolean flag to indicate if the omic is a regulator or not.
regulatorEffect Possible regulation effects of the omic (enhancer, repressor or both).
idToGene Data frame with the association table between genes and other features. The structure must be 2 columns, one named "ID" and the other "Gene".
min Minimum value allowed in the omic.
max Maximum value allowed in the omic.
depth Sequencing depth to simulate.
depthRound Number of decimal places to round when adjusting depth.
depthAdjust Boolean indicating whether to adjust by sequencing depth or not.
totalFeatures Number of features to simulate. This will replace the data with a subset.
oiseFunction Noise function to apply when simulating counts. Must accept the parameter 'n' and return a vector of the same length. Defaults to 'rnorm'
increment Read-only. Minimum value to increase when simulating counts.
simData Contains the final simulated data.
pregenerated Indicates if the child class will generate the simulated data instead of the general process.
randData Auxiliary vector containing the original count data in random order with other adjustments.
MOSimulatorRegion-class

Virtual class containing general methods for simulators based on regions of the chromosomes, like DNase-seq, ChIP-seq or Methyl-seq

Description

Virtual class containing general methods for simulators based on regions of the chromosomes, like DNase-seq, ChIP-seq or Methyl-seq

Class to simulate RNA-seq data
Class to simulate transcription factor data
Class to simulate miRNA-seq
Class to simulate ChIP-seq data
Class to simulate DNase-seq data
Class to simulate Methyl-seq data.

Slots

- `locs` Vector containing the list of locations of the sites.
- `locsName` Type of the site to simulate, only for debug.
- `splitChar` Character symbol used to split identifiers in chr/start/end
- `nCpG` numeric. Number of CpG sites to simulate.
- `pSuccessDemethReg` numeric. Probability of success in non methylated region
- `errorMethReg` numeric. Error rate in methylated region
- `errorDemethReg` numeric. Error rate in methylated region
- `nReadsMethReg` numeric. Mean number of reads in methylated region.
- `nReadsDemethReg` numeric. Mean number of reads in non methylated regions.
- `phaseDiff` numeric. Phase difference in the differentially methylated regions between two samples.

- `noiseParams` Noise parameters to be used with noise function.
- `roundDigits` Number of digits to round the simulated count values.
- `minMaxQuantile` Numeric vector of length 2 indicating the quantiles to use in order to retrieve the absolute minimum and maximum value that a differentially expressed feature can have.
- `minMaxFC` Numeric vector of length 2 indicating the minimum and maximum fold-change that a differentially expressed feature can have.
- `minMaxDist` Named list containing different minimum and maximum constraints values calculated at the beginning of the simulation process.
- `replicateParams` Named list containing the parameters a and b to be used in the replicates generation process, see the vignette for more info.
omicData  

Set customized data for an omic.

**Description**

Set customized data for an omic.

**Usage**

```r
omicData(omic, data = NULL, associationList = NULL)
```

**Arguments**

- `omic`  
  The name of the omic to provide data.

- `data`  
  Data frame with the omic identifiers as row names and just one column named Counts containing numeric values used as initial sample for the simulation.

- `associationList`  
  Only for regulatory omics, a data frame with 2 columns, the first called containing the regulator ID and the second called Gene with the gene identifier.

**Value**

Initialized simulation object with the given data.

**Examples**

```r
# Take a subset of the included dataset for illustration purposes. We could also load it from a csv file or RData, # as long as we transform it to have 1 column named "Counts" # and the identifiers as row names.

data(sampleData)

custom_rnaseq <- head(sampleData$SimRNAseq$data, 100)
```
# In this case, 'custom_rnaseq' is a data frame with
# the structure:
head(custom_rnaseq)
## Counts
## ENSMUSG0000000001 6572
## ENSMUSG0000000003 0
## ENSMUSG0000000028 4644
## ENSMUSG0000000031 8
## ENSMUSG0000000037 0
## ENSMUSG0000000049 0

# The helper 'omicData' returns an object with our custom data.
rnaseq_customdata <- omicData("RNA-seq", data = custom_rnaseq)

omicResults

## omicResults

### omicResults

Retrieves the simulated data.

**Description**

Retrieves the simulated data.

**Usage**

omicResults(simulation, omics = NULL, format = "data.frame")

**Arguments**

- **simulation**: A MOSimulation object.
- **omics**: List of the omics to retrieve the simulated data.
- **format**: Type of object to use for returning the results

**Value**

A list containing an element for every omic specific, with the simulation data in the format indicated, or a numeric matrix with simulated data if the omic name is directly provided.

**Examples**

omic_list <- c("RNA-seq")
rnaseq_simulation <- mosim(omics = omic_list)
#' # This will be a data frame with RNA-seq counts
rnaseq_simulated <- omicResults(rnaseq_simulation, "RNA-seq")

# Group1.Time0.Rep1 Group1.Time0.Rep2 Group1.Time0.Rep3 ...
# ENSMUSG00000073155 4539 5374 5808 ...
# ENSMUSG000000261251 0 0 0 ...
omicSettings

Retrieves the settings used in a simulation

Description

Retrieves the settings used in a simulation

Usage

omicSettings(
    simulation, 
    omics = NULL, 
    association = FALSE, 
    reverse = FALSE, 
    only.linked = FALSE, 
    prefix = FALSE, 
    include.lagged = TRUE 
)

Arguments

simulation
omics
association
reverse
only.linked
prefix
include.lagged

A MOSimulation object.
List of omics to retrieve the settings.
A boolean indicating if the association must also be returned for the regulators.
A boolean, swap the column order in the association list in case we want to use the output directly and the program requires a different ordering.
Return only the interactions that have an effect.
Logical indicating if the name of the omic should prefix the name of the regulator.
Logical indicating if interactions with transitory profile and different minimum/maximum time point between gene and regulator should be included or not.

Value

A list containing a data frame with the settings used to simulate each of the indicated omics. If association is TRUE, it will be a list with 3 keys: 'associations', 'settings' and 'regulators', with the first two keys being a list containing the information for the selected omics and the last one a global data frame giving the merged information.
omicSim

Set the simulation settings for an omic.

Description
Set the simulation settings for an omic.

Usage
omicSim(omic, depth = NULL, totalFeatures = NULL, regulatorEffect = NULL)

Arguments
omic Name of the omic to set the settings.
depth Sequencing depth in millions of counts. If not provided will take the global parameter passed to mosim function.
totalFeatures Limit the number of features to simulate. By default include all present in the dataset.
regulatorEffect only for regulatory omics. Associative list containing the percentage of effects over the total number of regulator, including repressor, association and no effect (NE).

Value
A list with the appropriate structure to be given as options in mosim function.

Examples
omic_list <- c("RNA-seq", "miRNA-seq")
rnaseq_options <- c(omicSim("miRNA-seq", totalFeatures = 2500))

# The return value is an associative list compatible with
# 'omicsOptions'

rnaseq_simulation <- mosim(omics = omic_list,
plotProfile

Generate a plot of a feature’s profile for one or two omics.

Description
Generate a plot of a feature’s profile for one or two omics.

Usage
plotProfile(simulation, omics, featureIDS, drawReps = FALSE, groups = NULL)

Arguments
- simulation: A MOSimulation object
- omics: Character vector of the omics to simulate.
- featureIDS: List containing the feature to show per omic. Must have the omics as the list names and the features as values.
- drawReps: Logical to enable/disable the representation of the replicates inside the plot.
- groups: Character vector indicating the groups to plot in the form "GroupX" (i.e. Group1)

Value
A ggplot2 object.

Examples
omic_list <- c("RNA-seq", "miRNA-seq")

rnaseq_options <- c(omicSim("miRNA-seq", totalFeatures = 2500))
rnaseq_simulation <- mosim(omics = omic_list, omicsOptions = rnaseq_options)

plotProfile(rnaseq_simulation, omics = c("RNA-seq", "miRNA-seq"), featureIDS = list("RNA-seq"="ENSMUSG00000007682", "miRNA-seq"="mmu-miR-320-3p")
)
sampleData  

**Default data**

### Description

Dataset with base counts and id-gene tables.

### Usage

`sampleData`

### Format

An object of class `list` of length 6.

### Details

List with 6 elements:

- **SimRNAseq data** Dataframe with base counts with gene id as rownames.  
  - `geneLength` Length of every gene.
- **SimChIPseq data** Dataframe with base counts with regions as rownames.  
  - `idToGene` Dataframe with region as "ID" column and gene name on "Gene" column.
- **SimDNaseq data** Dataframe with base counts with regions as rownames.  
  - `idToGene` Dataframe with region as "ID" column and gene name on "Gene" column.
- **SimMiRNAseq data** Dataframe with base counts with miRNA id as rownames.  
  - `idToGene` Dataframe with miRNA as "ID" column and gene name on "Gene" column.
- **SimMethylseq idToGene** Dataframe with region as "ID" column and gene name on "Gene" column.
- **CpGisland** Dataframe of CpG to be used as initialization data, located on "Region" column.
Index

* datasets
  sampleData, 14
* internal
  MOSimulation-class, 6
  MOSimulator-class, 7
  MOSimulatorRegion-class, 8

discretize, 2

experimentalDesign, 3

is.declared, 4

mosim, 4, 5
MOSim-package, 2
MOSimulation-class, 6
MOSimulator-class, 7
MOSimulatorRegion-class, 8

omicData, 5, 9
omicResults, 10
omicSettings, 11
omicSim, 5, 12

plotProfile, 13

sampleData, 14
SimChIPseq-class
  (MOSimulatorRegion-class), 8
SimDNaseseq-class
  (MOSimulatorRegion-class), 8
SimMethylseq-class
  (MOSimulatorRegion-class), 8
SimmiRNAseq-class
  (MOSimulatorRegion-class), 8
SimRNAseq-class
  (MOSimulatorRegion-class), 8
SimTF-class (MOSimulatorRegion-class), 8