Package ‘RJMCMCNucleosomes’

April 9, 2019

Version 1.7.0  
Date 2015-05-01  
Title Bayesian hierarchical model for genome-wide nucleosome positioning with high-throughput short-read data (MNase-Seq)  
Description This package does nucleosome positioning using informative Multinomial-Dirichlet prior in a t-mixture with reversible jump estimation of nucleosome positions for genome-wide profiling.  
Depends R (>= 3.4), IRanges, GenomicRanges  
Imports Rcpp (>= 0.12.5), consensusSeeker, BiocGenerics, GenomeInfoDb, S4Vectors, BiocParallel, stats, graphics, methods, grDevices  
Suggests BiocStyle, knitr, rmarkdown, nucleoSim, RUnit  
LinkingTo Rcpp  
SystemRequirements Rcpp  
Encoding UTF-8  
License Artistic-2.0  
URL https://github.com/ArnaudDroitLab/RJMCMCNucleosomes  
BugReports https://github.com/ArnaudDroitLab/RJMCMCNucleosomes/issues  
VignetteBuilder knitr  
biocViews BiologicalQuestion, ChIPSeq, NucleosomePositioning, Software, StatisticalMethod, Bayesian, Sequencing, Coverage  
RoxygenNote 6.0.1  
git_url https://git.bioconductor.org/packages/RJMCMCNucleosomes  
git_branch master  
git_last_commit 19999a2  
git_last_commit_date 2018-10-30  
Date/Publication 2019-04-08  
Author Pascal Belleau [aut], Rawane Samb [aut], Astrid Deschênes [cre, aut], Khader Khadraoui [aut], Lajmi Lakhal-Chaieb [aut], Arnaud Droit [aut]  
Maintainer Astrid Deschênes <adeschen@hotmail.com>
R topics documented:

RJMCMCNucleosomes-package ........................................... 2
mergeAllRDSFilesFromDirectory ........................................ 3
mergeRDSFiles .......................................................... 4
plotNucleosomes ......................................................... 5
postTreatment ............................................................ 6
print.rjmcmcNucleosomes ................................................ 7
print.rjmcmcNucleosomesBeforeAndAfterPostTreatment ............ 8
print.rjmcmcNucleosomesMerge ......................................... 9
reads_demo_01 .......................................................... 10
reads_demo_02 .......................................................... 10
rjmcmc ................................................................. 11
rjmcmcCHR ............................................................ 13
RJMCMC_result .......................................................... 15
segmentation ............................................................ 16
syntheticNucleosomeReads ............................................... 17

Index 19

RJMCMCNucleosomes-package

RJMCMCNucleosomes: Bayesian hierarchical model for genome-
wide nucleosome positioning with high-throughput short-read data
(MNase-Seq)

Description

This package does nucleosome positioning using informative Multinomial-Dirichlet prior in a t-
mixture with reversible jump estimation of nucleosome positions for genome-wide profiling.

Author(s)

Pascal Belleau, Rawane Samb, Astrid Deschénes, Khader Khadraoui, Lajmi Lakhal and Arnaud Droit

Maintainer: Astrid Deschenes <adeschen@hotmail.com>

See Also

- rjmcmc for profiling of nucleosome positions for a segment
- rjmcmcCHR for profiling of nucleosome positions for a large region. The function will take
care of splitting and merging.
- segmentation for splitting a GRanges containing reads in a list of smaller segments for the
rjmcmc function.
- postTreatment for merging closely positioned nucleosomes
- mergeRDSFiles for merging nucleosome information from selected RDS files.
- plotNucleosomes for generating a graph containing the nucleosome positions and the read
coverage.
mergeAllRDSFilesFromDirectory

Merge nucleosome information from all RDS files present in a same directory. Beware that only nucleosome information from same chromosome should be merged together.

Description

Merge nucleosome information, from all RDS files present in a same directory, into one object of class "rjmcmcNucleosomesMerge".

Usage

mergeAllRDSFilesFromDirectory(directory)

Arguments

directory a character, the name of the directory (relative or absolute path) containing RDS files. The RDS files must contain R object of class "rjmcmcNucleosomes" or "rjmcmcNucleosomesMerge".

Value

a list of class "rjmcmcNucleosomesMerge" containing:

- k a integer, the number of nucleosomes.
- mu a GRanges containing the positions of the nucleosomes.

Author(s)

Pascal Belleau, Astrid Deschenes

Examples

```r
## Use a directory present in the RJMCMC package
directoryWithRDSFiles <- system.file("extdata", package = "RJMCMCNucleosomes")

## Merge nucleosomes info from RDS files present in directory
## It is assumed that all files present in the directory are nucleosomes
## result for the same chromosome
result <- mergeAllRDSFilesFromDirectory(directoryWithRDSFiles)

## Print the number and the position of the nucleosomes
result$k
result$mu

## Class of the output object
class(result)
```
mergeRDSFiles

Merge nucleosome information from selected RDS files.

Description

Merge nucleosome information present in RDS files into one object of class "rjmcmcNucleosomesMerge".

Usage

mergeRDSFiles(RDSFiles)

Arguments

RDSFiles a array, the names of all RDS used to merge nucleosome information. The files must contain R object of class "rjmcmcNucleosomes" or "rjmcmcNucleosomesMerge".

Value

a list of class "rjmcmcNucleosomesMerge" containing:

- k a integer, the number of nucleosomes.
- mu a GRanges containing the positions of the nucleosomes.

Author(s)

Pascal Belleau, Astrid Deschenes

Examples

## Use RDS files present in the RJMCMC package
RDSFiles <- dir(system.file("extdata", package = "RJMCMCNucleosomes"),
full.names = TRUE, pattern = "*.RDS")

## Merge nucleosomes info from RDS files present in directory
result <- mergeRDSFiles(RDSFiles)

## Print the number and the position of the nucleosomes
result$k
result$mu

## Class of the output object
class(result)
Generate a graph for a GRanges or a GRangesList of nucleosome positions. In presence of only one prediction (with multiples nucleosome positions), a GRanges is used. In presence of more than one predictions (as example, before and after post-treatment or results from different software), a GRangesList with one entry per prediction is used. All predictions must have been obtained using the same reads.

Usage

plotNucleosomes(nucleosomePositions, reads, seqName = NULL, xlab = "position", ylab = "coverage", names = NULL)

Arguments

nucleosomePositions
  a GRanges or a GRangesList containing the nucleosome positions for one or multiples predictions obtained using the same reads. In presence of only one prediction (with multiples nucleosome positions), a GRanges is used. In presence of more than one predictions (as example, before and after post-treatment or results from different software), a GRangesList with one entry per prediction is used.

reads
  a GRanges containing forward and reverse reads. The GRanges should contain at least one read.

seqName
  a character string containing the label of the chromosome, present in the GRanges object, that will be used. The NULL value is accepted when only one seqname is present in the GRanges; the only seqname present will be used. Default: NULL.

xlab
  a character string containing the label of the x-axis.

ylab
  a character string containing the label of the y-axis.

names
  a vector of a character string containing the label of each prediction set. The vector must be the same length of the nucleosomePositions list or 1 in presence of a vector. When NULL, the name of the elements of the list are used or the string "Nucleosome" for a vector are used. Default: NULL.

Value

a graph containing the nucleosome positions and the read coverage

Author(s)

Astrid Deschenes
postTreatment

Examples

```r
## Load reads dataset
data(reads_demo_01)

## Run RJMCMC method
result <- rjmcmc(reads = reads_demo_01,
                 seqName = "chr_SYNTHETIC",
                 nbrIterations = 4000, lambda = 2, kMax = 30,
                 minInterval = 146, maxInterval = 292, minReads = 5,
                 vSeed = 10213)

## Create graph using the synthetic map
plotNucleosomes(nucleosomePositions = result$mu, seqName = "chr_SYNTHETIC",
                 reads = reads_demo_01)
```

---

**postTreatment**

A post-treatment function to merge closely positioned nucleosomes, from the same chromosome, identified by the `rjmcmc` function.

**Description**

A helper function which merges closely positioned nucleosomes to rectify the over splitting and provide a more conservative approach. Beware that each chromosome must be treated separately.

**Usage**

```r
postTreatment(reads, seqName = NULL, resultRJMCMC, extendingSize = 74L,
             chrLength)
```

**Arguments**

- `reads` a GRanges containing forward and reverse reads. Beware that the start position of a reverse read is always higher than the end position.
- `seqName` a character string containing the label of the chromosome, present in the GRanges object, that will be used. The NULL value is accepted when only one seqname is present in the GRanges; the only seqname present will be used. Default: NULL.
- `resultRJMCMC` an object of class "rjmcmcNucleosomes" or "rjmcmcNucleosomesMerge", the information about nucleosome positioning for an entire chromosome or a region that must be treated as one unit.
- `extendingSize` a positive numeric or a positive integer indicating the size of the consensus region used to group closely positioned nucleosomes. The minimum size of the consensus region is equal to twice the value of the `extendingSize` parameter. The numeric will be treated as an integer. Default: 74.
- `chrLength` a positive numeric or a positive integer indicating the length of the current chromosome. The length of the chromosome is used to ensure that the consensus positions are all located inside the chromosome.
**Value**

a GRanges, the updated nucleosome positions. When no nucleosome is present, NULL is returned.

**Author(s)**

Pascal Belleau, Astrid Deschenes

**Examples**

```r
## Loading dataset
data(reads_demo_02)

## Nucleosome positioning, running both merge and split functions
result <- rjmcmc(reads = reads_demo_02,
     seqName = "chr_SYNTHETIC", nbrIterations = 1000,
     lambda = 2, kMax = 30, minInterval = 146,
     maxInterval = 490, minReads = 3, vSeed = 11)

## Before post-treatment
result

##Post-treatment function which merged closely positioned nucleosomes
postResult <- postTreatment(reads = reads_demo_02,
     seqName = "chr_SYNTHETIC", result, 100, 73500)

## After post-treatment
postResult
```

---

**print.rjmcmcNucleosomes**

*Formated output of predicted nucleosomes*

**Description**

Generated a formated output of a list marked as an `rjmcmcNucleosomes` class

**Usage**

```r
## S3 method for class 'rjmcmcNucleosomes'
print(x, ...)
```

**Arguments**

- **x**  
  the output object from `rjmcmc` function to be printed
- **...**  
  arguments passed to or from other methods

**Value**

An object of class `rjmcmcNucleosomes`
Author(s)
Astrid Deschenes

Examples

```r
## Loading dataset
data(RJMCMC_result)

print(RJMCMC_result)
```

---

**print.rjmcmcNucleosomesBeforeAndAfterPostTreatment**

*Formatted output of predicted nucleosomes*

**Description**

Generated a formatted output of a list marked as an `rjmcmcNucleosomesBeforeAndAfterPostTreatment` class.

**Usage**

```r
## S3 method for class 'rjmcmcNucleosomesBeforeAndAfterPostTreatment'
print(x, ...)
```

**Arguments**

- `x`: the output object from `rjmcmcCHR` function to be printed
- `...`: arguments passed to or from other methods

**Value**

An object of class `rjmcmcNucleosomesBeforeAndAfterPostTreatment`

Author(s)
Astrid Deschenes

Examples

```r
## Load synthetic dataset of reads
data(syntheticNucleosomeReads)

## Use dataset of reads to create GRanges object
sampleGRanges <- GRanges(syntheticNucleosomeReads$dataIP)

## Run nucleosome detection on the entire sample
## Not run: result <- rjmcmcCHR(reads = sampleGRanges, zeta = 147, delta=50, maxLength=1200, nbrIterations = 1000, lambda = 3, kMax = 30, minInterval = 146, maxInterval = 292, minReads = 5, vSeed = 10113, nbCores = 2, saveAsRDS = FALSE)
```
### Description
Generated a formatted output of a list marked as an `rjmcmcNucleosomesMerge` class

### Usage
```r
## S3 method for class 'rjmcmcNucleosomesMerge'
print(x, ...)
```

### Arguments
- `x` the output object from `mergeAllRDSFilesFromDirectory` function to be printed
- `...` arguments passed to or from other methods

### Value
an object of class `mergeAllRDSFilesFromDirectory`

### Author(s)
Astrid Deschenes

### Examples
```r
## Use a directory present in the RJMCMC package
directoryWithRDSFiles <- system.file("extdata", package = "RJMCMCNucleosomes")

## Merge nucleosomes info from RDS files present in directory
## It is assumed that all files present in the directory are nucleosomes
## result for the same chromosome
result <- mergeAllRDSFilesFromDirectory(directoryWithRDSFiles)

## Show resulting nucleosomes
print(result)

## or simply
result
```
reads_demo_01  
Forward reads and reverse reads in GRanges format (for demo purpose).

Description
A group of forward and reverse reads, in a GRanges, that can be used to test the rjmcmc function.

Usage
data(reads_demo_01)

Format
A GRanges containing forward and reverse reads.

Value
A GRanges containing forward and reverse reads.

See Also
• rjmcmc for profiling of nucleosome positions

Examples

## Loading dataset
data(reads_demo_01)

## Nucleosome positioning
rjmcmc(reads = reads_demo_01, nbrIterations = 100, lambda = 3, kMax = 30, 
minInterval = 146, maxInterval = 292, minReads = 5)

reads_demo_02  
Forward reads and reverse reads in GRanges format (for demo purpose).

Description
A group of forward and reverse reads that can be used to test the rjmcmc function.

Usage
data(reads_demo_02)

Format
A GRanges containing forward and reverse reads.
Value

A GRanges containing forward and reverse reads.

See Also

- `rjmcmc` for profiling of nucleosome positions
- `rjmcmcCHR` for profiling of nucleosome positions for a large region. The function will take care of splitting and merging.
- `segmentation` for splitting a GRanges containing reads in a list of smaller segments for the rjmcmc function.
- `postTreatment` for merging closely positioned nucleosomes
- `mergeRDSFiles` for merging nucleosome information from selected RDS files.
- `plotNucleosomes` for generating a graph containing the nucleosome positions and the read coverage.

Examples

```r
## Loading dataset
data(reads_demo_02)

## Nucleosome positioning
## Since there is only one chromosome present in reads_demo_02, the name
## of the chromosome does not need to be specified
rjmcmc(reads = reads_demo_02, nbrIterations = 150, lambda = 3, kMax = 30,
        minInterval = 144, maxInterval = 290, minReads = 6)
```

Description

Use of a fully Bayesian hierarchical model for chromosome-wide profiling of nucleosome positions based on high-throughput short-read data (MNase-Seq data). Beware that for a genome-wide profiling, each chromosome must be treated separately. This function is optimized to run on segments that are smaller sections of the chromosome.

Usage

```
rjmcmc(reads, seqName = NULL, nbrIterations, kMax, lambda = 3, minInterval,
        maxInterval, minReads = 5, adaptIterationsToReads = TRUE, vSeed = -1,
        saveAsRDS = FALSE)
```

Arguments

- `reads` a GRanges containing forward and reverse reads. Beware that the start position of a reverse read is always higher that the end position.
seqName: a character string containing the label of the chromosome, present in the GRanges object, that will be used. The NULL value is accepted when only one seqname is present in the GRanges; the only seqname present will be used. Default: NULL.

nbrIterations: a positive integer or numeric, the number of iterations. Non-integer values of nbrIterations will be casted to integer and truncated towards zero.

kMax: a positive integer or numeric, the maximum number of degrees of freedom per region. Non-integer values of kMax will be casted to integer and truncated towards zero.

lambda: a positive numeric, the theoretical mean of the Poisson distribution. Default: 3.

minInterval: a numeric, the minimum distance between two nucleosomes.

maxInterval: a numeric, the maximum distance between two nucleosomes.

minReads: a positive integer or numeric, the minimum number of reads in a potential candidate region. Non-integer values of minReads will be casted to integer and truncated towards zero. Default: 5.

adaptIterationsToReads: a logical indicating if the number of iterations must be modified in function of the number of reads. Default: TRUE.

vSeed: a integer. A seed used when reproducible results are needed. When a value inferior or equal to zero is given, a random integer is used. Default: -1.

saveAsRDS: a logical. When TRUE, a RDS file containing the complete output of the c++ rjmcmc() function is created. Default: FALSE.

Value:

a list of class "rjmcmcNucleosomes" containing:

- call the matched call.
- k a integer, the final estimation of the number of nucleosomes. 0 when no nucleosome is detected.
- mu a GRanges containing the positions of the nucleosomes and '*' as strand. The seqnames of the GRanges correspond to the seqName input value. NA when no nucleosome is detected.
- k_max a integer, the maximum number of nucleosomes obtained during the iteration process. NA when no nucleosome is detected.

Author(s)
Rawane Samb, Pascal Belleau, Astrid Deschenes

Examples

```r
## Loading dataset
data(reads_demo_01)

## Nucleosome positioning, running both merge and split functions
result <- rjmcmc(reads = reads_demo_01, seqName = "chr_SYNTHETIC",
nbrIterations = 1000, lambda = 2, kMax = 30,
minInterval = 146, maxInterval = 292, minReads = 5,
vSeed = 10113, saveAsRDS = FALSE)
```
rjmcmcCHR

## Print the final estimation of the number of nucleosomes
result$k

## Print the position of nucleosomes
result$mu

## Print the maximum number of nucleosomes obtained during the iteration
## process
result$k_max

### Description

Use of a fully Bayesian hierarchical model for chromosome-wide profiling of nucleosome positions based on high-throughput short-read data (MNase-Seq data). Beware that for a genome-wide profiling, each chromosome must be treated separately. This function is optimized to run on an entire chromosome.

The function will process by splitting the GRanges of reads (as example, the reads from a chromosome) in a list of smaller GRanges segments that can be run by the rjmcmc function. All those steps are done automatically.

### Usage

```
rjmcmcCHR(reads, seqName = NULL, zeta = 147, delta, maxLength, 
nbrIterations, kMax, lambda = 3, minInterval, maxInterval, minReads = 5, 
adaptIterationsToReads = TRUE, vSeed = -1, nbCores = 1, 
dirOut = "out", saveAsRDS = FALSE, saveSEG = TRUE)
```

### Arguments

- **reads**: a GRanges, the forward and reverse reads that need to be segmented.
- **seqName**: a character string containing the label of the chromosome, present in the GRanges object, that will be used. The NULL value is accepted when only one seqname is present in the GRanges; the only seqname present will be used. Default: NULL.
- **zeta**: a positive integer or numeric, the length of the nucleosomes. Default: 147.
- **delta**: a positive integer or numeric, the accepted range of overlapping section between segments. The overlapping section being zeta + delta.
- **maxLength**: a positive integer or numeric, the length of each segment.
- **nbrIterations**: a positive integer or numeric, the number of iterations. Non-integer values of nbrIterations will be casted to integer and truncated towards zero.
- **kMax**: a positive integer or numeric, the maximum number of degrees of freedom per region. Non-integer values of kMax will be casted to integer and truncated towards zero.
- **lambda**: a positive numeric, the theoretical mean of the Poisson distribution. Default: 3.
minInterval: a numeric, the minimum distance between two nucleosomes.
maxInterval: a numeric, the maximum distance between two nucleosomes.
minReads: a positive integer or numeric, the minimum number of reads in a potential candidate region. Non-integer values of minReads will be casted to integer and truncated towards zero. Default: 5.

adaptIterationsToReads: a logical indicating if the number of iterations must be modified in function of the number of reads. Default: TRUE.
vSeed: a integer. A seed used when reproducible results are needed. When a value inferior or equal to zero is given, a random integer is used. Default: -1.

nbCores: a positive integer, the number of cores used to run in parallel. Default: 1.

dirOut: a character string. The name of the directory where 2 directories are created (if they don’t already exists). The directory “dirOut/results” contains the rjmcmc results for each segment. The directory “dirOut/done” contains file a log file for each segment in RData format. If the log file for a segment is in the directory, the program considers that it has been processed and run the next segment. Default: “out”.

saveAsRDS: a logical. When TRUE, a RDS file containing the complete output of the rjmcmc function is created. Default: FALSE.

saveSEG: a logical. When TRUE, a RDS file containing the segments generated by segmentation function is saved in directory named from parameter dirOut. Default: FALSE.

Value:
a list of class “rjmcmcNucleosomesBeforeAndAfterPostTreatment” containing:
  • k: a integer, the number of nucleosomes.
  • mu: a GRanges containing the positions of the nucleosomes.
  • kPost: a integer, the number of nucleosomes after post-treatment and ‘*’ as strand. The seqnames of the GRanges correspond to the seqName input value. NA when no nucleosome is detected.
  • muPost: a GRanges containing the positions of the nucleosomes after post-treatment and ‘*’ as strand. The seqnames of the GRanges correspond to the seqName input value. NA when no nucleosome is detected.

Author(s)
Pascal Belleau, Astrid Deschenes

Examples

```r
## Load synthetic dataset of reads
data(syntheticNucleosomeReads)

## Use dataset of reads to create GRanges object
sampleGRanges <- GRanges(syntheticNucleosomeReads$dataIP)

## Run nucleosome detection on the entire sample
## Not run: result <- rjmcmcCHR(reads = sampleGRanges, zeta = 147, delta=50,
```
### Description

A list of class "rjmcNucleosomes" which contains the information about the detected nucleosomes.

### Usage

```r
data(RJMCMC_result)
```

### Format

A list of class "rjmcNucleosomes" containing:

- `call` the matched call.
- `k` a integer, the final estimation of the number of nucleosomes. 0 when no nucleosome is detected.
- `mu` a vector of numeric of length k, the positions of the nucleosomes. NA when no nucleosome is detected.
- `k_max` a integer, the maximum number of nucleosomes obtained during the iteration process. NA when no nucleosome is detected.

### Value

A list of class "rjmcNucleosomes" containing:

- `call` the matched call.
- `k` a integer, the final estimation of the number of nucleosomes. 0 when no nucleosome is detected.
- `mu` a vector of numeric of length k, the positions of the nucleosomes. NA when no nucleosome is detected.
- `k_max` a integer, the maximum number of nucleosomes obtained during the iteration process. NA when no nucleosome is detected.

### See Also

- `rjmc` for profiling of nucleosome positions
- `rjmcCHR` for profiling of nucleosome positions for a large region. The function will take care of splitting and merging.
- `segmentation` for splitting a GRanges containing reads in a list of smaller segments for the `rjmc` function.
• **postTreatment** for merging closely positioned nucleosomes

• **mergeRDSFiles** for merging nucleosome information from selected RDS files.

• **plotNucleosomes** for generating a graph containing the nucleosome positions and the read coverage.

**Examples**

```r
## Loading dataset
data(RJMCMC_result)
data(reads_demo_02)

## Results before post-treatment
RJMCMC_result$mu

## Post-treatment function which merged closely positioned nucleosomes
postResult <- postTreatment(reads = reads_demo_02,
                           extendingSize = 60, chrLength = 100000, resultRJMCMC = RJMCMC_result)

## Results after post-treatment
postResult
```

---

### segmentation

Split a GRanges containing reads in a list of smaller segments for the rjmcmc function.

**Description**

Split a GRanges of reads (as example, the reads from a chromosome) in a list of smaller GRanges so that the rjmcmc function can be run on each segments.

**Usage**

```r
segmentation(_reads, _zeta = 147, _delta, _maxLength)
```

**Arguments**

- **reads**
  - a GRanges, the reads that need to be segmented.

- **_zeta**
  - a positive integer or numeric, the length of the nucleosomes. Default: 147.

- **_delta**
  - a positive integer or numeric, the accepted range of overlapping section between segments. The overlapping section being _zeta + _delta.

- **_maxLength**
  - a positive integer or numeric, the length of each segment.

**Value**

a GRangesList containing all the segments.

**Author(s)**

Pascal Belleau, Astrid Deschenes
syntheticNucleosomeReads

Examples

```r
## Load synthetic dataset of reads
data(syntheticNucleosomeReads)

## Use dataset of reads to create GRanges object
sampleGRanges <- GRanges(seqnames = syntheticNucleosomeReads$dataIP$chr,
                          ranges = IRanges(start = syntheticNucleosomeReads$dataIP$start,
                                          end = syntheticNucleosomeReads$dataIP$end),
                          strand = syntheticNucleosomeReads$dataIP$strand)

# Segmentation of the reads
segmentation(reads = sampleGRanges, zeta = 147, delta = 50,
             maxLength = 1000)
```

---

**syntheticNucleosomeReads**

*Simulated dataset of reads generated by nucleoSim package (for demo purpose).*

---

Description

A list of class "syntheticNucReads" which contains the information about synthetic reads related to nucleosomes. The dataset has been created using a total of 300 well-positioned nucleosomes, 30 fuzzy nucleosomes with variance of reads following a Normal distribution.

Usage

```r
data(syntheticNucleosomeReads)
```

Format

A list containing:

- `call` the called that generated the dataset.
- `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.
Value

A list containing:

- call the called that generated the dataset.
- 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.
Index

*Topic datasets
  reads_demo_01, 10
  reads_demo_02, 10
  RJMCMC_result, 15
  syntheticNucleosomeReads, 17
*Topic package
  RJMCMCNucleosomes-package, 2
mergeAllRDSFilesFromDirectory, 3
mergeRDSFiles, 2, 4, 11, 16
plotNucleosomes, 2, 5, 11, 16
postTreatment, 2, 6, 11, 16
print.rjmcmcNucleosomes, 7
print.rjmcmcNucleosomesBeforeAndAfterPostTreatment, 8
print.rjmcmcNucleosomesMerge, 9
reads_demo_01, 10
reads_demo_02, 10
rjmc, 2, 6, 10, 11, 11, 15
RJMCMC_result, 15
rjmcmcCHR, 2, 11, 13, 15
RJMCMCNucleosomes
  (RJMCMCNucleosomes-package), 2
RJMCMCNucleosomes-package, 2
segmentation, 2, 11, 14, 15, 16
syntheticNucleosomeReads, 17