Package 'ChIPSeqSpike'

October 16, 2020

Type Package

Title ChIP-Seq data scaling according to spike-in control

Version 1.8.0

Date 2018-04-11

Author Nicolas Descostes

Maintainer Nicolas Descostes <nicolas.descostes@gmail.com>

Description Chromatin Immuno-Precipitation followed by Sequencing (ChIP-Seq) is used to determine the binding sites of any protein of interest, such as transcription factors or histones with or without a specific modification, at a genome scale. The many steps of the protocol can introduce biases that make ChIP-Seq more qualitative than quantitative. For instance, it was shown that global histone modification differences are not caught by traditional downstream data normalization techniques. A case study reported no differences in histone H3 lysine-27 trimethyl (H3K27me3) upon Ezh2 inhibitor treatment. To tackle this problem, external spike-in control were used to keep track of technical biases between conditions. Exogenous DNA from a different non-closely related species was inserted during the protocol to infer scaling factors that enabled an accurate normalization, thus revealing the inhibitor effect.
ChIPSeqSpike offers tools for ChIP-Seq spike-in normalization. Ready to use scaled bigwig files and scaling factors values are obtained as output. ChIPSeqSpike also provides tools for ChIP-Seq spike-in assessment and analysis through a versatile collection of graphical functions.

License Artistic-2.0

Imports tools, stringr, Rsamtools, GenomicRanges, IRanges, seqplots, ggplot2, LSD, corrplot, methods, stats, grDevices, graphics, utils, BiocGenerics, S4Vectors

VignetteBuilder knitr

Suggests BiocStyle, knitr, rmarkdown, testthat

Depends R (>= 3.5), rtracklayer (>= 1.37.6)

biocViews ImmunoOncology, ChIPSeq, Sequencing, Normalization, Transcription, Coverage, DifferentialMethylation, Epigenetics, DataImport, HistoneModification

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

git_branch RELEASE_3_11

git_last_commit be9b9b0

git_last_commit_date 2020-04-27
Date/Publication 2020-10-16

R topics documented:

`averageBindingValues` *Set the binding values used to perform profiles and heatmaps*

Description

Set a plotSetArray object to the plotSetArrayList slot of the given object.

Usage

```
averageBindingValues(theObject) <- value


## S4 replacement method for signature 'ChIPSeqSpikeDataset'
averageBindingValues(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeDatasetBoost'
averageBindingValues(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeCore'
averageBindingValues(theObject) <- value
```

Arguments

`theObject`	A `ChIPSeqSpikeDataset`, `ChIPSeqSpikeDatasetBoost` or `ChIPSeqSpikeCore` object
`value`	A `PlotSetArray` object

Value

The modified object is returned

Author(s)

Nicolas Descostes

See Also

`plotProfile` `plotTransform` `plotHeatmaps` `PlotSetArray-class`

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
```

```
                              "input_0-filtered.bw",
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")

## Copying example files
dir.create("./test_chipseqspike")
result <- file.copy(bigwig_files, "test_chipseqspike")

if(.Platform$OS.type != 'windows') {
    csds <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
                      genome_name, verbose = TRUE,
                      outputFolder = output_folder)
    test <- getAverageBindingValues(csds[[1]])
    averageBindingValues(csds[[1]]) <- test
}

unlink("test_chipseqspike/", recursive = TRUE)
```

`bigWigFile` *Set the path to the bigwig file*

Description

Set the path to the input or experiment bigwig file.

Usage

```
bigWigFile(theObject) <- value


## S4 replacement method for signature 'ChIPSeqSpikeDataset'
bigWigFile(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeDatasetBoost'
bigWigFile(theObject) <- value

## S4 replacement method for signature 'Experiment'
bigWigFile(theObject) <- value

## S4 replacement method for signature 'ExperimentLoaded'
bigWigFile(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeCore'
bigWigFile(theObject) <- value
```

Arguments

`theObject`	A `ChIPSeqSpikeCore`, `ChIPSeqSpikeDataset`, `ChIPSeqSpikeDatasetBoost`, `Experiment` or `ExperimentLoaded` object
`value`	A string representing a path to a bigwig file

Details

If the object is ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost, bigWigFile will set the path to the input DNA experiment bigwig file corresponding to all experiments defined by the object.

If the object is Experiment or ExperimentLoaded, bigWigFile will set the path to the experiment bigwig file.

Value

The modified object is returned

Author(s)

Nicolas Descostes

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
                              "input_0-filtered.bw",
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")

## Copying example files
dir.create("./test_chipseqspike")
result <- file.copy(bigwig_files, "test_chipseqspike")

if(.Platform$OS.type != 'windows') {
    csds <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
                      genome_name, verbose = TRUE,
                      outputFolder = output_folder)
    write("toto", file="./test_chipseqspike/toto.bw")
    getBigWigFile(csds[[1]][[1]])
    bigWigFile(csds[[1]][[1]]) <- "test_chipseqspike/toto.bw"
    getBigWigFile(csds[[1]][[1]])
}

unlink("test_chipseqspike/", recursive = TRUE)
```

---

boxplotSpike *plot boxplots of ChIP-seq experiments*

Description

plot boxplots of the mean values of ChIP-seq experiments on the annotations given to the extractBinding method

Usage

```
boxplotSpike(theObject, col = NULL, rawFile = FALSE,
             rpmFile = FALSE, bgsubFile = FALSE, revFile = FALSE,
             spiked = TRUE, ylab = NULL, outline = TRUE, violinPlot = FALSE,
             notch = TRUE, mean_with_sd = FALSE, mean = FALSE,
             median = FALSE, boxplot = FALSE, jitter = FALSE, plot = TRUE,
             verbose = FALSE)

## S4 method for signature 'ChIPSeqSpikeDataset'
boxplotSpike(theObject, col = NULL, rawFile =
FALSE, rpmFile = FALSE, bgsubFile = FALSE, revFile = FALSE, spiked = TRUE, ylab
= NULL, outline = TRUE, violinPlot = FALSE, notch = TRUE, mean_with_sd = FALSE,
 mean = FALSE, median = FALSE, boxplot = FALSE, jitter = FALSE, plot = TRUE,
verbose = FALSE)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
boxplotSpike(theObject, col = NULL,
rawFile = FALSE, rpmFile = FALSE, bgsubFile = FALSE, revFile = FALSE,
spiked = TRUE, ylab = NULL, outline = TRUE, violinPlot = FALSE, notch = TRUE,
mean_with_sd = FALSE, mean = FALSE, median = FALSE, boxplot = FALSE,
jitter = FALSE, plot = TRUE, verbose = FALSE)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
boxplotSpike(theObject, col = NULL, rawFile
= FALSE, rpmFile = FALSE, bgsubFile = FALSE, revFile = FALSE, spiked = TRUE,
ylab = NULL, outline = TRUE, violinPlot = FALSE, notch = TRUE, mean_with_sd =
FALSE, mean = FALSE, median = FALSE, boxplot = FALSE, jitter = FALSE,
plot = TRUE, verbose = FALSE)

## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
boxplotSpike(theObject, col = NULL,
rawFile = FALSE, rpmFile = FALSE, bgsubFile = FALSE, revFile = FALSE,
spiked = TRUE, ylab = NULL, outline = TRUE, violinPlot = FALSE, notch = TRUE,
mean_with_sd = FALSE, mean = FALSE, median = FALSE, boxplot = FALSE,
jitter = FALSE, plot = TRUE, verbose = FALSE)
```

Arguments

theObject	ChIPSeqSpike dataset (see ?spikeDataset)
col	Vector of colors for each experiment. Default is NULL.
rawFile	If TRUE, use the untransformed data. Not available in boost mode. Default is FALSE. (see details)
rpmFile	If TRUE, use the RPM scaled data. Not available in boost mode. Default is FALSE. (see details)
bgsubFile	If TRUE, use the input subtracted data. Not available in boost mode. Default is FALSE. (see details)
revFile	If TRUE, use the RPM reverted data. Not available in boost mode. Default is FALSE. (see details)
spiked	If TRUE, use the spiked data. Default is TRUE. (see details)
ylab	Character string of the name of the y-axis. Default is NULL.

outline	Logical indicating if outliers are shown. Default is TRUE.
violinPlot	Logical indicating if a violin plot representation is used. Default is FALSE.
notch	Logical indicating if confidence intervals are shown. Default is TRUE.
mean_with_sd	Logical indicating if the mean and standard deviation are shown on the violin plot. Default is FALSE.
mean	Logical indicating if the mean is shown on the violin plot. Default is FALSE.
median	Logical indicating if the median is shown on the violin plot. Default is FALSE.
boxplot	Logical indicating if boxplot is shown on the violin plot. Default is FALSE
jitter	Logical indicating if each mean values is represented as a point on the violin plot. Default is FALSE.
plot	Logical indicating if the boxplot should be plotted. Default is TRUE.
verbose	Logical indicating if processing messages are shown. Default is FALSE.

Details

The 'rawFile', 'rpmFile', 'bgsubFile', 'revFile', 'spiked' parameters indicate if the untransformed, RPM scaled, input DNA subtracted, RPM reversed or spiked data should be plotted. This option is only available if not in boost mode (see ?spikePipe for details on what these steps are).

Value

Return a list with the components 'stats', 'n', 'conf', 'out', 'group' and 'names'. See ?boxplot for details.

Methods (by class)

- `ChIPSeqSpikeDataset`: Method for signature theObject= 'ChIPSeqSpikeDataset'
- `ChIPSeqSpikeDatasetBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- `ChIPSeqSpikeDatasetList`: Method for signature theObject= 'ChIPSeqSpikeDatasetList'
- `ChIPSeqSpikeDatasetListBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

See Also

`spikeDataset` `spikePipe` `plotProfile` `plotTransform` `plotHeatmaps` `plotCor`

Examples

```
data("result_extractBinding")
boxplotSpike(csds)
boxplotSpike(csds, outline = FALSE, violinPlot = TRUE)
boxplotSpike(csds, outline = FALSE, violinPlot = TRUE, mean = TRUE, jitter =
TRUE)
boxplotSpike(csds, rawFile = TRUE, rpmFile = TRUE, bgsubFile = TRUE, revFile =
TRUE, spiked = TRUE, outline = FALSE, violinPlot = TRUE)
```

`ChIPSeqSpikeCore-class`
ChIPSeqSpikeCore Reference Class

Description

Main class containing input DNA file paths, scores and binding values. The classes ChIPSeqSpikeDataset and ChIPSeqSpikeDatasetBoost inherit from this class. This class is not used directly in the package but is part of ChIPSeqSpikeDataset and ChIPSeqSpikeDatasetBoost. The constructor should not be used directly.

Fields

inputBam: File path to the input control BAM file
inputBigWig: File path to the input control BigWig file
inputScalingFactor: Input control scaling factor
inputCount: Input control reads count
plotSetArrayList: List of average binding values
matBindingValList: List of binding values matrices

Constructor

ChIPSeqSpikeCore(inputBamFile, inputBigWigFile, inputSF = 0, inputNb = 0, SetArrayList = list(), matBindingList = list())

Arguments

inputBamFile String representing the file path to the input control BAM file.
inputBigWigFile String representing the file path to the input control bigWig file. (see details)
inputSF Numeric scaling factor. Default is 0. (see details)
inputNb Numeric read counts. Default is 0. (see details)
SetArrayList List of PlotSetArray objects. Default is an empty list. (see details)
matBindingList List of binding value matrices. Default is an empty list. (see details)

Getters

`getBam` Returns the input BAM path
`getBigWigFile` Returns the input bigWig path
`getScalingFactor` Returns the input scaling factor
`getCount` Returns the number of reads contained in the input BAM file
`getAverageBindingValues` Returns a list of PlotSetArray objects. (see details)
`getMatBindingValues` Returns a list of matrices containing binding values. (see details)

Setters

`scalingFactor` Modifies the input scaling factor value
`count` Modifies the input count value
`bigWigFile` Modifies the input bigWig file path
`averageBindingValues` Modifies the PlotSetArray list. (see details)
`matBindingValues` Modifies the list of binding value matrices. (see details)

Details

'inputSF' is the scaling factor that will be applied to the input bigWigFile before input subtraction of the different experiments. 'inputNb' which holds the number of aligned reads is used to calculate the aforementioned factor. Only the endogenous count and factor are needed for the input conversely to the experiments for which both the endogenous/exogenous scaling factors and counts are needed to perform spike-in normalization.

'SetArrayList' contains PlotSetArray objects. The PlotSetArray class is defined in the Bioconductor package 'seqplots' and holds the values that are necessary to plot profiles and heatmaps. These values can be retrieved with the 'getAverageBindingValues' function.

'matBindingList' contains matrices of binding values for each experiment. These values are used to generate boxplots and correlations plots. They are retrieved by calling the function 'BWGFile_summary' of the bioconductor package 'rtracklayer'.

Author(s)

Nicolas Descostes

See Also

`Experiment-class` `ExperimentLoaded-class` `ChIPSeqSpikeDatasetBoost-class` `ChIPSeqSpikeDataset-class` `spikeDataset` `PlotSetArray-class`

ChIPSeqSpikeDataset-class

ChIPSeqSpikeDataset Reference Class

Description

Main class containing file paths, scores and values for spike normalization. It inherits from ChIPSeqSpikeCore.

Fields

experimentList: A list of Experiment objects

Constructor

ChIPSeqSpikeDataset(endogenousBam_vec, exogenousBam_vec, bigWigFile_endogenous_vec, inputBigWigFile, inputBamFile, expnames, inputSF = 0, inputNb = 0, SetArrayList = list(), matBindingList = list())

Arguments

endogenousBam_vec Character vector of file paths to the BAM files aligned to the reference genome.

exogenousBam_vec Character vector of file paths to the BAM files aligned to the exogenous genome.

bigWigFile_endogenous_vec Character vector of file paths to the bigWig files aligned to the reference genome.

inputBigWigFile String representing the file path to the input control bigWig file. (see details)

inputBamFile String representing the file path to the input control BAM file.

expnames Character vector of experiment names. (see details)

inputSF Numeric scaling factor. Default is 0. (see details)

inputNb Numeric read counts. Default is 0. (see details)

SetArrayList List of PlotSetArray objects. Default is an empty list. (see details)

matBindingList List of binding value matrices. Default is an empty list. (see details)

Getters

`getBam` Returns the input BAM path

`getBigWigFile` Returns the input bigWig path

`getExperimentListBigWigs` Returns a character vector of paths to the experiment bigWig files

`getExpName` Returns a character vector of experiment names

`getScalingFactor` Returns the input scaling factor

`getCount` Returns the number of reads contained in the input BAM file

`getAverageBindingValues` Returns a list of PlotSetArray objects. (see details)

`getMatBindingValues` Returns a list of matrices containing binding values. (see details)

`x[[i]]` Get the Experiment object at index i

Setters

`scalingFactor` Modifies the input scaling factor value

`count` Modifies the input count value

`bigWigFile` Modifies the input bigWig file path

`averageBindingValues` Modifies the PlotSetArray list. (see details)

`matBindingValues` Modifies the list of binding value matrices. (see details)

`x[[i]] <- value` Set `value` to ChIPSeqSpikeDataset i

Details

'expnames' character vector is used to define the names of the experiment list and are used as labels in plotting, summary and getRatio functions.

'inputSF' is the scaling factor that will be applied to the input bigWigFile before input subtraction of the different experiments. 'inputNb' which holds the number of aligned reads is used to calculate the aforementioned factor. Only the endogenous count and factor are needed for the input conversely to the experiments for which both the endogenous/exogenous scaling factors and counts are needed to perform spike-in normalization.

'SetArrayList' contains PlotSetArray objects. The PlotSetArray class is defined in the Bioconductor package 'seqplots' and holds the values that are necessary to plot profiles and heatmaps. These values can be retrieved with the 'getAverageBindingValues' function.

'matBindingList' contains matrices of binding values for each experiment. These values are used to generate boxplots and correlations plots. They are retrieved by calling the function 'BWGFile_summary' of the bioconductor package 'rtracklayer'.

If the dataset contains more than one input, one would want to use the ChIPSeqSpikeDatasetList class. Boost mode classes (ChIPSeqSpikeDatasetBoost and ChIPSeqSpikeDatasetListBoost) can also be considered to speed up the analysis.

Author(s)

Nicolas Descostes

See Also

Experiment-class ChIPSeqSpikeCore-class ChIPSeqSpikeDatasetBoost-class ChIPSeqSpikeDatasetList-cla
spikeDataset PlotSetArray-class spikeSummary getRatio

Examples

```
file_vec <- system.file("extdata",
                        c("bam_files/H3K79me2_0_dm3-filtered.bam",
                        "bam_files/H3K79me2_0_hg19-filtered.bam",
                        "bigwig_files/H3K79me2_0-filtered.bw",
                        "bigwig_files/input_0-filtered.bw",
                        "bam_files/input_0_hg19-filtered.bam"),
                        package="ChIPSeqSpike")

csds <- ChIPSeqSpikeDataset(endogenousBam_vec = file_vec[2],
                         exogenousBam_vec = file_vec[1],
                         bigWigFile_endogenous_vec = file_vec[3],
                         inputBigWigFile = file_vec[4],
                         inputBamFile = file_vec[5],
                         expnames = "H3K79me2_0")
csds
```

ChIPSeqSpikeDatasetBoost-class

ChIPSeqSpikeDatasetBoost Reference Class

Description

Boost version of ChIPSeqSpikeDataset class keeping data in GRanges form. It inherits from ChIPSeqSpikeCore.

Fields

experimentListLoaded: A list of ExperimentLoaded-class objects

inputBigWigLoaded: A GRanges object of input binding scores

Constructor

ChIPSeqSpikeDatasetBoost(endogenousBam_vec, exogenousBam_vec, bigWigFile_endogenous_vec, inputBigWigFile, inputBamFile, expnames, inputSF = 0, inputNb = 0, SetArrayList = list(), matBindingList = list(), verbose = TRUE)

Arguments

endogenousBam_vec Character vector of file paths to the BAM files aligned to the reference genome.

exogenousBam_vec Character vector of file paths to the BAM files aligned to the exogenous genome.

bigWigFile_endogenous_vec Character vector of file paths to the bigWig files aligned to the reference genome.

inputBigWigFile String representing the file path to the input control bigWig file. (see details)

inputBamFile String representing the file path to the input control BAM file.

expnames Character vector of experiment names. (see details)

inputSF Numeric scaling factor. Default is 0. (see details)

inputNb Numeric read counts. Default is 0. (see details)

SetArrayList List of PlotSetArray objects. Default is an empty list. (see details)

matBindingList List of binding value matrices. Default is an empty list. (see details)

Getters

`getBam` Returns the input BAM path

`getBigWigFile` Returns the input bigWig path

`getExperimentListBigWigs` Returns a character vector of paths to the experiment bigWig files

`getExpName` Returns a character vector of experiment names

`getScalingFactor` Returns the input scaling factor

`getCount` Returns the number of reads contained in the input BAM file

`getAverageBindingValues` Returns a list of PlotSetArray objects. (see details)

`getMatBindingValues` Returns a list of matrices containing binding values. (see details)

`getLoadedData` Returns the GRanges object of input DNA binding scores

Setters

`scalingFactor` Modifies the input scaling factor value

`count` Modifies the input count value

`bigWigFile` Modifies the input bigWig file path

`averageBindingValues` Modifies the PlotSetArray list. (see details)

`matBindingValues` Modifies the list of binding value matrices. (see details)

Details

'expnames' character vector is used to define the names of the experiment list and are used as labels in plotting, summary and getRatio functions.

'inputSF' is the scaling factor that will be applied to the input bigWigFile before input subtraction of the different experiments. 'inputNb' which holds the number of aligned reads is used to calculate the aforementioned factor. Only the endogenous count and factor are needed for the input conversely to the experiments for which both the endogenous/exogenous scaling factors and counts are needed to perform spike-in normalization.

'SetArrayList' contains PlotSetArray objects. The PlotSetArray class is defined in the Bioconductor package 'seqplots' and holds the values that are necessary to plot profiles and heatmaps. These values can be retrieved with the 'getAverageBindingValues' function.

'matBindingList' contains matrices of binding values for each experiment. These values are used to generate boxplots and correlations plots. They are retrieved by calling the function 'BWGFile_summary' of the bioconductor package 'rtracklayer'.

If the dataset contains more than one input, one would want to use the ChIPSeqSpikeDatasetList class. Boost mode classes (ChIPSeqSpikeDatasetBoost and ChIPSeqSpikeDatasetListBoost) can also be considered to speed up the analysis.

'exportBigWigs' output the binding values from the GRanges objects contained in inputBigWigLoaded and experimentListLoaded slots.

On Windows operating system, due to the Bioconductor package rtracklayer >= 1.37.6 not supporting bigWig files, this class is not available.

Author(s)

Nicolas Descostes

See Also

ExperimentLoaded-class ChIPSeqSpikeDataset-class ChIPSeqSpikeCore-class ChIPSeqSpikeDatasetListBoo
spikeDataset PlotSetArray-class spikeSummary getRatio

Examples

```
file_vec <- system.file("extdata",
                        c("bam_files/H3K79me2_0_dm3-filtered.bam",
                        "bam_files/H3K79me2_0_hg19-filtered.bam",
                        "bigwig_files/H3K79me2_0-filtered.bw",
                        "bigwig_files/input_0-filtered.bw",
                        "bam_files/input_0_hg19-filtered.bam"),
                        package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {
    csds <- ChIPSeqSpikeDatasetBoost(endogenousBam_vec = file_vec[2],
                                exogenousBam_vec = file_vec[1],
                                bigWigFile_endogenous_vec = file_vec[3],
                                inputBigWigFile = file_vec[4],
                                inputBamFile = file_vec[5],
                                expnames = "H3K79me2_0")
    csds
}
```

ChIPSeqSpikeDatasetList-class

ChIPSeqSpikeDatasetList Reference Class

Description

Class containing a list of ChIPSeqSpikeDataset objects

Fields

datasetList: A list of ChIPSeqSpikeDataset objects

Constructor

ChIPSeqSpikeDatasetList(dataset_list, verbose)

Arguments

dataset_list A properly formatted list of information needed to create the object (see details).

verbose Indicate if processing messages should be output.

Getters

`getBigWigFile` Returns a vector of all bigWig paths corresponding to all files specified in info.csv

`x[[i]]` Get the ChIPSeqSpikeDataset object at index i

Setters

`x[[i]] <- value` Set `value` to ChIPSeqSpikeDatasetList i

Details

This class enables to process datasets containing different input files. It will creates a list of ChIPSeqSpikeDataset objects, each of them containing a different input.

The above indicated constructor should not be used directly. One would rather create the object by calling the "meta-constructor" spikeDataset function which takes a info.csv file as input. spikeDataset function formats properly the different information into a dataset_list which is submitted to the ChIPSeqSpikeDatasetList constructor.

Author(s)

Nicolas Descostes

See Also

`ChIPSeqSpikeDataset-class` `spikeDataset` `spikeSummary` `getRatio`

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file(c("extdata/bam_files"), package="ChIPSeqSpike")
bigwig_path <- system.file(c("extdata/bigwig_files"), package="ChIPSeqSpike")

csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
                     bigWigPath = bigwig_path)

is(csds)
csds
```

ChIPSeqSpikeDatasetListBoost-class

ChIPSeqSpikeDatasetListBoost Reference Class

Description

Class containing a list of ChIPSeqSpikeDatasetBoost objects

Fields

datasetList: A list of ChIPSeqSpikeDatasetBoost objects

Constructor

ChIPSeqSpikeDatasetListBoost(dataset_list, verbose)

Arguments

dataset_list A properly formatted list of information needed to create the object (see details).

verbose Indicate if processing messages should be output.

Getters

`exportBigWigs` Output all bigwig files corresponding to the previously performed transformations. The list of bigwig files is given in info.csv (see details).

`x[[i]]` Get the ChIPSeqSpikeDatasetBoost object at index i

Details

This class enables to process datasets containing different input files in boost mode. It will creates a list of ChIPSeqSpikeDatasetBoost objects, each of them containing a different input.

The above indicated constructor should not be used directly. One would rather create the object by calling the "meta-constructor" spikeDataset function, which takes a info.csv file as input, setting boost = TRUE. spikeDataset function formats properly the different information into a dataset_list which is submitted to the ChIPSeqSpikeDatasetListBoost constructor.

Author(s)

Nicolas Descostes

See Also

`ChIPSeqSpikeDatasetBoost-class` `spikeDataset` `spikeSummary` `getRatio`

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file(c("extdata/bam_files"), package="ChIPSeqSpike")
bigwig_path <- system.file(c("extdata/bigwig_files"), package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {
    csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
                         bigWigPath = bigwig_path, boost = TRUE)
    is(csds)
    csds
}
```

count *Set the number of reads associated to an experiment*

Description

Set the number of endogenous reads associated to an experiment. This is used to compute scaling factors.

Usage

```
count(theObject) <- value


## S4 replacement method for signature 'ChIPSeqSpikeDataset'
count(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeDatasetBoost'
count(theObject) <- value

## S4 replacement method for signature 'Experiment'
count(theObject) <- value

## S4 replacement method for signature 'ExperimentLoaded'
count(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeCore'
count(theObject) <- value
```

Arguments

theObject	A ChIPSeqSpikeDataset, ChIPSeqSpikeDatasetBoost, Experiment, ExperimentLoaded or ChIPSeqSpikeCore object
value	A numeric representing the number of mapped reads

Details

If the object is ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost, count will set the number of input DNA mapped reads to the endogenous reference genome.

If the object is Experiment or ExperimentLoaded, count will set the number of experiment mapped reads to the endogenous reference genome.

Value

The modified object is returned

Author(s)

Nicolas Descostes

See Also

exoCount estimateScalingFactors

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
                              "input_0-filtered.bw",
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")


if(.Platform$OS.type != 'windows') {
    ## Copying example files
    dir.create("./test_chipseqspike")
    result <- file.copy(bigwig_files, "test_chipseqspike")

    csds <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
    genome_name, verbose = TRUE, outputFolder = output_folder)
    getCount(csds[[1]][[1]])
    count(csds[[1]][[1]]) <- 10
    getCount(csds[[1]][[1]])
    unlink("test_chipseqspike/", recursive = TRUE)
}
```

`datasetList` *Set the list of dataset representing experiments*

Description

Set the list of ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost objects.

Usage

```
datasetList(theObject) <- value


## S4 replacement method for signature 'ChIPSeqSpikeDatasetList'
datasetList(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeDatasetListBoost'
datasetList(theObject) <- value
```

Arguments

`theObject`	A ChIPSeqSpikeDatasetList or ChIPSeqSpikeDatasetListBoost object
`value`	A list of ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost objects

Value

The modified object is returned

Author(s)

Nicolas Descostes

See Also

getDatasetList

Examples

```
data(result_extractBinding)
datasetList(csds) <- getDatasetList(csds)
```

`estimateScalingFactors` *Compute scaling factors to perform spike-in normalization*

Description

Compute scaling factors for endogenous and exogenous experiment from 'Experiment', 'ExperimentLoaded', 'ChIPSeqSpikeDataset', 'ChIPSeqSpikeDatasetBoost', 'ChIPSeqSpikeDatasetList', and 'ChIPSeqSpikeDatasetListBoost'

Usage

```
estimateScalingFactors(theObject, paired = FALSE, verbose = TRUE)

## S4 method for signature 'Experiment'
estimateScalingFactors(theObject, paired = FALSE,
verbose = TRUE)

## S4 method for signature 'ExperimentLoaded'
estimateScalingFactors(theObject, paired = FALSE,
verbose = TRUE)

## S4 method for signature 'ChIPSeqSpikeDataset'
estimateScalingFactors(theObject,
paired = FALSE, verbose = TRUE)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
estimateScalingFactors(theObject,
paired = FALSE, verbose = TRUE)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
estimateScalingFactors(theObject,
paired = FALSE, verbose = TRUE)
```

```
## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
estimateScalingFactors(theObject,
paired = FALSE, verbose = TRUE)
```

Arguments

theObject	ChIPSeqSpike dataset (see ?spikeDataset)
paired	Indicate if sequences are single- or paired-ended. Default is FALSE
verbose	If False, do not output processing messages. Default is TRUE

Details

Estimating scaling factors is the first step to perform on a dataset. A scaling factor is defined as:

- 1/(bam_count/1000000)

bam_count being the number of reads aligned to the genome. The count is determined for the endogenous and exogenous experiments.

Scaling factors will the be applied to the bigwig files in the following steps of the procedure. After estimating scaling factors, RPM normalization should be performed.

Value

Return an object of the same class of the input object containing computed scaling factors.

Methods (by class)

- Experiment: Method for signature theObject='Experiment'
- ExperimentLoaded: Method for signature theObject='ExperimentLoaded '
- ChIPSeqSpikeDataset: Method for signature theObject= 'ChIPSeqSpikeDataset'
- ChIPSeqSpikeDatasetBoost: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- ChIPSeqSpikeDatasetList: Method for signature theObject= 'ChIPSeqSpikeDatasetList'
- ChIPSeqSpikeDatasetListBoost: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

See Also

spikeSummary scaling spikePipe

Examples

```
## Mock example on a restricted number of reads
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file(c("extdata/bam_files"), package="ChIPSeqSpike")
bigwig_path <- system.file(c("extdata/bigwig_files"), package="ChIPSeqSpike")

csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
                     bigWigPath = bigwig_path)
```

```
csds <- estimateScalingFactors(csds)
```

---

exoCount *Set the number of reads associated to an experiment*

Description

Set the number of exogenous reads associated to an experiment. This is used to compute scaling factors.

Usage

```
exoCount(theObject) <- value


## S4 replacement method for signature 'Experiment'
exoCount(theObject) <- value

## S4 replacement method for signature 'ExperimentLoaded'
exoCount(theObject) <- value
```

Arguments

theObject	An Experiment or ExperimentLoaded object
value	A numeric representing the number of mapped reads to the exogenous reference genome

Value

The modified object is returned

Author(s)

Nicolas Descostes

See Also

count estimateScalingFactors

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
                              "input_0-filtered.bw",
```

```
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {
    ## Copying example files
    dir.create("./test_chipseqspike")
    result <- file.copy(bigwig_files, "test_chipseqspike")

    csds <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
                      genome_name, verbose = TRUE,
                      outputFolder = output_folder)
    getExoCount(csds[[1]][[1]])
    exoCount(csds[[1]][[1]]) <- 5
    getExoCount(csds[[1]][[1]])
    unlink("test_chipseqspike/", recursive = TRUE)
}
```

`exogenousScalingFactor` *Set the exogenous scaling factor associated to an experiment*

Description

Set the exogenous scaling factor associated to an experiment

Usage

```
exogenousScalingFactor(theObject) <- value


## S4 replacement method for signature 'Experiment'
exogenousScalingFactor(theObject) <- value

## S4 replacement method for signature 'ExperimentLoaded'
exogenousScalingFactor(theObject) <- value
```

Arguments

`theObject`	An `Experiment` or `ExperimentLoaded` object
`value`	A numeric representing the exogenous scaling factor

Details

A scaling factor is defined as:

- 1/(bam_count/1000000)

Value

The modified object is returned

Author(s)

Nicolas Descostes

See Also

scalingFactor estimateScalingFactors

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
                              "input_0-filtered.bw",
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {

    ## Copying example files
    dir.create("./test_chipseqspike")
    result <- file.copy(bigwig_files, "test_chipseqspike")

    csds <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
                      genome_name, verbose = TRUE,
                      outputFolder = output_folder)
    getExogenousScalingFactor(csds[[1]][[1]])
    exogenousScalingFactor(csds[[1]][[1]]) <- 5
    getExogenousScalingFactor(csds[[1]][[1]])
    unlink("test_chipseqspike/", recursive = TRUE)
}
```

Experiment-class *Experiment Reference Class*

Description

Object containing information about experiment. It constitutes an element of a list held by ChIPSeqSpikeDataset class objects

Fields

endogenousBam: File path to the experiment BAM file aligned to the reference genome

exogenousBam: File path to the experiment BAM file aligned to the exogenous genome

bigWigFile: File path to the experiment BIGWIG file aligned to the reference genome

expName: Experiment name

endogenousScalingFactor: Experiment scaling factor

exogenousScalingFactor: Scaling factor obtained from exogenous DNA

endoCount: Number of reads contained in the endogenous BAM file

exoCount: Number of reads contained in the exogenous BAM file

Constructor

Experiment(endogenousBamFilePath, exogenousBamFilePath, bigWigFilePath, name, endoScalingFactor = 0, exoScalingFactor = 0, endoNb = 0, exoNb = 0)

Arguments

endogenousBamFilePath Character vector of path to experiment BAM file aligned to the reference genome.

exogenousBamFilePath Character vector of path to experiment BAM file aligned to the exogenous genome.

bigWigFilePath Character vector of path to experiment bigWig file aligned to the reference genome.

name Character vector of the experiment name.

endoScalingFactor Numeric scaling factor computed from the number of reads aligned to the reference genome. Default is 0.

exoScalingFactor Numeric scaling factor computed from the number of reads aligned to the exogenous genome. Default is 0.

endoNb Number of reads aligned to the reference genome. Default is 0.

endoNb Number of reads aligned to the exogenous genome. Default is 0.

Getters

`getBam` Returns the endogenous BAM path

`getExogenousBam` Returns the exogenous BAM path

`getBigWigFile` Returns the endogenous bigWig path

`getExpName` Returns a character vector of the experiment name

`getScalingFactor` Returns the endogenous scaling factor

`getExogenousScalingFactor` Returns the exogenous scaling factor

`getCount` Returns the number of reads aligned to the reference genome

`getExoCount` Returns the number of reads aligned to the exogenous genome

Setters

`scalingFactor` Modifies the endogenous scaling factor value

`exogenousScalingFactor` Modifies the exogenous scaling factor value

`count` Modifies the endogenous count value

`exoCount` Modifies the exogenous count value

`bigWigFile` Modifies the endogenous bigWig file path

`x[[i]] <- value` Set `value` to experiment i

Author(s)

Nicolas Descostes

See Also

`ExperimentLoaded-class` `ChIPSeqSpikeDataset-class` `spikeSummary` `getRatio`

Examples

```
file_vec <- system.file("extdata",
                        c("bam_files/H3K79me2_0_dm3-filtered.bam",
                        "bam_files/H3K79me2_0_hg19-filtered.bam",
                        "bigwig_files/H3K79me2_0-filtered.bw"),
                        package="ChIPSeqSpike")

exp <- Experiment(endogenousBamFilePath = file_vec[2],
                  exogenousBamFilePath = file_vec[1],
                  bigWigFilePath = file_vec[3],
                  name = "H3K79me2_0")
```

---

experimentList *Set the list of Experiment and ExperimentLoaded objects*

Description

Set the list of Experiment and ExperimentLoaded objects to the ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost object

Usage

```
experimentList(theObject) <- value


## S4 replacement method for signature 'ChIPSeqSpikeDataset'
experimentList(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeDatasetBoost'
experimentList(theObject) <- value
```

Arguments

theObject	A ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost object
value	A list of Experiment and ExperimentLoaded objects

Value

The modified object is returned

Author(s)

Nicolas Descostes

See Also

getExperimentList

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
                              "input_0-filtered.bw",
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {

    ## Copying example files
    dir.create("./test_chipseqspike")
    result <- file.copy(bigwig_files, "test_chipseqspike")

    csds <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
                      genome_name, verbose = TRUE,
                      outputFolder = output_folder)

    experimentList(csds[[1]]) <- getExperimentList(csds[[1]])

    unlink("test_chipseqspike/", recursive = TRUE)
}
```

ExperimentLoaded-class

ExperimentLoaded Reference Class

Description

Class inheriting from Experiment and containing loaded binding values

Fields

loadedBigWigFile: GRanges object representing binding values
endogenousBam: File path to the experiment BAM file aligned to the reference genome
exogenousBam: File path to the experiment BAM file aligned to the exogenous genome
bigWigFile: File path to the experiment BIGWIG file aligned to the reference genome
expName: Experiment name
endogenousScalingFactor: Experiment scaling factor
exogenousScalingFactor: Scaling factor obtained from exogenous DNA
endoCount: Number of reads contained in the endogenous BAM file
exoCount: Number of reads contained in the exogenous BAM file

Constructor

ExperimentLoaded(endogenousBamFilePath, exogenousBamFilePath, bigWigFilePath, name, endoScalingFactor = 0, exoScalingFactor = 0, endoNb = 0, exoNb = 0, verbose = TRUE)

Arguments

endogenousBamFilePath Character vector of path to experiment BAM file aligned to the reference genome.

exogenousBamFilePath Character vector of path to experiment BAM file aligned to the exogenous genome.

bigWigFilePath Character vector of path to experiment bigWig file aligned to the reference genome.

name Character vector of the experiment name.

endoScalingFactor Numeric scaling factor computed from the number of reads aligned to the reference genome. Default is 0.

exoScalingFactor Numeric scaling factor computed from the number of reads aligned to the exogenous genome. Default is 0.

endoNb Number of reads aligned to the reference genome. Default is 0.

exoNb Number of reads aligned to the exogenous genome. Default is 0.

verbose Indicate if processing messages should be output. Default is TRUE.

Getters

`getBam` Returns the endogenous BAM path

`getExogenousBam` Returns the exogenous BAM path

`getBigWigFile` Returns the endogenous bigWig path

`getExpName` Returns a character vector of the experiment name

`getScalingFactor` Returns the endogenous scaling factor

`getExogenousScalingFactor` Returns the exogenous scaling factor

`getCount` Returns the number of reads aligned to the reference genome

`getExoCount` Returns the number of reads aligned to the exogenous genome

`getLoadedData` Returns the GRanges object of binding values

Setters

`bigWigFile` Modifies the endogenous bigWig file path

`loadedData` Modifies the GRanges object of binding values

Details

On Windows operating system, due to the Bioconductor package rtracklayer >= 1.37.6 not supporting bigWig files, this class is not available.

Author(s)

Nicolas Descostes

See Also

`Experiment-class` `ChIPSeqSpikeDatasetBoost-class` `spikeSummary` `getRatio`

Examples

```
file_vec <- system.file("extdata",
                        c("bam_files/H3K79me2_0_dm3-filtered.bam",
                        "bam_files/H3K79me2_0_hg19-filtered.bam",
                        "bigwig_files/H3K79me2_0-filtered.bw"),
                        package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {
    exp <- ExperimentLoaded(endogenousBamFilePath = file_vec[2],
                            exogenousBamFilePath = file_vec[1],
                            bigWigFilePath = file_vec[3],
                            name = "H3K79me2_0")
}
```

`exportBigWigs` *Export bigwig files from values contained in a boost mode object*

Description

In boost mode, binding values of experiments are stored in the form of GRanges tables in the object. exportBigWigs output these values giving a proper bigwig file name corresponding to the transformations already performed.

Usage

```
exportBigWigs(theObject, verbose = TRUE)


## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
exportBigWigs(theObject, verbose = TRUE)

## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
exportBigWigs(theObject,
verbose = TRUE)
```

Arguments

`theObject`	`ChIPSeqSpike` dataset in boost mode (see ?spikeDataset)
`verbose`	If FALSE, do not output processing messages. Default is TRUE

Value

Output bigwig files of binding values.

The suffix of the bigwig file reflects the transformation steps performed on the object. If all steps were performed, the file name will be of the form: 'expName-RPM-BGSub-reverse-spiked.bw'. The suffixes 'RPM', 'BGSub', 'reverse' and 'spiked' stands for RPM scaling, input subtraction, RPM scaling reversal and exogenous scaling respectively.

Methods (by class)

- `ChIPSeqSpikeDatasetBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- `ChIPSeqSpikeDatasetListBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

Examples

```
## Mock example on a restricted number of reads
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file(c("extdata/bam_files"), package="ChIPSeqSpike")
bigwig_path <- system.file(c("extdata/bigwig_files"), package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {
    csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
                         bigWigPath = bigwig_path, boost = TRUE)

    ## Creating test folder
    dir.create("./test_chipseqspike")

    csds <- estimateScalingFactors(csds)

    ## Apply RPM scaling
    csds <- scaling(csds, outputFolder = "test_chipseqspike")

    ## output RPM scaled files
    exportBigWigs(csds)

    ## Apply input subtraction
    csds <- inputSubtraction(csds)

    ## output input subtracted files
    exportBigWigs(csds)

    ## Delete all files generated in this example
    unlink("test_chipseqspike/", recursive = TRUE)
}
```

`extractBinding` *Extract binding values for graphical representations*

Description

Extracts and formats binding scores for each experiment into structures adapted to performing different graphical representations.

Usage

```
extractBinding(theObject, gff_vec, genome, binsize = 50, before = 2000,
               after=2000, mean_or_median = "mean", interpolation_number = 100,
               interpolation_average = 10000, ignore_strand = FALSE,
               verbose = FALSE)

## S4 method for signature 'ChIPSeqSpikeDataset'
extractBinding(theObject, gff_vec, genome,
                binsize = 50, before = 2000, after=2000,
                mean_or_median = "mean", interpolation_number = 100,
                interpolation_average = 10000, ignore_strand = FALSE,
                verbose = FALSE)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
extractBinding(theObject, gff_vec, genome,
                binsize = 50, before = 2000, after=2000,
                mean_or_median = "mean", interpolation_number = 100,
                interpolation_average = 10000, ignore_strand = FALSE,
                verbose = FALSE)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
extractBinding(theObject, gff_vec, genome,
                binsize = 50, before = 2000, after=2000,
                mean_or_median = "mean", interpolation_number = 100,
                interpolation_average = 10000, ignore_strand = FALSE,
                verbose = FALSE)

## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
extractBinding(theObject, gff_vec,
                genome, binsize = 50, before = 2000, after=2000,
                mean_or_median = "mean", interpolation_number = 100,
                interpolation_average = 10000, ignore_strand = FALSE,
                verbose = FALSE)
```

Arguments

theObject	ChIPSeqSpike dataset (see ?spikeDataset)
gff_vec	File in GFF format containing annotations used to plot information
genome	The UCSC code of reference genome, e.g. 'hg19' for Homo sapiens (see details)
binsize	Binning size used to create bigwig files. Default is 50.
before	Length in bp of the interval upstream annotation. Default is 2000.
after	Length in bp of the interval downstream annotation. Default is 2000.
mean_or_median	For average profiles, should the 'mean' or 'median' values be used. Default is 'mean'.
interpolation_number	Number of interpolated points to create matrices (see details). Default is 100.
interpolation_average	Number of interpolated points of profiles and heatmaps (see details). Default is 10000.

ignore_strand	If TRUE, the directionality is ignored, that is all features strands, regardless of annotation in GFF file, are treated as undetermined ("*"). Default is FALSE.
verbose	If TRUE, output processing messages. Default is FALSE.

Details

This method should be called before performing any graphical analysis. It updates two slots of theObject:

- SetArrayList: Contains the binding values to perform meta-profile (see ?plotProfile); transformation profiles if not in boost mode (see ?plotTransform) and heatmaps (see ?plotHeatmaps). These values are stored in a plotSetArray object. This object is created by the method getPlotSetArray of the 'seqplots' package.
- matBindingList: Contains list of matrices for each experiment. Each row correspond to an annotation given by gff_vec and the number of columns is defined by the interpolation_number parameter. These matrices are used to perform boxplots (see ?boxplotSpike) and correlation plots (see ?plotCor).

The SetArrayList contains values for 4 kind of representations (profiles and heatmaps): Representation at the start of the annotation (-before/ +after parameters); at the midpoint of the annotation; at the end of the annotation (-before/+after) or at the entire annotation (-before/+after). For representations using the entire annotations and upstream (before)/ downstream intervals, the number of points used for the within annotation interpolation is defined by the interpolation_average parameter.

For details on installing reference genomes, see details of the function 'getPlotSetArray' of the 'seqplots' package.

Value

Returns the same object with binding values in the form of plotSetArray and matrices (see details).

Methods (by class)

- ChIPSeqSpikeDataset: Method for signature theObject= 'ChIPSeqSpikeDataset'
- ChIPSeqSpikeDatasetBoost: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- ChIPSeqSpikeDatasetList: Method for signature theObject= 'ChIPSeqSpikeDatasetList'
- ChIPSeqSpikeDatasetListBoost: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

See Also

spikeDataset plotProfile plotTransform plotHeatmaps boxplotSpike plotCor getPlotSetArray

Examples

```
## Mock example on a restricted number of reads
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file(c("extdata/bam_files"), package="ChIPSeqSpike")
bigwig_path <- system.file(c("extdata/bigwig_files"), package="ChIPSeqSpike")
```

```
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"

if(.Platform$OS.type != 'windows') {
    csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
                         bigWigPath = bigwig_path)

    ## Copying test files to the current folder
    originalBW_vec <- as.character(getBigWigFile(csds))
    dir.create("./test_chipseqspike")
    result <- file.copy(originalBW_vec, "test_chipseqspike")

    csds <- estimateScalingFactors(csds)

    ## Apply RPM scaling
    csds <- scaling(csds, outputFolder = "test_chipseqspike")

    ## Perform input subtraction
    csds <- inputSubtraction(csds)

    ## Reverse RPM scaling after input subtraction
    csds <- scaling(csds, reverse = TRUE)

    ## Apply exogenous scaling factors
    csds <- scaling(csds, type = "exo")

    ## Extract binding values
    csds <- extractBinding(csds, gff_vec, genome_name)

    ## Delete all files generated in this example
    unlink("test_chipseqspike/", recursive = TRUE)
}
```

`getAverageBindingValues` *Get the average binding values associated to a dataset*

Description

Accessor returning the average binding values associated to a dataset.

Usage

```
getAverageBindingValues(theObject)


## S4 method for signature 'ChIPSeqSpikeDataset'
getAverageBindingValues(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getAverageBindingValues(theObject)

## S4 method for signature 'ChIPSeqSpikeCore'
getAverageBindingValues(theObject)
```

Arguments

theObject	A ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost object.

Details

Each element of the list contains the binding values to perform meta-profile (see ?plotProfile); transformation profiles if not in boost mode (see ?plotTransform) and heatmaps (see ?plotHeatmaps). These values are stored in a plotSetArray object. This object is created by the method getPlotSetArray of the 'seqplots' package.

Value

A list of plotSetArray objects.

Author(s)

Nicolas Descostes

See Also

plotProfile plotTransform plotHeatmaps getMatBindingValues getPlotSetArray

Examples

```
data(result_extractBinding)
getAverageBindingValues(csds[[1]])
```

---

getBam *Get the path to an endogenous experiment bam file*

Description

Access and returns the path to the bam file containing the reads of an experiment aligned to the endogenous reference genome.

Usage

```
getBam(theObject)


## S4 method for signature 'ChIPSeqSpikeDataset'
getBam(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getBam(theObject)

## S4 method for signature 'Experiment'
getBam(theObject)

## S4 method for signature 'ExperimentLoaded'
getBam(theObject)
```

```
## S4 method for signature 'ChIPSeqSpikeCore'
getBam(theObject)
```

Arguments

theObject	A ChIPSeqSpikeCore, ChIPSeqSpikeDataset, ChIPSeqSpikeDatasetBoost, Experiment or ExperimentLoaded object

Details

If the object is ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost, getBam returns the path to the input DNA experiment bam file containing reads aligned to the endogenous reference genome.

If the object is Experiment or ExperimentLoaded, getBam returns the path to the experiment bam file containing reads aligned to the endogenous reference genome.

Value

A string of the path to the endogenous bam file

Author(s)

Nicolas Descostes

See Also

getExogenousBam

Examples

```
data(result_extractBinding)
getBam(csds[[1]])
```

---

getBigWigFile *Get the path to an endogenous experiment bigwig file*

Description

Access and returns the path to the endogenous bigwig file of an experiment

Usage

```
getBigWigFile(theObject)


## S4 method for signature 'ChIPSeqSpikeDataset'
getBigWigFile(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
getBigWigFile(theObject)
```

```
## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getBigWigFile(theObject)

## S4 method for signature 'Experiment'
getBigWigFile(theObject)

## S4 method for signature 'ExperimentLoaded'
getBigWigFile(theObject)

## S4 method for signature 'ChIPSeqSpikeCore'
getBigWigFile(theObject)
```

Arguments

theObject	A ChIPSeqSpikeCore, ChIPSeqSpikeDataset, ChIPSeqSpikeDatasetList, ChIPSeqSpikeDatasetBoost, Experiment or ExperimentLoaded object

Details

If the object is ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost, getBigWigFile returns the path to the endogenous input DNA experiment bigwig file.

If the object is Experiment or ExperimentLoaded, getBigWigFile returns the path to the endogenous experiment bigwig file.

If the object is ChIPSeqSpikeDatasetList, all bigwig files associated with all dataset are returned at once.

Value

A string of the path to the endogenous bigwig file

Author(s)

Nicolas Descostes

Examples

```
data(result_extractBinding)
getBigWigFile(csds)
```

---

getCount *Get the number of reads aligned to the endogenous reference genome*

Description

Access and returns the number of reads of an experiment or input DNA experiment that were aligned to the endogenous reference genome.

Usage

```
getCount(theObject)


## S4 method for signature 'ChIPSeqSpikeDataset'
getCount(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getCount(theObject)

## S4 method for signature 'Experiment'
getCount(theObject)

## S4 method for signature 'ExperimentLoaded'
getCount(theObject)

## S4 method for signature 'ChIPSeqSpikeCore'
getCount(theObject)
```

Arguments

theObject	A ChIPSeqSpikeCore, ChIPSeqSpikeDataset, ChIPSeqSpikeDatasetBoost, Experiment or ExperimentLoaded object

Details

If the object is ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost, getCount returns the number of reads of the input DNA experiment that were aligned to the endogenous reference genome.

If the object is Experiment or ExperimentLoaded, getCount returns the number of reads of the experiment that were aligned to the endogenous reference genome.

Value

A numeric of the number of reads aligned to the endogenous reference genome

Author(s)

Nicolas Descostes

See Also

getExoCount

Examples

```
data(result_extractBinding)
getCount(csds[[1]])
```

`getDatasetList` *Get the list of ChIPSeqSpike objects*

Description

Access and returns the list of ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost objects contained in the structure

Usage

```
getDatasetList(theObject)


## S4 method for signature 'ChIPSeqSpikeDatasetList'
getDatasetList(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
getDatasetList(theObject)
```

Arguments

`theObject`	A `ChIPSeqSpikeDatasetList` or `ChIPSeqSpikeDataseListBoost` object

Value

A list of `ChIPSeqSpikeDataset` or `ChIPSeqSpikeDatasetBoost` objects

Author(s)

Nicolas Descostes

See Also

`datasetList`

Examples

```
data(result_extractBinding)
result <- getDatasetList(csds)
is(result)
is(result[[1]])
```

getExoCount *Get the number of reads aligned to the exogenous reference genome*

Description

Access and returns the number of reads of an experiment that were aligned to the exogenous reference genome.

Usage

```
getExoCount(theObject)


## S4 method for signature 'Experiment'
getExoCount(theObject)

## S4 method for signature 'ExperimentLoaded'
getExoCount(theObject)
```

Arguments

theObject	An Experiment or ExperimentLoaded object

Value

A numeric of the number of reads aligned to the exogenous reference genome

Author(s)

Nicolas Descostes

See Also

getCount

Examples

```
data(result_extractBinding)
getExoCount(csds[[1]][[1]])
```

getExogenousBam *Get the path to an exoogenous experiment bam file*

Description

Accesses and returns the path to the bam file containing the reads of an experiment aligned to the exogenous reference genome.

Usage

```
getExogenousBam(theObject)


## S4 method for signature 'Experiment'
getExogenousBam(theObject)

## S4 method for signature 'ExperimentLoaded'
getExogenousBam(theObject)
```

Arguments

theObject	An Experiment or ExperimentLoaded object

Value

A string of the path to the exogenous bam file

Author(s)

Nicolas Descostes

See Also

getBam

Examples

```
data(result_extractBinding)
getExogenousBam(csds[[1]][[1]])
```

---

getExogenousScalingFactor

Get the exogenous scaling factor

Description

Accesses and returns the experiment exogenous scaling factor.

Usage

```
getExogenousScalingFactor(theObject)


## S4 method for signature 'Experiment'
getExogenousScalingFactor(theObject)

## S4 method for signature 'ExperimentLoaded'
getExogenousScalingFactor(theObject)
```

Arguments

theObject	An Experiment or ExperimentLoaded object

Value

A numeric of the exogenous scaling factor

Author(s)

Nicolas Descostes

See Also

getScalingFactor

Examples

```
data(result_extractBinding)
getExogenousScalingFactor(csds[[1]][[1]])
```

---

getExperimentList *Get all Experiment or ExperimentLoaded objects associated with a dataset*

Description

Accesses and returns all Experiment or ExperimentLoaded objects associated with a dataset.

Usage

```
getExperimentList(theObject)


## S4 method for signature 'ChIPSeqSpikeDataset'
getExperimentList(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getExperimentList(theObject)
```

Arguments

theObject	A ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost object

Value

A list of Experiment or ExperimentLoaded objects

Author(s)

Nicolas Descostes

See Also

experimentList

Examples

```
data(result_extractBinding)
getExperimentList(csds[[1]])
```

`getExperimentListBigWigs` *Get all paths to the bigwig files associated with a dataset*

Description

Accesses and returns all paths to the bigwig files associated with a dataset.

Usage

```
getExperimentListBigWigs(theObject)


## S4 method for signature 'ChIPSeqSpikeDataset'
getExperimentListBigWigs(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getExperimentListBigWigs(theObject)
```

Arguments

`theObject`	A `ChIPSeqSpikeDataset` or `ChIPSeqSpikeDatasetBoost` object

Value

A character vector of all paths to the bigwig files associated with a dataset

Author(s)

Nicolas Descostes

Examples

```
data(result_extractBinding)
getExperimentListBigWigs(csds[[1]])
```

getExpName *Get the experiment name*

Description

Accesses and returns the experiment names associated with a dataset.

Usage

```
getExpName(theObject)


## S4 method for signature 'ChIPSeqSpikeDataset'
getExpName(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getExpName(theObject)

## S4 method for signature 'Experiment'
getExpName(theObject)

## S4 method for signature 'ExperimentLoaded'
getExpName(theObject)
```

Arguments

theObject	A ChIPSeqSpikeDataset, ChIPSeqSpikeDatasetBoost, Experiment or ExperimentLoaded object

Value

A string or character vector of the names of the experiments

Author(s)

Nicolas Descostes

Examples

```
data(result_extractBinding)
getExpName(csds[[1]])
```

getLoadedData *Get the endogenous reference genome binding scores of an experiment*

Description

Accesses and returns the binding scores of an experiment or input DNA experiment on the endogenous reference genome. (only available in boost mode).

Usage

```
getLoadedData(theObject)


## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getLoadedData(theObject)

## S4 method for signature 'ExperimentLoaded'
getLoadedData(theObject)
```

Arguments

theObject	A ChIPSeqSpikeDatasetBoost or ExperimentLoaded object

Details

If the object is ChIPSeqSpikeDatasetBoost, getLoadedData returns a GRanges object of binding scores of the input DNA experiment.

If the object is ExperimentLoaded, getLoadedData returns a GRanges object of binding scores of the experiment.

Value

A GRanges object of binding scores

Author(s)

Nicolas Descostes

Examples

```
file_vec <- system.file("extdata",
                        c("bam_files/H3K79me2_0_dm3-filtered.bam",
                        "bam_files/H3K79me2_0_hg19-filtered.bam",
                        "bigwig_files/H3K79me2_0-filtered.bw",
                        "bigwig_files/input_0-filtered.bw",
                        "bam_files/input_0_hg19-filtered.bam"),
                        package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {
    csds <- ChIPSeqSpikeDatasetBoost(endogenousBam_vec = file_vec[2],
                             exogenousBam_vec = file_vec[1],
                             bigWigFile_endogenous_vec = file_vec[3],
```

```
                            inputBigWigFile = file_vec[4],
                            inputBamFile = file_vec[5],
                            expnames = "H3K79me2_0")
    getLoadedData(csds)
}
```

---

getMatBindingValues *Get the list of matrices of binding scores*

Description

Accesses and returns the list of binding scores matrices of all experiments associated with a dataset. These matrices are used to plot boxplots and correlation plots.

Usage

```
getMatBindingValues(theObject)


## S4 method for signature 'ChIPSeqSpikeDataset'
getMatBindingValues(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getMatBindingValues(theObject)

## S4 method for signature 'ChIPSeqSpikeCore'
getMatBindingValues(theObject)
```

Arguments

theObject	A ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost object.

Value

A list of matrices containing binding scores

Author(s)

Nicolas Descostes

See Also

getAverageBindingValues boxplotSpike plotCor

Examples

```
data(result_extractBinding)
getMatBindingValues(csds[[1]])
```

`getRatio` *Output the percentage of exogenous DNA compared to that of endogenous DNA*

Description

Output the percentage of exogenous DNA compared to that of endogenous DNA

Usage

```
    getRatio(theObject)

## S4 method for signature 'Experiment'
getRatio(theObject)
## S4 method for signature 'ExperimentLoaded'
getRatio(theObject)
## S4 method for signature 'ChIPSeqSpikeDataset'
getRatio(theObject)
## S4 method for signature 'ChIPSeqSpikeDatasetList'
getRatio(theObject)
## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getRatio(theObject)
## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
getRatio(theObject)
```

Arguments

`theObject`	ChIPSeqSpike dataset (see ?spikeDataset)

Details

The rows represent, for each experiment of the dataset, the percentage of exogenous DNA defined as the number of aligned exogenous reads compared to the total number of reads.

The method 'getRatio' will throw a warning if the percentage of exogenous DNA represents less than 2 percent or more than 25 percent of the endogenous DNA. Less than 2 percent of exogenous DNA does not guarantee a proper scaling. Large amount of exogenous DNA should not impact the scaling procedure but is worth notifying to the user[1].

Value

A numeric matrix

Methods (by class)

- `Experiment`: Method for signature theObject = 'Experiment'
- `ExperimentLoaded`: Method for signature theObject = 'ExperimentLoaded'
- `ChIPSeqSpikeDataset`: Method for signature theObject= 'ChIPSeqSpikeDataset'
- `ChIPSeqSpikeDatasetBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- `ChIPSeqSpikeDatasetList`: Method for signature theObject= 'ChIPSeqSpikeDatasetList'

- `ChIPSeqSpikeDatasetListBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

References

[1] Orlando et al, "Quantitative ChIP-Seq normalization reveals global modulation of the epigenome", Cell Rep, 2014.

See Also

`spikeSummary`

Examples

```
## Mock example on files samples
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file(c("extdata/bam_files"), package="ChIPSeqSpike")
bigwig_path <- system.file(c("extdata/bigwig_files"), package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"

csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
                     bigWigPath = bigwig_path)
csds <- estimateScalingFactors(csds)
getRatio(csds)

## Results on the complete files
data("ratio")
print(ratio)
```

`getScalingFactor` *Get the endogenous scaling factor*

Description

Accesses and returns the experiment or input DNA experiment endogenous scaling factor.

Usage

```
getScalingFactor(theObject)


## S4 method for signature 'ChIPSeqSpikeDataset'
getScalingFactor(theObject)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
getScalingFactor(theObject)

## S4 method for signature 'Experiment'
```

```
getScalingFactor(theObject)

## S4 method for signature 'ExperimentLoaded'
getScalingFactor(theObject)

## S4 method for signature 'ChIPSeqSpikeCore'
getScalingFactor(theObject)
```

Arguments

theObject	A ChIPSeqSpikeCore, ChIPSeqSpikeDataset, ChIPSeqSpikeDatasetBoost, Experiment or ExperimentLoaded object

Details

If the object is ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost, getScalingFactor returns the input DNA experiment endogenous scaling factor.

If the object is Experiment or ExperimentLoaded, getScalingFactor returns the experiment endogenous scaling factor.

Value

A numeric of the endogenous scaling factor

Author(s)

Nicolas Descostes

See Also

getExogenousScalingFactor

Examples

```
data(result_extractBinding)
getScalingFactor(csds[[1]])
```

---

inputSubtraction *Substracts binding scores of input DNA to experiment binding scores*

Description

Substracts binding scores of input DNA to experiment binding scores. This step enables to remove artifactual signal.

Usage

```
inputSubtraction(theObject, verbose = TRUE)


## S4 method for signature 'ChIPSeqSpikeDataset'
inputSubtraction(theObject, verbose = TRUE)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
inputSubtraction(theObject,
verbose = TRUE)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
inputSubtraction(theObject, verbose = TRUE)

## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
inputSubtraction(theObject,
verbose = TRUE)
```

Arguments

theObject	ChIPSeqSpike dataset (see ?spikeDataset)
verbose	If FALSE, do not output processing messages. Default is TRUE

Details

When immunoprecipitating (IP) DNA bound by a given protein, a control is needed to distinguish background noise from true signal. This is typically achieved by performing a mock IP omitting the use of antibody. After sequencing, one can notice peaks of signal above background. These peaks have to be removed from analysis since they represent false positives.

The inputSubtraction function reads bigwig files into GRanges objects that are used to perform the subtraction. In boost mode (ChIPSeqSpikeDatasetBoost and ChIPSeqSpikeDatasetListBoost), The reading/writing steps are omitted.

If not in boost mode, the input DNA subtracted bigwig files are written to the folder containing the currently processed bigwig files. In boost mode, use the method exportBigWigs to output the transformed files.

On Windows operating system, due to the Bioconductor package rtracklayer >= 1.37.6 not supporting bigWig files, this method is not available.

Value

Return an object of the same class of the input object with subtracted experiment scores.

A 'BGSub' suffix is added to the bigwig file name.

Methods (by class)

- ChIPSeqSpikeDataset: Method for signature theObject= 'ChIPSeqSpikeDataset'
- ChIPSeqSpikeDatasetBoost: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- ChIPSeqSpikeDatasetList: Method for signature theObject= 'ChIPSeqSpikeDatasetList'
- ChIPSeqSpikeDatasetListBoost: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

See Also

spikeDataset exportBigWigs spikePipe scaling

Examples

```
## Mock example on a restricted number of reads
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file(c("extdata/bam_files"), package="ChIPSeqSpike")
bigwig_path <- system.file(c("extdata/bigwig_files"), package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {

    csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
                         bigWigPath = bigwig_path)

    ## Copying test files to the current folder
    originalBW_vec <- as.character(getBigWigFile(csds))
    dir.create("./test_chipseqspike")
    result <- file.copy(originalBW_vec, "test_chipseqspike")

    csds <- estimateScalingFactors(csds)

    ## Apply RPM scaling
    csds <- scaling(csds, outputFolder = "test_chipseqspike")

    ## Apply input subtraction
    csds <- inputSubtraction(csds)

    ## Delete all files generated in this example
    unlink("test_chipseqspike/", recursive = TRUE)
}
```

---

matBindingValues *Set the matrices of binding values*

Description

Set a list of matrices of binding values that are used to plot boxplots and correlation plots.

Usage

```
matBindingValues(theObject) <- value


## S4 replacement method for signature 'ChIPSeqSpikeDataset'
matBindingValues(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeDatasetBoost'
matBindingValues(theObject) <- value
```

```
## S4 replacement method for signature 'ChIPSeqSpikeCore'
matBindingValues(theObject) <- value
```

Arguments

theObject	A ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost object
value	A list of matrices

Value

The modified object is returned

Author(s)

Nicolas Descostes

See Also

plotCor boxplotSpike

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
                              "input_0-filtered.bw",
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {

    ## Copying example files
    dir.create("./test_chipseqspike")
    result <- file.copy(bigwig_files, "test_chipseqspike")

    csds <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
                      genome_name, verbose = FALSE,
                      outputFolder = output_folder)

    new_list <- list(matrix(seq_len(10)), matrix(2:18))
    matBindingValues(csds[[1]]) <- new_list
    unlink("test_chipseqspike/", recursive = TRUE)
}
```

plotCor *Plot the correlation between ChIP-seq experiments*

Description

Plot the correlation between ChIP-seq experiments using heatscatter plot or, if heatscatterplot = FALSE, correlation tables.

Usage

```
  plotCor(theObject, rawFile = FALSE, rpmFile = FALSE, bgsubFile = FALSE,
revFile = FALSE, spiked = TRUE, main = "", add_contour = FALSE, method_cor =
"spearman", nlevels = 10, color_contour = "black", show_cor = TRUE,
allOnPanel = TRUE, method_scale = "none", method_corrplot = "circle",
heatscatterplot = TRUE, type_corrplot = "upper", diag_corrplot = FALSE,
separateWindows = FALSE, verbose = FALSE, ...)

## S4 method for signature 'ChIPSeqSpikeDataset'
plotCor(theObject, rawFile = FALSE, rpmFile =
FALSE, bgsubFile = FALSE, revFile = FALSE, spiked = TRUE, main = "",
add_contour = FALSE, method_cor = "spearman", nlevels = 10, color_contour =
"black", show_cor = TRUE, allOnPanel = TRUE, method_scale = "none",
method_corrplot = "circle", heatscatterplot = TRUE, type_corrplot = "upper",
diag_corrplot = FALSE, separateWindows = FALSE, verbose = FALSE, ...)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
plotCor(theObject, rawFile = FALSE,
rpmFile = FALSE, bgsubFile = FALSE, revFile = FALSE, spiked = TRUE, main = "",
add_contour = FALSE, method_cor = "spearman", nlevels = 10,
color_contour = "black", show_cor = TRUE, allOnPanel = TRUE, method_scale =
"none", method_corrplot = "circle", heatscatterplot = TRUE, type_corrplot =
"upper", diag_corrplot = FALSE, separateWindows = FALSE, verbose = FALSE, ...)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
plotCor(theObject, rawFile = FALSE, rpmFile
= FALSE, bgsubFile = FALSE, revFile = FALSE, spiked = TRUE, main = "",
add_contour = FALSE, method_cor = "spearman", nlevels = 10, color_contour =
"black", show_cor = TRUE, allOnPanel = TRUE, method_scale = "none",
 method_corrplot = "circle", heatscatterplot = TRUE, type_corrplot = "upper",
 diag_corrplot = FALSE, separateWindows = FALSE, verbose = FALSE, ...)

## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
plotCor(theObject, rawFile = FALSE,
rpmFile = FALSE, bgsubFile = FALSE, revFile = FALSE, spiked = TRUE, main = "",
add_contour = FALSE, method_cor = "spearman", nlevels = 10,
color_contour = "black", show_cor = TRUE, allOnPanel = TRUE,
method_scale = "none", method_corrplot = "circle", heatscatterplot = TRUE,
type_corrplot = "upper", diag_corrplot = FALSE, separateWindows = FALSE,
verbose = FALSE, ...)
```

Arguments

theObject	ChIPSeqSpike dataset (see ?spikeDataset)
rawFile	If TRUE, use the untransformed data. Not available in boost mode. Default is FALSE. (see details)
rpmFile	If TRUE, use the RPM scaled data. Not available in boost mode. Default is FALSE. (see details)
bgsubFile	If TRUE, use the input subtracted data. Not available in boost mode. Default is FALSE. (see details)
revFile	If TRUE, use the RPM reverted data. Not available in boost mode. Default is FALSE. (see details)
spiked	If TRUE, use the spiked data. Default is TRUE. (see details)
main	Main title of the plot. No title is displayed by default.
add_contour	If TRUE, contours are added to the heatscatter. Default is FALSE.
method_cor	A character string indicating which correlation coefficient is to be computed. One of 'pearson' (default) , 'kendall' or 'spearman'.
nlevels	An integer giving the number of levels of the contour lines. Not used if heatscatteplot is FALSE. Default is 10
color_contour	Character string defining the color of the contour line. Not used if heatscatteplot is FALSE. Default is 'black'.
show_cor	Logical indicating if the correlation is added to the title. Not used if heatscatteplot is FALSE. Default is TRUE.
allOnPanel	Logical indicating if all correlations should be on the same panel. Not used if heatscatteplot is FALSE. Default is TRUE.
method_scale	Character string indicating the scaling to be applied to the data. Possible values are 'none', 'log', 'asinh', 'cuberoot' or 'zscore'. Not used if heatscatteplot is FALSE.
method_corrplot	If heatscatterplot is FALSE, define the graphical representation used for the correlation table. Possible values are 'circle', 'square', 'ellipse', 'number', 'pie', 'shade' and 'color'. See ?corrplot::corrplot for more details. Default is 'circle'.
heatscatterplot	If TRUE, use a heatscatter representation instead of correlation table. see ?LSD::heatscatter for more details. Default is TRUE.
type_corrplot	If heatscatter is FALSE, define if the full ('full'), the lower triangular ('lower') or upper triangular matrix is displayed. Default is 'upper'.
diag_corrplot	If heatscatter is FALSE, logical indicating if the correlation coefficients are displayed on the principal diagonal. Default is FALSE.
separateWindows	If heatscatterplot is TRUE, Logical indicating if each plot is output to a separate window. Default is FALSE.
verbose	Logical indicating if processing messages are displayed. Default is FALSE
...	Additional parameter to pass to the LSD::heatscatter or corrplot::corrplot functions.

Details

The 'rawFile', 'rpmFile', 'bgsubFile', 'revFile', 'spiked' parameters indicate if the untransformed, RPM scaled, input DNA subtracted, RPM reversed or spiked data should be used plotted. This option is only available if not in boost mode (see ?spikePipe for details on what these steps are).

Value

If heatscatterplot is FALSE, return the correlation matrix.

Methods (by class)

- `ChIPSeqSpikeDataset`: Method for signature theObject= 'ChIPSeqSpikeDataset'
- `ChIPSeqSpikeDatasetBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- `ChIPSeqSpikeDatasetList`: Method for signature theObject= 'ChIPSeqSpikeDatasetList'
- `ChIPSeqSpikeDatasetListBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

See Also

`spikeDataset` `heatscatter` `corrplot` `spikePipe` `plotTransform` `boxplotSpike` `plotHeatmaps` `plotProfile`

Examples

```
## Sub-sample
data("result_extractBinding")

## Heatscatter of spiked data using Spearman correlation
plotCor(csds)

## Pearson correlation of log transformed raw data
plotCor(csds,  rawFile = TRUE, spiked = FALSE,  main = "heatscatter",
method_cor = "pearson", method_scale = "log")

## Correlation table of all transformation steps with circle representation
plotCor(csds, rawFile = TRUE, rpmFile = TRUE, bgsubFile = TRUE, revFile = TRUE,
 spiked = TRUE, heatscatterplot = FALSE, verbose = TRUE)

## Correlation table of all transformation steps with number representation
plotCor(csds, rawFile = TRUE, rpmFile = TRUE, bgsubFile = TRUE, revFile = TRUE,
 spiked = TRUE, heatscatterplot = FALSE, verbose = TRUE, method_corrplot =
 "number")
```

`plotHeatmaps` *plot heatmaps of ChIP-seq experiments*

Description

Clusters and output binding values signal in the form of heatmaps

Usage

```
  plotHeatmaps(theObject, location = "start",
               transformType = "spiked", legend = TRUE, plot_scale = "no",
               sort_rows = "decreasing", nb_of_groups = 1,
               clustering_method = "none", include_exp_vec = NULL,
               auto_scale = FALSE, raster_value = FALSE, col_value = "blue",
               ...)

## S4 method for signature 'ChIPSeqSpikeDataset'
plotHeatmaps(theObject, location = "start",
transformType = "spiked", legend = TRUE, plot_scale = "no", sort_rows =
"decreasing", nb_of_groups = 1, clustering_method = "none", include_exp_vec =
NULL, auto_scale = FALSE, raster_value = FALSE, col_value = "blue", ...)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
plotHeatmaps(theObject, location =
"start", transformType = "spiked", legend = TRUE, plot_scale = "no",
sort_rows = "decreasing", nb_of_groups = 1, clustering_method = "none",
include_exp_vec = NULL, auto_scale = FALSE, raster_value = FALSE,
col_value = "blue", ...)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
plotHeatmaps(theObject, location = "start",
transformType = "spiked", legend = TRUE, plot_scale = "no", sort_rows =
"decreasing", nb_of_groups = 1, clustering_method = "none", include_exp_vec =
NULL, auto_scale = FALSE, raster_value = FALSE, col_value = "blue", ...)

## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
plotHeatmaps(theObject, location =
"start", transformType = "spiked", legend = TRUE, plot_scale = "no",
sort_rows = "decreasing", nb_of_groups = 1, clustering_method = "none",
include_exp_vec = NULL, auto_scale = FALSE, raster_value = FALSE,
col_value = "blue", ...)
```

Arguments

theObject	ChIPSeqSpike dataset (see ?spikeDataset)
location	Define the location at which heatmaps should be plotted. Possible values are 'start', 'midpoint', 'end' and 'composite' (see details). Default is 'start'.
transformType	Indicate what step of chip-seq spike-in normalization is used. Possible values are 'spiked', 'reverse', 'BGSub' 'RPM' and 'raw' (see details). Default is 'spiked'. Not available in boost mode.
legend	Logical indicating if legend should be indicated. Default is FALSE.
plot_scale	Indicate the transformation that is applied to the data before plotting. Possible values are 'no' (default), 'linear', 'log2' or 'zscore'. See ?seqplots::plotHeatmap for more details.
sort_rows	Indicate how rows of the heatmap should be sorted. Possible values are 'increasing', 'decreasing' or FALSE. See ?seqplots::plotHeatmap for more details. Default is 'decreasing'.

`nb_of_groups` If clustering_method is different than 'none', define the number of groups of the cluster. Default is 1.

`clustering_method` Determine the heatmap clustering algorithm. Possible values are 'k-means', 'hclust', 'ssom', 'bed_scores' and 'none'. See ?seqplots::plotHeatmap for more details. Default is 'none'.

`include_exp_vec` Logical vector indicating on which experiments clustering is performed. NULL meaning all experiments. See ?seqplots::plotHeatmap for more details. Default is NULL.

`auto_scale` Logical indicating if color scaled should be specific to each experiment. Default is FALSE.

`raster_value` Logical indicating if the bitmap raster is used. See ?seqplots::plotHeatmap for more details. Default is FALSE.

`col_value` The vector or list of colour values used generate sub-heatmaps colorspaces. See ?seqplots::plotHeatmap for more details. Default is 'blue'.

`...` Additional parameter to pass to seqplots::plotHeatmap.

Details

This method relies on the 'plotHeatmap' method of the 'seqplots' package and enables to generate different heatmaps on the chIPSeqSpike objects (see ?spikeDataset).

The 'transformType' indicates if the untransformed (raw), RPM scaled (RPM), input subtracted (BGSub), RPM reversed (reverse) or spiked (spiked) data should be used to generate the heatmaps. This option is only available if not in boost mode (see ?spikePipe for details on what these steps are).

Value

See ?seqplots::plotHeatmap for details.

Methods (by class)

- `ChIPSeqSpikeDataset`: Method for signature theObject= 'ChIPSeqSpikeDataset'
- `ChIPSeqSpikeDatasetBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- `ChIPSeqSpikeDatasetList`: Method for signature theObject= 'ChIPSeqSpikeDatasetList'
- `ChIPSeqSpikeDatasetListBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

See Also

`spikeDataset` `spikePipe` `plotHeatmap` `plotProfile` `plotTransform` `boxplotSpike` `plotCor`

Examples

```
data("result_extractBinding")

## Spiked data in decreasing order
plotHeatmaps(csds)

## Raw data in decreasing order
plotHeatmaps(csds, transformType = "raw")

## K-means clustering with 3 groups
plotHeatmaps(csds, nb_of_groups = 3, clustering_method = "kmeans")
```

`plotProfile` *Plots average profiles of ChIP-seq experiments*

Description

Plots average profiles of all experiments contained in a 'spikeDataset' at different locations

Usage

```
    plotProfile(theObject, legends = FALSE, colVec = NULL, notScaled = FALSE)

## S4 method for signature 'ChIPSeqSpikeDataset'
plotProfile(theObject, legends = FALSE,
colVec = NULL, notScaled = FALSE)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
plotProfile(theObject, legends = FALSE,
colVec = NULL)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
plotProfile(theObject, legends = FALSE,
colVec = NULL, notScaled = FALSE)

## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
plotProfile(theObject, legends =
 FALSE, colVec = NULL)
```

Arguments

`theObject`	`ChIPSeqSpike` dataset (see ?spikeDataset)
`legends`	Logical indicating if legend should be indicated. Default is FALSE.
`colVec`	Character vector indicating the colors to use. Default is NULL
`notScaled`	If not in boost mode, logical indicating if untransformed data should be plotted. Default is FALSE.

Details

plotProfile plots average signal over annotations that were given to the 'extractBinding' method. plotProfile is using the 'plotAverage' method of the 'seqplots' package.

By default, the spiked signal is plotted. If the 'notScaled' parameter is set to TRUE and the object is of type ChIPSeqSpikeDatasetList or ChIPSeqSpikeDataset (no boost mode), the RPM scaled and input subtracted values are also plotted. This option enables to visualize the effect of spike-in scaling.

The signal is plotted at four different annotation locations: 'start', 'end', 'midpoint' and 'composite' (pf, ef, mf and af options of 'plotAverage' method).

Value

Nothing

Methods (by class)

- `ChIPSeqSpikeDataset`: Method for signature theObject= 'ChIPSeqSpikeDataset'
- `ChIPSeqSpikeDatasetBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- `ChIPSeqSpikeDatasetList`: Method for signature theObject= 'ChIPSeqSpikeDatasetList'
- `ChIPSeqSpikeDatasetListBoost`: Method for signature theObject= ChIPSeqSpikeDatasetList-Boost'

Author(s)

Nicolas Descostes

See Also

`spikeDataset` `extractBinding` `plotAverage` `plotTransform` `boxplotSpike` `plotHeatmaps` `plotCor`

Examples

```
data("result_extractBinding")
plotProfile(csds)
plotProfile(csds, TRUE)
plotProfile(csds, TRUE, notScaled = TRUE)
```

`plotTransform` *Plots average profiles of steps of the spike-in normalization*

Description

For each experiment, plots average profiles of RPM scaled, input subtracted, RPM reversed and spiked data at different annotation locations (not available in boost mode)

Usage

```
  plotTransform(theObject, legends = FALSE, colVec = NULL,
                separateWindows = FALSE)

## S4 method for signature 'ChIPSeqSpikeDataset'
plotTransform(theObject, legends = FALSE,
colVec = NULL, separateWindows = FALSE)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
plotTransform(theObject, legends = FALSE,
colVec = NULL, separateWindows = FALSE)
```

Arguments

theObject	A ChIPSeqSpikeDatasetList or ChIPSeqSpikeDataset object
legends	Logical indicating if legend should be indicated. Default is FALSE.
colVec	Character vector indicating the colors to use. Default is NULL
separateWindows	Plot each experiment in a separate window. Default is FALSE.

Details

plotTransform plots average signal over annotations that were given to the 'extractBinding' method. plotTransform is using the 'plotAverage' method of the 'seqplots' package.

The signal is plotted at four different annotation locations: 'start', 'end', 'midpoint' and 'composite' (pf, ef, mf and af options of 'plotAverage' method).

As objects created in boost mode only hold the binding values in GRanges objects, the previously applied transformations are not kept in memory. Therefore, this method does not work with ChIPSeqSpikeDatasetBoost and ChIPSeqSpikeDatasetListBoost objects.

Value

Nothing

Methods (by class)

- ChIPSeqSpikeDataset: Method for signature theObject= 'ChIPSeqSpikeDataset'
- ChIPSeqSpikeDatasetList: Method for signature theObject= 'ChIPSeqSpikeDatasetList'

Author(s)

Nicolas Descostes

See Also

spikeDataset extractBinding plotAverage plotProfile boxplotSpike plotHeatmaps plotCor

Examples

```
data("result_extractBinding")
plotTransform(csds, TRUE)
plotTransform(csds, TRUE, separateWindows=TRUE)
```

`ratio` *Result of method getRatio on the complete dataset*

Description

Result of method getRatio on the complete dataset.

Usage

```
data(ratio)
```

Format

matrix

Details

Complete Dataset

The data used in this documentation represent a gold-standard example of the importance of using spike-in controls with ChIP-Seq experiments. It uses Drosophila Melanogaster chromatin as exogenous spike-in control to correct experimental biases. Without spike-in control and using only RPM normalization, proper differences of H3K79me2 histone modification in human Jurkat cells upon EPZ5676 inhibitor treatment are not observed [1].

This dataset is made of bigwig and bam files of H3K79me2 ChIP-Seq data and corresponding input DNA controls. Bam files contain data aligned to the Human reference genome Hg19 or to the Drosophila reference genome dm3. The latest is used to compute external spike-in scaling factors. All above mentioned data are available at 0, 50 and 100 percent EPZ5676 inhibitor treatment (see vignette for data references).

References

[1] Orlando et al, "Quantitative ChIP-Seq normalization reveals global modulation of the epigenome", Cell Rep, 2014.

result_data *Testing data for vignette*

Description

This dataset gives the result of calling the method estimateScalingFactors on the complete dataset[1] and the method result_extractBinding on the top 100 mostly bound genes

Usage

```
data(result_estimateScalingFactors)
data(result_extractBinding)
```

Format

A ChIPSeqSpikeDataset object

Details

Complete Data

The data used in this documentation represent a gold-standard example of the importance of using spike-in controls with ChIP-Seq experiments. It uses Drosophila Melanogaster chromatin as exogenous spike-in control to correct experimental biases. Without spike-in control and using only RPM normalization, proper differences of H3K79me2 histone modification in human Jurkat cells upon EPZ5676 inhibitor treatment are not observed [1].

This dataset is made of bigwig and bam files of H3K79me2 ChIP-Seq data and corresponding input DNA controls. Bam files contain data aligned to the Human reference genome Hg19 or to the Drosophila reference genome dm3. The latest is used to compute external spike-in scaling factors. All above mentioned data are available at 0, 50 and 100 percent EPZ5676 inhibitor treatment (see vignette for data references).

References

[1] Orlando et al, "Quantitative ChIP-Seq normalization reveals global modulation of the epigenome", Cell Rep, 2014.

scaling *Applies different type of scaling/normalization procedures*

Description

Applies or reverse different type of scaling/normalization to bigwig files contained in 'ChIPSeqSpikeDataset', 'ChIPSeqSpikeDatasetBoost', 'ChIPSeqSpikeDatasetList' or 'ChIPSeqSpikeDatasetListBoost' objects

Usage

```
scaling(theObject, reverse = FALSE, type = "endo", verbose = TRUE,
        outputFolder = NULL)

## S4 method for signature 'ChIPSeqSpikeDataset'
scaling(theObject, reverse = FALSE,
type = "endo", verbose = TRUE, outputFolder = NULL)

## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
scaling(theObject, reverse = FALSE,
type = "endo", verbose = TRUE, outputFolder = NULL)

## S4 method for signature 'ChIPSeqSpikeDatasetList'
scaling(theObject, reverse = FALSE,
type = "endo", verbose = TRUE, outputFolder = NULL)

## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
scaling(theObject, reverse = FALSE,
type = "endo", verbose = TRUE, outputFolder = NULL)
```

Arguments

theObject	ChIPSeqSpike dataset (see ?spikeDataset)
reverse	If TRUE, reverse the scaling applied previously (see details). Default is FALSE
type	Should be 'endo' or 'exo' to apply the endogenous or exogenous scaling factor respectively.
verbose	If FALSE, do not output processing messages. Default is TRUE
outputFolder	Define the folder where scaled bigwig are output. Default is NULL (see details).

Details

Scaling is performed after estimating scaling factors with '?estimateScalingFactors'.

Different type of scaling can be performed according to the steps of spike-in normalization. The first type is to transform the data by performing a RPM scaling. The second type is to reverse the RPM scaling after having performed input subtraction with '?inputSubtraction'. The third and last type is to apply the exogenous scaling factor which finalizes the spike-in normalization (see example below for parameter settings).

To perform all steps in one call and in the right order, one can consider using the '?spikePipe' function.

If not in boost mode, the bigwig files are written to the folder containing the currently processed bigwig files. In boost mode, use the method ?exportBigWigs to output the transformed files.

If outputFolder is not NULL, the original bigwig files should be copied to this folder before performing the analysis. This parameter was created to test the package with the provided files in extdata/.

On Windows operating system, due to the Bioconductor package rtracklayer >= 1.37.6 not supporting bigWig files, this method is not available.

Value

Return an object of the same class of the input object and output scaled bigwig files if not in boost mode (see ?ChIPSeqSpikeDatasetBoost).

'-RPM', 'reverse' or 'spiked' suffixes are added to the bigwig file name if performing RPM scaling, reversing RPM scaling or applying exogenous scaling factors respectively.

Methods (by class)

- `ChIPSeqSpikeDataset`: Method for signature theObject= 'ChIPSeqSpikeDataset'
- `ChIPSeqSpikeDatasetBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- `ChIPSeqSpikeDatasetList`: Method for signature theObject= 'ChIPSeqSpikeDatasetList'
- `ChIPSeqSpikeDatasetListBoost`: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

See Also

`spikeDataset` `ChIPSeqSpikeDatasetBoost` `spikePipe` `inputSubtraction` `exportBigWigs`

Examples

```
## Mock example on a restricted number of reads
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file(c("extdata/bam_files"), package="ChIPSeqSpike")
bigwig_path <- system.file(c("extdata/bigwig_files"), package="ChIPSeqSpike")

csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
                     bigWigPath = bigwig_path)

## Copying test files to the current folder
originalBW_vec <- as.character(getBigWigFile(csds))
dir.create("./test_chipseqspike")
result <- file.copy(originalBW_vec, "test_chipseqspike")

csds <- estimateScalingFactors(csds)

if(.Platform$OS.type != 'windows') {
    ## Apply RPM scaling
    csds <- scaling(csds, outputFolder = "test_chipseqspike")

    ## Apply input subtraction
    csds <- inputSubtraction(csds)

    ## Reverse RPM scaling after input subtraction
    csds <- scaling(csds, reverse = TRUE)

    ## Apply exogenous scaling factors
    csds <- scaling(csds, type = "exo")
}
```

```
## Delete all files generated in this example
unlink("test_chipseqspike/", recursive = TRUE)
```

`scalingFactor` *Set the endogenous scaling factor associated to an experiment*

Description

Set the endogenous scaling factor associated to an experiment or its corresponding input DNA experiment.

Usage

```
scalingFactor(theObject) <- value


## S4 replacement method for signature 'ChIPSeqSpikeDataset'
scalingFactor(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeDatasetBoost'
scalingFactor(theObject) <- value

## S4 replacement method for signature 'Experiment'
scalingFactor(theObject) <- value

## S4 replacement method for signature 'ExperimentLoaded'
scalingFactor(theObject) <- value

## S4 replacement method for signature 'ChIPSeqSpikeCore'
scalingFactor(theObject) <- value
```

Arguments

`theObject`	A `ChIPSeqSpikeCore`, `ChIPSeqSpikeDataset`, `ChIPSeqSpikeDatasetBoost`, `Experiment` or `ExperimentLoaded` object
`value`	A numeric representing the endogenous scaling factor

Details

A scaling factor is defined as:

- 1/(bam_count/1000000)

If the object is ChIPSeqSpikeCore, ChIPSeqSpikeDataset or ChIPSeqSpikeDatasetBoost, scalingFactor will set the endogenous scaling factor of the input DNA experiment of the dataset.

If the object is Experiment or ExperimentLoaded, scalingFactor will set the endogenous scaling factor of the experiment.

Value

The modified object is returned

Author(s)

Nicolas Descostes

See Also

exogenousScalingFactor estimateScalingFactors

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
                              "input_0-filtered.bw",
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {

    ## Copying example files
    dir.create("./test_chipseqspike")
    result <- file.copy(bigwig_files, "test_chipseqspike")

    csds <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec, genome_name,
                      verbose = TRUE, outputFolder = output_folder)

    getScalingFactor(csds[[1]][[1]])
    scalingFactor(csds[[1]][[1]]) <- 15
    getScalingFactor(csds[[1]][[1]])

    unlink("test_chipseqspike/", recursive = TRUE)
}
```

---

spikeDataset *spikeDataset constructors function*

Description

Function for calling the correct constructor using csv or txt input file

Usage

```
spikeDataset(infoFile, bamPath, bigWigPath, boost = FALSE,
      verbose = TRUE)
```

Arguments

infoFile	csv or txt file containing information about files (see details).
bamPath	Path to the folder containing bam files.
bigWigPath	Path to the folder containing bigwig files.
boost	If TRUE, the object created enables to perform the analysis in boost mode (see details). Default is FALSE.
verbose	If FALSE, do not print messages about object creation. Default is TRUE

Details

'infoFile' should be a csv or a tab separated txt file. The column names should be: expName, endogenousBam, exogenousBam, inputBam, bigWigEndogenous and bigWigInput. These columns indicate the experiment names; the bam file names of data aligned to the reference genome; the bam file names of data aligned to the exogenous genome; the input DNA bam file names corresponding to each experiment; the bigwig file names of data aligned to the reference genome and the bigwig file names of input DNA experiments.

If 'infoFile' contains only one input file (specified for each experiment), a ChIPSeqSpikeDataset (or ChIPSeqSpikeDatasetBoost) object is created. if 'infoFile' contains different input DNA files, an object of type 'list' is created (ChIPSeqSpikeDatasetList or ChIPSeqSpikeDatasetListBoost). Each element of the list will contain all experiments corresponding to a given input DNA one.

If boost = TRUE, either a ChIPSeqSpikeDatasetBoost or ChIPSeqSpikeDatasetListBoost object is created. The boost mode enables to store the binding values in the form of a GRanges object and avoid reading/writing files at each processing step. Even if faster, this mode however consumes much more memory and should be used with caution.

Value

A ChIPSeqSpikeDataset object if only one input DNA experiment is provided.

A ChIPSeqSpikeDatasetBoost object if only one input DNA experiment is provided and boost = TRUE.

A ChIPSeqSpikeDatasetList object if several input DNA experiments are provided.

A ChIPSeqSpikeDatasetListBoost object if several input DNA experiments are provided and boost = TRUE.

Author(s)

Nicolas Descostes

See Also

ChIPSeqSpikeDataset ChIPSeqSpikeDatasetBoost ChIPSeqSpikeDatasetList ChIPSeqSpikeDatasetListBoost spikePipe

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file(c("extdata/bam_files"), package="ChIPSeqSpike")
bigwig_path <- system.file(c("extdata/bigwig_files"), package="ChIPSeqSpike")

## Returns ChIPSeqSpikeDatasetList
csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
```

```
                    bigWigPath = bigwig_path)
is(csds)
csds

## Returns ChIPSeqSpikeDatasetListBoost

if(.Platform$OS.type != 'windows') {
    csds <- spikeDataset(infoFile = info_file_csv, bamPath = bam_path,
                         bigWigPath = bigwig_path, boost = TRUE)
    is(csds)
    csds
}
```

---

spikePipe *ChIP-seq spike-in normalization wrapper function*

Description

This function performs all steps of spike-in normalization: Dataset creation, RPM scaling, input DNA subtraction, RPM scaling reversal, exogenous DNA scaling (spike) and binding values extraction.

Usage

```
spikePipe(infoFile, bamPath, bigWigPath, anno, genome_version,
          paired = FALSE, binsize = 50, profile_length_before = 2000,
          profile_length_after= 2000, mean_or_median = "mean",
          interpolation_number = 100, interpolation_average = 10000,
          ignore_strand = FALSE, verbose = FALSE, boost = FALSE,
          outputFolder = NULL)
```

Arguments

infoFile	csv or tab separated txt file containing information about files (see details)
bamPath	Path to the folder containing bam files
bigWigPath	Path to the folder containing bigwig files
anno	File in GFF format containing annotations used to plot information
genome_version	The UCSC code of reference genome, e.g. 'hg19' for Homo sapiens (see details)
paired	Indicate if sequences are single- or paired-ended. Default is FALSE
binsize	Binning size used to create bigwig files. Default is 50.
profile_length_before	Length in bp of the interval upstream annotation (see details). Default is 2000.
profile_length_after	Length in bp of the interval downstream annotation (see details). Default is 2000.
mean_or_median	For average profiles, should the 'mean' or 'median' values be used. Default is 'mean'.
interpolation_number	Number of interpolated points to create matrices (see details). Default is 100.

`interpolation_average`
: Number of interpolated points of profiles and heatmaps (see details). Default is 10000.

`ignore_strand`
: If TRUE the directionality is ignored, that is all features strands, regardless of annotation in GFF file, are treated as undetermined ("*"). default is FALSE.

`verbose`
: If TRUE, output processing messages. Default is FALSE.

`boost`
: If TRUE, the object created enables to perform the analysis in boost mode (see details). Default is FALSE

`outputFolder`
: Define the folder where scaled bigwig are output. Default is NULL (see details).

Details

'infoFile' should be a csv or a tab separated txt file. The column names should be: expName, endogenousBam, exogenousBam, inputBam, bigWigEndogenous and bigWigInput. These columns indicate the experiment names; the bam file names of data aligned to the reference genome; the bam file names of data aligned to the exogenous genome; the input DNA bam file names corresponding to each experiment; the bigwig file names of data aligned to the reference genome and the bigwig file names of input DNA experiments.

If 'infoFile' contains only one input file (specified for each experiment), a ChIPSeqSpikeDataset (or ChIPSeqSpikeDatasetBoost) object is created. if 'infoFile' contains different input DNA files, an object of type 'list' is created (ChIPSeqSpikeDatasetList or ChIPSeqSpikeDatasetListBoost). Each element of the list will contain all experiments corresponding to a given input DNA one.

This function calls different processing steps that overall perform ChIP-seq spike-in normalization. The steps and functions are called in the following order: Dataset creation (see ?spikeDataset), RPM scaling (see ?scaling), input DNA subtraction (see ?inputSubtraction), RPM scaling reversal (see ?scaling), exogenous DNA scaling (see ?scaling) and binding values extraction (see ?extractBinding).

For details on installing reference genomes, see details of the function '?getPlotSetArray' of the 'seqplots' package.

For more details on parameters profile_length_before, profile_length_after, mean_or_median, interpolation_number, interpolation_average and ignore_strand, see ?extractBinding.

If boost = TRUE, either a ?ChIPSeqSpikeDatasetBoost or ?ChIPSeqSpikeDatasetListBoost object is created. The boost mode enables to store the binding values in the form of a GRanges object and avoid reading/ writing files at each processing step. Even if faster, this mode however consumes much more memory and should be used with caution.

If outputFolder is not NULL, the original bigwig files should be copied to this folder before performing the analysis. This parameter was created to test the package with the provided files in extdata/.

On Windows operating system, this function is not available due to the Bioconductor package rtracklayer >= 1.37.6 which does not support bigWig files. This function will return null.

Value

Returns a spike-in normalized object with extracted binding values that can be used to perform graphical representations (see ?plotProfile, ?plotTransform, ?plotHeatmaps, ?boxplotSpike and ?plotCor).

According to the files provided in 'infoFile', different objects are returned:

- A ChIPSeqSpikeDataset object if only one input DNA experiment is provided.

- A ChIPSeqSpikeDatasetBoost object if only one input DNA experiment is provided and boost = TRUE.
- A ChIPSeqSpikeDatasetList object if several input DNA experiments are provided.
- A ChIPSeqSpikeDatasetListBoost object if several input DNA experiments are provided and boost = TRUE.

Author(s)

Nicolas Descostes

See Also

ChIPSeqSpikeDataset ChIPSeqSpikeDatasetList ChIPSeqSpikeDatasetBoost ChIPSeqSpikeDatasetListBoost spikeDataset scaling inputSubtraction extractBinding plotProfile plotTransform plotHeatmaps boxplotSpike plotCor getPlotSetArray

Examples

```
info_file_csv <- system.file("extdata/info.csv", package="ChIPSeqSpike")
bam_path <- system.file("extdata/bam_files", package="ChIPSeqSpike")
bigwig_path <- system.file("extdata/bigwig_files", package="ChIPSeqSpike")
gff_vec <- system.file("extdata/test_coord.gff", package="ChIPSeqSpike")
genome_name <- "hg19"
output_folder <- "test_chipseqspike"
bigwig_files <- system.file("extdata/bigwig_files",
                            c("H3K79me2_0-filtered.bw",
                              "H3K79me2_100-filtered.bw",
                              "H3K79me2_50-filtered.bw",
                              "input_0-filtered.bw",
                              "input_100-filtered.bw",
                              "input_50-filtered.bw"), package="ChIPSeqSpike")

if(.Platform$OS.type != 'windows') {

    ## Copying example files
    dir.create("./test_chipseqspike")
    result <- file.copy(bigwig_files, "test_chipseqspike")

    csds <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
                      genome_name, verbose = TRUE,
                      outputFolder = output_folder)

    csds2 <- spikePipe(info_file_csv, bam_path, bigwig_path, gff_vec,
                       genome_name, boost = TRUE, verbose = TRUE,
                       outputFolder = output_folder)

    unlink("test_chipseqspike/", recursive = TRUE)
    is(csds)
    is(csds2)
}
```

spikeSummary *Output dataset summary information*

Description

Output a table giving endogenous/exogenous scaling scores and read counts for each experiment contained in a dataset

Usage

```
    spikeSummary(theObject)

## S4 method for signature 'Experiment'
spikeSummary(theObject)
## S4 method for signature 'ExperimentLoaded'
spikeSummary(theObject)
## S4 method for signature 'ChIPSeqSpikeDataset'
spikeSummary(theObject)
## S4 method for signature 'ChIPSeqSpikeDatasetList'
spikeSummary(theObject)
## S4 method for signature 'ChIPSeqSpikeDatasetBoost'
spikeSummary(theObject)
## S4 method for signature 'ChIPSeqSpikeDatasetListBoost'
spikeSummary(theObject)
```

Arguments

theObject	ChIPSeqSpike dataset (see ?spikeDataset)

Details

The rows represent an experiment (with corresponding input below) and the columns represent the endogenous scaling factor (endoScalFact), the exogenous scaling factor (exoScalFact), the number of reads aligned to the reference genome (endoCount) and the number of reads aligned to the exogenous genome (exoCount).

Value

a matrix of information about each experiment

Methods (by class)

- Experiment: Method for signature theObject = 'Experiment'
- ExperimentLoaded: Method for signature theObject = 'ExperimentLoaded'
- ChIPSeqSpikeDataset: Method for signature theObject= 'ChIPSeqSpikeDataset'
- ChIPSeqSpikeDatasetBoost: Method for signature theObject= 'ChIPSeqSpikeDatasetBoost'
- ChIPSeqSpikeDatasetList: Method for signature theObject= 'ChIPSeqSpikeDatasetList'
- ChIPSeqSpikeDatasetListBoost: Method for signature theObject= 'ChIPSeqSpikeDatasetListBoost'

Author(s)

Nicolas Descostes

See Also

getRatio

Examples

```
data("result_estimateScalingFactors")
spikeSummary(csds)
```

Index