mosaics

March 24, 2012

`BinData-class` *Class "BinData"*

Description

This class represents bin-level ChIP-seq data.

Objects from the Class

Objects can be created by calls of the form `new("BinData", ...)`.

Slots

`chrID`: Object of class `"character"`, a vector of chromosome IDs.

`coord`: Object of class `"numeric"`, a vector of coordinates.

`tagCount`: Object of class `"numeric"`, a vector of tag counts of ChIP sample.

`mappability`: Object of class `"numeric"`, a vector of mappability score.

`gcContent`: Object of class `"numeric"`, a vector of GC content score.

`input`: Object of class `"numeric"`, a vector of tag counts of control sample.

`dataType`: Object of class `"character"`, indicating how reads were processed. Possible values are "unique" (only uniquely aligned reads were retained) and "multi" (reads aligned to multiple locations were also retained).

Methods

mosaicsFit `signature(object = "BinData")`: fit MOSAiCS model from a bin-level ChIP-seq data.

plot `signature(x = "BinData", y = "missing", plotType = NULL )`: provide exploratory plots of mean ChIP tag counts. This method plots mean ChIP tag counts versus mappability score, GC content score, and input tag counts, with 95% confidence intervals, for `plotType="M"`, `plotType="GC"`, and `plotType="input"`, respectively. `plotType="M|input"` and `plotType="GC|input"` provide plots of mean ChIP tag counts versus mappability and GC content score, respectively, conditional on input tag counts. If `plotType` is not specified, this method plots histogram of ChIP tag counts.

print `signature(x = "BinData")`: return bin-level data in data frame format.

show `signature(object = "BinData")`: provide brief summary of the object.

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, G Pan, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

`readBins`, `mosaicsFit`.

Examples

```
showClass("BinData")
## Not run:
library(mosaicsExample)
data(exampleBinData)

exampleBinData
print(exampleBinData)[1:10,]
plot(exampleBinData)
plot( exampleBinData, plotType="M" )
plot( exampleBinData, plotType="GC" )
plot( exampleBinData, plotType="input" )
plot( exampleBinData, plotType="M|input" )
plot( exampleBinData, plotType="GC|input" )

exampleFit <- mosaicsFit( exampleBinData, analysisType="TS" )

## End(Not run)
```

`MosaicsFit-class` *Class "MosaicsFit"*

Description

This class represents MOSAiCS model fit.

Objects from the Class

Objects can be created by calls of the form `new("MosaicsFit", ...)`.

Slots

`mosaicsEst`: Object of class `"MosaicsFitEst"`, representing estimates of MOSAiCS model fit.

`mosaicsParam`: Object of class `"MosaicsFitParam"`, representing tuning parameters for fitting MOSAiCS model.

`chrID`: Object of class `"character"`, a vector of chromosome IDs.

`coord`: Object of class `"numeric"`, a vector of coordinates.

`tagCount`: Object of class `"numeric"`, a vector of tag counts of ChIP sample.

`bic1S`: Object of class `"numeric"`, Bayesian Information Criterion (BIC) value of one-signal-component model.

`bic2S`: Object of class `"numeric"`, Bayesian Information Criterion (BIC) value of two-signal-component model.

Methods

estimates `signature(object = "MosaicsFit")`: extract estimates from MOSAiCS model fit.

mosaicsPeak `signature(object = "MosaicsFit")`: call peaks using MOSAiCS model fit.

plot `signature(x = "MosaicsFit", y = "missing")`: draw Goodness of Fit (GOF) plot.

print `signature(x = "MosaicsFit")`: (not supported yet)

show `signature(object = "MosaicsFit")`: provide brief summary of the object.

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, JA Thomson, R Stewart, and S Keles (2010), "A Statistical Framework for the Analysis of ChIP-Seq Data", To appear in *Journal of the American Statistical Association* (http://pubs.amstat.org/doi/abs/10.1198/jasa.2011.ap09706).

See Also

`mosaicsFit`, `mosaicsPeak`, `estimates`.

Examples

```
showClass("MosaicsFit")
## Not run:
library(mosaicsExample)
data(exampleFit)

exampleFit
plot(exampleFit)
estimates(exampleFit)

examplePeak <- mosaicsPeak( exampleFit, signalModel = "2S", FDR = 0.05 )

## End(Not run)
```

`MosaicsPeak-class` *Class "MosaicsPeak"*

Description

This class represents peak calling results.

Objects from the Class

Objects can be created by calls of the form `new("MosaicsPeak", ...)`.

Slots

`peakList`: Object of class `"MosaicsPeakList"`, representing peak list.

`peakParam`: Object of class `"MosaicsPeakParam"`, representing parameters for peak calling.

`bdBin`: Object of class `"numeric"`, a vector of bounded bins.

`empFDR`: Object of class `"numeric"`, empirical FDR.

Methods

export `signature(object = "MosaicsPeak")`: export peak list into text files.

print `signature(x = "MosaicsPeak")`: return peak list in data frame format.

show `signature(object = "MosaicsPeak")`: provide brief summary of the object.

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, G Pan, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

`mosaicsPeak`, `export`.

Examples

```
showClass("MosaicsPeak")
## Not run:
library(mosaicsExample)
data(exampleFit)
examplePeak <- mosaicsPeak( exampleFit, signalModel = "2S", FDR = 0.05 )

examplePeak
print(examplePeak)[1:10, ]
export( examplePeak, type = "txt", fileLoc = "./", fileName = "TSpeakList.txt" )
export( examplePeak, type = "bed", fileLoc = "./", fileName = "TSpeakList.bed" )
```

```
export( examplePeak, type = "gff", fileLoc = "./", fileName = "TSpeakList.gff" )

## End(Not run)
```

`constructBins` *Construct bin-level ChIP-sep data from an aligned read file*

Description

Preprocess and construct bin-level ChIP-sep data from an aligned read file.

Usage

```
constructBins( infileLoc=NULL, infileName=NULL, fileFormat=NULL, outfileLoc=infi
    byChr=FALSE, fragLen=200, binSize=fragLen, capping=0, perl = "perl" )
```

Arguments

`infileLoc`	Directory of the aligned read file to be processed.
`infileName`	Name of the aligned read file to be processed.
`fileFormat`	Format of the aligned read file to be processed. Currently, `constructBins` permits the following aligned read file formats: `"eland_result"` (Eland result), `"eland_extended"` (Eland extended), `"eland_export"` (Eland export), `"bowtie"` (default Bowtie), `"sam"` (SAM), and `"bed"` (BED).
`outfileLoc`	Directory of processed bin-level files. By default, processed bin-level files are exported to the directory that the aligned read file is located.
`byChr`	Construct separate bin-level file for each chromosome? Possible values are `TRUE` or `FALSE`. If `byChr=FALSE`, all chromosomes are exported into one file. Default is `FALSE`.
`fragLen`	Average fragment length. Default is 200.
`binSize`	Size of bins. By default, bin size equals to `fragLen` (average fragment length).
`capping`	Maximum number of reads allowed to start at each nucleotide position. To avoid potential PCR amplification artifacts, the maximum number of reads that can start at a nucleotide position is capped at `capping`. Capping is not applied if non-positive `capping` is used. Default is 0 (no capping).
`perl`	Name of the perl executable to be called. Default is `"perl"`.

Details

Bin-level files are constructred from the aligned read file and exported to `outfileLoc`. If `byChr=FALSE`, bin-level files are named as `[infileName]_fragL[fragLen]_bin[binSize].txt`, If `byChr=TRUE`, bin-level files are named as `[chrID]_[infileName]_fragL[fragLen]_bin[binSize].tx` where `[chrID]` is chromosome ID that reads align to. These chromosome IDs are extracted from the aligned read file. Constructed bin-level files can be loaded into the R environment using the method `readBins`.

`constructBins` currently supports the following aligned read file formats: Eland result (`"eland_result"`), Eland extended (`"eland_extended"`), Eland export (`"eland_export"`), default Bowtie (`"bowtie"`), SAM (`"sam"`), and BED (`"bed"`). This method assumes that these aligned read files are obtained from single-end tag (SET) experiments and retains only reads mapping uniquely to the reference genome.

Value

Processed bin-level files are exported to the directory specified in `outfileLoc`.

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

`readBins`, `BinData`.

Examples

```
## Not run:
constructBins( infileLoc="/scratch/eland/",
    infileName="STAT1_eland_results.txt",
    fileFormat="eland_result", outfileLoc=infileLoc,
    byChr=FALSE, fragLen=200, binSize=fragLen, capping=0 )

## End(Not run)
```

---

estimates *Extract estimates of the fitted MOSAiCS model*

Description

Extract estimates from `MosaicsFit` class object, which is a fitted MOSAiCS model.

Usage

```
estimates( object, ... )
## S4 method for signature 'MosaicsFit'
estimates( object )
```

Arguments

`object`	Object of class `MosaicsFit`, which represents fitted MOSAiCS model obtained using method `mosaicsFit`.
`...`	Other parameters to be passed through to generic `estimates`.

Value

Returns a list with components:

`pi0`	Mixing proportion of background component and signal components.
`a`	Parameter for background component.
`betaEst`	Parameter for background component (coefficient estimates).
`muEst`	Parameter for background component.
`b`	Parameter for one-signal-componenet model.
`c`	Parameter for one-signal-componenet model.
`p1`	Parameter for two-signal-componenet model (mixing proportion of signal components).
`b1`	Parameter for two-signal-componenet model (the first signal component).
`c1`	Parameter for two-signal-componenet model (the first signal component).
`b2`	Parameter for two-signal-componenet model (the second signal component).
`c2`	Parameter for two-signal-componenet model (the second signal component).
`analysisType`	Analysis type. Possible values are "OS" (one-sample analysis), "TS" (two-sample analysis using mappability and GC content), and "IO" (two-sample analysis without using mappability and GC content).

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

`mosaicsFit`, `MosaicsFit`.

Examples

```
## Not run:
library(mosaicsExample)
data(exampleFit)

estimates(exampleFit)

## End(Not run)
```

export *Export peak calling results to text files*

Description

Export peak calling results to text files in TXT, BED, or GFF file format.

Usage

```
export(object, ...)
## S4 method for signature 'MosaicsPeak'
export( object, type=NA, fileLoc=NA, fileName=NA )
```

Arguments

object	Object of class MosaicsPeak, peak calling results obtained using method mosaicsPeak.
type	File format. Possible values are "txt", "bed", and "gff". See Details.
fileLoc	Directory of the exported file.
fileName	Name of the exported file.
...	Other parameters to be passed through to generic export.

Details

TXT file format (type="txt") exports peak calling results in the most informative way. Columns include chromosome ID, peak start position, peak end position, peak width, average posterior probability, minimum posterior probability, average ChIP tag count, maximum ChIP tag count, average input tag count, average input tag count scaled by sequencing depth, average log base 2 ratio of ChIP over input tag counts, average mappability score, and average GC content score in each peak. type="bed" and type="gff" export peak calling results in standard BED and GFF file formats, respectively, where score is the average ChIP tag counts in each peak. If no peak is detected, files will not be exported.

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

mosaicsPeak, MosaicsPeak.

Examples

```
## Not run:
library(mosaicsExample)
data(exampleFit)

examplePeak <- mosaicsPeak( exampleFit, signalModel = "2S", FDR = 0.05 )
export( examplePeak, type = "txt", fileLoc = "./", fileName = "TSpeakList.txt" )
export( examplePeak, type = "bed", fileLoc = "./", fileName = "TSpeakList.bed" )
export( examplePeak, type = "gff", fileLoc = "./", fileName = "TSpeakList.gff" )

## End(Not run)
```

`mosaics-package` *MOSAiCS (MOdel-based one and two Sample Analysis and Inference for ChIP-Seq)*

Description

This package provides functions for fitting MOSAiCS, a statistical framework to analyze one-sample or two-sample ChIP-seq data.

Details

Package:	mosaics
Type:	Package
Version:	1.2.5
Date:	2012-02-15
License:	GPL (>= 2)
LazyLoad:	yes

This package contains three main classes, `BinData`, `MosaicsFit`, and `MosaicsPeak`, which represent bin-level ChIP-seq data, MOSAiCS model fit, and MOSAiCS peak calling results, respectively. This package contains three main methods,`readBins`, `mosaicsFit`, and `mosaicsPeak`. `constructBins` method constructs bin-level files from the aligned read file. `readBins` method imports bin-level data and construct `BinData` class object. `mosaicsFit` method fits MOSAiCS model using `BinData` class object and constructs `MosaicsFit` class object. `mosaicsPeak` method calls peaks using `MosaicsFit` class object and construct `MosaicsPeak` class object. `MosaicsPeak` class object can be exported as text files or transformed into data frame and can be used for the downstream analysis. This package also provides methods for simple exploratory analysis.

The `mosaics` package companion website, http://www.stat.wisc.edu/~keles/Software/mosaics/, provides preprocessing scripts, preprocessed files for diverse reference genomes, and easy-to-follow instructions. We encourage questions or requests regarding `mosaics` package to be posted on our Google group, http://groups.google.com/group/mosaics_user_group. Please check the vignette for further details on the `mosaics` package and these websites.

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

Maintainer: Dongjun Chung <chungdon@stat.wisc.edu>

References

Kuan, PF, D Chung, G Pan, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

constructBins, readBins, mosaicsFit, mosaicsPeak , BinData, MosaicsFit, MosaicsPeak.

Examples

```
## Not run:
library(mosaicsExample)
exampleBinData <- readBins( type=c("chip","input","M","GC","N"),
    fileName=c( system.file("extdata/chip_chr21.txt", package="mosaicsExample"),
    system.file("extdata/input_chr21.txt", package="mosaicsExample"),
    system.file("extdata/M_chr21.txt", package="mosaicsExample"),
    system.file("extdata/GC_chr21.txt", package="mosaicsExample"),
    system.file("extdata/N_chr21.txt", package="mosaicsExample") ) )

exampleBinData
print(exampleBinData)[1:10, ]
plot(exampleBinData)
plot( exampleBinData, plotType="M" )
plot( exampleBinData, plotType="GC" )
plot( exampleBinData, plotType="input" )
plot( exampleBinData, plotType="M|input" )
plot( exampleBinData, plotType="GC|input" )

exampleFit <- mosaicsFit( exampleBinData, analysisType="TS" )

exampleFit
plot(exampleFit)
estimates(exampleFit)

examplePeak <- mosaicsPeak( exampleFit, signalModel = "2S", FDR = 0.05 )

examplePeak
print(examplePeak)[1:10, ]
export( examplePeak, type = "txt", fileLoc = "./", fileName = "TSpeakList.txt" )
export( examplePeak, type = "bed", fileLoc = "./", fileName = "TSpeakList.bed" )
export( examplePeak, type = "gff", fileLoc = "./", fileName = "TSpeakList.gff" )

## End(Not run)
```

`mosaicsFit` *Fit MOSAiCS model*

Description

Fit one-sample or two-sample MOSAiCS model with one signal component and two signal components.

Usage

```
mosaicsFit( object, ... )
## S4 method for signature 'BinData'
mosaicsFit( object, analysisType=NULL, bgEst=NA,
    k=3, meanThres=NA, s=2, d=0.25, truncProb=0.999 )
```

Arguments

`object`	Object of class `BinData`, bin-level ChIP-seq data imported using method `readBins`.
`analysisType`	Analysis type. Possible values are "OS" (one-sample analysis), "TS" (two-sample analysis using mappability and GC content), and "IO" (two-sample analysis without using mappability and GC content). If `analysisType` is not specified, this method tries to guess its best for `analysisType`, based on the data provided.
`bgEst`	Parameter to determine background estimation approach. Possible values are "matchLow" (estimation using bins with low tag counts) and "rMOM" (estimation using robust method of moment (MOM)). If `bgEst` is not specified, this method tries to guess its best for `bgEst`, based on the data provided.
`k`	Parameter for estimating background distribution. It is not recommended for user to change this value.
`meanThres`	Parameter for estimating background distribution. Default is 1 for `analysisType="TS"` and 0 for `analysisType="OS"`. Not relevant when `analysisType="IO"`.
`s`	Parameter for estimating background distribution. Relevant only when `analysisType="TS"`. Default is 2.
`d`	Parameter for estimating background distribution. Relevant only when `analysisType="TS"` or `analysisType="IO"`. Default is 0.25.
`truncProb`	Parameter for estimating background distribution. It is not recommended for user to change this value.
`...`	Other parameters to be passed through to generic `mosaicsFit`.

Details

The imported data type constraints the analysis that can be implemented. If there is no control data (i.e., `type=c("chip", "M", "GC", "N")` was used in method `readBins`), only one-sample analysis (`analysisType="OS"`) is permitted. If mappability score, GC content score, or sequence ambiguity score are missing (i.e., either `type=c("chip", "input")` or `type=c("chip", "input", "N")` was used in method `readBins`), only two-sample analysis without using mappability and GC content (`analysisType="IO"`) is possible. If control data is available with mappability score, GC content score, or sequence ambiguity score, (i.e., `type=c("chip", "input", "M", "GC", "N")` was used in method `readBins`), user can do either one- or two-sample analysis (`analysisType="OS"`, `analysisType="TS"`, or `analysisType="IO"`).

`meanThres`, `s`, and `d` are the tuning parameters for estimating background distribution. The vignette and Kuan et al. (2010) provide further details about these tuning parameters. Do not change `k` or `truncProb`.

Value

Construct `MosaicsFit` class object.

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

`readBins`, `MosaicsFit`.

Examples

```
## Not run:
library(mosaicsExample)
data(exampleBinData)

exampleFit <- mosaicsFit( exampleBinData, analysisType="TS" )

## End(Not run)
```

---

mosaicsPeak *Call peaks based on fitted MOSAiCS model*

Description

Call peaks using `MosaicsFit` class object, which is a fitted MOSAiCS model.

Usage

```
mosaicsPeak( object, ... )
## S4 method for signature 'MosaicsFit'
mosaicsPeak( object, signalModel="2S", FDR=0.05, maxgap=200, minsize=50, thres=1
```

Arguments

`object`	Object of class `MosaicsFit`, a fitted MOSAiCS model obtained using function `mosaicsFit`.
`signalModel`	Signal model. Possible values are "1S" (one-signal-component model) and "2S" (two-signal-component model). Default is "2S".
`FDR`	False discovery rate. Default is 0.05.
`maxgap`	Initial nearby peaks are merged if the distance (in bp) between them is less than `maxgap`. Default is 200.
`minsize`	An initial peak is removed if its width is narrower than `minsize`. Default is 50.

thres	A bin within initial peak is removed if its ChIP tag counts are less than thres. Default is 10.
...	Other parameters to be passed through to generic mosaicsPeak.

Details

When peaks are called, proper signal model needs to be specified. The optimal choice of the number of signal components depends on the characteristics of ChIP-seq data. In order to support users in the choice of optimal signal model, Bayesian Information Criterion (BIC) values and Goodness of Fit (GOF) plot are provided. BIC values and GOF plot can be obtained by applying show and plot methods to the MosaicsFit class object, which is a fitted MOSAiCS model. maxgap, minsize, and thres are for refining initial peaks called using specified signalModel and FDR.

If you use a bin size shorter than the average fragment length of the experiment, set maxgap to the average fragment length and minsize to the bin size. If you set the bin size to the average fragment length or if bin size is larger than the average fragment length, set maxgap to the average fragment length and minsize to a value smaller than the average fragment length. See the vignette for further details.

Value

Construct MosaicsPeak class object.

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, G Pan, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

mosaicsFit , MosaicsPeak, MosaicsFit.

Examples

```
## Not run:
library(mosaicsExample)
data(exampleFit)

examplePeak <- mosaicsPeak( exampleFit, signalModel = "2S", FDR = 0.05 )

## End(Not run)
```

mosaicsRunAll *Analyze ChIP-seq data using the MOSAiCS framework*

Description

Construct bin-level ChIP-sep data from aligned read files of ChIP and control samples, fit MOSAiCS model, call peaks, and export peak calling results and reports for diagnostics.

Usage

```
mosaicsRunAll( chipDir=NULL, chipFileName=NULL, chipFileFormat=NULL,
    controlDir=NULL, controlFileName=NULL, controlFileFormat=NULL,
    binfileDir=NULL, peakDir=NULL, peakFileName=NULL, peakFileFormat=NULL,
    reportSummary=FALSE, summaryDir=NULL, summaryFileName=NULL,
    reportExploratory=FALSE, exploratoryDir=NULL, exploratoryFileName=NULL,
    reportGOF=FALSE, gofDir=NULL, gofFileName=NULL, byChr=FALSE,
    excludeChr=NULL, FDR=0.05, fragLen=200, binSize=fragLen, capping=0,
    analysisType="IO", bgEst=NA, d=0.25,
    signalModel="BIC", maxgap=fragLen, minsize=50, thres=10, parallel=FALSE, nCo
```

Arguments

chipDir	Directory of the aligned read file of ChIP sample to be processed.
chipFileName	Name of the aligned read file of ChIP sample to be processed.
chipFileFormat	Format of the aligned read file of ChIP sample to be processed. Currently, mosaicsRunAll permits the following aligned read file formats: "eland_result" (Eland result), "eland_extended" (Eland extended), "eland_export" (Eland export), "bowtie" (default Bowtie), and "sam" (SAM).
controlDir	Directory of the aligned read file of control sample to be processed.
controlFileName	Name of the aligned read file of control sample to be processed.
controlFileFormat	Format of the aligned read file of control sample to be processed. Currently, mosaicsRunAll permits the following aligned read file formats: "eland_result" (Eland result), "eland_extended" (Eland extended), "eland_export" (Eland export), "bowtie" (default Bowtie), and "sam" (SAM).
binfileDir	Directory to store processed bin-level files.
peakDir	Directory to store the peak list generated from the analysis.
peakFileName	Name of the peak list generated from the analysis.
peakFileFormat	Format of the peak list generated from the analysis. Possible values are "txt", "bed", and "gff".
reportSummary	Report the summary of model fitting and peak calling? Possible values are TRUE and FALSE. Default is FALSE.
summaryDir	Directory to store the summary report of model fitting and peak calling.

summaryFileName
: Name of the summary report of model fitting and peak calling. The summary report is a text file.

reportExploratory
: Report the exploratory analysis plots? Possible values are TRUE and FALSE. Default is FALSE.

exploratoryDir
: Directory to store the exploratory analysis plots.

exploratoryFileName
: Name of the file for exploratory analysis plots. The exploratory analysis results are exported as PDF.

reportGOF
: Report the goodness of fit (GOF) plots? Possible values are TRUE and FALSE. Default is FALSE.

gofDir
: Directory to store the goodness of fit (GOF) plots.

gofFileName
: Name of the file for goodness of fit (GOF) plots. The exploratory analysis results are exported as PDF.

byChr
: Analyze ChIP-seq data for each chromosome separately or analyze it genome-wide? Possible values are TRUE or FALSE. byChr=TRUE and byChr=FALSE mean chromosome-wise and genome-wide analysis, respectively. Default is FALSE (genome-wide analysis).

excludeChr
: Vector of chromosomes that are excluded from the analysis.

FDR
: False discovery rate. Default is 0.05.

fragLen
: Average fragment length. Default is 200.

binSize
: Size of bins. By default, bin size equals to fragLen (average fragment length).

capping
: Maximum number of reads allowed to start at each nucleotide position. To avoid potential PCR amplification artifacts, the maximum number of reads that can start at a nucleotide position is capped at capping. Capping is not applied if non-positive capping is used. Default is 0 (no capping).

analysisType
: Analysis type. Currently, only "IO" is supported.

bgEst
: Parameter to determine background estimation approach. Possible values are "matchLow" (estimation using bins with low tag counts) and "rMOM" (estimation using robust method of moment (MOM)). If bgEst is not specified, this method tries to guess its best for bgEst, based on the data provided.

d
: Parameter for estimating background distribution. Default is 0.25.

signalModel
: Signal model. Possible values are "BIC" (automatic model selection using BIC), "1S" (one-signal-component model), and "2S" (two-signal-component model). Default is "BIC".

maxgap
: Initial nearby peaks are merged if the distance (in bp) between them is less than maxgap. By default, maxgap equals to fragLen (average fragment length).

minsize
: An initial peak is removed if its width is narrower than minsize. Default is 50.

thres
: A bin within initial peak is removed if its ChIP tag counts are less than thres. Default is 10.

parallel
: Utilize multiple CPUs for parallel computing using "multicore" package? Possible values are TRUE (use "multicore") or FALSE (not use "multicore"). Default is FALSE (not use "multicore").

nCore
: Number of maximum number of CPUs used for the analysis. Default is 8.

Details

This method implements the work flow to analyze ChIP-seq data using the MOSAiCS framework. It imports aligned read files of ChIP and control samples, process them into bin-level files, fit MOSAiCS model, call peaks, and export the peak lists. This method is a wrapper function of `constructBins`, `readBins`, `mosaicsFit`, `mosaicsPeak`, `export`, and methods of classes `BinData`, `MosaicsFit`, and `MosaicsPeak`.

See the vignette of the package for the illustration of the work flow and the description of employed methods and their options. Exploratory analysis plots and goodness of fit (GOF) plots are generated using the methods `plot` of the classes `BinData` and `MosaicsFit`, respectively. See the help of `constructBins` for details of the options `chipFileFormat`, `controlFileFormat`, `byChr`, `fragLen`, `binSize`, and `capping`. See the help of `readBins` for details of the option `excludeChr`. See the help of `mosaicsFit` for details of the options `analysisType`, `bgEst`, and `d`. See the help of `mosaicsPeak` for details of the options `FDR`, `signalModel`, `maxgap`, `minsize`, and `thres`. See the help of `export` for details of the option `peakFileFormat`.

When the data contains multiple chromosomes, parallel computing can be utilized for faster preprocessing and model fitting if `parallel=TRUE` and `multicore` package is installed. `nCore` determines number of CPUs used for parallel computing.

Value

Processed bin-level files are exported to the directory specified in `binfileDir`. If `byChr=FALSE` (genome-wide analysis), one bin-level file is exported for each of ChIP and control samples, where file names are `[chipFileName]_fragL[fragLen]_bin[binSize].txt` and `[controlFileName]_frag` respectively. If `byChr=TRUE` (chromosome-wise analysis), bin-level files are exported for each chromosome of each of ChIP and control samples, where file names are `[chrID]_[chipFileName]_fragL[frag` and `[chrID]_[controlFileName]_fragL[fragLen]_bin[binSize].txt` (`[chrID]` is chromosome ID that reads align to). The peak list generated from the analysis are exported to the directory specified in `peakDir` with the file name specified in `peakFileName`. If `reportSummary=TRUE`, the summary of model fitting and peak calling is exported to the directory specified in `summaryDir` with the file name specified in `summaryFileName` (text file). If `reportExploratory=TRUE`, the exploratory analysis plots are exported to the directory specified in `exploratoryDir` with the file name specified in `exploratoryFileName` (PDF file). If `reportGOF=TRUE`, the goodness of fit (GOF) plots are exported to the directory specified in `gofDir` with the file name specified in `gofFileName` (PDF file).

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, G Pan, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

`constructBins`, `readBins`, `mosaicsFit`, `mosaicsPeak`, `export`, `BinData`, `MosaicsFit`, `MosaicsPeak`.

Examples

```
## Not run:
# minimal input (without any reports for diagnostics)

mosaicsRunAll(
    chipDir="/scratch/eland/",
    chipFileName="STAT1_eland_results.txt",
    chipFileFormat="eland_result",
    controlDir="/scratch/eland/",
    controlFileName="input_eland_results.txt",
    controlFileFormat="eland_result",
    binfileDir="/scratch/bin/",
    peakDir="/scratch/peak/",
    peakFileName="STAT1_peak_list.txt",
    peakFileFormat="txt" )

# generate all reports for diagnostics

mosaicsRunAll(
    chipDir="/scratch/eland/",
    chipFileName="STAT1_eland_results.txt",
    chipFileFormat="eland_result",
    controlDir="/scratch/eland/",
    controlFileName="input_eland_results.txt",
    controlFileFormat="eland_result",
    binfileDir="/scratch/bin/",
    peakDir="/scratch/peak/",
    peakFileName="STAT1_peak_list.txt",
    peakFileFormat="txt",
    reportSummary=TRUE,
    summaryDir="/scratch/reports/",
    summaryFileName="mosaics_summary.txt",
    reportExploratory=TRUE,
    exploratoryDir="/scratch/reports/",
    exploratoryFileName="mosaics_exploratory.pdf",
    reportGOF=TRUE,
    gofDir="/scratch/reports/",
    gofFileName="mosaics_GOF.pdf",
    byChr=FALSE,
    FDR=0.05,
    fragLen=200,
    capping=0,
    parallel=FALSE,
    nCore=8 )

## End(Not run)
```

---

readBins *Import bin-level ChIP-sep data*

Description

Import and preprocess all or subset of bin-level ChIP-sep data, including ChIP data, control data, mappability score, GC content score, and sequence ambiguity score.

Usage

```
readBins( type = c("chip", "M", "GC", "N"), fileName = NULL,
    excludeChr=NULL, dataType = "unique", rounding = 100, parallel=FALSE, nCore=
```

Arguments

type	Character vector indicating data types to be imported. This vector can contain "chip" (ChIP data), "input" (input data), "M" (mappability score), "GC" (GC content score), and "N" (sequence ambiguity score). Currently, readBins permits only the following combinations: c("chip", "input", "M", "GC", "N"),c("chip", "M", "GC", "N"),c("chip", "input", "N"), and c("chip", "input"). Default is c("chip", "M", "GC", "N").
fileName	Character vector of file names, each of which matches each element of type. type and fileName should have the same length and corresponding elements in two vectors should appear in the same order.
excludeChr	Vector of chromosomes that are excluded from the analysis.
dataType	How reads were processed? Possible values are either "unique" (only uniquely aligned reads were retained) or "multi" (reads aligned to multiple locations were also retained).
rounding	How are mappability score and GC content score rounded? Default is 100 and this indicates rounding of mappability score and GC content score to the nearest hundredth.
parallel	Utilize multiple CPUs for parallel computing using "multicore" package? Possible values are TRUE (use "multicore") or FALSE (not use "multicore"). Default is FALSE (not use "multicore").
nCore	Number of CPUs when parallel computing is utilized.

Details

Bin-level ChIP and input data can be generated from the aligned read files for your samples (e.g., files obtained from the ELAND aligner) using the method constructBins. In mosaics package companion website, http://www.stat.wisc.edu/~keles/Software/mosaics/, we provide preprocessed mappability score, GC content score, and sequence ambiguity score files for diverse reference genomes. Please check the website and the vignette for further details.

The imported data type constraints the analysis that can be implemented. If type=c("chip", "M", "GC", "N"), only one-sample analysis is permitted. If type=c("chip", "input") or c("chip", "input", "N"), only two-sample analysis without using mappability and GC content is possible. For type=c("chip", "input", "M", "GC", "N"), user can do all the one- or two-sample analysis. See also help page of mosaicsFit.

When the data contains multiple chromosomes, parallel computing can be utilized for faster preprocessing if parallel=TRUE and multicore package is installed. nCore determines number of CPUs used for parallel computing.

Value

Construct BinData class object.

Author(s)

Dongjun Chung, Pei Fen Kuan, Sunduz Keles

References

Kuan, PF, D Chung, G Pan, JA Thomson, R Stewart, and S Keles (2011), "A Statistical Framework for the Analysis of ChIP-Seq Data", *Journal of the American Statistical Association*, Vol. 106, pp. 891-903.

See Also

constructBins, mosaicsFit, BinData.

Examples

```
## Not run:
library(mosaicsExample)
exampleBinData <- readBins( type=c("chip","input","M","GC","N"),
    fileName=c( system.file("extdata/chip_chr21.txt", package="mosaicsExample"),
    system.file("extdata/input_chr21.txt", package="mosaicsExample"),
    system.file("extdata/M_chr21.txt", package="mosaicsExample"),
    system.file("extdata/GC_chr21.txt", package="mosaicsExample"),
    system.file("extdata/N_chr21.txt", package="mosaicsExample") ) )

## End(Not run)
```

Index