

mosaics

October 25, 2011

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

`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, JA Thomson, R Stewart, and S Keles (2010), "A Statistical Framework for the Analysis of CHIP-Seq Data", submitted (http://works.bepress.com/sunduz_keles/19/).

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.

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", submitted (http://works.bepress.com/sunduz_keles/19/).

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, JA Thomson, R Stewart, and S Keles (2010), "A Statistical Framework for the Analysis of CHIP-Seq Data", submitted (http://works.bepress.com/sunduz_keles/19/).

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", chrID = "
export( examplePeak, type = "bed", fileLoc = "./", fileName = "TSpeakList.bed", chrID = "
```

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

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

Value

Returns a list with components:

<code>pi0</code>	Mixing proportion of background component and signal components.
<code>a</code>	Parameter for background component.
<code>betaEst</code>	Parameter for background component (coefficient estimates).
<code>muEst</code>	Parameter for background component.
<code>b</code>	Parameter for one-signal-component model.
<code>c</code>	Parameter for one-signal-component model.
<code>p1</code>	Parameter for two-signal-component model (mixing proportion of signal components).
<code>b1</code>	Parameter for two-signal-component model (the first signal component).
<code>c1</code>	Parameter for two-signal-component model (the first signal component).
<code>b2</code>	Parameter for two-signal-component model (the second signal component).
<code>c2</code>	Parameter for two-signal-component model (the second signal component).
<code>analysisType</code>	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 (2010), "A Statistical Framework for the Analysis of ChIP-Seq Data", submitted (http://works.bepress.com/sunduz_keles/19/).

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, chrID=NA )
```

Arguments

<code>object</code>	Object of class <code>MosaicsPeak</code> , peak calling results obtained using method <code>mosaicsPeak</code> .
<code>type</code>	File format. Possible values are "txt", "bed", and "gff". See Details.
<code>fileLoc</code>	Directory of the exported file.
<code>fileName</code>	Name of the exported file.
<code>chrID</code>	Chromosome ID, which is included in the first column in the exported file, e.g. <code>chrID="chr21"</code> .
<code>...</code>	Other parameters to be passed through to generic <code>export</code> .

Details

TEXT file format (`type="txt"`) exports peak calling results in the most informative way. Columns include 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 (2010), "A Statistical Framework for the Analysis of ChIP-Seq Data", submitted (http://works.bepress.com/sunduz_keles/19/).

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", chrID = "chr1" )
export( examplePeak, type = "bed", fileLoc = "./", fileName = "TSpeakList.bed", chrID = "chr1" )
export( examplePeak, type = "gff", fileLoc = "./", fileName = "TSpeakList.gff", chrID = "chr1" )

## End(Not run)
```

mosaics-package *MOSAiCS (MOdel-based one and two Sample Analysis and Inference for*

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:   1.0.1
Date:      2011-05-05
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`. `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, JA Thomson, R Stewart, and S Keles (2010), "A Statistical Framework for the Analysis of ChIP-Seq Data", submitted (http://works.bepress.com/sunduz_keles/19/).

See Also

[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", chrID = " " )
export( examplePeak, type = "bed", fileLoc = "./", fileName = "TSpeakList.bed", chrID = " " )
export( examplePeak, type = "gff", fileLoc = "./", fileName = "TSpeakList.gff", chrID = " " )

## 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, k=3,
            meanThres=NA, s=2, d=0.25, truncProb=0.9999 )

```

Arguments

<code>object</code>	Object of class <code>BinData</code> , bin-level ChIP-seq data imported using method <code>readBins</code> .
<code>analysisType</code>	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 <code>analysisType</code> is not specified, this method tries to guess its best for <code>analysisType</code> , based on the data provided.
<code>k</code>	Parameter for estimating background distribution. It is not recommended for user to change this value.
<code>meanThres</code>	Parameter for estimating background distribution. Default is 1 for <code>analysisType="TS"</code> and 0 for <code>analysisType="OS"</code> . Not relevant when <code>analysisType="IO"</code> .
<code>s</code>	Parameter for estimating background distribution. Relevant only when <code>analysisType="TS"</code> . 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 (2010), "A Statistical Framework for the Analysis of ChIP-Seq Data", submitted (http://works.bepress.com/sunduz_keles/19/).

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

<code>object</code>	Object of class <code>MosaicsFit</code> , a fitted MOSAiCS model obtained using function <code>mosaicsFit</code> .
<code>signalModel</code>	Signal model. Possible values are "1S" (one-signal-component model) and "2S" (two-signal-component model). Default is "2S".
<code>FDR</code>	False discovery rate. Default is 0.05.
<code>maxgap</code>	Initial nearby peaks are merged if the distance (in bp) between them is less than <code>maxgap</code> . Default is 200.
<code>minsize</code>	An initial peak is removed if its width is narrower than <code>minsize</code> . Default is 50.
<code>thres</code>	A bin within initial peak is removed if its ChIP tag counts are less than <code>thres</code> . Default is 10.
<code>...</code>	Other parameters to be passed through to generic <code>mosaicsPeak</code> .

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, JA Thomson, R Stewart, and S Keles (2010), "A Statistical Framework for the Analysis of ChIP-Seq Data", submitted (http://works.bepress.com/sunduz_keles/19/).

See Also

`mosaicsFit`, `MosaicsPeak`, `MosaicsFit`.

Examples

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

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

## End(Not run)
```

readBins

Import bin-level ChIP-seq data

Description

Import and preprocess all or subset of bin-level ChIP-seq 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,
          dataType = "unique", rounding = 100 )
```

Arguments

<code>type</code>	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, <code>readBins</code> permits only the following combinations: <code>c("chip", "input", "M", "GC", "N")</code> , <code>c("chip", "M", "GC", "N")</code> , <code>c("chip", "input", "N")</code> , and <code>c("chip", "input")</code> . Default is <code>c("chip", "M", "GC", "N")</code> .
<code>fileName</code>	Character vector of file names, each of which matches each element of <code>type</code> . <code>type</code> and <code>fileName</code> should have the same length and corresponding elements in two vectors should appear in the same order.
<code>dataType</code>	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).
<code>rounding</code>	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.

Details

Bin-level ChIP and input data can be generated from the aligned files for your samples (e.g., files obtained from the ELAND aligner) using the scripts we provide. You can download these scripts from the `mosaics` package companion website, <http://www.stat.wisc.edu/~keles/Software/mosaics/>. In this website, we also 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`.

Value

Construct `BinData` class 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", submitted (http://works.bepress.com/sunduz_keles/19/).

See Also

`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

*Topic classes

BinData-class, 1
MosaicsFit-class, 2
MosaicsPeak-class, 4

*Topic methods

estimates, 5
export, 6
mosaicsFit, 9
mosaicsPeak, 11
readBins, 12

*Topic models

estimates, 5
export, 6
mosaicsFit, 9
mosaicsPeak, 11
readBins, 12

*Topic package

mosaics-package, 7

bdBin, MosaicsPeak-method
(*MosaicsPeak-class*), 4

BinData, 8, 13

BinData-class, 1

coord, BinData-method
(*BinData-class*), 1

empFDR, MosaicsPeak-method
(*MosaicsPeak-class*), 4

estimates, 3, 5

estimates, MosaicsFit-method
(*estimates*), 5

export, 4, 6

export, MosaicsPeak-method
(*export*), 6

gcContent, BinData-method
(*BinData-class*), 1

input, BinData-method
(*BinData-class*), 1

mappability, BinData-method
(*BinData-class*), 1

mosaics (*mosaics-package*), 7

mosaics-package, 7

MosaicsFit, 6, 8, 10, 12

mosaicsFit, 2, 3, 6, 8, 9, 12, 13

mosaicsFit, BinData-method
(*mosaicsFit*), 9

MosaicsFit-class, 2

MosaicsPeak, 7, 8, 12

mosaicsPeak, 3, 4, 7, 8, 11

mosaicsPeak, MosaicsFit-method
(*mosaicsPeak*), 11

MosaicsPeak-class, 4

plot, BinData, missing-method
(*BinData-class*), 1

plot, MosaicsFit, ANY-method
(*MosaicsFit-class*), 2

print, BinData-method
(*BinData-class*), 1

print, MosaicsFit-method
(*MosaicsFit-class*), 2

print, MosaicsPeak-method
(*MosaicsPeak-class*), 4

readBins, 2, 8, 10, 12

show, BinData-method
(*BinData-class*), 1

show, MosaicsFit-method
(*MosaicsFit-class*), 2

show, MosaicsPeak-method
(*MosaicsPeak-class*), 4

tagCount, BinData-method
(*BinData-class*), 1