Package 'GenomicInteractions'

April 10, 2015

Type Package

Title R package for handling genomic interaction data

Version 1.0.3

Date 2014-07-14

Author Harmston, N., Ing-Simmons, E., Perry, M., Baresic A., Lenhard B.

Maintainer Nathan Harmston <nathan.harmston07@csc.mrc.ac.uk>

Imports Rsamtools, GenomicRanges, IRanges, data.table, stringr, rtracklayer, ggplot2, gridExtra, methods, igraph, plotrix

Suggests knitr, BiocStyle, BSgenome.Hsapiens.UCSC.hg19, BSgenome.Mmusculus.UCSC.mm9

VignetteBuilder knitr

Description R package for handling Genomic interaction data, such as ChIA-PET/Hi-C, annotating genomic features with interaction information and producing various plots / statistics

biocViews Software,Infrastructure,DataImport,DataRepresentation

License GPL-3

Depends R (>= 2.10)

R topics documented:

GenomicInteractions-package *A package for looking at genomic interaction data.*

Description

A package for looking at genomic interaction data.

---

annotateAnchors *Annotate anchors*

Description

This function directly annotates a single set of anchors using the GRanges elementMetadata.

Usage

```
annotateAnchors(GIObject, oneOrTwo, name, dat)


  ## S4 method for signature GenomicInteractions,numeric,character,vector
annotateAnchors(GIObject,
  oneOrTwo, name, dat)
```

Arguments

`GIObject` A GenomicInteractions object

`oneOrTwo` An integer indicating which anchor to annotate

`name` Character. Will be used as a column name for the elementMetadata of the annotated anchor.

`dat` Vector of the same length as the GenomicInteractions object, containing data with which to annotate the object.

Value

invisible(1)

`annotateInteractions` *Annotate the interactions in a GenomicInteractions object*

Description

This function will annotate both anchors with a list of named GRanges objects. Each metadata column is labeled "name.id" and contains the id of the genomic interval(s) it overlaps. Anonymous lists will be given names "FEATURE#.id" where # is the position in the list.

Usage

```
annotateInteractions(GIObject, annotations)

## S4 method for signature GenomicInteractions,list
annotateInteractions(GIObject, annotations)
```

Arguments

`GIObject` A GenomicInteractions object to be annotated

`annotations` A list containing GRanges (or GRangesList) objects with which to annotate the GenomicInteractions object.

Details

For each anchor a "node.class" metadata column will also be added, containing the name of the list element which was *first* annotated to each range. Ranges with no overlaps will be classified as "distal". The identifiers for each individual feature/annotation are taken from either the name of the list item in the case of a GRangesList or from either the names of a the provided GRanges or an id column in its associated metadata.

Value

invisible(1)

Examples

```
data(hic_example_data)
data(mm9_refseq_promoters)
## Not run:
annotateInteractions(hic_example_data, list(promoter=mm9_refseq_promoters))

## End(Not run)
```

calculateDistances *Calculate interaction distances*

Description

This function takes a GenomicInteractions object and calculates the distances between the anchors according to the value of `method`. The distances returned follow the same convention as distance(x, y) in GenomicRanges where the distance between adjacent regions is 0. Note that if anchors are overlapping this method will print a warning and return the distance as 0.

Usage

```
calculateDistances(GIObject, method = "midpoint", floor = TRUE)

## S4 method for signature GenomicInteractions
calculateDistances(GIObject,
  method = "midpoint", floor = TRUE)
```

Arguments

`GIObject`	A GenomicInteractions object
`method`	Character vector indicating how to calculate distances, must be one of ‘midpoint’, ‘outer’, ‘inner’.
`floor`	A logical specifying whether to round down distances to nearest base pair or not. Default TRUE.

Value

An vector containing the distances between anchors/GRanges, NA if on different chromosomes, rounded down to the nearest bp.

Examples

```
library(BSgenome.Mmusculus.UCSC.mm9)
anchor.one = GRanges(c(”chr1”, ”chr1”, ”chr1”, ”chr1”), IRanges(c(10, 20, 30, 20), width=5), seqlengths=seqlengths
anchor.two = GRanges(c(”chr1”, ”chr1”, ”chr1”, ”chr2”), IRanges(c(100, 200, 300, 50), width=5), seqlengths=seqleng
test <- new(”GenomicInteractions”, experiment_name=”test”, description=”this is a test”,
                 genome_name=”BSgenome.Mmusculus.UCSC.mm9”, anchor_one = anchor.one,
           anchor_two = anchor.two, counts=as.integer(c(2,1,2,3)), pvalue=c(0.1, 0.3, 0.1, 0.08))
calculateDistances(test, method=”midpoint”)
```

capitalize *Capitalize first letter of string*

Description

This function will capitalize the first letter of each string in a character vector, and lowercase following letters.

Usage

```
capitalize(x)
```

Arguments

x	A character vector

Value

a string with the first letter capitalised

categoriseInteractions *Get the numbers of interaction types existing in your data*

Description

Get the numbers of interaction types existing in your data

Usage

```
categoriseInteractions(GIObject, node.classes = NULL, viewpoints = NULL)
```

Arguments

`GIObject` A GenomicInteractions object

`node.classes` Optional. All node.classes to include in the analysis. Default: all node classes.

`viewpoints` Optional. If set will only consider interactions where at least one anchor is of this node class. Default: all classes in node.classes.

Value

A data.frame.

Examples

```
data(hic_example_data)
categoriseInteractions(hic_example_data)
```

`export.bed12` *Export interactions in BED12 format.*

Description

Export interactions in BED12 format.

Usage

```
export.bed12(GIObject, fn = NULL)

## S4 method for signature GenomicInteractions
export.bed12(GIObject, fn = NULL)
```

Arguments

`GIObject` A GenomicInteractions object.

`fn` A filename to write the object to

Exports a GenomicInteractions object to BED12 format, and writes to a specified file. If filename is not specified, then a data.frame containing the information is returned. Please note some large datasets may take a long time to export.

Value

invisible(1) if outputting to file or a data.frame containing all of the corresponding information

export.bedpe — *Export interactions in BED Paired-End format.*

Description

#' Exports a GenomicInteractions object to BED-PE format, and writes to a specified file. If filename is not specified, then a data.frame containing the information is returned. The value of the score parameter defines which field is used to populate the score field.

Usage

```
export.bedpe(GIObject, fn = NULL, score = "counts")

## S4 method for signature GenomicInteractions
export.bedpe(GIObject, fn = NULL,
  score = "counts")
```

Arguments

`GIObject`	A GenomicInteractions object.
`fn`	A filename to write the interactions data to
`score`	Which metadata column to use as the score: counts, pvalue, fdr, normalised

Value

invisible(1) if outputting to file or a data.frame containing all of the corresponding information

export.igraph — *Export interactions to an igraph object.*

Description

Exports a GenomicInteractions object to graph.data.frame for use by igraph package. This uses unique anchors as nodes and generates edges between them. For the resulting graph to be easily interpretable, anchors should be non-overlapping. This should already be the case for HiC data (either binned or restriction fragments), however ChIA-PET data can contain overlapping anchors, which may need to be reduced to non-overlapping regions before graph export.

Usage

```
export.igraph(GIObject)

## S4 method for signature GenomicInteractions
export.igraph(GIObject)
```

Arguments

`GIObject`	A GenomicInteractions object.

Value

a graph.data.frame representation of the GenomicInteractions object

`findOverlaps` *Find overlaps between a GRanges and a GenomicInteractions object*

Description

This function calls findOverlaps separately on each anchor and returns a list. See 'findOverlaps' in the GenomicRanges package for detailed documentation for this function.

Usage

```
## S4 method for signature GenomicInteractions,GRanges
findOverlaps(query, subject,
  maxgap = 0L, minoverlap = 1L, type = c("any", "start", "end", "within",
  "equal"), select = c("all", "first", "last", "arbitrary"))

## S4 method for signature GRanges,GenomicInteractions
findOverlaps(query, subject,
  maxgap = 0L, minoverlap = 1L, type = c("any", "start", "end", "within",
  "equal"), select = c("all", "first", "last", "arbitrary"))
```

Arguments

`query`	GenomicInteractions or GRanges
`subject`	GRanges or GenomicInteractions
`maxgap,minoverlap,type,select`	See 'findOverlaps' in the IRanges package.

Value

A list containing Hits objects for anchors one and two.

GenomicInteractions *Function to create GenomicInteraction objects.*

Description

Function to create GenomicInteraction objects from a variety of files. The resulting objects contain information on which genomic regions are interacting with each other, and the number of counts supporting each interaction. It is also possible to store information on associated p-values and false-discovery rates (FDR). It is possible to create GenomicInteractions objects for various datasets including Hi-C and ChIA-PET. It is possible to read interactions from a variety of files including BAM files, bed files (BED12 and BEDPE) and from the output from standard processing pipelines, such as HOMER and ChIA-PET tool. GenomicInteractions objects can also be created using calls of the form `new("GenomicInteractions", ...)`. For hiclib, it expects the directory in which the files extracted using h5dictToTxt.py from the hdf5 file are located, where as for all of the other file types it expects the full filename.

Usage

```
GenomicInteractions(fn, type, experiment_name, description, gname)
```

Arguments

`fn` Filename or, if type="hiclib", folder

`type` One of "chiapet.tool", "chiapet.encode", "bed12", "bedpe", "hiclib", "homer", "bam".

`experiment_name` Experiment name.

`description` Description of experiment.

`gname` Genome name to use for constructing the GenomicInteractions object.

Value

a GenomicInteractions object

Examples

```
library(BSgenome.Hsapiens.UCSC.hg19)

k562.rep1 = GenomicInteractions(file.path(system.file(package=”GenomicInteractions”), ”extdata”, ”k562.rep1.clus
                                     type=”chiapet.tool”,
                                     experiment_name=”k562”,
                                     description=”k562 pol2 8wg16”,
                                     gname=”BSgenome.Hsapiens.UCSC.hg19”)

k562.rep1
```

GenomicInteractions-class

A S4 class to represent interactions between genomic regions.

Description

A S4 class to represent interactions between genomic regions.

Slots

`experiment_name` Character. Experiment name.

`description` Character. Longer description of experiment.

`genome_name` Character. Genome version for experiment data, should correspond to a BSgenome data package.

`anchor_one,anchor_two` GRanges. Set of anchors of interactions.

`counts` Numeric. Counts of reads supporting each interaction.

`normalised_counts` Numeric. Normalised counts of reads supporting each interaction.

`pvalue` Numeric. P-values for individual interactions.

`fdr` Numeric. FDRs for individual interactions.

This class is used to store information on which genomic regions are interacting with each other, the number of counts supporting each interaction, and associated p-values and false-discovery rates (FDR). Objects of this class contain information of the genomic coordinates of the interacting regions and the strength of these interactions, and associated metadata such as the name of the dataset and a brief description of the dataset. Interacting regions are stored as a pair of GenomicRanges: each set of anchor regions is stored as a separate GenomicRanges object, accessed by `getAnchorOne` and `getAnchorTwo`.

Examples

```
showClass("GenomicInteractions")

library(BSgenome.Mmusculus.UCSC.mm9)
anchor.one = GRanges(c("chr1", "chr1", "chr1", "chr1"), IRanges(c(10, 20, 30, 20), width=5), seqlengths=seqlengths
anchor.two = GRanges(c("chr1", "chr1", "chr1", "chr2"), IRanges(c(100, 200, 300, 50), width=5), seqlengths=seqleng
test <- new("GenomicInteractions", experiment_name="test", description="this is a test",
                 genome_name="BSgenome.Mmusculus.UCSC.mm9", anchor_one = anchor.one,
           anchor_two = anchor.two, counts=as.integer(c(2,1,2,3)), pvalue=c(0.1, 0.3, 0.1, 0.08))
```

getters *Functions to access data held in a GenomicInteractions object.*

Description

Use these functions to access data stored in each of the slots of a GenomicInteractions object.

Usage

```
name(GIObject)

anchorOne(GIObject)

anchorTwo(GIObject)

count(GIObject)

pValue(GIObject)

FDR(GIObject)

normalisedCount(GIObject)

description(GIObject)

genomeName(GIObject)

annotationFeatures(GIObject)

## S4 method for signature GenomicInteractions
name(GIObject)

## S4 method for signature GenomicInteractions
anchorOne(GIObject)

## S4 method for signature GenomicInteractions
anchorTwo(GIObject)

## S4 method for signature GenomicInteractions
count(GIObject)

## S4 method for signature GenomicInteractions
normalisedCount(GIObject)

## S4 method for signature GenomicInteractions
pValue(GIObject)
```

```
## S4 method for signature GenomicInteractions
FDR(GIObject)

## S4 method for signature GenomicInteractions
description(GIObject)

## S4 method for signature GenomicInteractions
genomeName(GIObject)

## S4 method for signature GenomicInteractions
annotationFeatures(GIObject)
```

Arguments

`GIObject`	A GenomicInteractions object

Value

For 'anchorOne' and 'anchorTwo', a GRanges. For 'counts', 'normalisedCount', pValue', 'FDR', a numeric vector with counts, normalised counts, p-values or FDRs for each interaction in the object. For 'description','name', and 'genomeName', a character vector with length 1. For 'annotationFeatures', a character vector of features with which the object was previously annotated, or 'NA' if the object is unannotated.

Examples

```
library(BSgenome.Mmusculus.UCSC.mm9)
anchor.one = GRanges(c("chr1", "chr1", "chr1", "chr1"), IRanges(c(10, 20, 30, 20), width=5), seqlengths=seqlengths
anchor.two = GRanges(c("chr1", "chr1", "chr1", "chr2"), IRanges(c(100, 200, 300, 50), width=5), seqlengths=seqleng
test <- new("GenomicInteractions", experiment_name="test", description="this is a test",
                 genome_name="BSgenome.Mmusculus.UCSC.mm9", anchor_one = anchor.one,
           anchor_two = anchor.two, counts=as.integer(c(2,1,2,3)), pvalue=c(0.1, 0.3, 0.1, 0.08))

name(test)
description(test)
anchorOne(test)
anchorTwo(test)
count(test)
pValue(test)
genomeName(test)
```

hg19.refseq.transcripts

Human Refseq transcripts from chr 17-18

Description

This dataset contains a subset of the transcripts from the Refseq annotation for mouse genome build hg19 See the ChIA-PET analysis vignette (`vignettes(GenomicInteractions)`) for more information on how this dataset was created.

Usage

```
data(hg19.refseq.transcripts)
```

Format

A GRanges object with length 2441.

`hic_example_data` *Example HiC dataset*

Description

This dataset contains HiC data from Seitan et al. 2013. The data was analysed using HOMER (Heinz et al. 2010) at a resolution of 100kb to find significant interactions. This example dataset has been filtered to retain only interactions on chromosomes 14 and 15 with a FDR < 0.1. The data has also been annotated for overlaps with Refseq promoters. See the HiC analysis vignette (`vignettes(GenomicInteractions)`) for more information on how this dataset was created.

Usage

```
data(hic_example_data)
```

Format

A GenomicInteractions object with length 8171.

Value

GenomicInteractions object

References

Seitan, V. C. et al. Cohesin-based chromatin interactions enable regulated gene expression within pre-existing architectural compartments. Genome Res. 23, 2066-77 (2013).

Heinz S, Benner C, Spann N, Bertolino E et al. Simple Combinations of Lineage-Determining Transcription Factors Prime cis-Regulatory Elements Required for Macrophage and B Cell Identities. Mol Cell 2010 May 28;38(4):576-589.

is.pp *Interaction Type Helpers*

Description

Functions to classify interactions within GenomicInteractions objects.

- "isInteractionType" takes two character arguments which are annotated node classes and returns interactions between them.
- "is.pp", "is.pd" etc. are bindings for common annotations:

  p promoter

  d distal

  t terminator
- "is.trans" & "is.cis" select trans-chromosomal and intra-chromosomal interactions, respectively

Usage

```
is.pp(GIObject)

is.pd(GIObject)

is.pt(GIObject)

is.dd(GIObject)

is.dt(GIObject)

is.tt(GIObject)

isInteractionType(GIObject, x, y)

is.trans(GIObject)

is.cis(GIObject)

## S4 method for signature GenomicInteractions
is.pp(GIObject)

## S4 method for signature GenomicInteractions
is.pd(GIObject)

## S4 method for signature GenomicInteractions
is.pt(GIObject)

## S4 method for signature GenomicInteractions
```

```
is.dd(GIObject)

## S4 method for signature GenomicInteractions
is.dt(GIObject)

## S4 method for signature GenomicInteractions
is.tt(GIObject)

## S4 method for signature GenomicInteractions
isInteractionType(GIObject, x, y)

## S4 method for signature GenomicInteractions
is.trans(GIObject)

## S4 method for signature GenomicInteractions
is.cis(GIObject)
```

Arguments

`GIObject`	A GenomicInteractions object
`x,y`	Names of annotated node classes

Value

A logical vector

`length,GenomicInteractions-method`

Get the length of a GenomicInteractions GIObject

Description

Get the length of a GenomicInteractions GIObject

Usage

```
## S4 method for signature GenomicInteractions
length(x)
```

Arguments

`x`	GenomicInteractions GIObject

Value

A numeric vector containing the length of the GIObject

`mm9_refseq_promoters` *Mouse Refseq promoters from chr 14-15*

Description

This dataset contains a subset of the promoters from the Refseq annotation for mouse genome build mm9. See the HiC analysis vignette (`vignettes(GenomicInteractions)`) for more information on how this dataset was created.

Usage

```
data(mm9_refseq_promoters)
```

Format

A GRanges object with length 2441.

`plotCisTrans` *Plots the percentages of cis and trans interactions for a GenomicInteractions object as a donut plot.*

Description

Plots the percentages of cis and trans interactions for a GenomicInteractions object as a donut plot.

Usage

```
plotCisTrans(GIObject)
```

Arguments

`GIObject`	A GenomicInteractions object

Value

A ggplot2 plot

Examples

```
data(hic_example_data)
plotCisTrans(hic_example_data)
```

plotCounts

Plot a bar chart of the number of interactions supported by different numbers of reads in your data.

Description

Plot a bar chart of the number of interactions supported by different numbers of reads in your data.

Usage

```
plotCounts(GIObject, normalise = FALSE, cut = 10)
```

Arguments

`GIObject` A GenomicInteractions object.

`normalise` Logical. If TRUE, plots proportion of total reads instead of count.

`cut` Numeric, can be NULL. Default: 10. All interactions with counts > cut are consolidated into a single category.

Value

A ggplot2 plot

Examples

```
data(hic_example_data)
plotCounts(hic_example_data)
plotCounts(hic_example_data, normalise=TRUE)
```

plotDists

Plots a histogram of interaction distances for a GenomicInteractions Object

Description

Plots a histogram of interaction distances for a GenomicInteractions Object

Usage

```
plotDists(GIObject, breaks = c(0, 1000, 5000, 10000, 50000, 1e+05, 5e+05,
  1e+06, 2e+06), method = "midpoint")
```

Arguments

`GIObject` A GenomicInteractions object

`breaks` A numeric vector of breaks for the histogram

`method` Method used for distance between anchors. Passed to calculateDistances. One of "midpoint", "inner", or "outer".

Value

A ggplot2 plot

Examples

```
data(hic_example_data)
plotDists(hic_example_data)
```

---

plotInteractionAnnotations

Plot a donut plot of interaction types for an annotated GenomicInteractions object

---

Description

Plot a donut plot of interaction types for an annotated GenomicInteractions object

Usage

```
plotInteractionAnnotations(GIObject, node.classes = NULL, viewpoints = NULL,
  other = 0, keep.order = FALSE, legend = FALSE)
```

Arguments

`GIObject` A GenomicInteractions object

`node.classes` Optional. All node.classes to include in the analysis. Default: all node classes.

`viewpoints` Optional. If set will only consider interactions where at least one anchor is of this node class. Default: all classes in node.classes.

`other` Optional. Interaction types making up fewer than "other" percent of the total interactions will be consolidated into a single "other" category.

`keep.order` Optional. Logical. Keep original order of node.classes for plotting or not. Default: FALSE, alphabetical order.

`legend` Optional. Logical. If TRUE, legend is plotted to right of donut plot. If FALSE, donut plot is annotated with category names.

Value

A ggplot2 plot

Examples

```
data(hic_example_data)
plotInteractionAnnotations(hic_example_data)
```

---

plotRegion *Plot interactions within a specified region.*

Description

This is function allows the plotting of an interactions between annotated features in a specified area. The resulting plot shows unique interactions as curves between interaction anchor points with the number of counts supporting that interaction proportional to the thickness of that line. It is also possible to add cis-interactions which are not within the window/region and to also plot regions that are involved in trans-interactions. Plotting the data this way makes it possible to examine a region and easily examine which regions are highly interacting with each other. It is not recommended to use this style of plot to examine regions larger than 5Mb.

Usage

```
plotRegion(GIObject, region, annotation.features, annotation.cols = NULL,
  reduce.anchors = TRUE, plot.trans = TRUE, plot.cis = TRUE,
  order.cis = TRUE, plot.cis.names = TRUE, plot.header = TRUE,
  plot.lines = TRUE, anchor.col = "darkred", plot.ids = FALSE)

## S4 method for signature GenomicInteractions,GRanges,list
plotRegion(GIObject, region,
  annotation.features, annotation.cols = NULL, reduce.anchors = TRUE,
  plot.trans = TRUE, plot.cis = TRUE, order.cis = TRUE,
  plot.cis.names = TRUE, plot.header = TRUE, plot.lines = TRUE,
  anchor.col = "darkred", plot.ids = FALSE)
```

Arguments

`GIObject`	GenomicInteractions object
`region`	A GRanges specifying the genomic region to plot
`annotation.features`	a list of GRanges specifying the features within the region to plot
`annotation.cols`	a named vector specifying which colour to plot the individual tracks
`reduce.anchors`	a logical specifying whether to reduce the anchor GRanges
`plot.trans`	a logical specifying whether to show trans-interactions
`plot.cis`	a logical specifying whether to plot cis-interactions that are outside of the specified region
`order.cis`	logical specifying whether to order cis-interactions by their distances to the specified region

`plot.cis.names` a logical specifying whether to plot textual information on the other anchor region of cis-interactions

`plot.header` a logical specifying whether to plot a header describing the genomic region plotted

`plot.lines` a logical specifying whether to plot dashed lines indicating anchor regions and associated features

`anchor.col` colour for anchor regions

`plot.ids` a logical specifying whether to plot ids for features in annotation.features. Looks for the presence of an id or name

Value

invisible(1)

Examples

```
## Not run:
k562.rep1 = GenomicInteractions(file.path(system.file(package="GenomicInteractions"), "extdata", "k562.rep1.clus
                                          type="chiapet.tool",
                                          experiment_name="k562",
                                          description="k562 pol2 8wg16",
                                          gname="BSgenome.Hsapiens.UCSC.hg19")
annotation.features = list(promoter=refseq.promoters,
                                terminator=refseq.terminators,
                                gene.body=refseq.transcripts)
plotRegion(k562.rep1, GRanges("chr18", IRanges(9000000, 10000000)), annotation.features)

## End(Not run)
```

`plotSummaryStats` *Plot summary statistics for a GenomicInteractions object*

Description

Makes summary plots of the counts, interaction distances, interaction annotations, and percentage of cis and trans interactions for a GenomicInteractions object using ‘plotCounts‘, ‘plotDists‘, ‘plotCisTrans‘, and ‘plotInteractionAnnotations‘.

Usage

```
plotSummaryStats(GIObject, other = 5, cut = 10)
```

Arguments

`GIObject`	A GenomicInteractions object
`other`	Default 5. Passed to plotInteractionAnnotations. Interaction types making up fewer than "other" percent of the total interactions will be consolidated into a single "other" category.
`cut`	Default 10. Passed to plotCounts.All interactions with counts > cut are consolidated into a single category.

Value

invisible(1)

Examples

```
data(hic_example_data)
plotSummaryStats(hic_example_data)
```

`print,GenomicInteractions-method` *Print function for GenomicInteractions*

Description

Print function for GenomicInteractions

Usage

```
## S4 method for signature GenomicInteractions
print(x)
```

Arguments

`x`	GenomicInteractionsObject

Value

invisible(1)

`resetAnnotations` *Reset annotations made to a GenomicInteractions object*

Description

This function removes all annotations from a GenomicInteractions object by deleting all of the metadata columns associated with both anchors.

Usage

```
resetAnnotations(GIObject)

## S4 method for signature GenomicInteractions
resetAnnotations(GIObject)
```

Arguments

`GIObject`	An annotated GenomicInteractions object

Value

invisible(1)

`setters` *Functions to set data held in a GenomicInteractions object.*

Description

Use these functions to set data stored in each of the slots of a GenomicInteractions object.

Usage

```
name(GIObject) <- value

description(GIObject) <- value

pValue(GIObject) <- value

FDR(GIObject) <- value

normalisedCount(GIObject) <- value

## S4 replacement method for signature GenomicInteractions
name(GIObject) <- value

## S4 replacement method for signature GenomicInteractions
```

```
normalisedCount(GIObject) <- value

## S4 replacement method for signature GenomicInteractions
pValue(GIObject) <- value

## S4 replacement method for signature GenomicInteractions
FDR(GIObject) <- value

## S4 replacement method for signature GenomicInteractions
description(GIObject) <- value
```

Arguments

`GIObject`	A GenomicInteractions object
`value`	A vector to replace a slot in the object

Value

GenomicInteractions object

Examples

```
library(BSgenome.Mmusculus.UCSC.mm9)
anchor.one = GRanges(c("chr1", "chr1", "chr1", "chr1"), IRanges(c(10, 20, 30, 20), width=5), seqlengths=seqlengths
anchor.two = GRanges(c("chr1", "chr1", "chr1", "chr2"), IRanges(c(100, 200, 300, 50), width=5), seqlengths=seqleng
test <- new("GenomicInteractions", experiment_name="test", description="this is a test",
                 genome_name="BSgenome.Mmusculus.UCSC.mm9", anchor_one = anchor.one,
                 anchor_two = anchor.two, counts=as.integer(c(2,1,2,3)) )

name(test) <- "Mouse test"
name(test)

description(test) <- "This is a test using the mouse genome"
description(test)

pValue(test) = c(0.1, 0.3, 0.1, 0.08)
pValue(test)

FDR(test) = p.adjust(pValue(test), "bonferroni")
FDR(test)

normalisedCount(test) = count(test) / sum(test)
normalisedCount(test)
```

show,GenomicInteractions-method

Representation function for GenomicInteractions

Description

Representation function for GenomicInteractions

Usage

```
## S4 method for signature GenomicInteractions
show(object)
```

Arguments

`object`	A GenomicInteractionsObject

Value

invisible(1)

`subsetByFeatures` *Subset a GenomicInteractions object by features*

Description

Subsets interactions for which at least one of the anchors overlaps with a given GRanges object. Alternatively, subsets interactions based on annotated feature IDs for a particular feature.

Usage

```
subsetByFeatures(GIObject, features, feature.class = NULL)

## S4 method for signature GenomicInteractions,GRanges,missing
subsetByFeatures(GIObject,
  features, feature.class = NULL)

## S4 method for signature GenomicInteractions,GRangesList,missing
subsetByFeatures(GIObject,
  features, feature.class = NULL)

## S4 method for signature GenomicInteractions,character,character
subsetByFeatures(GIObject,
  features, feature.class = NULL)
```

Arguments

`GIObject`	A GenomicInteractions object
`features`	A GRanges or GRangesList object, or a character vector containing IDs of annotated features, e.g. promoter IDs.
`feature.class`	If ‘features’ is a character vector, the corresponding feature name, e.g. "promoter".

Value

a subsetted GenomicInteractions object

sum,GenomicInteractions-method

Return the total number of interactions in a GenomicInteractions GIObject

Description

Return the total number of interactions in a GenomicInteractions GIObject

Usage

```
## S4 method for signature GenomicInteractions
sum(x)
```

Arguments

x	GenomicInteractions GIObject

Value

The sum of the counts in GIObject

summariseByFeatures

Summary statistics of interactions for a given feature set

Description

This function will calculate summary statistics for each element in the given feature set, including the number of interactions (the sum of all interaction counts), number of unique interactions and number of trans- (interchromosomal) interations. It also returns some statistics for the distances of interactions for all interactions of the feature, and for the different interaction types e.g. promoter-distal.

Usage

```
summariseByFeatures(GIObject, features, feature.name,
  distance.method = "midpoint", annotate.self = FALSE)

## S4 method for signature GenomicInteractions
summariseByFeatures(GIObject, features,
  feature.name, distance.method = "midpoint", annotate.self = FALSE)
```

Arguments

GIObject	An annotated GenomicInteractions object
features	A GRanges object containing the feature set
feature.name	The name of the feature set
distance.method	Method for calculating distances between anchors, see ?calculateDistances
annotate.self	Logical. Indicates whether to annotate self interactions, i.e. where a feature in ‘features‘ overlaps both anchors of an interaction. Default: FALSE.

Value

A data frame with one line for each range in ‘features’

---

viewPoint *Plot coverage of interactions originating at a given viewpoint.*

Description

Plot coverage of interactions originating at a given viewpoint.

Usage

```
viewPoint(pos, GIObject, leftflank, rightflank, plot = TRUE)
```

Arguments

pos	A single region in GRanges format.
GIObject	A GenomicInteractions object.
leftflank	An integer; flank size in bp upstream of pos centre.
rightflank	An integer; flank size in bp downstream of pos centre.
plot	Logical. Whether to plot vector (default), or just return vector of coverage.

Value

Plot of coverage or Rle-vector of coverage, depending on plot parameter.

Examples

```
## Not run:
data(hic_data)
pos <- GRanges(seqnames="chr5", ranges=IRanges(start=115938063, end=115941352))
viewPoint(pos,hic_data, 100000, 100000, plot=FALSE)

## End(Not run)
```

viewPointAverage *Plot coverage of interactions originating at a given set of viewpoints.*

Description

Plot coverage of interactions originating at a given set of viewpoints.

Usage

```
viewPointAverage(positions, GIObject, leftflank, rightflank, plot = TRUE,
  flip = FALSE)
```

Arguments

`positions`	A set of regions in GRanges format.
`GIObject`	A GenomicInteractions object.
`leftflank`	An integer; flank size in bp upstream of position centre.
`rightflank`	An integer; flank size in bp downstream of position centre.
`plot`	Logical. Whether to plot vector (default), or just return vector of coverage.
`flip`	Logical. If TRUE, flip window of coverage around a position on the minus strand. Use e.g. to plot around promoters. Default: FALSE.

Value

Plot of mean coverage around all positions or Rle-vector of mean coverage, depending on plot parameter.

[*Standard subsetting methods for GenomicInteractions objects*

Description

Standard subsetting methods for GenomicInteractions objects

Usage

```
## S4 method for signature GenomicInteractions,logical,missing
x[i]

## S4 method for signature GenomicInteractions,numeric,missing
x[i]

## S4 method for signature GenomicInteractions,Rle,missing
x[i]

## S4 method for signature GenomicInteractions,rle,missing
x[i]
```

Arguments

`x`	A genomicInteractions object
`i`	A numeric, logical or Rle vector

Value

A GenomicInteractions object containing only the features specified by ‘i‘.

Index