Package 'multiHiCcompare'

December 5, 2025

Title Normalize and detect differences between Hi-C datasets when replicates of each experimental condition are available

Version 1.29.0

Description multiHiCcompare provides functions for joint normalization and difference detection in multiple Hi-C datasets. This extension of the original HiCcompare package now allows for Hi-C experiments with more than 2 groups and multiple samples per group. multiHiCcompare operates on processed Hi-C data in the form of sparse upper triangular matrices. It accepts four column (chromosome, region1, region2, IF) tab-separated text files storing chromatin interaction matrices. multiHiCcompare provides cyclic loess and fast loess (fastlo) methods adapted to jointly normalizing Hi-C data. Additionally, it provides a general linear model (GLM) framework adapting the edgeR package to detect differences in Hi-C data in a distance dependent manner.

Depends R (>= 4.0.0)

Imports data.table, dplyr, HiCcompare, edgeR, BiocParallel, qqman, pheatmap, methods, GenomicRanges, graphics, stats, utils, pbapply, GenomeInfoDbData, GenomeInfoDb, aggregation

biocViews Software, HiC, Sequencing, Normalization

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, testthat, BiocStyle

RoxygenNote 7.1.1

VignetteBuilder knitr

BugReports https://github.com/dozmorovlab/multiHiCcompare/issues

URL https://github.com/dozmorovlab/multiHiCcompare

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

git_branch devel

git_last_commit 31f6781

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-12-04

Author Mikhail Dozmorov [aut, cre] (ORCID:
<https://orcid.org/0000-0002-0086-8358>),
John Stansfield [aut]

Maintainer Mikhail Dozmorov <mikhail.dozmorov@gmail.com>

Contents

cyclic_loess *Cyclic Loess normalization for Hi-C data*

Description

Cyclic Loess normalization for Hi-C data

Usage

```
cyclic_loess(
  hicexp,
  iterations = 3,
  span = NA,
  parallel = FALSE,
  verbose = FALSE
)
```

Arguments

hicexp	A hicexp object
iterations	The number of iterations (cycles) of loess normalization to perform. Defaults to 3.
span	The span for loess normalization. Defaults to NA indicating that span will be automatically calculated using generalized cross validation.
parallel	Logical. Should parallel processing be used?
verbose	Logical. Should messages about loess normalization be printed to the screen.

Details

This function performs cyclic loess normalization on a Hi-C experiment. multiHiCcompare's cyclic loess procedure is a modified version of Ballman's (2004) cyclic loess and the joint loess normalization used in the original HiCcompare. For each unique pair of samples in the hicexp object an MD plot is generated. A loess curve is fit to the MD plot and then the fitted values are used to adjust the data. This is performed on all unique pairs and then repeated until convergence.

Value

A hicexp object that has been normalized.

Examples

```
#' data("hicexp2")
hicexp2 <- cyclic_loess(hicexp2, span = 0.7)
```

exportJuicebox *Export multiHiCcompare results for visualization in Juicebox*

Description

Export multiHiCcompare results for visualization in Juicebox

Usage

```
exportJuicebox(
  hicexp,
  logfc_cutoff = 1,
  logcpm_cutoff = 1,
  p.adj_cutoff = 0.01,
  D_cutoff = 1,
  file_name = "juiceboxAnnotations.txt",
  color = "0,0,255"
)
```

Arguments

hicexp	A hicexp object which has been compared.
logfc_cutoff	The logFC value you wish to filter by. Defaults to 1.
logcpm_cutoff	The logCPM cutoff you wish to filter by. Defaults to 1.
p.adj_cutoff	The adjusted p-value cutoff you wish to filter by. Defaults to 0.01.
D_cutoff	The distance cutoff you wish to filter by. Interactions with a D < D_cutoff will be filtered. Defaults to 1.
file_name	The file name of the text file to be saved.
color	A decimal RGB color code. Should be a character value in form of "0,0,255". Defaults to color code for blue. This will determine the color of the annotations on the Juicebox heatmap.

Details

This function is meant to filter the results of multiHiCcompare and export the significant differentially interacting regions into a text file which can be imported into Juicebox as a 2D annotations file. This will allow you to visualize where your DIRs occur on the heatmap of the interactions. Please see the included vignette on using Juicebox to visualize multiHiCcompare results. This can be accessed with browseVignettes("multiHiCcompare").

Value

A text file containing annotations for input into Juicebox.

Examples

```
data('hicexp_diff')
exportJuicebox(hicexp_diff, file_name = "juiceboxAnnotations.txt")
```

`fastlo` *Perform fast loess normalization on a Hi-C experiment*

Description

Perform fast loess normalization on a Hi-C experiment

Usage

```
fastlo(
  hicexp,
  iterations = 3,
  span = 0.7,
  parallel = FALSE,
  verbose = FALSE,
  max.pool = 0.7
)
```

Arguments

`hicexp`	A hicexp object
`iterations`	The number of iterations (cycles) for fastlo to proceed through.
`span`	The span of loess fitting. Defaults to 0.7. To automatically calculate a span using the GCV set span = NA. However note that setting span = NA will significantly slow down the normalization process.
`parallel`	Logical. Should parallel processing be used?
`verbose`	Logical, should messages about the normalization be printed?
`max.pool`	The proportion of unit distances after which all further distances will be pooled. Distances before this value will be progressively pooled and any distances after this value will be combined into a single pool. Defaults to 0.7. Warning: do not adjust this value from the default unless you are getting errors related to the lfproc function or due to sparsity in fastlo normalization. If these errors occur it is due to either sparsity or low variance and max.pool will need to be lowered; typically to 0.5 or 0.6.

Details

This function performs the fast loess (fastlo) normalization procedure on a hicexp object. the fast linear loess ("fastlo") method of Ballman (2004) that is adapted to Hi-C data on a per-distance basis. To perform "fastlo" on Hi-C data we first split the data into p pooled matrices. The "progressive pooling" is used to split up the Hi-C matrix by unit distance. Fastlo is then performed on the MA plots for each distance pool. See Stansfield et al (2018) for full description.

Value

A hicexp object that is normalized.

Examples

```
data("hicexp2")
hicexp2 <- fastlo(hicexp2)
```

HCT116_r1 *A 4 column sparse matrix for a Hi-C matrix.*

Description

A matrix object with 4 columns and 56603 rows. Sample 1 of 7 included as example Hi-C data. This is replicate 1 of HCT-116 cell line at 100KB resolution from chHCT116_r22.

Usage

```
HCT116_r1
```

Format

An object of class `data.frame` with 56603 rows and 4 columns.

Value

A matrix

Source

Data from Rao 2017. See their website at http://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE104888

HCT116_r2 *A 4 column sparse matrix for a Hi-C matrix.*

Description

A matrix object with 4 columns and 57010 rows. Sample 2 of 7 included as example Hi-C data. This is replicate 2 of HCT-116 cell line at 100KB resolution from chHCT116_r22.

Usage

```
HCT116_r2
```

Format

An object of class `data.frame` with 57010 rows and 4 columns.

Value

A matrix

Source

Data from Rao 2017. See their website at http://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE104888

`HCT116_r3` *A 4 column sparse matrix for a Hi-C matrix.*

Description

A matrix object with 4 columns and 56744-C data. Sample 3 of 7 included as example Hi-C data. This is replicate 3 of HCT-116 cell line at 100KB resolution from chHCT116_r22.

Usage

```
HCT116_r3
```

Format

An object of class `data.frame` with 56744 rows and 4 columns.

Value

A matrix

Source

Data from Rao 2017. See their website at http://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE104888

HCT116_r4 *A 4 column sparse matrix for a Hi-C matrix.*

Description

A matrix object with 4 columns and 54307 rows. Sample 4 of 7 included as example Hi-C data. This is replicate 4 of HCT-116 cell line at 100KB resolution from chHCT116_r22.

Usage

```
HCT116_r4
```

Format

An object of class `data.frame` with 54307 rows and 4 columns.

Value

A matrix

Source

Data from Rao 2017. See their website at http://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE104888

HCT116_r5 *A 4 column sparse matrix for a Hi-C matrix.*

Description

A matrix object with 4 columns and 55092 rows. Sample 5 of 7 included as example Hi-C data. This is replicate 5 of HCT-116 cell line at 100KB resolution from chHCT116_r22.

Usage

```
HCT116_r5
```

Format

An object of class `data.frame` with 55092 rows and 4 columns.

Value

A matrix

Source

Data from Rao 2017. See their website at http://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE104888

`HCT116_r6` *A 4 column sparse matrix for a Hi-C matrix.*

Description

A matrix object with 4 columns and 55581 rows. Sample 6 of 7 included as example Hi-C data. This is replicate 6 of HCT-116 cell line at 100KB resolution from chHCT116_r22.

Usage

```
HCT116_r6
```

Format

An object of class `data.frame` with 55581 rows and 4 columns.

Value

A matrix

Source

Data from Rao 2017. See their website at http://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE104888

`hg19_cyto` *A GenomicRanges object containing centromeric, gvar, and stalk regions.*

Description

A GRanges object with 2 metadata columns and 70 rows. These ranges indicate the locations of centromeres, stalks, and gvar regions from hg19. Use this for filtering out these regions from your data.

Usage

```
hg19_cyto
```

Format

An object of class `GRanges` of length 70.

Value

A GRanges object

Source

Data from UCSC http://hgdownload.cse.ucsc.edu/goldenPath/hg19/database/cytoBand.txt.gz

---

`hg38_cyto` *A GenomicRanges object containing centromeric, gvar, and stalk regions.*

Description

A GRanges object with 2 metadata columns and 70 rows. These ranges indicate the locations of centromeres, stalks, and gvar regions from hg38. Use this for filtering out these regions from your data.

Usage

```
hg38_cyto
```

Format

An object of class `GRanges` of length 70.

Value

A GRanges object

Source

Data from UCSC http://hgdownload.cse.ucsc.edu/goldenPath/hg38/database/cytoBand.txt.gz

Hicexp-class *An S4 class for working with Hi-C data*

Description

An S4 class for working with Hi-C data

Value

Hicexp class

Slots

`hic_table` A data.table containing the sparse upper triangular matrix for your Hi-C data.

`comparison` The results of a multiHiCcompare comparison.

`metadata` Data.frame for covariate information.

`resolution` The resolution of the dataset.

`normalized` Indicator for if data has been normalized.

Examples

```
data('hicexp2')
hicexp2
```

hicexp2 *hicexp object with 4 samples from two groups.*

Description

A hicexp object with a hic_table slot containing 666 rows from chromosome 22 at 1MB resolution.

Usage

```
hicexp2
```

Format

An object of class `Hicexp` of length 1.

Value

A hicexp object

Source

Data from Rao 2017. See their website at http://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE104888

hicexp_diff *hicexp object with 7 samples from two groups.*

Description

A hicexp object with a hic_table slot containing 666 rows and a metadata slot containing 3 covariates. Same data as from "hicexp2" object but has been normalized and tested for differences with the hic_exactTest.

Usage

```
hicexp_diff
```

Format

An object of class `Hicexp` of length 1.

Value

A hicexp object

Source

Data from Rao 2017. See their website at http://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE104888

hic_exactTest *Perform exact test based difference detection on a Hi-C experiment*

Description

Perform exact test based difference detection on a Hi-C experiment

Usage

```
hic_exactTest(hicexp, parallel = FALSE, p.method = "fdr", max.pool = 0.7)
```

Arguments

hicexp	A hicexp object.
parallel	Logical, should parallel processing be used?
p.method	Charact string to be input into p.adjust() as the method for multiple testing correction. Defaults to "fdr".

max.pool — The proportion of unit distances after which all further distances will be pooled. Distances before this value will be progressively pooled and any distances after this value will be combined into a single pool. Defaults to 0.7. Warning: do not adjust this value from the default unless you are getting errors related to the lfproc function or due to sparsity in fastlo normalization. If these errors occur it is due to either sparsity or low variance and max.pool will need to be lowered; typically to 0.5 or 0.6.

Details

This function performs the edgeR exact test on a per distance basis for Hi-C data. It tests for differences between two groups when the groups are the only variable of interest. This is an application of the negative binomial exact test proposed by Robinson and Smyth (2008) for a difference in mean between the groups. These exact tests are applied to the Hi-C data on a distance group basis using "progressive pooling" of distances.

Value

A hicexp object with the comparison slot filled.

Examples

```
## Not run:
data("hicexp_diff")
hicexp_diff <- hic_exactTest(hicexp_diff)
## End(Not run)
```

---

hic_filter *Perform filtering on a Hi-C experiment*

Description

Perform filtering on a Hi-C experiment

Usage

```
hic_filter(hicexp, zero.p = 0.8, A.min = 5, remove.regions = hg19_cyto)
```

Arguments

hicexp — A hicexp object.

zero.p — The proportion of zeros in a row to filter by. If the proportion of zeros in a row is <= zero.p the row will be filtered out, i.e. zero.p = 1 means nothing is filtered based on zeros and zero.p = 0 will filter rows that have any zeros.

A.min — The minimum average expression value (row mean) for an interaction pair. If the interaction pair has an average expression value less than A.min the row will be filtered out.

`remove.regions` A GenomicRanges object indicating specific regions to be filtered out. By default this is the hg19 centromeric, gvar, and stalk regions. Also included in the package is hg38_cyto. If your data is not hg19 you will need to substitute this file. To choose not to filter any regions set regions = NULL.

Details

This function is used to filter out the interactions that have low average IFs or large numbers of 0 IF values. If you have already performed filtering when making your hicexp object do not use this again. As these interactions are not very interesting and are commonly false positives during difference detection it is better to remove them from the dataset. Additionally, filtering will help speed up the run time of multiHiCcompare. Filtering can be performed before or after normalization, however the best computational speed gain will occur when filtering is done before normalization.

Value

A hicexp object.

Examples

```
data("hicexp2")
hicexp2 <- hic_filter(hicexp2)
```

---

hic_glm *Function to perform GLM differential analysis on Hi-C experiment*

Description

Function to perform GLM differential analysis on Hi-C experiment

Usage

```
hic_glm(
  hicexp,
  design,
  contrast = NA,
  coef = NA,
  method = "QLFTest",
  M = 1,
  p.method = "fdr",
  parallel = FALSE,
  max.pool = 0.7
)
```

Arguments

hicexp	A hicexp object,
design	A design matrix for the GLM.
contrast	Numeric vector or matrix specifying one or more contrasts of the linear model coefficients to be tested equal to zero.
coef	integer or character index vector indicating which coefficients of the linear model are to be tested equal to zero.
method	The test method to be performed. Should be one of "QLFTest", "LRTest", or "Treat".
M	The log2 fold change value for a TREAT analysis.
p.method	p-value adjustment method to be used. Defaults to "fdr". See ?p.adjust for other adjustment options.
parallel	Logical, Should parallel processing be used?
max.pool	The proportion of unit distances after which all further distances will be pooled. Distances before this value will be progressively pooled and any distances after this value will be combined into a single pool. Defaults to 0.7. Warning: do not adjust this value from the default unless you are getting errors related to the lfproc function or due to sparsity in fastlo normalization. If these errors occur it is due to either sparsity or low variance and max.pool will need to be lowered; typically to 0.5 or 0.6.

Details

This function performs the specified edgeR GLM based test on a per distance basis on the Hi-C data. Distances groups are pooled using "progressive pooling". There are 3 options for the type of GLM based test to be used which is specified with the method option.
`QLFTest` will use edgeR's glmQLFit and glmQLFTest functions which makes use of quasi-likelihood methods described in Lund et al (2012).
`LRTest` uses edgeR's glmFit and glmLRT functions which uses a interaction-wise negative binomial general linear model. This method uses a likelihood ratio test for the coefficients specified in the model.
`Treat` uses edgeR's glmTreat function which performs a test for differential expression with a minimum required fold-change threshold imposed. It tests whether the absolute value of the log2 fold change is greater than the value specified as the `M` option.

Value

A hicexp object with a filled in comparison slot.

Examples

```
## Not run:
data("hicexp_diff")
d <- model.matrix(~factor(meta(hicexp_diff)$group) + factor(c(1,2,1,2)))
hicexp_diff <- hic_glm(hicexp_diff, design = d, coef = 2)
## End(Not run)
```

hic_scale *Perform library scaling on a hicexp object*

Description

Perform library scaling on a hicexp object

Usage

```
hic_scale(hicexp)
```

Arguments

hicexp	A hicexp object.

Details

This function will perform library scaling on a hicexp object. Scaling is performed separately for each chromosome. This is an alternative normalization method to the cyclic loess and fastlo methods also provided in multiHiCcompare. Use this normalization method if for some reason you do not want to remove trends in the data and only want to normalize based on library size.

Value

A hicexp object.

Examples

```
data("hicexp2")
hicexp2 <- hic_scale(hicexp2)
```

hic_table *Print the hic_table*

Description

Print the hic_table

Usage

```
hic_table(x)

## S4 method for signature 'Hicexp'
hic_table(x)
```

Arguments

x	The Hicexp object

Value

Hicexp class information

Examples

```
data('hicexp2')
hic_table(hicexp2)
```

---

make_hicexp *Make Hi-C experiment object from data*

Description

Make Hi-C experiment object from data

Usage

```
make_hicexp(
  ...,
  data_list = NA,
  groups,
  covariates = NULL,
  remove_zeros = FALSE,
  zero.p = 0.8,
  A.min = 5,
  filter = TRUE,
  remove.regions = hg19_cyto
)
```

Arguments

...	Hi-C data. Data must in sparse upper triangular format with 4 columns: chr, region1, region2, IF or in 7 column BEDPE format with columns chr, start1, end1, chr, start2, end2, IF.
data_list	Alternate way to enter data. If you have your Hi-C data in the form of a list already with each entry of the list representing a sample use this option.
groups	A vector of the experimental groups corresponding to each Hi-C data object entered. If it is not in factor form when entered it will be converted to a factor.
covariates	Optional data.frame containing covariate information for your Hi-C experiment. Some examples are enzyme used, batch number, etc. Should have the same number of rows as the number of Hi-C data objects entered and columns corresponding to covariates.

remove_zeros	Logical, should rows with 1 or more zero IF values be removed?
zero.p	The proportion of zeros in a row to filter by. If the proportion of zeros in a row is <= zero.p the row will be filtered out, i.e. zero.p = 1 means nothing is filtered based on zeros and zero.p = 0 will filter rows that have any zeros.
A.min	The minimum average expression value (row mean) for an interaction pair. If the interaction pair has an average expression value less than A.min the row will be filtered out.
filter	Logical, should filtering be performed? Defaults to TRUE. If TRUE it will filter out the interactions that have low average IFs or large numbers of 0 IF values. As these interactions are not very interesting and are commonly false positives during difference detection it is better to remove them from the dataset. Additionally, filtering will help speed up the run time of multiHiCcompare. Filtering can be performed before or after normalization, however the best computational speed gain will occur when filtering is done before normalization. Filtering parameters are controlled by the zero.p and A.min options.
remove.regions	A GenomicRanges object indicating specific regions to be filtered out. By default this is the hg19 centromeric, gvar, and stalk regions. Also included in the package is hg38_cyto. If your data is not hg19 you will need to substitute this file. To choose not to filter any regions set regions = NULL. NOTE: if you set filter = FALSE these regions will NOT be removed. This occurs in conjuction with the filtering step.

Details

Use this function to create a hicexp object for analysis in multiHiCcompare. Filtering can also be performed in this step if the filter option is set to TRUE. Filtering parameters are controlled by the zero.p and A.min options.

Value

A hicexp object.

Examples

```
# load data in sparse upper triangular format
data("HCT116_r1", "HCT116_r2", "HCT116_r3", "HCT116_r4",
    "HCT116_r5", "HCT116_r6")
# make groups & covariate input
groups <- factor(c(1, 1, 1, 2, 2, 2))
covariates <- data.frame(enzyme = factor(c('mobi', 'mboi', 'mboi',
 'dpnii', 'dpnii', 'dpnii')), batch = c(1, 2, 1, 2, 1, 2))
# make the hicexp object
hicexp <- make_hicexp(HCT116_r1, HCT116_r2, HCT116_r3, HCT116_r4,
     HCT116_r5, HCT116_r6, groups = groups,
     covariates = covariates)
```

manhattan_hicexp *Manhattan plot function for results of multiHiCcompare*

Description

Manhattan plot function for results of multiHiCcompare

Usage

```
manhattan_hicexp(
  hicexp,
  pval_aggregate = "standard",
  return_df = FALSE,
  p.adj_cutoff = 0.05,
  plot.chr = NA
)
```

Arguments

hicexp	A hicexp object that has had differences detected
pval_aggregate	string denoting the p-value method to use for plotting. Options are "standard", "fisher", "lancaster", "sidak", and "count". "standard" plots a manhattan plot using all individual p-values (very slow, use with caution). "fisher", "lancaster", or "sidak" methods use the Fisher's, Lancaster, or the Sidak method, respectively, for combining p-values for each region which are then plotted on the -log10(p-value) Y-axis. "count" summarizes the number of times a region was detected as significant (see "p.adj_cutoff" parameter), plotted on Y-axis. The higher the dots are, the more significant/more frequent a region was detected as significantly differentially interacting. See ?topDirs
return_df	Logical, should the data.frame used to generate the plot be returned?
p.adj_cutoff	The adjusted p-value cutoff to be used for calling an interaction significant. This is only used if method = 'count'. Defaults to 0.05.
plot.chr	A numeric value indicating a specific chromosome number to subset the plot to. Defaults to NA indicating that all chromosomes will be plotted.

Details

This function is used to create a manhattan plot for the significance of all genomic regions in the dataset. These correspond to the rows (or columns) of the upper triangle of the full Hi-C matrix. Each genomic region of the Hi-C dataset has multiple interactions it is involved in and the significance of all of these can be visualized with `pval_aggregate = "standard"`. Alternatively the p-values for all these interactions can be combined using either Fisher's, or the Lancaster or the Sidac method of combining p-values. Additionally the "count" option will plot based on the number of times each region was found to be involved in a signficantly different interaction. The manhattan plot can be used to identify "hotspot" regions of the genome where major differences seem to be located based on the results of a multiHiCcompare analysis.

Value

A manhattan plot and optionally the data.frame used to generate the manhattan plot.

Examples

```
data("hicexp_diff")
manhattan_hicexp(hicexp_diff, pval_aggregate = "fisher")
```

`MD_composite` *Plot a composite MD plot with the results of a comparison*

Description

Plot a composite MD plot with the results of a comparison

Usage

```
MD_composite(hicexp, plot.chr = NA, D.range = 1)
```

Arguments

`hicexp`	A hicexp object which has had a multiHiCcompare comparison step performed on it.
`plot.chr`	A specific chromosome or set of chromosome which you want to plot. This should be a numeric value, i.e. to plot chromosome 1 set plot.chr = 1, to plot chromosomes 1 and 5 set plot.chr = c(1, 5). Defaults to NA indicating that all chromosomes present in the hicexp will be plotted.
`D.range`	Allows for subsetting of the plot by Distance. Set to proportion of total distance that you want to be displayed. Value of 1 indicates that the entire distance range will be displayed. Defaults to 1.

Value

An MD plot

Examples

```
data("hicexp_diff")
MD_composite(hicexp_diff)
```

MD_hicexp *Make MD plots for all combinations of a condition*

Description

Make MD plots for all combinations of a condition

Usage

```
MD_hicexp(hicexp, prow = 3, pcol = 3, plot.chr = NA, plot.loess = FALSE)
```

Arguments

hicexp	A hicexp object.
prow	The number of rows to use for the grid of MD plots. Defaults to 3.
pcol	The number of columns to use for the grid of MD plots. Defaults to 3.
plot.chr	A specific chromosome or set of chromosome which you want to plot. This should be a numeric value, i.e. to plot chromosome 1 set plot.chr = 1, to plot chromosomes 1 and 5 set plot.chr = c(1, 5). Defaults to NA indicating that all chromosomes present in the hicexp will be plotted.
plot.loess	Logical, should a loess curve be plotted over the MD plots. Note setting this to TRUE will increase the computational time for plotting.

Value

A set of MD plots.

Examples

```
data("hicexp2")
MD_hicexp(hicexp2)
```

meta *Print the metadata*

Description

Print the metadata

Usage

```
meta(x)

## S4 method for signature 'Hicexp'
meta(x)
```

Arguments

x	The Hicexp object

Value

Hicexp class information

Examples

```
data('hicexp2')
meta(hicexp2)
```

---

normalized *Print the indicator for if the data is normalized*

Description

Print the indicator for if the data is normalized

Usage

```
normalized(x)

## S4 method for signature 'Hicexp'
normalized(x)
```

Arguments

x	The Hicexp object

Value

Hicexp class information

Examples

```
data('hicexp2')
normalized(hicexp2)
```

perm_test *Perform a permutation test to check enrichment of a genomic feature with DIRs detected by multiHiCcompare*

Description

Perform a permutation test to check enrichment of a genomic feature with DIRs detected by multiHiCcompare

Usage

```
perm_test(
  hicexp,
  feature,
  p.adj_cutoff = 10^-10,
  logfc_cutoff = 1,
  num.perm = 1000,
  pval_aggregate = "max"
)
```

Arguments

hicexp	A Hicexp object which has been compared.
feature	A GRanges object containing locations for a genomic feature you would like to test for enrichment in the differentially interacting regions (DIRs).
p.adj_cutoff	The adjusted p-value cutoff for declaring a region significant. See ?topDirs for more information. Defaults to 10^-10
logfc_cutoff	The log fold change cutoff for a region to be declared significant. See ?topDirs for more information. Defaults to 1.
num.perm	The number of permutations to run. Defaults to 1000.
pval_aggregate	Method to aggregate region-specific p-values. If a region differentially interacts with several other regions, the p-values are aggregated using a 'max' method (Default, select maximum p-value, most conservative), or the Fisher ('fisher'), Lancaster ('lancaster'), or Sidak ('sidak') methods (see 'aggregate' package). regions, it is assigned a single p-value aggregated from several. See ?topDirs

Value

The permutation p-value

Examples

```
## Not run:
data("hicexp_diff")
data("hg19_cyto")
perm_test(hicexp_diff, hg19_cyto)
```

```
## End(Not run)
```

`plot_counts` *Plot the count results from topDirs*

Description

Plot the count results from topDirs

Usage

```
plot_counts(dirs, plot.chr = NA)
```

Arguments

`dirs`	The output of the topDirs function when the return_df option is set to "bed".
`plot.chr`	A numeric value indicating a specific chromosome number to subset the plot to. Defaults to NA indicating that all chromosomes will be plotted.

Value

A plot.

Examples

```
data('hicexp_diff')
dirs <- topDirs(hicexp_diff, return_df = 'bed')
plot_counts(dirs)
```

`plot_pvals` *Plot the p-value results from topDirs*

Description

Plot the p-value results from topDirs

Usage

```
plot_pvals(dirs, plot.chr = NA)
```

Arguments

`dirs`	The output of the topDirs function when the return_df option is set to "bed".
`plot.chr`	A numeric value indicating a specific chromosome number to subset the plot to. Defaults to NA indicating that all chromosomes will be plotted.

Value

A plot.

Examples

```
data('hicexp_diff')
dirs <- topDirs(hicexp_diff, return_df = 'bed')
plot_pvals(dirs)
```

---

pval_heatmap *Function to visualize p-values from multiHiCcompare results*

Description

Function to visualize p-values from multiHiCcompare results

Usage

```
pval_heatmap(hicexp, alpha = NA, chr = 0)
```

Arguments

hicexp	A hicexp object that has been normalized and has had differences detected.
alpha	The alpha level at which you will call a p-value significant. If this is set to a numeric value then any p-values >= alpha will be set to 1 for the visualization in the heatmap. Defaults to NA for visualization of all p-values.
chr	The numeric value for the chromosome that you want to plot. Set to 0 to plot all chromosomes in the dataset.

Details

The goal of this function is to visualize where in the Hi-C matrix the differences are occuring between two experimental conditions. The function will produce a heatmap of the -log10(p-values) * sign(logFC) to visualize where the significant differences between the datasets are occuring on the genome.

Value

A heatmap

Examples

```
data("hicexp_diff")
pval_heatmap(hicexp_diff, chr = 22)
```

resolution *Print the resolution*

Description

Print the resolution

Usage

```
resolution(x)

## S4 method for signature 'Hicexp'
resolution(x)
```

Arguments

x	The Hicexp object

Value

Hicexp class information

Examples

```
data('hicexp2')
resolution(hicexp2)
```

results *Print the results*

Description

Print the results

Usage

```
results(x)

## S4 method for signature 'Hicexp'
results(x)
```

Arguments

x	The Hicexp object

Value

Hicexp class information

Examples

```
data('hicexp2')
results(hicexp2)
```

---

show,Hicexp-method *Print information about a HiCexp object*

Description

Print information about a HiCexp object

Usage

```
## S4 method for signature 'Hicexp'
show(object)
```

Arguments

object	A Hicexp object

Value

HiCexp information

---

smartApply *Function to apply either biocParallel or standard lapply*

Description

Function to apply either biocParallel or standard lapply

Usage

```
smartApply(parallel, x, FUN, ...)
```

Arguments

parallel	Logical, should parallel processing be used?
x	The main list object which the function will be applied to.
FUN	The function to be applied.
...	Additional arguments for bplapply or lapply.

Value

results of lapply or bplapply

`topDirs` *Filter results of multiHiCcompare*

Description

Filter results of multiHiCcompare

Usage

```
topDirs(
  hicexp,
  logfc_cutoff = 1,
  logcpm_cutoff = 1,
  p.adj_cutoff = 0.01,
  D_cutoff = 1,
  return_df = "pairedbed",
  pval_aggregate = "max"
)
```

Arguments

`hicexp`	A hicexp object which has been compared.
`logfc_cutoff`	The logFC value you wish to filter by. Defaults to 1.
`logcpm_cutoff`	The logCPM cutoff you wish to filter by. Defaults to 1.
`p.adj_cutoff`	The adjusted p-value cutoff you wish to filter by. Defaults to 0.01.
`D_cutoff`	The distance cutoff you wish to filter by. Interactions with a D < D_cutoff will be filtered. Defaults to 1.
`return_df`	The format for the data.frame returned by the function. Options are "bed" and "pairedbed" (Default).
`pval_aggregate`	Method to aggregate region-specific p-values. If a region differentially interacts with several other regions, the p-values are aggregated using a 'max' method (Default, select maximum p-value, most conservative), or the Fisher ('fisher'), Lancaster ('lancaster'), or Sidak ('sidak') methods (see 'aggregate' package). regions, it is assigned a single p-value aggregated from several

Details

This function is meant to filter the results of multiHiCcompare. The top differentially interacting regions (DIRs) can be returned by using this function. When the return_df = "bed" option is set the resulting data.frame can be input into the plot_pvals or plot_counts functions to visualize the top DIRs.

Value

A data.table containing the filtered results.

Examples

```
data('hicexp_diff')
topDirs(hicexp_diff)
```

Index