Package ‘HiCcompare’

April 23, 2024

Title HiCcompare: Joint normalization and comparative analysis of multiple Hi-C datasets

Version 1.25.0

Description

HiCcompare provides functions for joint normalization and difference detection in multiple Hi-C datasets.
HiCcompare operates on processed Hi-C data in the form of chromosome-specific chromatin interaction matrices.
It accepts three-column tab-separated text files storing chromatin interaction matrices in a sparse matrix format which are available from several sources. HiCcompare is designed to give the user the ability to perform a comparative analysis on the 3-Dimensional structure of the genomes of cells in different biological states.`HiCcompare` differs from other packages that attempt to compare Hi-C data in that it works on processed data in chromatin interaction matrix format instead of pre-processed sequencing data. In addition, `HiCcompare` provides a non-parametric method for the joint normalization and removal of biases between two Hi-C datasets for the purpose of comparative analysis.
`HiCcompare` also provides a simple yet robust method for detecting differences between Hi-C datasets.

Depends R (>= 3.4.0), dplyr

Imports data.table, ggplot2, gridExtra, mgcv, stats, InteractionSet, GenomicRanges, IRanges, S4Vectors, BiocParallel, KernSmooth, methods, utils, graphics, pheatmap, gtools, rhdf5

Suggests knitr, rmarkdown, testthat, multiHiCcompare

biocViews Software, HiC, Sequencing, Normalization

License MIT + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.3

VignetteBuilder knitr

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

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

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

git_branch devel

git_last_commit c7e8c86

git_last_commit_date 2023-10-24

Repository Bioconductor 3.19

Date/Publication 2024-04-23

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

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

Contents

HiCcompare-package . . . 3
brain_table . . . 4
centromere_locations . . . 4
cooler . . . 5
cooler2bedpe . . . 5
cooler2sparse . . . 6
create.hic.table . . . 7
filter_params . . . 9
full2sparse . . . 10
hg19_blacklist . . . 11
hg38_blacklist . . . 11
hicpro2bedpe . . . 12
hic_compare . . . 13
hic_diff . . . 14
hic_loess . . . 16
hic_simulate . . . 18
HMEC.chr10 . . . 20
HMEC.chr22 . . . 21
hmec.IS . . . 22
KRnorm . . . 22
make_InteractionSet . . . 23
manhattan_plot . . . 24
MA_norm . . . 25
MD.plot1 . . . 26
MD.plot2 . . . 27
NHEK.chr10 . . . 28
NHEK.chr22 . . . 28
nhek.IS . . . 29

remove_centromere 30
SCN 30
sim.other.methods 31
sim_matrix 32
sparse2full 34
split_centromere 35
total_sum 36
visualize_pvals 37

Index **38**

`HiCcompare-package` *HiCcompare*

Description

HiCcompare provides functions for joint normalization and difference detection in multiple Hi-C datasets. HiCcompare operates on processed Hi-C data in the form of chromosome-specific chromatin interaction matrices. It accepts three-column tab-separated text files storing chromatin interaction matrices in a sparse matrix format which are available from several sources. HiCcompare is designed to give the user the ability to perform a comparative analysis on the 3-Dimensional structure of the genomes of cells in different biological states.

Details

To learn more about HiCcompare, start with the vignettes: ‘browseVignettes(package = "HiCcompare")‘

Author(s)

Maintainer: Mikhail Dozmorov <mikhail.dozmorov@gmail.com> (ORCID)

Authors:

- Kellen Cresswell <cresswellkg@vcu.edu>
- John Stansfield <stansfieldjc@vcu.edu>

See Also

Useful links:

- https://github.com/dozmorovlab/HiCcompare
- Report bugs at https://github.com/dozmorovlab/HiCcompare/issues

brain_table *Hi-C data from two regions of the brain at 100KB resolution*

Description

An hic.table object containing data from chromosomes 18 & 22. hic_loess has already been used to jointly normalize the datasets.

Usage

```
brain_table
```

Format

An object of class `list` of length 2.

Value

A hic.table list

centromere_locations *Locations of the centromeres for hg19*

Description

A BED file

Usage

```
centromere_locations
```

Format

A data.frame with 3 columns and 24 rows

Value

A data.frame

cooler — *Hi-C data in the cooler format*

Description

Contains Hi-C data for the entire genome from the Dixon2012-H1hESC-HindIII-allreps-filtered.1000kb.cool file.

Usage

```
cooler
```

Format

A matrix with 4011620 rows and 7 columns in BEDPE format.

Value

A matrix.

Source

Data from the Index of Coolers. ftp://cooler.csail.mit.edu/coolers/hg19/Dixon2012-H1hESC-HindIII-allreps-
1000kb.cool

cooler2bedpe — *Read a .cool file into R and output the data in BEDPE format*

Description

Read a .cool file into R and output the data in BEDPE format

Usage

```
cooler2bedpe(path)
```

Arguments

path	The path to a .cool file on your disk.

Details

.cool files are HDF5 containers that store Hi-C data. Many public Hi-C datasets are available in .cool format on the mirnylab ftp site ftp://cooler.csail.mit.edu/coolers. To use these files in HiCcompare simply download the .cool file and read it into R using this function. This function will dump the contents of the file and format them into BEDPE format in R. The resulting object cant then be used in HiCcompare.

Value

A list with two items. Item 1, "cis" contains the intra-chromosomal contact matrices, one per chromosome. Item 2, "trans" contains the inter-chromsomal contact matrix.

Examples

```
## Not run:
 dat <- cooler2bedpe(path = "path/to/cool/file.cool")

## End(Not run)
```

---

cooler2sparse — *Transform a .cool file to a sparse upper triangular matrix for input into hic_loess*

Description

Transform a .cool file to a sparse upper triangular matrix for input into hic_loess

Usage

```
cooler2sparse(cooler)
```

Arguments

`cooler`	The plain text file from a .cool file loaded into an R data.frame object. See vignette for more details.

Details

The .cool format is linked a database of Hi-C experiments and allows access to many sets of Hi-C data which can be found at the Index of Coolers `ftp://cooler.csail.mit.edu/coolers`. Once a .cool file is dumped into a contact matrix in plain text it can be read into R. This function provides a method for converting the .cool matrix into a sparse upper triangular matrix ready to be entered into hic_loess.

Value

A Sparse upper triangular matrix or a list of sparse upper triangular matrices. If the .cool file contains data for more than one chromosome The function will split the data up into a list of matrices, one per chromosome.

Examples

```
data('cooler')
sparse <- cooler2sparse(cooler)
head(sparse)
```

`create.hic.table` *Create hic.table object from a sparse upper triangular Hi-C matrix*

Description

Create hic.table object from a sparse upper triangular Hi-C matrix

Usage

```
create.hic.table(
  sparse.mat1,
  sparse.mat2,
  chr = NA,
  scale = TRUE,
  include.zeros = FALSE,
  subset.dist = NA,
  subset.index = NA,
  exclude.regions = NA,
  exclude.overlap = 0.2
)
```

Arguments

`sparse.mat1` Required, sparse upper triangular Hi-C matrix, 7 column BEDPE format of the upper triangle of the matrix, OR InteractionSet object with the genomic ranges of the interacting regions for the upper triangle of the Hi-C matrix and a single metadata column containing the interaction frequencies for each interacting pair for the first dataset you wish to jointly normalize.

`sparse.mat2` Required, sparse upper triangular Hi-C matrix, 7 column BEDPE format of the upper triangle of the matrix, OR InteractionSet object with the genomic ranges of the interacting regions for the upper triangle of the Hi-C matrix and a single metadata column containing the interaction frequencies for each interacting pair for the second dataset you wish to jointly normalize.

`chr` The chromosome name for the matrices being entered i.e 'chr1' or 'chrX'. Only needed if using sparse upper triangular matrix format. If using BEDPE format leave set to NA.

`scale` Logical, should scaling be applied to the matrices to adjust for total read counts. If TRUE the IFs of the second sparse matrix will be adjusted as follows: IF2_scaled = IF2 / (sum(IF2)/sum(IF1)).

`include.zeros` Logical, If set to TRUE the function will include pairwise interactions where one of the interaction frequencies is 0.

`subset.dist` Should the matrix be subset to only include interactions up to a user specified matrix unit distance? i.e. to only include the cells of the matrix which are at a unit distance less than or equal to 100 set `subset.dist = 100`. Subsetting the matrix by distance will cut out any interactions occuring at a unit distance greater

than the specified value. This could be used to speed up computation time or if there is only interest in the interactions occuring below a specific distance in the matrix. Warning: If you subset by distance do NOT to transform the subsetted hic.table into a full matrix using ‘sparse2full‘. If you plan on trasforming the matrix to a full contact matrix use subset.index instead.

`subset.index` Should the matrix be subset by a user specified distance? Input as a vector of 4 numbers (i.start, i.end, j.start, j.end). i.e. to only include a subset of the matrix with row numbers 20 <= i <= 40 and column numbers 30 <= j <= 50 set as `subset.index = c(20, 40, 30, 50)`. This can be used to speed up computation time if only a subset of the matrix is of interest. The indices used here correspond to the indices of the full Hi-C contact matrix. The ‘sparse2full‘ function can be used to view the full contact matrix and make a decision about subsetting based on index.

`exclude.regions`
A data.frame or genomic ranges object in the form of chr start end. Regions contained in the object will be removed from the hic.table object. Could be useful for excluding regions with a known CNV, blacklist regions, or some other a priori known difference.

`exclude.overlap`
The proportion of overlap required to exclude a region. Defaults to 0.2, indicating 20% or more overlap will be enough for exclusion. To exclude any amount of overlap set to 0. If set to 1, only a 100% overlap with an excluded regions will result in exclusion.

Details

This function is used to transform two sparse upper triangular Hi-C matrices into an object usable in the `hic_loess` function. Sparse upper triangular Hi-C matrix format is typical of the Hi-C data available from the Aiden Lab https://www.aidenlab.org/. If you have a full Hi-C contact matrix, first transform it to sparse upper triangular format using the `full2sparse` function. Sparse matrices should have 3 columns in the following order: Start location of region 1, Start location of region 2, Interaction Frequency. Matrices in 7 column BEDPE format should have 7 columns in the following order: Chromosome name of the first region, Start location of first region, End location of first region, Chromosome name of the second region, Start location of the second region, End location of the second region, Interaction Frequency. Please enter either two sparse matrices or two matrices in 7 column BEDPE format or two InteractionSet objects; do not mix and match.

Value

A hic.table object.

Examples

```
# Create hic.table object using included Hi-C data in sparse upper
# triangular matrix format
data('HMEC.chr22')
data('NHEK.chr22')
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr = 'chr22')
# View result
hic.table
```

`filter_params` *Determine the A quantile cutoff to be used*

Description

Determine the A quantile cutoff to be used

Usage

```
filter_params(
  hic.table,
  SD = 2,
  numChanges = 300,
  FC = 3,
  alpha = 0.05,
  Plot = FALSE
)
```

Arguments

`hic.table`	A hic.table object
`SD`	The standard deviation of the fuzzing used to produce a Hi-C matrix from your data with few true differences.
`numChanges`	The number of changes to add into the Hi-C matrix created. This should be proportional to the resolution of the data. High resolution data should use more changes i.e. 1MB resolution - 300 changes, 100KB resolution - 1000 changes, etc.
`FC`	The fold change of the changes added to the Hi-C matrix.
`alpha`	The alpha level for hypothesis testing.
`Plot`	logical, should MD plots for the normalization and difference detection be plotted?

Details

This function will take your data and produce an additional Hi-C matrix using the IF1 vector. Random normal noise will be added to the vector to create a "fuzzed" matrix with few true differences. Then the specified number of true changes will be added at the specified fold change level to the matrices. The HiCcompare procedure is run on the data and a plot of the MCC, TPR, and FPR based on the A minimum value filtered out will be produced. This is to aid you in determining what value you should use when analyzing your data with the hic_compare() function.

Value

A plot of the Mathews Correlation Coefficient (MCC), true positive rate (TPR), and false positive rate (FPR) over the A minimum value filtered.

Examples

```
data('HMEC.chr22')
data('NHEK.chr22')
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr = 'chr22')
filter_params(hic.table)
```

`full2sparse` *Transfrom a full Hi-C contact matrix to a sparse upper triangular matrix*

Description

Transfrom a full Hi-C contact matrix to a sparse upper triangular matrix

Usage

```
full2sparse(mat)
```

Arguments

`mat`	A matrix. Must have column names equal to the start location for each bin. i.e. for a 6x6 Hi-C matrix where the first region starts at 0 kb and the final region starts at 500KB and the resolution is 100kb, the column names of the matrix should be as follows: colnames(mat) = c(0, 100000, 200000, 300000, 400000, 500000)

Value

A sparse upper triangular matrix.

Examples

```
m <- matrix(1:100, 10, 10)
colnames(m) <- 1:10
sparse <- full2sparse(m)
sparse
```

hg19_blacklist *BED file for hg19 blacklisted regions*

Description

A BED file with regions which may be excluded from your Hi-C data analysis.

Usage

```
hg19_blacklist
```

Format

A data.frame with 3 columns and 1649 rows:

chr chromosome for the region

start start location of the region

end end location of the region

Value

A data.frame

Source

Data from UCSC http://genome.ucsc.edu/cgi-bin/hgFileUi?db=hg19&g=wgEncodeMapability

hg38_blacklist *BED file for hg38 blacklisted regions*

Description

A BED file with regions which may be excluded from your Hi-C data analysis.

Usage

```
hg38_blacklist
```

Format

A data.frame with 3 columns and 38 rows:

chr chromosome for the region

start start location of the region

end end location of the region

Value

A data.frame

Source

Data from http://mitra.stanford.edu/kundaje/akundaje/release/blacklists/hg38-human/

---

hicpro2bedpe *Convert HiC-Pro results to BEDPE format*

Description

Convert HiC-Pro results to BEDPE format

Usage

```
hicpro2bedpe(mat, bed)
```

Arguments

mat	The 3 column sparse upper triangular matrix from HiC-Pro.
bed	The BED file containing the mappings for for the matrix.

Details

HiC-Pro will produce a .matrix file and a .bed file as the final aligned product of the alignment process. These files should be read into R using read.table() or similar and then entered as the mat and bed inputs to this function. The function will convert the data into a format useable for HiCcompare. The cis matrices in the results can be directly input into create.hic.table() as sparse matrices.

Value

A list with two items. Item 1, "cis" contains the intra-chromosomal contact matrices, one per chromosome. Item 2, "trans" contains the inter-chromsomal contact matrix.

Examples

```
## Not run:
 # read in data
 mat <- read.table("hic_1000000.matrix")
 bed <- read.table("hic_1000000_abs.bed")
 # convert to BEDPE
 dat <- hicpro2bedpe(mat, bed)

## End(Not run)
```

`hic_compare` *Detect differences between two jointly normalized Hi-C datasets.*

Description

Detect differences between two jointly normalized Hi-C datasets.

Usage

```
hic_compare(
  hic.table,
  A.min = NA,
  adjust.dist = TRUE,
  p.method = "fdr",
  Plot = FALSE,
  Plot.smooth = TRUE,
  parallel = FALSE,
  BP_param = bpparam()
)
```

Arguments

`hic.table`	A hic.table or list of hic.tables output from the `hic_loess` function. hic.table must be jointly normalized before being entered.
`A.min`	The required value of A in order for a differences to be considered. All Z-scores where the corresponding A value is < A.min will be set to 0. Defaults to NA. If NA, then the 10th percentile of A will automatically be calculated and set as the A.min value. To better determine how to set A.min see the help for ?filter_params().
`adjust.dist`	Logical, should the p-value adjustment be performed on a per distance basis. i.e. The p-values at distance 1 will be grouped and the p-value adjustment will be applied. This process is repeated for each distance. The highest 15 if you matrix has a maximum distance of 100, then distances 85-100 will be pooled together for p-value adjustment.
`p.method`	The method for p-value adjustment. See ?p.adjust() help for options and more information. Defaults to "fdr". Can be set to "none" for no p-value adjustments.
`Plot`	Logical, should the MD plot showing before/after loess normalization be output?
`Plot.smooth`	Logical, defaults to TRUE indicating the MD plot will be a smooth scatter plot. Set to FALSE for a scatter plot with discrete points.
`parallel`	Logical, set to TRUE to utilize the `parallel` package's parallelized computing. Only works on unix operating systems. Only useful if entering a list of hic.tables.
`BP_param`	Parameters for BiocParallel. Defaults to bpparam(), see help for BiocParallel for more information http://bioconductor.org/packages/release/bioc/vignettes/BiocParallel/inst/doc/Introduction_To_BiocParallel.pdf

Details

The function takes in a hic.table or a list of hic.table objects created with the `hic_loess` function. If you wish to perform difference detection on Hi-C data for multiple chromosomes use a list of hic.tables. The process can be parallelized using the `parallel` setting. The adjusted IF and adjusted M calculated from `hic_loess` are used for difference detection. Difference detection is performed by converting adjusted M values to Z-scores. Any M value with a corresponding average expression level (A; mean of IF1 and IF2) less than the specified A.quantile is not considered for Z-score calculation. This throws out the untrustworthy interactions that tend to produce false positives. The Z-scores are assumed to follow a roughly standard normal distribution and p-values are obtained. P-value adjustment for multiple testing is then performed on a per distance basis (or on all p-values, optionally). i.e. at each distance the vector of p-values corresponding to the interactions occuring at that distance have the selected multiple testing correction applied to them. See methods of Stansfield & Dozmorov 2017 for more details.

Value

A hic.table with additional columns containing a p-value for the significance of the difference and the raw fold change between the IFs of the two datasets.

Examples

```
# Create hic.table object using included Hi-C data in sparse upper triangular
# matrix format
data('HMEC.chr22')
data('NHEK.chr22')
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr = 'chr22')
# Plug hic.table into hic_loess()
result <- hic_loess(hic.table, Plot = TRUE)
# perform difference detection
diff.result <- hic_compare(result, Plot = TRUE)
```

`hic_diff` *Detect differences between two jointly normalized Hi-C datasets. OLD METHOD; USE hic_compare() instead*

Description

Detect differences between two jointly normalized Hi-C datasets. OLD METHOD; USE hic_compare() instead

Usage

```
hic_diff(
  hic.table,
  diff.thresh = "auto",
  iterations = 10000,
  Plot = FALSE,
```

```
  Plot.smooth = TRUE,
  parallel = FALSE,
  BP_param = bpparam()
)
```

Arguments

`hic.table`	A hic.table or list of hic.tables output from the `hic_loess` function. hic.table must be jointly normalized before being entered.
`diff.thresh`	Fold change threshold desired to call a detected difference significant. Set to 'auto' by default to indicate that the difference threshold will be automatically calculated as 2 standard deviations of all the adjusted M values. For no p-value adjustment set diff.thresh = NA. To set your own threshold enter a numeric value i.e. diff.thresh = 1. If set to 'auto' or a numeric value, a check will be made as follows: if permutation p-value < 0.05 AND M < diff.thresh (the log2 fold change for the difference between IF1 and IF2) then the p-value will be set to 0.5.
`iterations`	Number of iterations for the permuation test.
`Plot`	Logical, should the MD plot showing before/after loess normalization be output?
`Plot.smooth`	Logical, defaults to TRUE indicating the MD plot will be a smooth scatter plot. Set to FALSE for a scatter plot with discrete points.
`parallel`	Logical, set to TRUE to utilize the `parallel` package's parallelized computing. Only works on unix operating systems. Only useful if entering a list of hic.tables.
`BP_param`	Parameters for BiocParallel. Defaults to bpparam(), see help for BiocParallel for more information http://bioconductor.org/packages/release/bioc/vignettes/BiocParallel/inst/doc/Introduction_To_BiocParallel.pdf

Details

This is the old method for detecting difference. The function is left in for legacy reasons and it is recommended to use the new function, hic_compare(), instead. The function takes in a hic.table or a list of hic.table objects created with the `hic_loess` function. If you wish to perform difference detection on Hi-C data for multiple chromosomes use a list of hic.tables. The process can be parallelized using the `parallel` setting. The adjusted IF and adjusted M calculated from `hic_loess` are used for difference detection. A permutation test is performed to test the significance of the difference between each IF of the two datasets. Permutations are broken in blocks for each unit distance. See methods section of Stansfield & Dozmorov 2017 for more details.

Value

A hic.table with additional columns containing a p-value for the significance of the difference and the raw fold change between the IFs of the two datasets.

Examples

```
# Create hic.table object using included Hi-C data in sparse upper triangular
# matrix format
data('HMEC.chr22')
data('NHEK.chr22')
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr = 'chr22')
# Plug hic.table into hic_loess()
result <- hic_loess(hic.table, Plot = TRUE)
# perform difference detection
diff.result <- hic_diff(result, diff.thresh = 'auto', Plot = TRUE)
```

---

hic_loess *Perform joint loess normalization on two Hi-C datasets*

Description

Perform joint loess normalization on two Hi-C datasets

Usage

```
hic_loess(
  hic.table,
  degree = 1,
  span = NA,
  loess.criterion = "gcv",
  Plot = FALSE,
  Plot.smooth = TRUE,
  parallel = FALSE,
  BP_param = bpparam()
)
```

Arguments

hic.table	hic.table or a list of hic.tables generated from the create.hic.table function. list of hic.tables generated from the create.hic.table function. If you want to perform normalization over multiple chromosomes from each cell line at once utilizing parallel computing enter a list of hic.tables and set parallel = TRUE.
degree	Degree of polynomial to be used for loess. Options are 0, 1, 2. The default setting is degree = 1.
span	User set span for loess. If set to NA, the span will be selected automatically using the setting of loess.criterion. Defaults to NA so that automatic span selection is performed. If you know the span, setting it manually will significantly speed up computational time.

`loess.criterion`
: Automatic span selection criterion. Can use either 'gcv' for generalized cross-validation or 'aicc' for Akaike Information Criterion. Span selection uses a slightly modified version of the `loess.as()` function from the `fANCOVA` package. Defaults to 'gcv'.

`Plot`
: Logical, should the MD plot showing before/after loess normalization be output? Defaults to FALSE.

`Plot.smooth`
: Logical, defaults to TRUE indicating the MD plot will be a smooth scatter plot. Set to FALSE for a scatter plot with discrete points.

`parallel`
: Logical, set to TRUE to utilize the `parallel` package's parallelized computing. Only works on unix operating systems. Only useful if entering a list of hic.tables. Defauts to FALSE.

`BP_param`
: Parameters for BiocParallel. Defaults to bpparam(), see help for BiocParallel for more information http://bioconductor.org/packages/release/bioc/vignettes/BiocParallel/inst/doc/Introduction_To_BiocParallel.pdf

Details

The function takes in a hic.table or a list of hic.table objects created with the `create.hic.loess` function. If you wish to perform joint normalization on Hi-C data for multiple chromosomes use a list of hic.tables. The process can be parallelized using the `parallel` setting. The data is fist transformed into what is termed an MD plot (similar to the MA plot/Bland-Altman plot). M is the log difference log2(x/y) between the two datasets. D is the unit distance in the contact matrix. The MD plot can be visualized with the `Plot` option. Loess regression is then performed on the MD plot to model any biases between the two Hi-C datasets. An adjusted IF is then calculated for each dataset along with an adjusted M. See methods section of Stansfield & Dozmorov 2017 for more details. Note: if you receive the warning "In simpleLoess(y, x, w, span, degree = degree, parametric = parametric, ... :pseudoinverse used..." it should not effect your results, however it can be avoided by manually setting the span to a larger value using the span option.

Value

An updated hic.table is returned with the additional columns of adj.IF1, adj.IF2 for the respective normalized IFs, an adj.M column for the adjusted M, mc for the loess correction factor, and A for the average expression value between adj.IF1 and adj.IF2.

Examples

```
# Create hic.table object using included Hi-C data in sparse upper
# triangular matrix format
data("HMEC.chr22")
data("NHEK.chr22")
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr= 'chr22')
# Plug hic.table into hic_loess()
result <- hic_loess(hic.table, Plot = TRUE)
# View result
result
```

hic_simulate *Simulate a Hi-C matrix and perform HiCcompare analysis on it*

Description

Simulate a Hi-C matrix and perform HiCcompare analysis on it

Usage

```
hic_simulate(
  nrow = 100,
  medianIF = 50000,
  sdIF = 14000,
  powerlaw.alpha = 1.8,
  sd.alpha = 1.9,
  prop.zero.slope = 0.001,
  centromere.location = NA,
  CNV.location = NA,
  CNV.proportion = 0.8,
  CNV.multiplier = 0,
  biasFunc = .normal.bias,
  fold.change = NA,
  i.range = NA,
  j.range = NA,
  Plot = TRUE,
  scale = TRUE,
  alpha = 0.05,
  diff.thresh = NA,
  include.zeros = FALSE
)
```

Arguments

nrow	Number of rows and columns of the full matrix
medianIF	The starting value for a power law distribution for the interaction frequency of the matrix. Should use the median value of the IF at distance = 0. Typical values for 1MB data are around 50,000. For 500kb data typical values are 25,000. For 100kb data, 4,000. For 50kb data, 1,800.
sdIF	The estimated starting value for a power law distriubtion for the standard deviaton of the IFs. Should use the SD of the IF at distance = 0. Typical value for 1MB data is 19,000.
powerlaw.alpha	The exponential parameter for the power law distribution for the median IF. Typical values are 1.6 to 2. Defaults to 1.8.
sd.alpha	The exponential parameter for the power law distribution for the SD of the IF. Typical values are 1.8 to 2.2. Defaults to 1.9.

`prop.zero.slope`
: The slope to be used for a linear function of the probability of zero in matrix = slope * distance

`centromere.location`
: The location for a centromere to be simulated. Should be entered as a vector of 2 numbers; the start column number and end column number. i.e. to put a centromere in a 100x100 matrix starting at column 47 and ending at column 50 enter centromere.location = c(47, 50). Defaults NA indicating no simulated centromere will be added to the matrix.

`CNV.location`
: The location for a copy number variance (CNV). Should be entered as a vector of 2 numbers; the start column number and end column number. i.e. to put a CNV in a 100x100 matrix starting at column 1 and ending at column 50 enter CNV.location = c(1, 50). Defaults NA indicating no simulated CNV will be added to the matrices. If a value is entered one of the matrices will have a CNV applied to it.

`CNV.proportion`
: The proportion of 0's to be applied to the CNV location specified. Defaults to 0.8.

`CNV.multiplier`
: A multiplyer to be applied as the CNV. To approximate deletion set to 0, to increase copy numbers set to a value > 1. Defaults to 0.

`biasFunc`
: A function used for adding bias to one of the simulated matrices. Should take an input of unit distance and generally have the form of 1 + Probability Density Function with unit distance as the random variable. Can also use a constant as a scaling factor to add a global offset to one of the matrices. The output of the bias function will be multiplied to the IFs of one matrix. Included are a normal kernel bias and a no bias function. If no function is entered, a normal kernel bias with an additional global scaling factor of 4 will be used. To use no bias set biasFunc = .no.bias, see examples section.

`fold.change`
: The fold change you want to introduce for true differences in the simulated matrices. Defaults to NA for no fold change added.

`i.range`
: The row numbers for the cells that you want to introduce true differences at. Must be same length as j.range. Defaults to NA for no changes added.

`j.range`
: The column numbers for the cells that you want to introduce true differences at. Must be same length as Defaults to NA for no changes added.

`Plot`
: Logical, should the HiCdiff plots be output? Defaults to TRUE.

`scale`
: Logical, Should scaling be applied for the HiCdiff procedure? Defaults to TRUE.

`alpha`
: Type I error rate parameter. At what level should a significant difference be defined. Defaults to 0.05.

`diff.thresh`
: Parameter for hic_diff procedure. See ?hic_diff for more help. Defaults to NA.

`include.zeros`
: Should partial zero interactions be included? Defaults to FALSE.

Value

A list containing the true positive rate (TPR), the specificity (SPC), the p-values, the hic.table object, true differences - a data.table of the rows of the hic.table where a true difference was applied, the

truth vector - a vector of 0's and 1's where 1 indicates a true difference was applied to that cell, sim.table - the hic.table object for the simulate matrices before hic_loess and hic_compare was run on it.

Examples

```
# simulate two matrices with no fold changes introduced using default values
sim <- hic_simulate()

# example of bias functions
## the default function used
.normal.bias = function(distance) {
  (1 + exp(-((distance - 20)^2) / (2*30))) * 4
}

## an additional bias function
.no.bias = function(distance) {
  1
}

# simulate matrices with 200 true differences using no bias
i.range = sample(1:100, replace=TRUE)
j.range = sample(1:100, replace=TRUE)
sim2 <- hic_simulate(nrow=100, biasFunc = .no.bias, fold.change = 5,
                     i.range = i.range, j.range = j.range)
```

`HMEC.chr10` *Hi-C data from HMEC cell line - chromosome 10 at 500kb resolution*

Description

A sparse upper triangular matrix containing the interacting regions and the corresponding Interaction Frequency (IF).

Usage

```
HMEC.chr10
```

Format

A matrix with 35386 rows and 3 columns:

region1 The first interacting region - corresponds to the row name of a full Hi-C contact matrix

region2 The second interacting region - corresponds to the column name of a full Hi-C contact matrix

IF The Interaction Frequency - number of read counts found for the interaction between region1 and region2

Value

A matrix

Source

Data from the Aiden Lab. See their website at https://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525

HMEC.chr22 — *Hi-C data from HMEC cell line - chromosome 22 at 500kb resolution*

Description

A sparse upper triangular matrix containing the interacting regions and the corresponding Interaction Frequency (IF).

Usage

```
HMEC.chr22
```

Format

A matrix with 2490 rows and 3 columns:

region1 The first interacting region - corresponds to the row name of a full Hi-C contact matrix

region2 The second interacting region - corresponds to the column name of a full Hi-C contact matrix

IF The Interaction Frequency - number of read counts found for the interaction between region1 and region2

Value

A matrix

Source

Data from the Aiden Lab. See their website at https://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525

hmec.IS — *Hi-C data from HMEC cell line - chromosome 22 at 500kb resolution*

Description

An InteractionSet object 2490 interactions and 1 metadata column containing the Interaction Frequencies

Usage

```
hmec.IS
```

Format

An InteractionSet with 2490 rows and 1 metadata column:

IF The Interaction Frequency - number of read counts found for the interaction between region1 and region2

Value

A matrix

Source

Data from the Aiden Lab. See their website at https://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525

KRnorm — *Performs KR (Knight-Ruiz) normalization on a Hi-C matrix*

Description

Performs KR (Knight-Ruiz) normalization on a Hi-C matrix

Usage

```
KRnorm(A)
```

Arguments

A	the matrix to be normalized - any columns/rows of 0's will be removed before being normalized.

Details

Performs KR normalization. The function is a translation of the ‘Matlab‘ code provided in the 2012 manuscript. Knight PA, Ruiz D. A fast algorithm for matrix balancing. IMA Journal of Numerical Analysis. Oxford University Press; 2012; drs019.

Value

A KR normalized matrix

Examples

```
m <- matrix(rpois(100, 5), 10, 10)
KRnorm(m)
```

---

make_InteractionSet *Convert HiCdiff results to InteractionSet object*

Description

Convert HiCdiff results to InteractionSet object

Usage

```
make_InteractionSet(hic.table)
```

Arguments

`hic.table`	A hic.table object.

Details

This function will convert data from HiCdiff results in the hic.table object format to the InteractionSet format which makes use of GRanges objects.

Value

An object of class InteractionSet

Examples

```
# create hic.table
data(HMEC.chr22)
data(NHEK.chr22)
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr='chr22')
# convert to InteractionSet
gi <- make_InteractionSet(hic.table)
```

manhattan_plot *Create a Manhattan plot for the results of HiCcompare*

Description

Create a Manhattan plot for the results of HiCcompare

Usage

```
manhattan_plot(hic.table, adj.p = TRUE, alpha = 0.05, return_df = FALSE)
```

Arguments

hic.table	a hic.table object that has been normalized and had differences detected.
adj.p	Logical, should the adjusted p-value be used (TRUE) of the raw p-value (FALSE)?
alpha	The alpha level for calling a p-value significant.
return_df	Logical, should the data.frame built to be used for plotting be returned? If TRUE then the data.frame will be returned and the plot will only be printed.

Details

This function will produce a manhattan plot of the results of hic_compare(). Can be used to display which regions around found to be significantly different on the linear genome.

Value

A manhattan plot.

Examples

```
# Create hic.table object using included Hi-C data in
# sparse upper triangular matrix format
data('HMEC.chr22')
data('NHEK.chr22')
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr = 'chr22')
# Plug hic.table into hic_loess()
result <- hic_loess(hic.table, Plot = TRUE)
# perform difference detection
diff.result <- hic_compare(result, Plot = TRUE)
# make manhattan plot
manhattan_plot(diff.result)
```

MA_norm *Perform MA normalization on a hic.table object*

Description

Perform MA normalization on a hic.table object

Usage

```
MA_norm(
  hic.table,
  degree = 2,
  Plot = FALSE,
  span = NA,
  loess.criterion = "gcv"
)
```

Arguments

hic.table	A hic.table object
degree	The degree for loess normalization
Plot	logical, should the MA plot be output?
span	The span for loess. If left as the default value of NA the span will be calculated automatically
loess.criterion	The criterion for calculating the span for loess

Details

Performs loess normalization on the MA plot of the data.

Value

An extended hic.table with adjusted IFs and M columns.

Examples

```
# create hic.table
data("HMEC.chr22")
data("NHEK.chr22")
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr= 'chr22')
# Plug hic.table into MA_norm()
MA_norm(hic.table)
```

MD.plot1 *Visualize the MD plot before and after loess normalization*

Description

Visualize the MD plot before and after loess normalization

Usage

```
MD.plot1(M, D, mc, smooth = TRUE)
```

Arguments

M	The M component of the MD plot. Available from the hic.table object.
D	The D component of the MD plot. The unit distance of the interaction. Available from the hic.table object.
mc	The correction factor. Calculated by `hic_loess`. Available from the hic.table object after running `hic_loess`.
smooth	Should smooth scatter plots be used? If set to FALSE ggplot scatter plots will be used. When option is TRUE, plots generate quicker. It is recommend to use the smooth scatter plots.

Value

An MD plot.

Examples

```
# Create hic.table object using included Hi-C data in sparse upper
# triangular matrix format
data('HMEC.chr22')
data('NHEK.chr22')
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr = 'chr22')
# Plug hic.table into hic_loess()
result <- hic_loess(hic.table)
MD.plot1(result$M, result$D, result$mc)
```

MD.plot2 *Visualize the MD plot.*

Description

Visualize the MD plot.

Usage

```
MD.plot2(M, D, p.val = NA, diff.thresh = NA, smooth = TRUE)
```

Arguments

M	The M component of the MD plot. Available from the hic.table object.
D	The D component of the MD plot. The unit distance of the interaction. Available from the hic.table object.
p.val	An optional p-value vector to provide color to the plot based on the significance of the differences between the IFs.
diff.thresh	A difference threshold used for calculating p-values. If set to a value will add dotted horizontal lines to the plot to display the threshold cutoffs. See ‘hic_loess‘ or ‘hic_diff‘ functions help for more information on this parameter.
smooth	Should smooth scatter plots be used? If set to FALSE ggplot scatter plots will be used. When option is TRUE, plots generate quicker. It is recommend to use the smooth scatter plots.

Value

An MD plot.

Examples

```
data('HMEC.chr22')
data('NHEK.chr22')
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr='chr22')
# Plug hic.table into hic_loess()
result <- hic_loess(hic.table)
# perform difference detection
diff.result <- hic_compare(result)
MD.plot2(diff.result$M, diff.result$D, diff.result$p.value)
```

NHEK.chr10 — *Hi-C data from NHEK cell line - chromosome 10 at 500kb resolution*

Description

A sparse upper triangular matrix containing the interacting regions and the corresponding Interaction Frequency (IF).

Usage

```
NHEK.chr10
```

Format

A matrix with 35510 rows and 3 columns:

region1 The first interacting region - corresponds to the row name of a full Hi-C contact matrix

region2 The second interacting region - corresponds to the column name of a full Hi-C contact matrix

IF The Interaction Frequency - number of read counts found for the interaction between region1 and region2

Value

A matrix

Source

Data from the Aiden Lab. See their website at https://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525

NHEK.chr22 — *Hi-C data from NHEK cell line - chromosome 22 at 500kb resolution*

Description

A sparse upper triangular matrix containing the interacting regions and the corresponding Interaction Frequency (IF).

Usage

```
NHEK.chr22
```

Format

A matrix with 2508 rows and 3 columns:

region1 The first interacting region - corresponds to the row name of a full Hi-C contact matrix

region2 The second interacting region - corresponds to the column name of a full Hi-C contact matrix

IF The Interaction Frequency - number of read counts found for the interaction between region1 and region2

Value

A matrix

Source

Data from the Aiden Lab. See their website at https://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525

---

nhek.IS — *Hi-C data from NHEK cell line - chromosome 22 at 500kb resolution*

Description

An InteractionSet object 2508 interactions and 1 metadata column containing the Interaction Frequencies

Usage

```
nhek.IS
```

Format

An InteractionSet with 2508 rows and 1 metadata column:

IF The Interaction Frequency - number of read counts found for the interaction between region1 and region2

Value

A matrix

Source

Data from the Aiden Lab. See their website at https://www.aidenlab.org/ Or the the GEO link to download the data https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525

remove_centromere — *Function to remove centromere columns and rows from a full Hi-C contact matrix*

Description

Function to remove centromere columns and rows from a full Hi-C contact matrix

Usage

```
remove_centromere(mat)
```

Arguments

mat	A full Hi-C matrix

Value

A list of (1) the column/row numbers of the centromere and (2) the Hi-c matrix with the centromere removed

Examples

```
m <- matrix(rpois(100, 5), 10, 10)
m[5,] <- 0
m[,5] <- 0
remove_centromere(m)
```

SCN — *SCN normalization from Cournac 2012*

Description

SCN normalization from Cournac 2012

Usage

```
SCN(a, max.iter = 10)
```

Arguments

a	The matrix to be normalized. Any cols/rows that sum to 0 will be removed before normalization.
max.iter	maximum number of iterations to be performed

Details

Performs Sequential Component Normalization as described by Cournac. Coded using details in the manuscript. Cournac A, Marie-Nelly H, Marbouty M, Koszul R, Mozziconacci J. Normalization of a chromosomal contact map. BMC Genomics. 2012;13: 436. doi:10.1186/1471-2164-13-436

Value

An SCN normalized matrix

Examples

```
m <- matrix(rpois(100, 5), 10, 10)
SCN(m)
```

---

sim.other.methods *Compare other normalization methods on simulated data*

Description

Compare other normalization methods on simulated data

Usage

```
sim.other.methods(
  sim.table,
  i.range,
  j.range,
  Plot = TRUE,
  alpha = 0.05,
  diff.thresh = NA
)
```

Arguments

sim.table	the sim.table object output from hic_simulate
i.range	The row numbers for the cells that you want to introduce true differences at. Must be same length as j.range.
j.range	The column numbers for the cells that you want to introduce true differences at. Must be same length as i.range.
Plot	Logical, should the HiCdiff plots be output? Defaults to TRUE.
alpha	Type I error rate parameter. At what level should a significant difference be defined. Defaults to 0.05.
diff.thresh	Parameter for hic_diff procedure. see ?hic_diff for more details.

Value

A list containing the true positive rate (TPR), the specificity (SPC), the p-values, the hic.table object, true differences - a data.table of the rows of the hic.table where a true difference was applied, the truth vector - a vector of 0's and 1's where 1 indicates a true difference was applied to that cell.

Examples

```
i.range = sample(1:100, replace=TRUE)
j.range = sample(1:100, replace=TRUE)
sim <- hic_simulate(i.range = i.range, j.range = j.range, fold.change = 2)
mat1 <- sim$sim.table[, c('start1', 'start2', 'IF1'), with=FALSE]
mat2 <- sim$sim.table[, c('start1', 'start2', 'IF2'), with=FALSE]
mat1 <- sparse2full(mat1) %>% KRnorm
mat2 <- sparse2full(mat2) %>% KRnorm
colnames(mat1) <- 1:ncol(mat1)
colnames(mat2) <-1:ncol(mat2)
mat1 <- full2sparse(mat1)
mat2 <- full2sparse(mat2)
new.tab <- create.hic.table(mat1, mat2, chr= 'chrsim')
sim2 <- sim.other.methods(new.tab, i.range = i.range , j.range = j.range)
```

---

sim_matrix *Simulate 2 Hi-C matrices with differences*

Description

Simulate 2 Hi-C matrices with differences

Usage

```
sim_matrix(
  nrow = 100,
  medianIF = 50000,
  sdIF = 14000,
  powerlaw.alpha = 1.8,
  sd.alpha = 1.9,
  prop.zero.slope = 0.001,
  centromere.location = NA,
  CNV.location = NA,
  CNV.proportion = 0.8,
  CNV.multiplier = 0,
  biasFunc = .normal.bias,
  fold.change = NA,
  i.range = NA,
  j.range = NA
)
```

Arguments

`nrow` Number of rows and columns of the full matrix

`medianIF` The starting value for a power law distribution for the interaction frequency of the matrix. Should use the median value of the IF at distance = 0. Typical values for 1MB data are around 50,000. For 500kb data typical values are 25,000. For 100kb data, 4,000. For 50kb data, 1,800.

`sdIF` The estimated starting value for a power law distriubtion for the standard deviaton of the IFs. Should use the SD of the IF at distance = 0. Typical value for 1MB data is 19,000.

`powerlaw.alpha` The exponential parameter for the power law distribution for the median IF. Typical values are 1.6 to 2. Defaults to 1.8.

`sd.alpha` The exponential parameter for the power law distribution for the SD of the IF. Typical values are 1.8 to 2.2. Defaults to 1.9.

`prop.zero.slope` The slope to be used for a linear function of the probability of zero in matrix = slope * distance

`centromere.location` The location for a centromere to be simulated. Should be entered as a vector of 2 numbers; the start column number and end column number. i.e. to put a centromere in a 100x100 matrix starting at column 47 and ending at column 50 enter centromere.location = c(47, 50). Defaults NA indicating no simulated centromere will be added to the matrix.

`CNV.location` The location for a copy number variance (CNV). Should be entered as a vector of 2 numbers; the start column number and end column number. i.e. to put a CNV in a 100x100 matrix starting at column 1 and ending at column 50 enter CNV.location = c(1, 50). Defaults NA indicating no simulated CNV will be added to the matrices. If a value is entered one of the matrices will have a CNV applied to it.

`CNV.proportion` The proportion of 0's to be applied to the CNV location specified. Defaults to 0.8.

`CNV.multiplier` A multiplyer to be applied as the CNV. To approximate deletion set to 0, to increase copy numbers set to a value > 1. Defaults to 0.

`biasFunc` A function used for adding bias to one of the simulated matrices. Should take an input of unit distance and generally have the form of 1 + Probability Density Function with unit distance as the random variable. Can also use a constant as a scaling factor to add a global offset to one of the matrices. The output of the bias function will be multiplied to the IFs of one matrix. Included are a normal kernel bias and a no bias function. If no function is entered, a normal kernel bias with an additional global scaling factor of 4 will be used. To use no bias set biasFunc = .no.bias, see examples section.

`fold.change` The fold change you want to introduce for true differences in the simulated matrices. Defaults to NA for no fold change added.

`i.range` The row numbers for the cells that you want to introduce true differences at. Must be same length as j.range. Defaults to NA for no changes added.

`j.range` The column numbers for the cells that you want to introduce true differences at. Must be same length as Defaults to NA for no changes added.

Value

A hic.table object containing simulated Hi-C matrices.

Examples

```
# simulate two matrices with no fold changes introduced using default values
sim <- hic_simulate()

# example of bias functions
## the default function used
.normal.bias = function(distance) {
  (1 + exp(-((distance - 20)^2) / (2*30))) * 4
}

## an additional bias function
.no.bias = function(distance) {
  1
}

# simulate matrices with 200 true differences using no bias
i.range = sample(1:100, replace=TRUE)
j.range = sample(1:100, replace=TRUE)
sim2 <- hic_simulate(nrow=100, biasFunc = .no.bias, fold.change = 5,
                     i.range = i.range, j.range = j.range)
```

---

sparse2full — *Transform a sparse upper triangular matrix to a full Hi-C contact matrix*

Description

sparse2full will transform a sparse upper triangular Hi-C matrix to a full Hi-C chromatin contact matrix. If you are entering a simple sparse matrix, i.e. there are only 3 columns leave hic.table = FALSE and column.name = NA. If you wish to transform a Hi-C matrix in hic.table object format into a full matrix then set hic.table = TRUE. You will then need to specify the column name that you wish to be entered as the values for the cells in the full matrix using the column.name option.

Usage

```
sparse2full(sparse.mat, hic.table = FALSE, column.name = NA)
```

Arguments

`sparse.mat`	A matrix in sparse upper triangular format.
`hic.table`	Logical, is your sparse.mat a hic.table?
`column.name`	Character, Required if hic.table set to TRUE; The column name of the hic.table that you want placed into the cells of the full matrix. i.e. IF1, or p.value.

Value

A full Hi-C contact Matrix.

Examples

```
data('NHEK.chr22')
full.mat <- sparse2full(NHEK.chr22)
```

---

split_centromere *Function to split hic.table into 2 subsets at the centromere*

Description

Function to split hic.table into 2 subsets at the centromere

Usage

```
split_centromere(hic.table)
```

Arguments

hic.table	A hic.table or list of hic.table objects

Value

A list of hic.tables in which the matrices have been split at the centromere

Examples

```
data('HMEC.chr22')
data('NHEK.chr22')
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr = 'chr22')
split <- split_centromere(hic.table)
```

`total_sum` *Total sum normalization for a list of hic.table objects*

Description

Total sum normalization for a list of hic.table objects

Usage

```
total_sum(hic.list)
```

Arguments

`hic.list`	A list of hic.table objects created by the create.hic.table function. When creating the hic.list to be entered into this function you must set the scale option to FALSE.

Details

This function will scale the IFs based on the total sum of the counts for the genome instead of on a per chromosome basis as done in the create.hic.table function when the scale option is set to TRUE. The idea behind this function to preserve more local CNV differences while still accounting for technical biases which can cause the total read counts to differ between sequencing runs.

Value

A list of hic.table objects.

Examples

```
data('HMEC.chr22')
data('NHEK.chr22')
data('HMEC.chr10')
data('NHEK.chr10')
hic.table1 <- create.hic.table(HMEC.chr22, NHEK.chr22,
    chr = 'chr22', scale = FALSE)
hic.table2 <- create.hic.table(HMEC.chr10, NHEK.chr10,
    chr = 'chr10', scale = FALSE)
hic.list <- list(hic.table1, hic.table2)
scaled_list <- total_sum(hic.list)
```

visualize_pvals — *Function to visualize p-values from HiCcompare results*

Description

Function to visualize p-values from HiCcompare results

Usage

```
visualize_pvals(hic.table, alpha = NA, adj.p = TRUE)
```

Arguments

hic.table	A hic.table 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.
adj.p	Logical, Should the multiple testing corrected p-values be used (TRUE) or the raw p-values (FALSE)?

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(adj.M) to visualize where the significant differences between the datasets are occuring on the genome.

Value

A heatmap

Examples

```
# Create hic.table object using included Hi-C data in sparse upper triangular
# matrix format
data('HMEC.chr22')
data('NHEK.chr22')
hic.table <- create.hic.table(HMEC.chr22, NHEK.chr22, chr = 'chr22')
# Plug hic.table into hic_loess()
result <- hic_loess(hic.table, Plot = TRUE)
# perform difference detection
diff.result <- hic_compare(result, Plot = TRUE)
# visualize p-values
visualize_pvals(diff.result)
```

Index

∗ **datasets**
    brain_table, 4
    centromere_locations, 4
    cooler, 5
    hg19_blacklist, 11
    hg38_blacklist, 11
    HMEC.chr10, 20
    HMEC.chr22, 21
    hmec.IS, 22
    NHEK.chr10, 28
    NHEK.chr22, 28
    nhek.IS, 29

brain_table, 4

centromere_locations, 4
cooler, 5
cooler2bedpe, 5
cooler2sparse, 6
create.hic.table, 7

filter_params, 9
full2sparse, 10

hg19_blacklist, 11
hg38_blacklist, 11
hic_compare, 13
hic_diff, 14
hic_loess, 16
hic_simulate, 18
HiCcompare *(HiCcompare-package)*, 3
HiCcompare-package, 3
hicpro2bedpe, 12
HMEC.chr10, 20
HMEC.chr22, 21
hmec.IS, 22

KRnorm, 22

MA_norm, 25
make_InteractionSet, 23
manhattan_plot, 24
MD.plot1, 26
MD.plot2, 27

NHEK.chr10, 28
NHEK.chr22, 28
nhek.IS, 29

remove_centromere, 30

SCN, 30
sim.other.methods, 31
sim_matrix, 32
sparse2full, 34
split_centromere, 35

total_sum, 36

visualize_pvals, 37