

Package ‘HiCcompare’

April 12, 2018

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

Version 1.0.0

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 permutation 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, QDNAseq, KernSmooth, methods, utils, graphics, pheatmap, gtools

biocViews Software, HiC, Sequencing, Normalization

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author John Stansfield [aut, cre],
Mikhail Dozmorov [aut]

Maintainer John Stansfield <stansfieldjc@vcu.edu>

R topics documented:

HiCcompare-package	2
brain_table	3
centromere_locations	3
cooler	4
cooler2sparse	4
create.hic.table	5
full2sparse	7
get_CNV	7
hg19_blacklist	8
hg38_blacklist	9
hicpro2bedpe	10
hic_diff	10
hic_loess	12
hic_simulate	13
HMEC.chr10	15
HMEC.chr22	16
hmec.IS	17
KRnorm	17
make_InteractionSet	18
MA_norm	19
MD.plot1	19
MD.plot2	20
NHEK.chr10	21
NHEK.chr22	22
nhek.IS	22
plot_A	23
remove_centromere	24
SCN	24
sim.other.methods	25
sparse2full	26
split_centromere	26
total_sum	27
visualize_pvals	28
weight_M	29

Index	31
--------------	-----------

HiCcompare-package	<i>HiCcompare</i>
--------------------	-------------------

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: John Stansfield <stansfieldjc@vcu.edu>

Authors:

- Mikhail Dozmorov <mikhail.dozmorov@vcuhealth.org>

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	<i>Hi-C data in the cooler format</i>
--------	---------------------------------------

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-filtered.1000kb.cool>

cooler2sparse	<i>Transform a .cool file to a sparse upper triangular matrix for input into hic_loess</i>
---------------	--------------------------------------------------------------------------------------------

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)
```

<code>create.hic.table</code>	<i>Create hic.table object from a sparse upper triangular Hi-C matrix</i>
-------------------------------	---------------------------------------------------------------------------

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

- | | |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sparse.mat1</code> | 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. |
| <code>sparse.mat2</code> | 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. |
| <code>chr</code> | 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. |
| <code>scale</code> | 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: $IF2_scaled = IF2 / (\text{sum}(IF2)/\text{sum}(IF1))$. |
| <code>include.zeros</code> | Logical, If set to TRUE the function will include pairwise interactions where one of the interaction frequencies is 0. |
| <code>subset.dist</code> | 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 <code>subset.dist = 100</code> . Subsetting the matrix by distance will cut out any interactions occurring 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 occurring 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 transforming 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 $20 \leq i \leq 40$ and column numbers $30 \leq j \leq 50$ 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% exclusion. To exclude any amount of overlap set to 0. If set to 1, only a 100% 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 <http://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
```

full2sparse	<i>Transform a full Hi-C contact matrix to a sparse upper triangular matrix</i>
-------------	---------------------------------------------------------------------------------

Description

Transform 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: <code>colnames(mat) = c(0, 100000, 200000, 300000, 400000, 500000)</code>
-----	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Value

A sparse upper triangular matrix.

Examples

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

get_CNV	<i>Function to get the locations of copy number variations</i>
---------	----------------------------------------------------------------

Description

Function to get the locations of copy number variations

Usage

```
get_CNV(path2bam, out.file = "CNV_calls", bin.size = 1000,
        genome = "hg19", CNV.level = 2)
```

Arguments

path2bam	The path to the folder containing the .bam files for the data
out.file	The prefix for the filename which will be exported.
bin.size	The bin.size (resolution) of the HiC data you will be using in kbp. e.g. for 1MB resolution enter 1000, for 500kb resolution enter 500.
genome	Character string for the genome. Defaults to 'hg19'.
CNV.level	The value require to call a region a CNV. Should be either 1, or 2. This setting will determine what level of CNV will be returned in the resulting BED file. 1 indicates any copy number variation will be returned. 2 indicates only deletions and amplifications will be returned. Defaults to 2.

Details

This function utilizes QDNAseq to detect CNV (copy number variations) in your Hi-C data. You must have the .bam files in order to use this function. Place the .bam files in a folder and set path2bam as the path to this folder. The function will then perform CNV detection and export a .txt file containing the CNV calls for each region of the genome. In addition the results will be returned from the function as a data.frame. It is recommended to exclude the regions with extreme CNV when using HiCcompare. Regions can be excluded using the create.hic.table() function. For more information about the methods used in this function see QDNAseq here: <https://bioconductor.org/packages/release/bioc/html/QDNAseq.html>

Value

A data.frame containing the regions for CNVs at the level specified with the CNV.level option. A data.frame containing the calls for all regions will also be written to the disk using the out.file prefix. The first 4 columns of the data.frame contain location information for the genomic regions. All following columns contain the CNV calls for each .bam file. CNV calls are defined as follows: -2 = deletion, -1 = loss, 0 = normal, 1 = gain, 2 = amplification. Additionally, a .bed file will be written to disk containing the same values as the returned data.frame.

Examples

```
## Not run:
cnv_calls <- get_CNV(path2bam = 'C:/bamfiles')

## End(Not run)
```

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	<i>BED file for hg38 blacklisted regions</i>
----------------	----------------------------------------------

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	<i>Convert HiC-Pro results to BEDPE format</i>
--------------	------------------------------------------------

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-chromosomal 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_diff	<i>Detect differences between two jointly normalized Hi-C datasets.</i>
----------	-------------------------------------------------------------------------

Description

Detect differences between two jointly normalized Hi-C datasets.

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 clinically 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 permutation 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

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(), check.differences = FALSE, diff.thresh = "auto",
  iterations = 10000)
```

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.
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. Defaults 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
check.differences	Logical, should difference detection be performed? If TRUE, the same procedure as hic_diff will be performed. If FALSE, only normalization will be performed on the entered data. Defaults to FALSE.
diff.thresh	Fold change threshold desired to call a detected difference clinically 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. Defaults to 'auto'.

`iterations` Number of iterations for the permutation test. Will only be used if `check.differences` set to TRUE. Defaults to 10000.

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 first transformed into what is termed an MD plot (similar to the MA plot/Bland-Altman plot). `M` is the log difference $\log_2(x/y)$ 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`, and `mc` for the loess correction factor. If `check.differences` is set to TRUE a column containing the p-values for the significance of the difference between the two datasets will also be returned.

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 hic_diff analysis on it

Description

Simulate a Hi-C matrix and perform `hic_diff` 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 distribution for the standard deviation 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 multiplier 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_diff 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)
```

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 <http://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 <http://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 <http://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)
```

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_diff(result, diff.thresh = 1)
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 <http://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 <http://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 <http://www.aidenlab.org/> Or the the GEO link to download the data <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525>

plot_A	<i>function to plot distribution of A to get input values for weight_M</i>
--------	----------------------------------------------------------------------------

Description

function to plot distribution of A to get input values for weight_M

Usage

```
plot_A(hic.table)
```

Arguments

`hic.table` A hic.table or list of hic.tables. You must run `hic_loess` on the hic.table before running `plot_A`.

Details

This function will plot 4 veivs of the distribution of A for use in determining the A.min and quant parameters for the `weight_M` function. The first plot is the densities of A for each chromosome cut off at the 75th quantile. The second plot is the densities of each chromosome cut at the 90th quantile. The third plot is the log2 histogram of the distribution of A. The fourth plot is the histogram of A cut at the 75th quantile. You should try to choose a value of A.min where the distribution starts to only descend after that point. You should choose the value for quant at the point where the density flattens out at the lowest values.

Value

Plots of A.

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()
hic.table <- hic_loess(hic.table, Plot = TRUE)
# plot_A
plot_A(hic.table)
```

remove_centromere	<i>Function to remove centromere columns and rows from a full Hi-C contact matrix</i>
-------------------	---------------------------------------------------------------------------------------

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 = 5)
```

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)
```

sparse2full	<i>Transform a sparse upper triangular matrix to a full Hi-C contact matrix</i>
-------------	---------------------------------------------------------------------------------

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	<i>Function to split hic.table into 2 subsets at the centromere</i>
------------------	---------------------------------------------------------------------

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	<i>Function to visualize p-values from HiCcompare results</i>
-----------------	---------------------------------------------------------------

Description

Function to visualize p-values from HiCcompare results

Usage

```
visualize_pvals(hic.table, alpha = NA)
```

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 \geq alpha will be set to 1 for the visualization in the heatmap. Defaults to NA for visualization of all p-values.

Details

The goal of this function is to visualize where in the Hi-C matrix the differences are occurring between two experimental conditions. The function will produce a heatmap of the $-\log_{10}(\text{p-values}) * \text{sign}(\text{adj.M})$ to visualize where the significant differences between the datasets are occurring 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_diff(result, diff.thresh = 'auto', Plot = TRUE)
# visualize p-values
visualize_pvals(diff.result)
```

weight_M	<i>Function to weight M values on the MD plot by Average expression</i>
----------	-------------------------------------------------------------------------

Description

Function to weight M values on the MD plot by Average expression

Usage

```
weight_M(hic.table, A.min = 1, quant = 0.75, Plot.diagnostic = TRUE,
         Plot.MD = TRUE)
```

Arguments

hic.table	A hic.table object or hic.table list. You must run hic_loess on the hic.table before performing M weighting.
A.min	The A value at which to start the powerlaw estimation of the density. This should be determined by looking at the plots of A from the plot_A function. See vignette for more details. Defaults to 1.
quant	The quantile value at which to stop weighting M values. Typical values should range from 0.6 to 0.9. This can be determined by looking at the plots of the distribution of A from the plot_A function. See vignette for more details. Defaults to 0.75.
Plot.diagnostic	Logical, should a histogram of the subset of A between A.min and A < quant be plotted with the powerlaw fit overlaid. It is recommend to check the fit of the powerlaw to the distribution. If a good fit is not achieved you may need to adjust A.min or quant parameters. Defaults to TRUE.
Plot.MD	Logical, should the MD plot of the weighted M values be plotted? Defaults to TRUE.

Details

Weighting of M is used as a way to account for the fact that IFs with low average values are less trustworthy compared to IFs with high average values. i.e. a comparison between IF1 = 1 and IF2 = 2 results in an M value of 1, but a comparison of IF1 = 100 and IF2 = 200 also results in an M value of 1. However the average expression of the first comparison is 1.5 while the second is 150. Naturally the first comparison is much more likely to have occurred as a fluke compared to a true biological difference between the conditions being tested. This function performs down-weighting of M values which arise from IFs exhibiting low average expression. For a more in-depth explanation with examples please see the vignette.

Value

A hic.table with weighted adj.M values.

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()
hic.table <- hic_loess(hic.table, Plot = TRUE)
# weight M-values
hic.table <- weight_M(hic.table, A.min = 75, quant = 0.9)
```

Index

*Topic **datasets**

- brain_table, 3
- centromere_locations, 3
- cooler, 4
- hg19_blacklist, 8
- hg38_blacklist, 9
- HMEC.chr10, 15
- HMEC.chr22, 16
- hmec.IS, 17
- NHEK.chr10, 21
- NHEK.chr22, 22
- nhek.IS, 22

brain_table, 3

centromere_locations, 3

cooler, 4

cooler2sparse, 4

create.hic.table, 5

full2sparse, 7

get_CNV, 7

hg19_blacklist, 8

hg38_blacklist, 9

hic_diff, 10

hic_loess, 12

hic_simulate, 13

HiCcompare (HiCcompare-package), 2

HiCcompare-package, 2

hicpro2bedpe, 10

HMEC.chr10, 15

HMEC.chr22, 16

hmec.IS, 17

KRnorm, 17

MA_norm, 19

make_InteractionSet, 18

MD.plot1, 19

MD.plot2, 20

NHEK.chr10, 21

NHEK.chr22, 22

nhek.IS, 22

plot_A, 23

remove_centromere, 24

SCN, 24

sim.other.methods, 25

sparse2full, 26

split_centromere, 26

total_sum, 27

visualize_pvals, 28

weight_M, 29