Package ‘HiTC’

March 28, 2017

Type     Package
Title    High Throughput Chromosome Conformation Capture analysis
Description The HiTC package was developed to explore high-throughput ‘C’ data such as 5C or Hi-C. Dedicated R classes as well as standard methods for quality controls, normalization, visualization, and further analysis are also provided.
Version  1.18.1
Date     2015-12-15
Depends  R (>= 2.15.0), methods, IRanges, GenomicRanges
Imports  Biostrings, graphics, grDevices, rtracklayer, RColorBrewer, Matrix, parallel, GenomeInfoDb
Suggests BiocStyle, HiCDataHumanIMR90
Author   Nicolas Servant
Maintainer Nicolas Servant <nicolas.servant@curie.fr>
License  Artistic-2.0
         normalize_5C.R normalize_hiC.R tads.R deprecated.R
LazyLoad yes
biocViews Sequencing, HighThroughputSequencing, HiC
InstallableEverywhere yes
NeedsCompilation no

R topics documented:

  binningC .................................................................  2
  CQC .................................................................  3
  directionalityIndex ..................................................  4
  discretize ............................................................  5
  export.my5C ..........................................................  6
  exportC ...............................................................  7
  extractRegion .........................................................  8
  getAnnotatedRestrictionSites .......................................  9
  getExpectedCounts .................................................. 10
  getPearsonMap ....................................................... 11
  getRestrictionFragmentsPerChromosome ............................ 12
**binningC**

Windowing of high-throughput 'C' contact matrix

**Description**

Binning of 'C' contact map

**Usage**

```r
binningC(x, binsize=100000, bin.adjust=TRUE, upa=TRUE,
    method=c("sum", "median","mean"), use.zero=TRUE, step=1, optimize.by = c("speed", "memory"))
```

**Arguments**

- `x` object that inherits from class HTCexp
- `binsize` size of the bin to consider for windowing
- `bin.adjust` logical; adjust the size of the bin to the size of the genomic region
- `upa` logical; unique primer assignment. Allow one primer to belong to one or several bins
- `method` the method used to combine the counts. Must be 'mean', 'median' or 'sum'
- `use.zero` logical; use the zero values in the method calculation
- `step` numeric; binning step size in n coverage i.e. window step
- `optimize.by` "speed" will use faster methods but more RAM, and "memory" will be slower, but require less RAM

**Details**

`bin.adjust` allows to work with bins of exactly the same size. Otherwise, the last bin is usually smaller than the others.

This function aims at changing the resolution of both 5C or Hi-C data. In case of 5C data (i.e. raw, not binned data), the contacts between all pairs of primers will be summarized per genomic bins (the median of all pairwise primers is usually used). In case of binned data (as Hi-C maps), the function can generate smaller resolution maps by aggregating bins. For instance, going from a 40kb resolution to a 1Mb resolution.
The method is used to combine the counts in a bin, must be ‘mean’, ‘median’ or ‘sum’. The step parameter allows to choose the overlap between the bins. A step of 2 means a 50% overlap between two bins, a step of 3 means a 60% overlap between two bins, etc.

Value

An HTCexp-class object with binned interaction data. In this case, the genomic intervals are converted into bins of fixed size. The contact matrix is symmetric.

Author(s)

N. Servant, B. Lajoie

See Also

HTCexp-class

Examples

data(Nora_5C)

## Data binning 100kb, with a 1/3 overlap
E14.bin <- binningC(E14$chrXchrX, binsize=100000, step=3)
show(E14.bin)

## Move to a lower resolution map
E14.bin2 <- binningC(E14.bin, binsize=500000, step=1)
show(E14.bin2)

CQC

Quality Control for high-throughput ‘C’ experiment

Description

Quality Control for high-throughput ‘C’ experiment

Usage

CQC(x, cis.trans.ratio = TRUE, hist.interac=TRUE, scat.interac.dist=TRUE, hist.dist=TRUE, trim.range=0.98, winsize=NA, dev.new=TRUE)

Arguments

x object that inherits from class HTCexp or HTClist
cis.trans.ratio logical; barplot of percentage of inter-intrachromosomal interactions
hist.interac logical; histogram of the interaction frequency
scat.interac.dist logical; scatter plot of interaction count versus the genomic distance between two elements
hist.dist logical; histogram of the distance between the ‘x’ and ‘y’ intervals
trim.range remove the extreme values by trimming the counts. Only use for plotting functions. [0,1]
directionalityIndex

winsize used for the scat.interac.dist. If specify, the data are windowed before plotting

dev.new if true, each graphs is plotted in a new device

Details

If x is a HTClst object, all HTCexp objects are merged. The zero values are not used to compute
the descriptive statistics and to display the data. If trim.range are lower than 1. The highest values
(quantile probability is equal to trim.range) are discarded.

Value

Create quality plots and return a matrix with some simple statistics on all, cis and trans data.

Author(s)

N. Servant, B. Lajoie

See Also

HTCexp-class

Examples

data(Nora_5C)

## Quality Control
CQC(E14)

---

directionalityIndex Directionality index calculation

Description

Calculate the directionality index as proposed by Dixon et al. 2012

Usage

directionalityIndex(x, winup = 2e+06, windown = 2e+06)

Arguments

x HTClist object
winup size of upstream window
windown size of downstream window

Details

Calculate the directionality index as proposed by Dixon et al. This index is then used to call topo-
logical domains in Hi-C/5C data.

Value

A numeric vector
discretize

Author(s)
N. Servant

See Also
HTClist-class

Examples
require(HiCDataHumanIMR90)
data(Dixon2012_IMR90)
hox <- extractRegion(hic_imr90_40$chr6chr6, chr="chr6", from=50e6, to=58e6)
di<-directionalityIndex(hox)

discretize

Transform matrix of counts data into discrete matrix

Description
Transform matrix of counts data into discrete matrix

Usage
discretize(x, nb.lev=4, quant=TRUE)

Arguments
x data matrix
nb.lev number of discretization level
quant logical; use quantile distribution or split data into equals `nb.lev` levels

Value
A discrete matrix

Author(s)
N. Servant

See Also
quantile
## Examples

```r
## Not run:
data(Nora_5C)

## Data binning
E14bin<-binningC(E14$chrXchrX)

## Discretize matrix
dismat<-discretize(intdata(E14bin))
mapC(dismat)

## End(Not run)
```

---

**export.my5C**  
*Export HTCexp object to my5C website format*

### Description

Export HTCexp object to my5C website format.

### Usage

```r
export.my5C(x, file, genome="mm9", per.chromosome=FALSE)
```

### Arguments

- `x` object that inherits from class HTCexp
- `file` character; the prefix of the output file
- `genome` The genome version. This information is only used for the `mat` export format. See details
- `per.chromosome` logical; export each contact maps in a different file (i.e one per chromosome pair)

### Details

A tab-delimited matrix file is generated with the row and colnames defined as follow as in the my5C web tool:

```
REV_2|mm9|chrX:98831149-98834145
```

### Author(s)

N. Servant

### See Also

- `exportC`
Examples

```r
## Not run:
data(Nora_5C)

## Data binning
E14.bin<-binningC(E14$chrXchrX)

## Export the new intervals definition
export.my5C(E14.bin, file="E14my5C"

## End(Not run)
```

---

**exportC**

*Export HTCexp object*

**Description**

Export HTCexp object to tab format

**Usage**

```
exportC(x, file, per.chromosome=FALSE)
```

**Arguments**

- `x`: object that inherits from class HTCexp
- `file`: character; the basename of the output file
- `per.chromosome`: logical; export each contact maps in a different file (i.e one per chromosome pair)

**Value**

Three output files will be created; 2 BED files for each genomic intervals, and one tab file. The standard format for 5C/Hi-C data is the following:

**One list file (tab delimited) bin1 bin2 x12 bin1 bin3 x13 ...**

**The BED file(s) describing the intervals ('xgi.bed' and 'ygi.bed' are usually the same for Hi-C but can be different for 5C data)**

```
chr1 1 1000000 bin1 chr1 1000001 2000000 bin2 ...
```

Note that this format is particularly interesting for sparse data as only non null values are stored. If per.chromosome=FALSE, the data will be exported in one genome scaled file.

**Author(s)**

N. Servant

**See Also**

- `export.my5C`, `importC`
extractRegion

Examples

```r
## Not run:
data(Nora_5C)

## Data binning
E14.bin<-binningC(E14$chrXchrX)

## Export the new intervals definition
exportC(E14.bin, file="E14")

## End(Not run)
```

---

**extractRegion**

**Extract a subset of the HTCexp object**

**Description**

Extract a subset of the HTCexp object based on genomic ranges

**Usage**

```r
extractRegion(x, MARGIN, chr, from, to, exact=FALSE)
```

**Arguments**

- `x` object that inherits from class HTCexp
- `MARGIN` a vector giving the subscripts which the function will be applied over as in `apply` function. E.g., `1` for the `x` intervals, and `2` for the `y` intervals. `c(1, 2)` indicates `x` and `y` intervals.
- `chr` character; the chromosome of the genomic region
- `from` numeric; start of the genomic region
- `to` numeric; end of the genomic region
- `exact` logical; exact genomic region

**Details**

By default, only the intervals fully included in the genomic ranges are returned. If exact is true, the overlapping intervals are also used, and forced to start/end at the specified position. If no intervals are overlapping, an interval with NA values is added.

**Value**

A HTCexp object

**Author(s)**

N. Servant

**See Also**

[GRanges-class](#)
getAnnotatedRestrictionSites

Examples

```r
data(Nora_5C)

## Focus on the genomic region chrX:98000000-100000000
E14sub<-extractRegion(E14$chrX, c(1,2), chr="chrX", from=98000000, to=100000000)
show(E14sub)
```

getAnnotatedRestrictionSites

**Annotation of restriction sites**

Description

Performs the annotation of all restriction sites of a given genome (i.e. GC content, mappability, effective fragment length)

Usage

```r
getAnnotatedRestrictionSites(resSite="AAGCTT", overhangs5=1, chromosomes=NULL, genomePack="BSgenome.Musculus.UCSC.mm9", mappability=NULL, wingc=200, winmap=500)
```

Arguments

- `resSite`: the sequence of the restriction site to look for. Default is HindIII restriction site
- `overhangs5`: 5’ overhangs on the DNA resulted from the cutting
- `chromosomes`: vector of chromosome number to focus on. Default all the chromosomes for the specified genome.
- `genomePack`: name of the genome package to access the DNA sequence
- `wingc`: size of the window upstream and downstream the restriction site used to calculate the GC content
- `mappability`: a GRanges with a ’score’ describing the mappability of the genome
- `winmap`: size of the window upstream and downstream the restriction site used to calculate the mappability

Details

This function automatically annotate all the restriction sites of a given chromosome. The mappability is optional but strongly advice for Hi-C contact map normalization. This information can be easily download from public ressources like ftp://hgdownload.cse.ucsc.edu/gbdb/mm9/bbi/.

Value

Returns a GRanges object annotation data upstream (U) and downstream (D) the restriction sites.

Author(s)

N. Servant

See Also

normLGF, setGenomicFeatures
Examples

```r
## Not run:
## Mappability data From http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeMapability/
map_hg18<- import("wgEncodeCrgMapabilityAlign100mer.bw",format="BigWig")

## 1- Example of restriction sites annotation
cutSites <- getAnnotatedRestrictionSites(resSite="AAGCTT", overhangs5=1, chromosomes="chr1", genomePack="BSg

## End(Not run)
```

getExpectedCounts

Estimate expected interaction counts of a High-Throughput C intra-chromosomal map based on the genomic distance between two loci

Description

The expected interaction is defined as the linear relationship between the interaction counts and the distance between two loci. See details for additional informations.

Usage

```r
getExpectedCounts(x, method=c("mean","loess"), asList=FALSE, ...)
```

Arguments

- `x`: object that inherits from class HTCexp
- `method`: method used to estimate the expected counts based on the genomic distance. See details
- `asList`: return the results as a list. Otherwise, return an object of class HTCexp
- `...`: arguments for mean or loess method, see below

Details

The expected value is the interaction frequency between two loci that one would expect based on a sole dependency on the genomic proximity of these fragments in the linear genome. This can be estimated using two different methods, mean or loess.

The first method (default) is simply based on the mean counts of each diagonal. If logbin is false, the expected counts will be estimated for each bin of the contact maps. If logbin is true, the binsize will change according to the distance to the diagonal. Short (resp. long) distance will be estimated with smaller (larger) bins. This method works for all resolutions.

The second method is based on a Lowess regression model and works usually fine with low resolution data (250Kb to 1Mb). At higher resolution, the lowess regression might be difficult to fit. The lowess smoothing has two parameters: span and bin. The span corresponds to the fraction of the data used for smoothing. Instead of computing the local polynomial fitting at each data point, a window of size delta (bin parameter) is applied on the data and a linear interpolation is used to fill in the fitted values within the window. The default is 1% of the range of x. If delta=0 all but identical x values are estimated independently. The variance is then estimated using the same span and bin parameter, at each interpolation points. The points inside a window are weighted so that nearby points get the most weight (tricube weight function).
**Value**

A list with the expected interaction map and the estimated variance

**For ’mean’ method**

- **logbin** if true logarithm based bins are used. In practice, it means that the bin size will change as we move away from the diagonal
- **step** multiplicative factor between each bin. Use if logbin is true
- **filter.low** fraction of low counts to filter before normalizing. default: 0.05.

**For ’loess’ method**

- **span** fraction of the data used for smoothing at each x point.
- **bin** interpolation parameter
- **stdev** logical, calculate the variance
- **plot** logical, display lowess smoothing and variance estimation points

**Author(s)**

N. Servant, B. Lajoie

**See Also**

HTCexp-class, normPerExpected, normPerExpected, lowess

**Examples**

```r
data(Nora_5C)

## Estimate expected interaction from distance between intervals
E14.exp<-getExpectedCounts(E14$chrXchrX, method="loess", stdev=TRUE, plot=FALSE)
mapC(E14.exp)
```

---

**getPearsonMap**

*Pearson correlation map*

**Description**

Generate pearson correlation map, usually used to call chromosomal compartments

**Usage**

```r
getPearsonMap(x, normPerExpected=TRUE, center=TRUE, ...)
```

**Arguments**

- **x** object that inherits from class HTCexp
- **normPerExpected** normalized by expected interaction using the loess calculation of distance dependency. see normPerExpected
- **center** default=true. center the observed/expected map before calculating the Pearson correlation
- **...** additional parameters passed to normPerExpected
getRestrictionFragmentsPerChromosome

Description

The function returns an HTCexp object with Pearson correlation map. This is usually the first step of the Principal Component Analysis (see pca.hic). Centering the rows of the observed/expected matrix allows to avoid bias due to ranges of interaction counts. If true, the correlation of small values should be as valuable as correlation of large values.

Value

A HTCexp object

Author(s)

N. Servant, B. Lajoie, R. McCord

See Also

normPerExpected

Examples

```r
## Get Lieberman-Aiden Hi-C data
exDir <- system.file("extdata", package="HiTC")
l <- sapply(list.files(exDir, pattern=paste("HIC_gm06690_"), full.names=TRUE), import.my5C)
hiC <- HTClist(l)

## get Pearson correlation map
pm <- getPearsonMap(hiC$chr14chr14)
mapC(HTClist(pm), maxrange=1, col.pos=c("black","red"), col.neg=c("black","blue"))
```

getRestrictionFragmentsPerChromosome

Get a list of DNA restriction fragments

Description

Performs the detection of restriction sites on a given genome and convert this information as a list of restriction fragments.

Usage

g RestrictionFragmentsPerChromosome(resSite="AAGCTT", overhangs5=1, chromosomes=NULL, genomePack="BSgenome.Mmusculus.UCSC.mm9")

Arguments

- resSite: the sequence of the restriction site to look for. Default is HindIII restriction site.
- overhangs5: 5' overhangs on the DNA resulted from the cutting.
- chromosomes: vector of chromosome number to focus on. Default all the chromosomes for the specified genome.
- genomePack: name of the genome package to access the DNA sequence.
Value

Returns a GRanges object with all restriction fragments for a given genome/chromosome.

Author(s)

N. Servant

See Also

normLGF, setGenomicFeatures, getAnnotatedRestrictionSites

Examples

```r
## Not run:
## Mappability data From http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeMapability/
## map_hg18<- import("wgEncodeCrgMapabilityAlign100mer.bw",format="BigWig")
## 1- Get the list of restriction fragments for Human hg18 after HindIII digestion
resFrag <- getRestrictionFragmentsPerChromosome(resSite="AAGCTT", overhangs5=1, chromosomes="chrX", genomePack="BSgenome.Hsapiens.UCSC.hg18")
## End(Not run)
```

---

**HTCexp-class**

Class `HTCexp`

**Description**

A class for representing high throughput Chromosome Conformation Capture data from next-generation sequencing experiments.

**Details**

The `normPerExpected` method estimates the expected interactions based on the dependency on the genomic proximity between two loci. See `getExpectedCounts` function for details.

The `normPerTrans` method is based on the assumption that all trans contacts should be the same. Thus, the cis contacts can be normalized by the interaction level of trans data. The `xtrans` trans map has to share its `xgi` ranges with the cis map, and the `ytrans` has to share its `ygi` ranges with the cismap. The method is used to combine the normalization factor from x and y ranges. Must be `sum`, `mult` or `mean`.

**Objects from the Class**

Objects can be created either by:

1. calls of the form `new("HTCexp", intdata, GRanges, GRanges)`.  
2. using the auxiliary function `HTCexp` and supplying contact Matrix with x and y intervals definition. The forceSymmetric option can used to force intra-chromosomal contact maps to be stored as symmetrical matrix.
**Slots**

- `intdata`: Dense or Sparse Matrix, holding the interaction level between each pairs of 'x-y' intervals. The 'y' intervals must be in rows, and the 'x' in columns.
- `ygi`: Genomic ranges of y intervals; see class `granges` for details
- `xgi`: Genomic ranges of x intervals; see class `granges` for details

**Methods**

c(x, ...) Combines 'x' and the signature("HTCexp") objects in `...' together. The results is an object of class signature("HTCList")
detail(x) signature("HTCexp"): a more detailed output of the experiment than provided by `show`
divide(x) comparison of two signature("HTCexp") objects. Perform the division of the two contact matrices on the common 'x' and 'y' intervals. The operation is done only on the common intervals of both objects. If one of the two objects has a count to zero, the divided value will be NA
intdata(x) return the `intdata` Matrix counts. Note that triangular matrices are always returned as symmetric matrices.
export(x) Defunct. See exportC method
isBinned return TRUE if the data are binned. The method tests if the 'x' and 'y' genome intervals are the same, if 90% of the bins have the same size and if the full genomic range is covered
forceSymmetric(x) force the interaction data to 'symmetricMatrix'
forceTriangular(x) force the interaction data to triangular, i.e. symmetric. Lower triangle of the matrix is set to zero, therefore reducing the size of the data in memory
isIntraChrom(x) return TRUE if the current signature("HTCexp") object contains intrachromosomal contact data
isSymmetric(x) return TRUE if the contact map is symmetrical, i.e inherits the `symmetricMatrix` class
normPerReads(x) normalize the contact matrix by the total number of reads of the matrix.
normPerExpected(x, ...) normalize the contact matrix by the expected number of reads based on the distance between two loci. See details.
normPerZscore(x) Defunct. See normPerExpected method
normPerTrans(x, xtrans, ytrans, method="sum") Normalize cis contact map based on the trans interactions. See details
plot(x) visualization method; Display an heatmap of the contact data. Refer to the documentation of `mapC` for more details of the plotting function
range(x) return the genomic range of the signature("HTCexp") object
seq_name(x) Defunct. See seqlevels method
seqlevels(x) return the sequence levels of the signature("HTCexp") object
show(x) summarized output of the experiment, with informations about the data dimension and the genomic region studied
subtract(x) comparison of two signature("HTCexp") objects. Perform the substraction of the two contact matrices on the common 'x' and 'y' intervals. The operation is done only on the common intervals of both objects. If one of the two objects has a count to zero, the divided value will be NA
summary(x) return descriptive summary statistics about the contact map
x_intervals(x) return the xgi GRanges object defining the x intervals
y_intervals(x) return the ygi GRanges object defining the y intervals
xy_intervals(x) return both xgi and ygi objects as a GRangesList object
HTCexp-class

Author(s)
Nicolas Servant

See Also
GRanges-class, GRangesList-class, Matrix-class

Examples
data(Nora_5C)

## HTCexp description
show(E14)
detail(E14)

## Is binned data?
isBinned(E14$chrXchrX)

## Is a inter or intrachromosomal experiment?
isIntraChrom(E14$chrXchrX)

## Bin the data
E14.bin <- binningC(E14$chrXchrX, binsize=100000, step=3)

## Divide by expected interaction counts
E14norm<-normPerExpected(E14.bin)

## Operation on HTCexp object
E14_d_MEF<-divide(normPerReads(E14$chrXchrX), normPerReads(MEF$chrXchrX))
E14_s_MEF<-substract(normPerReads(E14$chrXchrX), normPerReads(MEF$chrXchrX))

## Overlap with genomic annotation
require(rtracklayer)
gene <- import(file.path(system.file("extdata", package="HiTC"), "refseq_mm9_chrX_98831149_103425150.bed"), format="bed")
plot(E14$chrXchrX, tracks=list(RefSeqGene=gene))

## Not run:
## normPerTrans data normalization applied on \href{http://genome.ucsc.edu/cgi-bin/hgFileUi?db=hg19&g=wgEncodeUmassDekker5C}{ENCODE data}.
ENCODE=import.my5C("./ENM-GM12878-R1.matrix")

## Look at raw contact map
mapC(ENCODE$chr7chr7)

## look at normalize by trans contact map
mapC(normPerTrans(ENCODE$chr7chr7, xtrans=ENCODE$chr7chr5, ytrans=ENCODE$chr5chr7))

## End(Not run)

## Not run:
## Export
exportC(E14$chrXchrX, con="E14.csv")

## End(Not run)
Class 'HTClist'

Description

A class for representing a list of high throughput Chromosome Conformation Capture data from next-generation sequencing experiments.

Details

A signature("HTClist") is composed of a list of contact maps, representing the chromosomal interactions between pair of chromosomes.

The expected number of maps for a complete signature("HTClist") object should be equal to \(lchrs+(lchrs*(lchrs-1)/2)\). In this case, the chr1-chr2 map is stored once, but the dataset is complete.

If a signature("HTClist") object is composed of all pairwise interaction maps, it means that the chr1-chr2 AND the chr2-chr1 maps will be stored. This way of storing the data is less memory efficient but can ease the use of some genome-wide algorithm.

Note that the getCombinedContact method should be used carefully. This method merges all single contact maps in one genome-wide map, therefore creating a very big matrix requiring memory space. However, this method is useful for many genome-wide analysis. The normPerExpected method applied to HTClist object is only available with the 'mean' method (see getExpectedCounts for details). In this case, the mean counts per distance are calculated over all intra-chromosomal maps.

Constructor

The HTClist represents a list of HTCexp objects and can be created as follow:

HTClist(...) : Creates a HTClist object using HTCexp objects supplied in '...'

Methods

c(x, ...) Combines a signature("HTClist") object 'x' with signature("HTClist") or signature("HTCexp") objects in '...'. The results is an object of class signature("HTCList")
detail(x) signature("HTClist") : a more detailed output of the experiment than provided by show.
forcePairwise(x) return a signature("HTClist") with all the pairwise contact maps
forceSymmetric(x) return a signature("HTClist") with half of the pairwise contact maps
isComplete(x) Logical; true if 'x' contains all intra and interchromosomal maps
isPairwise(x) Logical; true if 'x' contains all interchromosomal pairs, i.e. chr1chr2 and chr2chr1
isBinned(x) applies 'isBinned' to each element in 'x'
isIntraChrom(x) applies 'isIntraChrom' to each element in 'x'
getCombinedContacts(x) merge all contact maps in a single big matrix
getCombinedIntervals(x, merge=FALSE) merge all x and y intervals in single GRangesList object, or in a single GRanges object if merge=TRUE
normPerExpected(x) normalized by genomic distance all intra-chromosomal maps. See details
ranges(x) applies 'range' to each element in 'x'
range(x) return the reduce range of all elements in 'x'

reduce(x, chr, cis=TRUE, trans=TRUE, extend=FALSE) reduce a HTClist object to the list of provided chromosomes. Intra/Interchromosomal maps are returned according to the cis and trans args. If extend = True, all maps involving one of the chromosomes are returned

seqlevels(x) return the sequence levels of all elements in 'x'

as.list(x) coercion to simple list object

names(x) get the names of the elements

show(x) summarized output of the experiment, with informations about the data dimension

summary(x) return descriptive summary statistics for each interaction map

x[i ] Get elements i from x. Can be the positional index or its name.

Author(s)

Nicolas Servant

See Also

GRangesList-class, HTCexp-class

Examples

exDir <- system.file("extdata", package="HiTC")
l <- sapply(list.files(exDir, pattern=paste("HIC_gm06690_"), full.names=TRUE),
    import.my5C)
hiC <- HTClist(l)
names(hiC)

## Methods
ranges(hiC)
range(hiC)
isComplete(hiC)
isPairwise(hiC)
isBinned(hiC)
isIntraChrom(hiC)
seqlevels(hiC)
**importC**

**Description**
Import high-throughput 'C' data

**Usage**
importC(con, xgi.bed, ygi.bed=NULL, allPairwise=FALSE, rm.trans=FALSE, lazyload=FALSE)

**Arguments**
- **file**: input file from the my5C webtool
- **allPairwise**: logical; generate all pairwise chromosomal contact maps, i.e chr1-chr2, chr2-chr1
- **rm.trans**: if true, only intra-chromosomal maps are loaded
- **lazyload**: logical; force the intra-chromosomal contact maps to be stored as triangular matrix

**Details**
This function allows data import from the my5C webtool. The matrix format is a tab-delimited format, corresponding to the contact map. The rownames and columnames are splitted to created the genome intervals (example: REV_2|mm9|chrX:98831149-98834145).

The allPairwise option is not necessary in case of symetric design. Otherwise, it will return all the pairwise contact maps.

The matrix will be stored as a matrix inheriting from Matrix class. If forcesymmetrical=TRUE, the intrachromosomal matrix as coerced to symmetricMatrix class allowing a much more efficient memory usage.

**Value**
A HTClist object

**Author(s)**
N. Servant

**See Also**
import, codeHTClist-class, codeMatrix-class, symmetricMatrix-class

**Examples**
```r
exDir <- system.file("extdata", package="HiTC")
## Load my5C matrix format
hiC<-import.my5C(file.path(exDir, "HIC_gm06690_chr14_chr14_1000000_obs.txt"))
detail(hiC)
```
## importC

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>con</td>
<td>input csv file. See details</td>
</tr>
<tr>
<td>xgi.bed</td>
<td>BED file describing the 'x' intervals (i.e. column names) of the contact map</td>
</tr>
<tr>
<td>ygi.bed</td>
<td>BED file describing the 'y' intervals (i.e. row names) of the contact map</td>
</tr>
<tr>
<td>allPairwise</td>
<td>logical; generate all pairwise chromosomal contact maps, i.e chr1-chr2, chr2-chr1</td>
</tr>
<tr>
<td>rm.trans</td>
<td>if true, returns only intra-chromosomal maps</td>
</tr>
<tr>
<td>lazyload</td>
<td>logical; see details</td>
</tr>
</tbody>
</table>

**Details**

This function import high-throughput data from a tab list file. The standard format for 5C/Hi-C data is the following:

** One list file (tab delimited) bin1 bin2 x12 bin1 bin3 x13 ... ** The BED file(s) describing the intervals ('xgi.bed' and 'ygi.bed' are usually the same for Hi-C but can be different for 5C data) chr1 1 1000000 bin1 chr1 1000001 2000000 bin2 ...

Note that this format is particularly interesting for sparse data as only non null values are stored.

The lazyload option allow to reduce the memory size of imported object. Therefore, only half of inter-chromosomal maps are stored. And intra-chromosomal maps are stored as sparse triangular matrix. Note that even if the contact maps are stored as triangular matrix, the indata method always returns a symmetrical map.

**Value**

A HTClist object

**Author(s)**

N. Servant

**See Also**

exportC,import.my5C,HTCexp-class

**Examples**

```r
## Not run:
data(Nora_5C)

## Data binning
e14.bin<-binningC(E14$chrXchrX)

## Export the new intervals definition
exportC(E14.bin, file="E14")

##Import
importC(con="E14.count", xgi="E14_xgi.bed", ygi="E14_ygi.bed")

## End(Not run)
```
Description

Compute the distance of intrachromosomal contacts of a 'C' experiment.

Usage

```r
intervalsDist(x, use.zero=TRUE)
```

Arguments

- `x`: object that inherits from class HTCexp.
- `use.zero`: if FALSE, the distance for non interacting regions (zero counts) are not reported.

Details

If $A$ and $B$ are the two sets of intervals and $s$ and $e$, the start and end of an interval, the distance is calculated as:

$$\min(|A_e - B_s|, |A_s - B_e|)$$

Only intrachromosomal contact maps can be use for this operation.

Value

A matrix of distances between genomic intervals.

Author(s)

N. Servant

See Also

HTCexp-class

Examples

```r
data(Nora_5C)
## Calculate distances between primers/intervals
d<-intervalsDist(E14$chrXchrX)
```
Description

Visualize 'C' contact map

Details

This function implements the plot method for objects of class HTCexp and HTClist.
By default, the trim.range value is fixed so that the 98th percentile (resp. 2th percentile) of each interaction matrix is discarded. It therefore allow to remove the extreme values from the matrix, but each map is plotted independently. If the maxrange argument is set, data higher that this threshold will be fixed to the maxrange value for all maps. In addition, color ranges are ajusted in a way that all maps are plotted within the same color range allowing visual maps comparison.

The HTCexp and HTClist are not represented in the same way. The heatmap view is used to display the HTClist objects in two dimension. This view is mainly useful to have an overview of the data, as Hi-C data. The triangle view is used for HTCexp only and represent the top-right part the interaction matrix. If two HTCexp objects are specified, they will be displayed in order to compare both contact maps. The two maps have to be binned to ensure comparison between genomic ranges.

Annotation tracks can be added to both views. In case of binned data, the exact genomic positions of each features are takken into account. Otherwise, the 'C' intervals which overlap with the annotation features are colored.

Value

Returns NULL; this function is called for the side-effect of creating the plot.

For HTCexp and HTClist objects

x object that inherits from class HTCexp or HTClist
tracks List of GRanges objects of data to display as annotation track(s)
minrange the minimum range of values used to define the color palette
maxrange the maximum range of values used to define the color palette
trim.range define the maxrange and minrange values using the percentile of the interaction matrix
show.zero logical; plot the zero values
show.na logical; show the NA values in gray
log.data logical; do you want to log the data before plotting the heatmap
col.pos color for (low,mid,high) positive contact counts. Must be a vectore of size 3. mid can be NA
col.neg color for (low,mid,high) negative contact counts. Must be a vectore of size 3. mid can be NA
col.na color for NA values
grid logical; add a grid on the heatmap
title character; add a title to the HTCexp plot(s)
value logical; display the contact values on the matrix. Useful for small matrices
For HTCexp objects only

- `y` optional. object that inherits from class HTCexp.

For HTClist objects only

- `names` logical; display the names of the intervals. Useful for small matrices

Author(s)

N. Servant, B. Lajoie

See Also

HTCexp-class, HTClist-class

Examples

data(Nora_5C)

```r
## Contact map
## HTClist view
mapC(E14)

## HTCexp view
mapC(E14$chrXchrX)

## Play with contrast and color
mapC(E14$chrXchrX, maxrange=100, col.pos=c("black","red","yellow"))

## Add annotation and change view
require(rtracklayer)
exDir <- system.file("extdata", package="HiTC")
gene <- import(file.path(exDir,"refseq_mm9_chrX_98831149_103425150.bed"), format="bed")
mapC(E14$chrXchrX, tracks=list(Refseq=gene))

## Compare two samples
mapC(binningC(E14$chrXchrX), binningC(MEF$chrXchrX), tracks=list(Refseq=gene))
```

---

**Description**

5C data described by Nora et al. (2012)

**Usage**

data(Nora_5C)

**Format**

Contains two HTClist objects (E14 and MEF). Each of them containing the ChrX intrachromosomal maps as a HTCexp object.
Details

This 5C dataset published by Nora et al (GSE35721), contains two different samples, a male undifferentiated ES cells (E14, GSM873935) and a mouse embryonic fibroblasts (MEF, GSM873924). This dataset is mainly used to describe the available functionalities of the HiTC package. The data provided with the package are count data.

Source


References


Examples

```r
data(Nora_5C)
show(E14)
show(MEF)
```

**normICE**

Iterative Correction of Hi-C data (ICE)

Description

Iterative correction leverages the unique pairwise genome-wide structure of Hi-C data to decompose the data into a set of biases and a map of relative contact probabilities between any two genomic loci, achieving equal visibility across all genomic regions.

Usage

```r
normICE(x, max_iter=50, eps=1e-4, sparse.filter=0.02)
```

Arguments

- `x`: object that inherits from class HTCexp
- `max_iter`: maximum number of iteration
- `eps`: the relative increment in the results before declaring convergence
- `sparse.filter`: Define which percentage of bins with high sparsity should be force to zero

Details

The normalization of Hi-C data is based on matrix balancing algorithm which consists of iteratively estimating the matrix bias using the l1 norm. The method implemented here is the Sinkhorn-Knopp algorithm as used in the Imakaev et al. paper. Note that the original method is applied on the genome-wide Hi-C map, but that the method could be applied on intra-chromosomal maps at high resolution.

Value

Returns a HTCexp object with a corrected contact map.
normLGF

Local Genomic Feature (LGF) normalization

Description

Parametric model to remove systematic biases in the raw contact maps

Usage

```r
normLGF(x, family=c("poisson", "nb"))
```

Arguments

- `x`: object that inherits from class HTCexp
- `family`: parametric model to fit (poisson or nb)

Details

This function implements the HiCNorm method proposed by Hu et al. Briefly, the method uses a generalized linear model to correct the systematic biases (effective fragment length, GC content, mappability) in a Hi-C contact map.

Value

Returns a HTCexp object with a normalized contact map.

Examples

```r
## Not run:
## Lieberman data
exDir <- system.file("extdata", package="HiTC")
l <- sapply(list.files(exDir, pattern=paste("HIC_gm06690_"), full.names=TRUE),
import.my5C)
hiC <- HTClist(l)
hiC_iced <- HTClist(lapply(hiC, normICE))
## End(Not run)
```
Perform Principle Component Analysis on Hi-C contact map

Perform Principle Component Analysis on Hi-C contact map

pca.hic(x, normPerExpected=TRUE, npc=2, asGRangesList=TRUE, gene.gr=NULL, ...)

Author(s)
N. Servant, M. Hu, S. Selvaraj

References

See Also
getAnnotatedRestrictionSites, setGenomicFeatures

Examples
## Not run:
require(HiTC)
require(BSgenome.Hsapiens.UCSC.hg18)

## Lieberman data
exDir <- system.file("extdata", package="HiTC")
l <- sapply(list.files(exDir, pattern=paste("HIC_gm06690_"), full.names=TRUE),
import.my5C)
hiC <- HTClist(l)
hiC <- hiC[isIntraChrom(hiC)]
names(hiC)

## Mappability data From http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeMapability/
map_hg18<- import("wgEncodeCrgMapabilityAlign100mer.bw", format="BigWig")

## Get the genomic feature of the chromosome 12
hiC_annot <- HTClist(lapply(hiC, setGenomicFeatures, resSite="AAGCTT", overhangs5=1, genomePack="BSgenome.Hs	numchr12chr12

## Normalize the data
hiCnorm <- HTClist(lapply(hiC_annot, normLGF))

## End(Not run)
removeIntervals

Arguments

- `x`: object that inherits from class HTCexp
- `normPerExpected`: normalized by expected interaction using the loess calculation of distance dependency. see normPerExpected
- `npc`: numeric; number of first principal component to return
- `asGRangesList`: if TRUE a GRangesList object is returned where the scores represent the eigenvector
- `gene.gr`: object of class GenomicRanges describing the genes position. If used, the A/B compartments classes are defined based on gene density
- `...`: additional parameters passed to normPerExpected function

Details

This method was apply by Lieberman-Aiden et al. 2009 to correlate the annotation profiles (genes, ChIP-seq, etc.) with the topological domains observed in Hi-C (see Fig3G of Lieberman-Aiden et al. 2009)

Value

A list with the eigen vector(s) of the npc first principal component(s), and their importance

Author(s)

N. Servant, B. Lajoie, R. McCord

See Also

normPerExpected

Examples

```r
## Get Lieberman-Aiden Hi-C data
exDir <- system.file("extdata", package="HiTC")
x <- sapply(list.files(exDir, pattern=paste("HIC_gm06690_"), full.names=TRUE),
import.my5C)
hiC <- HTClist(x)
## Performed PCA
pr <- pca.hic(hiC$chr14chr14, npc=1, asGRangesList=TRUE)
```

Description

Remove intervals from HTCexp object
Usage

removeIntervals(x, ids)

Arguments

x
  object that inherits from class HTCexp
ids
  character; vector of primers Ids to remove from the object

Value

A HTCexp object without the discarded intervals

Author(s)

N. Servant

See Also

GRanges-class

Examples

data(Nora_5C)

## Remove intervals from a HTCexp object
removeIntervals(E14$chrXchrX, ids=c("5C_938_XIC-3_REV_2", "5C_938_XIC-3_REV_4"))

---

setGenomicFeatures  Annotation of Hi-C contact map

Description

Annotate a Hi-C contact map with the genomic local features (i.e. GC content, mappability, effective fragment length)

Usage

setGenomicFeatures(x, cutSites, minFragMap=.5, effFragLen=1000)

Arguments

x
  HTCexp object to annotate
cutSites
  GRangesList or GRanges object with restriction sites annotation obtained using the getAnnotatedRestrictionSites function
minFragMap
  Minimum Fragment Mappability. All fragments with a lower mappability are not used for the annotation.
effFragLen
  Effective Fragment Length. Size of specific fragment ligation
setGenomicFeatures

Details

The function requires the restriction sites annotation as provided by the getAnnotatedRestrictionSites function. The restriction sites are first filtered according to their mappability. This threshold has to be defined according to the data pre-processing. All remaining restriction sites are then intersected with the genomic bins of the contact map. All restriction sites included within a bin are averaged. The effective fragment length is defined as the size of specific ligation product. (See Yaffe and Tanay, 2011). In this paper, the authors define specific ligation as sum of distance to cutter sites \((d_1 + d_2) \leq 500\) bp. Such criterion implies that \(d_1 \leq 500\) bp and \(d_2 \leq 500\) bp. So for each fragment end, only reads mapped within 500 bp to cutter sites are used for downstream analysis. All defaults parameters correspond to the ones used in the HiCNorm method.

Value

Returns a HTCFexp object with local genomic features annotations.

Author(s)

N. Servant

See Also

normLGF, setGenomicFeatures

Examples

```r
## Not run:
require(BSgenome.Hsapiens.UCSC.hg18)
require(rtracklayer)

## Lieberman data
exDir <- system.file("extdata", package="HiTC")
l <- sapply(list.files(exDir, pattern=paste("HIC_gm06690_"), full.names=TRUE),
           import.my5C)
hiC <- HTClist(l)
hiC <- hiC[isIntraChrom(hiC)]
names(hiC)

## Get the genomic feature of the HiC chr12 data
cutSites <- getAnnotatedRestrictionSites(resSite="AAGCTT", overhangs5=1, chromosomes=seqlevels(hiC), genomePack="BSgenome.Hsapiens.UCSC.hg18")
chr12_annot <- setGenomicFeatures(hiC$chr12, cutSites)

## End(Not run)
```
setIntervalScale

Set x and y interval of the HTCexp object

Description
Set x and y interval of the HTCexp object and update the contact map accordingly

Usage
```r
setIntervalScale(x, xgi, ygi, upa=TRUE, method=c("sum","median","mean"), use.zero=TRUE, optimize.by = c("speed","memory"))
```

Arguments
- `x`: object that inherits from class HTCexp
- `ygi`: y intervals; see class GRanges for details
- `xgi`: x intervals; see class GRanges for details
- `upa`: logical; unique primer assignment. Allow one primer to belong to one or several bins
- `method`: the method used to combine the counts. Must be 'mean', 'median' or 'sum'
- `use.zero`: logical; use the zero values in the `method` calculation
- `optimize.by`: "speed" will use faster methods but more RAM, and "memory" will be slower, but require less RAM

Details
Define new contact map based on the specified xgi and ygi intervals.

This function has to be used carefully and can has important impact on the contact map. It is important to note that the `setIntervalScale` function is different from the `binningC` function in the way that the output is not symmetrical.

Value
A HTCexp object

Author(s)
N. Servant

See Also
- HTCexp-class
Examples

```r
data(Nora_5C)

E14.bin<-binningC(E14$chrXchrX)

## I have two HTCexp samples defined with different intervals.
show(E14.bin)
show(MEF$chrXchrX)

## How to compare them?
## One idea is to force the intervals definition of one object using the
## intervals of the other.

setIntervalScale(MEF$chrXchrX, xgi=x_intervals(E14.bin), ygi=y_intervals(E14.bin))
```
Index

*Topic IO
  directionalityIndex, 4
  exportC, 7
  import.my5C, 17
  importC, 18

*Topic classes
  HTCexp-class, 13
  HTClist-class, 16

*Topic datasets
  Nora_5C, 22

*Topic hplot
  getAnnotatedRestrictionSites, 9
  getRestrictionFragmentsPerChromosome, 12
  mapC, 21

*Topic manip
  binningC, 2
  CQC, 3
  discretize, 5
  extractRegion, 8
  getExpectedCounts, 10
  getPearsonMap, 11
  intervalsDist, 20
  pca.hic, 25
  removeIntervals, 26
  setIntervalScale, 29

*Topic plot
  export.my5C, 6
  [,HTClist,ANY,ANY,ANY-method (HTClist-class), 16
  [,HTClist,ANY,ANY-method (HTClist-class), 16
  as.list,HTClist-method (HTClist-class), 16
  binningC, 2
  c,HTCexp-method (HTCexp-class), 13
  c,HTClist-method (HTClist-class), 16
  CQC, 3
  detail (HTCexp-class), 13
  detail,HTCexp-method (HTCexp-class), 13
  detail,HTClist-method (HTClist-class), 16
  detail-methods (HTCexp-class), 13
  directionalityIndex, 4
  discretize, 5
  divide (HTCexp-class), 13
  divide,HTCexp,HTCexp-method (HTCexp-class), 13
  divide-methods (HTCexp-class), 13
  E14 (Nora_5C), 22
  export-defunct (HTCexp-class), 13
  export.my5C, 6, 7
  exportC, 6, 7, 19
  extractRegion, 8
  forcePairwise (HTClist-class), 16
  forcePairwise,HTClist-method (HTClist-class), 16
  forceSymmetric,HTCexp,character-method (HTCexp-class), 13
  forceSymmetric,HTCexp,missing-method (HTCexp-class), 13
  forceSymmetric,HTClist,character-method (HTClist-class), 16
  forceSymmetric,HTClist,missing-method (HTClist-class), 16
  forceTriangular (HTCexp-class), 13
  forceTriangular,HTCexp-method (HTCexp-class), 13
  forceTriangular-method (HTCexp-class), 13
  getAnnotatedRestrictionSites, 9, 13, 25
  getCombinedContacts (HTClist-class), 16
  getCombinedContacts,HTClist-method (HTClist-class), 16
  getCombinedContacts-method (HTClist-class), 16
  getCombinedIntervals (HTClist-class), 16
  getCombinedIntervals,HTClist-method (HTClist-class), 16
  getCombinedIntervals-method (HTClist-class), 16
getExpectedCounts, 10, 13
getPearsonMap, 11
getRestrictionFragmentsPerChromosome, 12

HTCexp (HTCexp-class), 13
HTCexp-class, 13
HTClist (HTClist-class), 16
HTClist-class, 16, 18

id (HTCexp-class), 13
id, GRanges-method (HTCexp-class), 13
id-methods (HTCexp-class), 13
importC, 18
import.data, 7, 18
import, my5C, 17, 19
intdata (HTCexp-class), 13
intdata, HTCexp-method (HTCexp-class), 13
intdata-method (HTCexp-class), 13
intdata<-(HTCexp-class), 13
intdata<-, HTCexp, Matrix-method (HTCexp-class), 13
intdata<--methods (HTCexp-class), 13
intervalsDist, 20
isBinned (HTCexp-class), 13
isBinned, HTCexp-method (HTCexp-class), 13

isBinned, HTClist-method (HTClist-class), 16
isBinned-methods (HTCexp-class), 13
isComplete (HTClist-class), 16
isComplete, HTClist-method (HTClist-class), 16
isComplete-methods (HTClist-class), 16
isIntraChrom (HTCexp-class), 13
isIntraChrom, HTCexp-method (HTCexp-class), 13

isIntraChrom, HTClist-method (HTClist-class), 16
isIntraChrom-methods (HTCexp-class), 13
isPairwise (HTClist-class), 16
isPairwise, HTClist-method (HTClist-class), 16
isPairwise-method (HTCexp-class), 13
isPairwise-methods (HTCexp-class), 16
isSymmetric (HTCexp-class), 13
isSymmetric, HTCexp-method (HTCexp-class), 13
isSymmetric-methods (HTCexp-class), 13
isTriangular (HTCexp-class), 13
isTriangular-methods (HTCexp-class), 13

lowess, 11

mapC, 14, 21
mapC, HTCexp, ANY-method (mapC), 21
mapC, HTCexp, HTCexp-method (mapC), 21
mapC, HTClist, ANY-method (mapC), 21
mapC-methods (mapC), 21

Matrix-class, 18

MEF (Nora_5C), 22

Nora_5C, 22

normICE, 23

normLGF, 9, 13, 24, 28

normPerExpected, 11, 12, 26

normPerExpected (HTCexp-class), 13

normPerExpected, HTCexp-method (HTCexp-class), 13

normPerExpected, HTClist-method (HTClist-class), 16

normPerExpected-methods (HTCexp-class), 13

normPerReads (HTCexp-class), 13
normPerReads, HTCexp-method (HTCexp-class), 13

normPerReads-methods (HTCexp-class), 13
normPerTrans (HTCexp-class), 13

normPerTrans, HTCexp, HTCexp-method (HTCexp-class), 13

normPerTrans-methods (HTCexp-class), 13

normPerZscore-defunct (HTCexp-class), 13

pca.hic, 25

plot, HTCexp, ANY-method (HTCexp-class), 13

plot, HTCexp, HTCexp-method (HTCexp-class), 13

plot, HTClist, ANY-method (HTClist-class), 16

range (HTCexp-class), 13
range, HTCexp-method (HTCexp-class), 13

range, HTClist-method (HTClist-class), 16
range-method (HTCexp-class), 13
ranges (HTClist-class), 16
ranges, HTClist-method (HTClist-class), 16
ranges-methods (HTClist-class), 16
reduce (HTClist-class), 16

reduce, HTClist-method (HTClist-class), 16

reduce-methods (HTClist-class), 16
removeIntervals, 26

seq_name-deprecated (HTCexp-class), 13

seqlevels (HTCexp-class), 13
INDEX

seqlevels, HTCexp-method (HTCexp-class), 13
seqlevels, HTClist-method
   (HTClist-class), 16
seqlevels-method (HTCexp-class), 13
setGenomicFeatures, 9, 13, 25, 27, 28
setIntervalScale, 29
show, HTCexp-method (HTCexp-class), 13
show, HTClist-method (HTClist-class), 16
substract (HTCexp-class), 13
substract, HTCexp, HTCexp-method
   (HTCexp-class), 13
substract-methods (HTCexp-class), 13
summary, HTCexp-method (HTCexp-class), 13
summary, HTClist-method (HTClist-class), 16
x_intervals (HTCexp-class), 13
x_intervals, HTCexp-method
   (HTCexp-class), 13
x_intervals-methods (HTCexp-class), 13
x_intervals<- (HTCexp-class), 13
x_intervals<-, HTCexp, GRanges-method
   (HTCexp-class), 13
x_intervals<--methods (HTCexp-class), 13
xy_intervals (HTCexp-class), 13
xy_intervals, HTCexp-method
   (HTCexp-class), 13
xy_intervals-methods (HTCexp-class), 13
y_intervals (HTCexp-class), 13
y_intervals, HTCexp-method
   (HTCexp-class), 13
y_intervals-methods (HTCexp-class), 13
y_intervals<- (HTCexp-class), 13
y_intervals<-, HTCexp, GRanges-method
   (HTCexp-class), 13
y_intervals<--methods (HTCexp-class), 13