Package 'HiTC'

April 5, 2014

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.

Version 1.6.0

Date 2013-06-13

Depends R (>= 2.15.0), methods, GenomicRanges, IRanges, Matrix,RColorBrewer

Imports methods, Biobase, Biostrings, graphics, grDevices, rtracklayer

Suggests rtracklayer,BiocStyle

Author Nicolas Servant

Maintainer Nicolas Servant <nicolas.servant@curie.fr>

License Artistic-2.0

Collate AllGenerics.R HTCexp-class.R HTClist-class.R qualityControl.R mapC.R mapC_tracks.R normalize.R binningC.R import.R export.R pca.R deprecated.R

LazyLoad yes

biocViews Sequencing, HighThroughputSequencing

InstallableEverywhere yes

R topics documented:

`binningC` *Windowing of high-throughput 'C' interaction matrix*

Description

Windowing of 'C' interaction map

Usage

```
binningC(x, binsize=100000, bin.adjust=TRUE, upa=TRUE,
method=c("median","mean","sum"), 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 bin of the same size. Otherwise, the last bin will has a size different from `binsize`. A primer is assigned to a bin, if there is at least one base overlap between the bin and the primer region.
The `method` is used to combine the counts in a bin, must be 'mea', '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 intraction data. In this case, the genomic intervals are converted into bins of fixed size. The interaction matrix is symetric.

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)
```

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, dev.new=FALSE)
```

Arguments

x	object that inherits from class HTCexp or HTClist
cis.trans.ratio	logical; barplot of percentage of inter-intrachromsomal 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]
dev.new	logical; specifying if each plots must be in a separate graphical device

Details

If x is a HTClist 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)
```

---

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

```
## 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

```
export.my5C(x, file, format=c("mat","list"), genome="mm9", header=TRUE)
```

Arguments

x	object that inherits from class HTCexp
file	character; the name of the output file
format	Either 'list' or 'mat'. See details
genome	The genome version. This information is only used for the 'mat' export format. See details
header	if true; a header is added in the output file with the package version and the date

Details

If 'format=list',a my5C tabbed delimited file is created, with :
Y_INTERVAL_NAME/X_INTERVAL_NAME/INTERACTION_COUNT
Otherwise, 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

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

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

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

## End(Not run)
```

---

exportC *Export* HTCexp *object*

Description

Export HTCexp object to csv format

Usage

```
exportC(x, file)
```

Arguments

x	object that inherits from class HTCexp
file	character; the basename of the output file

Value

Three output files will be created ; 2 BED files for each genomic intervals, and one matrix file

Author(s)

N. Servant

See Also

export.my5C, importC

Examples

```
## 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

```
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

Examples

```
data(Nora_5C)

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

`getExpectedCounts` *Estimate expected interaction counts of a High-Throughput C intra-chromsomal 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

```
getExpectedCounts(x, span=0.01, bin=0.005, stdev=FALSE, plot=FALSE)
```

Arguments

`x`	object that inherits from class `HTCexp`
`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

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 a Lowess regression model. 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

Author(s)

N. Servant, B. Lajoie

See Also

HTCexp-class,normPerExpected, normPerExpected, lowess

Examples

```
data(Nora_5C)

## Estimate expected interaction from distance between intervals
E14.exp<-getExpectedCounts(E14$chrXchrX, stdev=TRUE, plot=FALSE)
mapC(HTCexp(E14.exp$exp.interaction, xgi=x_intervals(E14$chrXchrX), ygi=y_intervals(E14$chrXchrX)))
```

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 a the dependency on the genomic proximity between two loci. If stdev is false, the ratio observed/expected is returned, otherwise, the zscore ((observed-expected)/stdev) is returned Look at the getExpectedCounts function for details.

The normPerTrans method is based on the assumption that all trans interactions should be the same. Thus, the cis interactions 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 interaction Matrix with x and y intervals definition.

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 interaction 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

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

isIntraChrom(x) return TRUE if the current `signature("HTCexp")` object contains intrachromosomal interaction data

isSymmetric(x) return TRUE if the interaction map is symmetrical, i.e inherits the `symmetricMatrix` class

normPerReads(x) normalize the interaction matrix by the total number of reads of the matrix.

normPerExpected(x, stdev=TRUE) normalize the interaction 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 interaction map based on the trans interactions. See details

plot(x) visualization method; Display an heatmap of the interaction 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

substract(x) comparison of two `signature("HTCexp")` objects. Perform the substraction of the two interaction 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

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

Author(s)

Nicolas Servant

See Also

GRanges-class,GRangesList-class,Matrix-class

Examples

```
data(Nora_5C)

## HTCexp descriptio
show(E14)
detail(E14)

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

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

## Divide by expected interaction counts
E14norm<-normPerExpected(E14$chrXchrX)

## 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"), forma
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=wgEncodeUm
ENCODE=import.my5C("./ENM-GM12878-R1.matrix")

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

## look at normalize by trans interaction 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)
```

HTClist-class *Class 'HTClist'*

Description

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

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.

isBinned(x) applies 'isBinned' to each element in 'x'

isIntraChrom(x) applies 'isIntraChrom' to each element in 'x'

ranges(x) applies 'range' to each element in 'x'

range(x) return the reduce range of all elements in 'x'

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

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)
isBinned(hiC)
isIntraChrom(hiC)
seqlevels(hiC)
```

`import.my5C` *Import data from my5C webtool*

Description

Import data from my5C webtool

Usage

```
import.my5C(my5C.datafile, xgi.bed=NULL, ygi.bed=NULL, all.pairwise=TRUE, forceSymmetric=FALSE)
```

Arguments

`my5C.datafile`	input file from the my5C webtool
`xgi.bed`	BED file describing the 'x' Intervals (i.e. column names) of the interaction map. Required for the my5C list format
`ygi.bed`	BED file describing the 'y' intervals (i.e. row names) of the interaction map. Required for the my5C list format
`all.pairwise`	logical; generate all pairwise chromosomal interaction maps, i.e chr1-chr2, chr2-chr1
`forceSymmetric`	Force the Matrix to be symmetrical

Details

This function allows data import from the the my5C webtool.
Two input formats can be used :
- The list format is composed of three files; two BED files describing the genomic intervals (i.e. primers); and a tabbed delimited format to specify the interaction between each genomic regions, with :
FORWARD_PRIMER_NAME/REVERSE_PRIMER_NAME/INTERACTION_COUNT. In this case, BED files describing the genomic coordinates are required.

- The matrix format is a tab-delimited format, corresponding to the interaction map. The rownames and columnames are splitted to created the genome intervals (example : REV_2|mm9|chrX:98831149-98834145).

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

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

Value

A `HTClist` object(s)

Author(s)

N. Servant

See Also

`import`, codeHTClist-class, codeMatrix-class, `symmetricMatrix-class`

Examples

```
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` *Import high-htroughput 'C' data*

Description

Import 5C or Hi-C data from csv file

Usage

```
importC(con, xgi.bed, ygi.bed = NULL, all.pairwise=TRUE)
```

Arguments

`con`	input csv file. See details
`xgi.bed`	BED file describing the 'x' Intervals (i.e. column names) of the interaction map. Required for the my5C list format
`ygi.bed`	BED file describing the 'y' intervals (i.e. row names) of the interaction map. Required for the my5C list format
`all.pairwise`	logical; generate all pairwise chromosomal interaction maps, i.e chr1-chr2, chr2-chr1

Details

This function import high-throughput data from a matrix file.

Value

A list of HTCexp object(s)

Author(s)

N. Servant

See Also

exportC,import.my5C, HTCexp-class

Examples

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

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

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

##Import
importC("E14.csv")

## End(Not run)
```

---

intervalsDist *intervalsDist*

Description

Compute the distance of intrachromosomal interactions of a 'C' experiment

Usage

```
intervalsDist(x)
```

Arguments

x	object that inherits from class HTCexp

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 intrachromsomal interaction maps can be use for this operation.

Value

A matrix of distances between genomic intervals

Author(s)

N. Servant

See Also

HTCexp-class

Examples

```
data(Nora_5C)

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

---

mapC *Visualize 'C' interaction map*

Description

Visualize 'C' interaction map

Details

This function implements the `plot` method for objects of class `HTCexp` and `HTClist`.

By default, the maxrange and minrange values are fixed as the 98th percentile (resp. 2th percentile) of the interaction matrix. These values are useful to play with the contrast and remove the extreme values from the matrix.

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 interaction 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 interaction counts. Must be a vectore of size 3. mid can be NA

col.neg color for (low,mid,high) negative interaction 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 interaction 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)

## Interaction 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", asRangedData=FALSE)
mapC(E14$chrXchrX, tracks=list(Refseq=gene))

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

Nora_5C *HiTC - 5C data*

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

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE35721

References

Nora EP, Lajoie BR, Schulz EG, Giorgetti L et al. Spatial partitioning of the regulatory landscape of the X-inactivation centre. Nature 2012 Apr 11;485(7398):381-5. PMID: 22495304

Examples

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

`pca.hic` *Perform Principle Component Analysis on Hi-C interaction map*

Description

Perform Principle Component Analysis on Hi-C interaction map

Usage

```
pca.hic(x, normPerExpected=TRUE, npc=2, asGRangesList=TRUE)
```

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

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

```
## 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)

## Performed PCA
pr<-pca.hic(hiC$chr14chr14, npc=1, asGRangesList=TRUE)
```

`removeIntervals` *Remove intervals from HTCexp object*

Description

Remove primers 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"))
```

setIntervalScale *Set x and y interval of the HTCexp object*

Description

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

Usage

```
setIntervalScale(x, xgi, ygi, upa=TRUE, method=c("median","mean","sum"),
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 interaction map based on the specified xgi and ygi intervals.

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

Value

A HTCexp object

Author(s)

N. Servant

See Also

HTCexp-class

Examples

```
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