Package ‘deepC’

August 31, 2020

R topics documented:

binInteractionValues *binInteractionValues*

Description

Take hic interactions and bin them into [bins] distinct classes equally split will reserve one bin for true 0s.

Usage

```
binInteractionValues(hic, bins = 9)
```

Arguments

`hic`	hic object (list) as imported.
`bins`	number of to split values into. Will add +1 bin for zeros.

Value

hic object with values binned.

Examples

```
## Not run:
binInteractionValues(hic, bins=9)

## End(Not run)
```

calcInsulationScoreChange

calcInsulationScoreChange

Description

Calculate changes in the interaction score that pass the diagonals starting from the variant bin —> insulation score changes. The "ins" snsulation score change. Also calculate changes in the score that do not pass the diagonal but are pot. affected by the centre of a domain. The "cont" containment score change.

Usage

```
calcInsulationScoreChange(diff.df, bin.size = 10000)
```

Arguments

`diff.df`	differential data frame as from getDiffDeepC.
`bin.size`	Bin size in bp. default = 10000

Value

List of length two with the .$ins change and the .$cont change

Examples

```
## Not run:
calcInsulationScoreChange(diff.df, bin.size = 5000)

## End(Not run)
```

callInsulationBoundaries

callInsulationBoundaries

Description

Call insulation score based TAD boundaries on a deepC data frame. Calculate the insulation score and approximations of the first and second order derivative. Select minima in the insulation score by looking for zero crossings in teh first order derivative "delta". Choose minima based on neighborhood and filter them on based on their sharpness "delta2". Requires a deepC like data frame (HiC, Skeleton, Prediction) ins.distance - distance from sliding center position in which to calculate the insulation score. For predictions 25 kb works great for HiC need larger distance. Delta2 threshold for local minima to be classed as boundaries.

Usage

```
callInsulationBoundaries(d, ins.dist = 25000, delta2.thresh = 5)
```

Arguments

`d`	deepc interaction data frame
`ins.dist`	Genomic distance window based on which which to calculate the insulation. Default 25000 Default in Crane et al. is 250 k.
`delta2.thresh`	Threshold of the delta2 score, the sharpness, (second order derivative approximation) based on which to filter boundaries. Keep above threshold default = 5

Value

Returns a dataframe based on the HiC/deepC position (bin == 1) with new columns for insulationscore, delta and delta2 (frst and second derivative approcimations and a 0/1 flag if called as minimum also a vector of local minima positions. List("df.ins" = df.ins, "boundaries")

Examples

```
## Not run:
callInsulationBoundaries(df, ins.dist = 25000, delta2.thresh = 5)

## End(Not run)
```

`ConvertBinToPos` *Convert Bin IDs/Numbers to Genomic Position*

Description

Converts bin identifiers / counts to genomic positions, given the bin size and fgenomic starting position.

Usage

```
ConvertBinToPos(x, bin.size, start.pos, start.id = 1)
```

Arguments

`x`	Genomic bin id/count
`bin.size`	Size of bins in bp
`start.pos`	Starting position
`start.id`	Number of first bin id

Value

Genomic position of provided bin ID.

Examples

```
ConvertBinToPos(12, bin.size = 5000, start.pos = 1000000, start.id = 1)
```

custom_ceil *custom_ceil*

Description

Ceiling round to a custom base (nearest base)

Usage

```
custom_ceil(x, base)
```

Arguments

x	Input number to round
base	Base to round to (base=1 for ordinary ceiling rounding)
extra	Character vector specifying which columns to extract interactions etc. from. Default c("value"). Optional length

Value

Number x rounded to nearest base ceiling.

Examples

```
custom_ceil(1503, 500)
```

custom_floor *custom_floor*

Description

Floor round to a custom base (nearest base)

Usage

```
custom_floor(x, base)
```

Arguments

x	Input number to round
base	Base to round to (base=1 for ordinary floor rounding)
extra	Character vector specifying which columns to extract interactions etc. from. Default c("value"). Optional length

Value

Number x rounded to nearest base floor.

Examples

```
custom_floor(1503, 500)
```

`custom_round` *custom_round*

Description

Round to a custom base (nearest base) round(x/base)*base

Usage

```
custom_round(x, base)
```

Arguments

`x`	Input number to round
`base`	Base to round to (base=1 for ordinary rounding)
`extra`	Character vector specifying which columns to extract interactions etc. from. Default c("value"). Optional length

Value

Number x rounded to nearest base.

Examples

```
custom_round(1503, 500)
```

`deduplicate` *deduplicate*

Description

Remove duplicated rows from a tibble/data frame. "Unique" on tibbles works better and faster

Usage

```
deduplicate(df)
```

Arguments

`df`	data frame / tibble

Value

data frame with duplicate rows removed.

Examples

```
## Not run:
deduplicate(df)

## End(Not run)
```

FormatBp *Format basepairs*

Description

Format a single bp position to kilo (k) or Mega (M) given type=k/M

Usage

```
FormatBp(bp, type = "M")
```

Arguments

bp	Size of bins in bp
type	Type of formating into kilo ’k’ or Mega ’M’ (default=’M’)

Value

Basepairs formated in kilo or mega.

Examples

```
FormatBp(1000, type = "k")
```

getBinnedChrom *getBinnedChrom*

Description

helper function to get a binned genome using bedtools (requires bedtools available) Will Run bedtools on your default shell saving temporary files in your working directory. Create a binned genome data frame on the specified chromosome from start to end positions in window.sized bins and step sized steps. See "bedtools makewindows" which is the fucntion invoked.

Usage

```
getBinnedChrom(
  chr = "chr1",
  start = 0,
  end = 10000,
  window = 1000,
  step = 500,
  timestamp = paste0(gsub("\\s+", "_", Sys.time()), "_", runif(n = 1))
)
```

Arguments

`chr`	chromosome
`start`	start position
`end`	end postion
`window`	size of the window / bins
`step`	The step size between windows/bins. The increment.
`timestamp`	automatically produces a timestamp to uniquely identify temporary files defaut = paste0(gsub("\s+", "_", Sys.time()), "_", runif(n=1))

Value

data frame with genomic bins

Examples

```
## Not run:
getBinnedChrom(chr='chr1', start=0, end=10000, window=1000, step=500)

## End(Not run)
```

`GetBinsEnd` *GetBinsEnd*

Description

Helper function to retrieve the end (right most position) of the last bin of a hic object

Usage

```
GetBinsEnd(hic)
```

Arguments

`hic`	hic object

Value

End position (right most) of the last bin of the hic object

Examples

```
## Not run:
GetBinsEnd(my.hic)

## End(Not run)
```

GetBinsStart *GetBinsStart*

Description

Helper function to retrieve the start (left most position) of the first bin of a hic object

Usage

```
GetBinsStart(hic)
```

Arguments

hic	hic object

Value

Start position (left most) of the first bin of the hic object

Examples

```
## Not run:
GetBinsStart(my.hic)

## End(Not run)
```

getBinSubMatrixfunction *getBinSubMatrixfunction*

Description

Get A pruned submatrix from hic object given a region of interest as well as an interaction data frame.

Usage

```
getBinSubMatrixfunction(
  hic.obj,
  chr = "chr1",
  start = 0,
  end = 20000,
  bin.size = 10000
)
```

Arguments

chr	chromosome
start	start position
end	end postion
bin.size	Bin size in bp default = 10000 defaut = paste0(gsub("\s+", "_", Sys.time()), "_", runif(n=1))
step	The step size between windows/bins. The increment.

Value

List with two elements "hicp" the pruned Hic object and "idf" the interaction data frame.

Examples

```
## Not run:
getBinSubMatrixfunction(chr='chr1', start=0, end=10000, bin.size = 5000)

## End(Not run)
```

getDiffDeepC *getDiffDeepC*

Description

Calcualte difference between two hic/deepc data frames (df1 - df2). e.g reference - variant

Usage

```
getDiffDeepC(df1, df2)
```

Arguments

df1	Interaction data frame (reference)
df2	Interaction data frame (variant)

Value

data frame of (predicted) interaction difference

Examples

```
## Not run:
getDiffDeepC(df1, df2)

## End(Not run)
```

`GetDifferenceMatrix` *GetDifferenceMatrix*

Description

Helper function to unify and caluclate the difference of two matrices Matrices should be normalised towards each other, this function won't take care of normalisation.

Usage

```
GetDifferenceMatrix(a, b)
```

Arguments

`a`	hic matrix in long format
`b`	hic matrix in long format

Value

Hi-C matrix with delta = a - b column

Examples

```
## Not run:
GetDifferenceMatrix(hic.wt, hic.variant)

## End(Not run)
```

`getDistanceWindow` *getDistanceWindow*

Description

Helper fuction to pick a distance window with the same distance from center position from a hic/deepC data frame

Usage

```
getDistanceWindow(d, center, distance)
```

Arguments

`d`	hic data frame / tibble
`center`	center position of region of interest
`distance`	distance from center to extract

Value

Hic/deepc data frame from region of interest

Examples

```
## Not run:
getDistanceWindow(d, center = 1000000, distance = 500000)

## End(Not run)
```

`getVerticalWindowInteractions`

getVerticalWindowInteractions

Description

Extract desired vertical center pole of interactions over the center of every window of window.size. Older non zig zag encoding.

Usage

```
getVerticalWindowInteractions(hic.obj, bin.df, window, bin)
```

Arguments

`hic.obj`	hic object including bin coordinates and an interaction matrix
`bin.df`	Binned genome / genomic segment data frame as from getBinnedChrom
`window`	window size in bp of interest usually 1 Mb + 1x bin size
`bin`	Bin size in bp defaut = paste0(gsub("\s+", "_", Sys.time()), "_", runif(n=1))

Value

Dataframe/tibble of interactions encoded in a vertical pole over the center of the window.

Examples

```
## Not run:
getVerticalWindowInteractions(hic, binned.genome, window=1e6+5000, bin = 5000)

## End(Not run)
```

`getZigZagWindowInteractionsPerl`

getZigZagWindowInteractionsPerl

Description

Extract desired vertical zigzag pole of interactions over the center of every window of window.size. Uses helper perl script for faster processing. First creates a query data frame, then saves the dquery and the hic data frame in temporary files in the working directory. Then uses the helper perl script to query and match and read the output file into R again. REQUIRES: perl installed and available in your default shell. #'

Usage

```
getZigZagWindowInteractionsPerl(
  hic.obj,
  bin.df,
  window,
  bin,

   query.pl = "~/fusessh/scripts/machine_learning/epigenome_nets/deepC/match_query_table.pl",
  timestamp = paste0(gsub("\\s+", "_", Sys.time()), "_", runif(n = 1))
)
```

Arguments

`hic.obj`	hic object including bin coordinates and an interaction matrix
`bin.df`	Binned genome / genomic segment data frame as from getBinnedChrom
`window`	window size in bp of interest usually 1 Mb + 1x bin size
`bin`	Bin size in bp
`query.pl`	path to helper perl script as distributed with the deepC repository "match_query_tabel.pl"
`timestamp`	automatically produces a timestamp to uniquely identify temporary files defaut = paste0(gsub("\s+", "_", Sys.time()), "_", runif(n=1))

Value

Dataframe/tibble of interactions encoded in a vertical pole over the center of the window.

Examples

```
## Not run:
getZigZagWindowInteractionsPerl(hic, binned.genome, window=1e6+5000, bin = 5000, query.pl = '~/myscripts/deep

## End(Not run)
```

`getZigZagWindowInteractionsPerlMemoryFriendly` *getZigZagWindowInteractionsPerlMemoryFriendly*

Description

Extract desired vertical zigzag pole of interactions over the center of every window of window.size. Uses helper perl script for faster processing. First creates a query data frame, then saves the dquery and the hic data frame in temporary files in the working directory. Then uses the helper perl script to query and match and read the output file into R again. REQUIRES: perl installed and available in your default shell. Memory friendly version that uses a second perl helper script for query preparation. Slower then default version.

Usage

```
getZigZagWindowInteractionsPerlMemoryFriendly(
  hic.obj,
  bin.df,
  window,
  bin,

   query.pl = "~/fusessh/scripts/machine_learning/epigenome_nets/deepC/match_query_table.pl",

   prepare.pl = "~/fusessh/scripts/machine_learning/epigenome_nets/deepC/prepare_query_table.pl",
  timestamp = paste0(gsub("\\s+", "_", Sys.time()), "_", runif(n = 1))
)
```

Arguments

`hic.obj` hic object including bin coordinates and an interaction matrix

`bin.df` Binned genome / genomic segment data frame as from getBinnedChrom

`window` window size in bp of interest usually 1 Mb + 1x bin size

`bin` Bin size in bp

`query.pl` path to helper perl script as distributed with the deepC repository "match_query_tabel.pl"

`prepare.pl` path to helper perl script as distributed with the deepC repository "prepare_query_tabel.pl"

`timestamp` automatically produces a timestamp to uniquely identify temporary files defaut = paste0(gsub("\s+", "_", Sys.time()), "_", runif(n=1))

Value

Dataframe/tibble of interactions encoded in a vertical pole over the center of the window.

Examples

```
## Not run:
getZigZagWindowInteractionsPerlMemoryFriendly(hic,
binned.genome,
window=1e6+5000,
bin = 5000,
query.pl = '~/myscripts/deepC/match_query_table.pl',
prepare.pl = '~/myscripts/deepC/prepare_query_tabel.pl')

## End(Not run)
```

`ImportHicproMatrix` *ImportHicproMatrix*

Description

Wrapper function for importing a Hi-C matrix and bin coordinates. Built around HiC-Pro data format with long sparse matrix and bed bed file specifying bin coordiantes. Import a HiCPro matrix and the according bed coordinates to assign the bin ids. The coordinates bed file is optional if the matrix is encoded as genomic position rather than bin ids This function will handle it accordingly. Add a centric, single coord per bin region. Calculate bin size and extract first id and genomic position. Store as list including a ggplot2 friendly data frame/tibble.

Usage

```
ImportHicproMatrix(
  matrix,
  coords = "empty",
  chr = "empty",
  bin.size = 0,
  nrow = -1
)
```

Arguments

`matrix`	Path to matrix file. 3 column matrix as result from HiCPro or pruned version (id.x id.y, interaction). May be coordinates (left most of each bin)
`coords`	Path to bed file specifying the genomic location of genomic bins. 4 column bed file (chr, start, end, id). If left "empty" will assume coords can be read from the bins of the matrix and extract it from there
`chr`	Specify a chromosome to add to the matrix object if the coordinates bed file was left empty.
`bin.size`	Matrix bin size in bp.
`nrow`	Specify number of rows to read in from the matrix. Dont speficy or set to -1 to read entire matrix.

Value

"HiC object" a list of .$matrix.df .$bin.size .$start.pos .$start.id

Examples

```
## Not run:
ImportHicproMatrix(path.to.matrix, path.to.bed, bin.size = 5000)

## End(Not run)
```

`IntersectBedDataframe` *IntersectBedDataframe*

Description

Wrapper to subset/intersect a given BED format df with a region of interest.

Usage

```
IntersectBedDataframe(df, chr, start, end)
```

Arguments

`df`	Bedlike 4 or more column data frame
`chr`	Chromsome of interest
`start`	Start coordinate of interest
`end`	End coordinate of interest

Value

Bed like data frame with all rows that intersect with the region of interest.

Examples

```
## Not run:
IntersectBedDataframe(df, chr='chr1', start='1000000', end='5000000')

## End(Not run)
```

leftHandNotate *leftHandNotate*

Description

Function to change matrix notation to left most base of the bin notation

Usage

```
leftHandNotate(matrix, bin.size = 1000)
```

Arguments

`matrix`	hic matrix data frame
`bin.size`	binsize

Value

hic matrix with bins left hand notated.

Examples

```
## Not run:
leftHandNotate(matrix)

## End(Not run)
```

MakeAnnotationDataFrame *MakeAnnotationDataFrame*

Description

Adjust a bed like data frame for annotation for ggplot2 like plotting.

Usage

```
MakeAnnotationDataFrame(df, name = "annotation", row = "1")
```

Arguments

`df`	Bed like data frame (min 3 columns(chr, start, end, (name)))
`name`	Name for the annotation to appear
`row`	Row in which annotation should appear 1 lowest ++ (default=1)

Value

Adjusted bed like data frame with annotation row, up and down border and annotation name column.

Examples

```
## Not run:
MakeAnnotationDataFrame(gene.df, name="genes", row = "1")

## End(Not run)
```

`MakeBedGraphDataFrameForSignal` *MakeAnnotationDataFrame*

Description

Adjust a bed like data frame for annotation for ggplot2 like plotting.

Usage

```
MakeBedGraphDataFrameForSignal(bg)
```

Arguments

`bg`	Bed like data frame with columns chr start end value

Value

Dataframe in bedgraph format.

Examples

```
## Not run:
MakeBedGraphDataFrameForSignal(bed.df)

## End(Not run)
```

makeGaussianMask *makeGaussianMask*

Description

Make Gaussian Mask that weights interactions or interaction differences in the center and the diagonal of a given window. Use plotGaussianMask to visualise it.

Usage

```
makeGaussianMask(
  window.size = 1010000,
  bin.size = 10000,
  diag.sigma.pos = 75000,
  diag.sigma.bin = 25,
  cen.sigma.pos = 2e+05,
  cen.sigma.bin = 40
)
```

Arguments

`window.size`	Window size in bp normally 1 MB + 1x bin.size
`bin.size`	Bin size in bp default = 10000
`diag.sigma.pos`	Sigma parameter for gaussian for the diagonals along postion dimension (e.g. 75000 for a 1010000 window size and diagonal from the center)
`diag.sigma.bin`	Sigma parameter for gaussian for the diagonals along bin/genomic distance dimension (e.g. 25 for a 1010000 window size and diagonal from the center)
`cen.sigma.pos`	Sigma parameter for gaussian for the center along postion dimension (e.g. 200000 for a 1010000 window size and diagonal from the center)
`cen.sigma.bin`	igma parameter for gaussian for the diagonals along bin/genomic distance dimension (e.g. 40 for a 1010000 window size and diagonal from the center)

Value

Mask for weighting.

Examples

```
## Not run:
makeGaussianMask(window.size = 1010000,
bin.size = 10000,
diag.sigma.pos = 75000,
diag.sigma.bin = 25,
cen.sigma.pos = 200000,
cen.sigma.bin = 40)

## End(Not run)
```

`MakeMirrorMatrix` *MakeMirrorMatrix*

Description

Mirror a half full (sparse, HiC-Pro like) data frame formated for ggplot2 to get a full matrix df to plot

Usage

```
MakeMirrorMatrix(df)
```

Arguments

`df`	Hi-C data frame (sparse, half full)

Value

Full matrix HiC data frame.

Examples

```
## Not run:
MakeMirrorMatrix(hic.df)

## End(Not run)
```

`MakeTriangleMatrix` *MakeTriangleMatrix*

Description

Take a single hic matrix data frame (non mirrored) as input convert it to triangular like polygon df over genomic locus for plotting.

Usage

```
MakeTriangleMatrix(df)
```

Arguments

`df`	Hi-C data frame (sparse, half full)

Value

Triangularised matrix for plotting as polygons.

Examples

```
## Not run:
MakeTriangleMatrix(hic.df)

## End(Not run)
```

medianImputeZerosDataFrame *medianImputeZerosDataFrame*

Description

Sets zero values in an interaction matrix of a hic object in the 4th column onwards to the median of a 2xk x 2xk window padded with median of total matrix Smooth replace according to neighborhood (median of neighborhood) With k being the neighborhood size +- k around central position.

Usage

```
medianImputeZerosDataFrame(d, k)
```

Arguments

d	hic object including bin coordinates and an interaction matrix
k	Neighborhood size (kxk)

Value

Hic object with the interaciton matrix median imputed

Examples

```
## Not run:
medianImputeZerosDataFrame(hic, k=5)

## End(Not run)
```

PlotAnnotation *PlotAnnotation*

Description

Create an ggplot2 annotation plot, to plot under a triangular HiC plot and for multiplots.

Usage

```
PlotAnnotation(
  anno.df,
  xmin,
  xmax,
  break.number = 4,
  format = "s",
  pal = "Set1"
)
```

Arguments

`anno.df` Annotation version of a bed like data frame (result from MakeAnnotationDataFrame)

`xmin` minimum x value to plot for limit

`xmax` maximum x value to plot for limit

`break.number` number of breaks to print on x axis default=5

`format` options to format the genomic postion (s=single, k=in kilo, M=in Mega) default="s"

`pal` RColorBrewer palette to use default="YlOrRd"

`vector` Numeric vector.

Value

GGplot2 plot of annotation.

Examples

```
## Not run:
PlotAnnotation(anno.df, xmin = 1000000, xmax = 5000000, break.number = 4, formart = "M")

## End(Not run)
```

`plotDiffDeepC` *plotDiffDeepC*

Description

Plot difference between two hic/deepc data frames (df1 - df2). e.g reference - variant

Usage

```
plotDiffDeepC(df1, df2, bin.size = 10000, threshold = 0)
```

Arguments

`df1` Interaction data frame (reference)

`df2` Interaction data frame (variant)

`bin.size` Bin size in bp. default = 10000

`threshold` Set every interaction below "threshold" to zero

Value

ggplot2 plot of (predicted) interaction difference

Examples

```
## Not run:
plotDiffDeepC(df1, df2, bin.size = 5000, threshold = 2)

## End(Not run)
```

plotGaussianMask *plotGaussianMask*

Description

Plot a Gaussian Mask that weights interactions.

Usage

```
plotGaussianMask(mask, bin.size)
```

Arguments

mask	Gaussian amsk as produced by makeGaussianMask
bin.size	bin size in bp

Value

ggplot2 plot of Gaussian mask

Examples

```
## Not run:
plotGaussianMask(mask, 5000)

## End(Not run)
```

PlotSignalTrack *PlotSignalTrack*

Description

Print a Triangular Matrix over a genomic locus plot given a three column (y, x, value) and graphic options

Usage

```
PlotSignalTrack(
  bg.df,
  label = "",
  xlim,
  break.number = 5,
  format = "s",
  color = "black"
)
```

Arguments

`bg.df`	data frame of genomic signal. requries "pos" and "value" column.
`label`	Label to name the signal track
`xlim`	Required. Limits of signal on x-axis to plot. Numeric vector with length 2.
`break.number`	number of breaks to plot on x-axis. default = 5
`format`	options to format the genomic postion (s=single, k=in kilo, M=in Mega) default="s"
`color`	colour for the signal track.

Value

GGplot2 plot of signal 1d coverage.

Examples

```
## Not run:
PlotSignalTrack(signal.df, label="CTCF", xlim=c(1000000,5000000), break.number = 5, color="blue")

## End(Not run)
```

`PlotSquareMatrix` *PlotSquareMatrix*

Description

Wrapper function to plot a matrix given a 3 column dataframe as input Print a Square Matrix plot given a three column (y, x, value) and graphic options

Usage

```
PlotSquareMatrix(
  hic,
  break.number = 5,
  square = FALSE,
  format = "s",
  pal = "YlOrRd"
)
```

Arguments

`hic`	hic object list
`break.number`	number of breaks to print on axis default=5
`square`	FALSE/TRUE default=FALSE indicate if to mirror a triangular matrix to get a full matrix plot
`format`	options to format the genomic postion (s=single, k=in kilo, M=in Mega) default="s"
`pal`	RColorBrewer palette to use default="YlOrRd"

Value

GGplot2 plot of square Hi-C matrix.

Examples

```
## Not run:
PlotSquareMatrix(hic, break.number = 5, square =TRUE)

## End(Not run)
```

`PlotTriangleMatrix` *PlotTriangleMatrix*

Description

Print a Triangular Matrix over a genomic locus plot given a three column (y, x, value) and graphic options

Usage

```
PlotTriangleMatrix(hic, break.number = 5, format = "s", pal = "YlOrRd")
```

Arguments

`hic`	hic object list
`break.number`	number of breaks to print on axis default=5
`format`	options to format the genomic postion (s=single, k=in kilo, M=in Mega) default="s"
`pal`	RColorBrewer palette to use default="YlOrRd"

Value

GGplot2 plot of triangular Hi-C matrix.

Examples

```
## Not run:
PlotTriangleMatrix(hic, break.number = 5)

## End(Not run)
```

PruneHicproMatrix *MakeTriangleMatrix*

Description

Function to prune a matrix to zoom into a region of interest. Take a list object as imported with ImportHicproMatrix and prune it to include only interactions between the genomic coordinates chosen.

Usage

```
PruneHicproMatrix(hics, chrs, starts = 0, ends = 0)
```

Arguments

hics	Hi-C list object as imported.
chrs	Chromosome. Same formating as matrix.
starts	Start position of interest.
ends	End position of interest.

Value

HiC object pruned to region of interest. (Matix and coords pruned.)

Examples

```
## Not run:
PruneHicproMatrix(hic, chrs='chr1', starts'1000000', ends='5000000')

## End(Not run)
```

pyramidBin *pyramidBin*

Description

Percentile normalize a deepC matrix into a pyramid scheme of unequal percentiles. First quantile normalize per column with 5 into 2x20 Assumes data frame with with 4th + columns encoding the interactions. Apply quantile normailsation per column of that matrix.

Usage

```
pyramidBin(df, return.means = FALSE)
```

Arguments

df	deepC interaction data frame in pos, bin encoding
return.means	TRUE/FALSE if to return a data frame with the mean interaction values per matrix column. default = FALSE

Value

data frame with interaction in pyramid style percentile bins

Examples

```
## Not run:
pyramidBin(hic, return.means=FALSE)

## End(Not run)
```

readDeepcInputHicBed *readDeepcInputHicBed*

Description

Read in a hic data file with interactions in bins in a comma separated string in the 4th column. Use when coordinates are bin ids rather then genomic poistions/

Usage

```
readDeepcInputHicBed(
  file,
  prediction.bins = 101,
  bin.size = 10000,
  gather = TRUE,
  zigzag = TRUE
)
```

Arguments

`file`	HiC/DeepC file tab separated ("chr", "start", "end", "qbins"). Where qbins are hic interactions in comma separated string.
`prediction.bins`	Number of bins/values in the output vector, depends on bin size and context window. Default = 101
`bin.size`	Bin size in bp. default = 10000
`gather`	TRUE/FALSE if to gather the data in long format for tidyverse/ggplot default = TRUE.#' @param zigzag TRUE/FALSE if to correc thte positions for zigzag pole encoding (shifting every second bin position) default = TRUE

Value

Dataframe/tibble of quantized interactions.

Examples

```
## Not run:
readDeepcInputHicBed(df, prediction.bins = 201, bin.size = 5000, gather = TRUE, zigzag = TRUE)

## End(Not run)
```

readDeepcVariantFile *readDeepcVariantFile*

Description

Read in a deepC variant file from deploy net output. Assumes a deepC header in the input file.

Usage

```
readDeepcVariantFile(
  file,
  prediction.bins = 101,
  bin.size = 10000,
  gather = TRUE,
  tag.col = FALSE,
  zigzag = TRUE
)
```

Arguments

`file`	Path to variant output file. Tab separated ("chr", "start", "end", predictions ...)
`prediction.bins`	Number of bins/values in the output vector, depends on bin size and context window. Default = 101
`bin.size`	Bin size in bp. default = 10000
`gather`	TRUE/FALSE if to gather the data in long format for tidyverse/ggplot default = TRUE.
`tag.col`	TRUE/FALSE if a tagging or id column is present ("chr", "start", "end", "tag", predictions ...) default = FALSE
`zigzag`	TRUE/FALSE if to correc thte positions for zigzag pole encoding (shifting every second bin position) default = TRUE

Value

Dataframe/tibble of predicted interactions/

Examples

```
## Not run:
readDeepcVariantFile(file, prediction.bins = 201, bin.size = 5000, gather = TRUE, tag.col = FALSE, zigzag = TRUE

## End(Not run)
```

`readDeepcVariantFileNoHeader`

readDeepcVariantFileNoHeader

Description

Read in a deepC variant file from deploy net output. Assumes a no header present in the input file.

Usage

```
readDeepcVariantFileNoHeader(
  file,
  prediction.bins = 101,
  bin.size = 10000,
  gather = TRUE,
  tag.col = FALSE,
  zigzag = TRUE
)
```

Arguments

`file` Path to variant output file. Tab separated ("chr", "start", "end", predictions ...)

`prediction.bins` Number of bins/values in the output vector, depends on bin size and context window. Default = 101

`bin.size` Bin size in bp. default = 10000

`gather` TRUE/FALSE if to gather the data in long format for tidyverse/ggplot default = TRUE.

`tag.col` TRUE/FALSE if a tagging or id column is present ("chr", "start", "end", "tag", predictions ...) default = FALSE

`zigzag` TRUE/FALSE if to correc thte positions for zigzag pole encoding (shifting every second bin position) default = TRUE

Value

Dataframe/tibble of predicted interactions/

Examples

```
## Not run:
readDeepcVariantFileNoHeader(file, prediction.bins = 201, bin.size = 5000, gather = TRUE, tag.col = FALSE, zigz

## End(Not run)
```

readHicDataBins *readHicDataBins*

Description

Read in a hic data file with interactions in bins in a comma separated string in the 4th column..use when start and end columns are genomic positions/coordinates.

Usage

```
readHicDataBins(file, prediction.bins = 101, gather = TRUE)
```

Arguments

file	Path to variant output file. Tab separated ("chr", "start", "end", "qbins"). Where qbins are hic interactions in comma separated string.
prediction.bins	Number of bins/values in the output vector, depends on bin size and context window. Default = 101
gather	TRUE/FALSE if to gather the data in long format for tidyverse/ggplot default = TRUE.#' @param zigzag TRUE/FALSE if to correc thte positions for zigzag pole encoding (shifting every second bin position) default = TRUE
bin.size	Bin size in bp. default = 10000

Value

Dataframe/tibble of quantized hic interactions.

Examples

```
## Not run:
readHicDataBins(file, prediction.bins = 50, gather = TRUE)

## End(Not run)
```

SetValueRange *SetValueRange*

Description

Function to cap the values for a max and min value range Take a vector and a min and max numeric argument as input. Return the vector with every value below min = min and above max = max.

Usage

```
SetValueRange(vector, min = -Inf, max = Inf)
```

Arguments

`vector`	Numeric vector.
`min`	Maximum (numeric) to cap/prune to.

Value

Capped numeric vector.

Examples

```
## Not run:
SetValueRange(my.vector, min = 0, max = 5)

## End(Not run)
```

`Sobeln` *Sobeln*

Description

Calculate 1st derivative approximation of profile by applying 1D sobel filter

Usage

```
Sobeln(profile)
```

Arguments

`profile`	vector of values / signal / profile

Value

1st derivative approximation of profile (numeric vector)

Examples

```
## Not run:
Sobeln(in)

## End(Not run)
```

`triangularize` *triangularize*

Description

Triangularize a hic/deepc dataframe for polygon style plotting/ Take a df with distance bins (deepCregr ZigZag) convert it to triangular like polygon df over genomic locus Specify which extra cloumn names to extract.

Usage

```
triangularize(df, bin = 10000, extra = c("value"))
```

Arguments

`df`	Hic/deepc data frame with interactions/predictions in zig zag encoding via bins and genomic position.
`bin`	Bin size in bp. default = 10000
`extra`	Character vector specifying which columns to extract interactions etc. from. Default c("value"). Optional length

Value

Data frame specifying polygon coordinates for HiC ploting.

Examples

```
## Not run:
triangularize(df, bin.size = 5000, extra = c("value"))

## End(Not run)
```

`trimHicCoords` *trimHicRange*

Description

Function to trim a hic object to only maintain interactions within start and end coordinate

Usage

```
trimHicCoords(hic.obj, start = 1e+06, end = 2e+06)
```

Arguments

`hic.obj`	hic object (list) as imported.
`start`	Start position of region of interest to trim to
`end`	End position of region of interest to trim to

Value

hic object with interactions trimmed for a region of interest

Examples

```
## Not run:
trimHicCoords(hic, start=1000000, end=5000000)

## End(Not run)
```

trimHicRange *trimHicRange*

Description

Function to trim a hic object to only maintain interactions with a maximum interaction range

(duplicate from HiC helper scripts) Function to trim a hic object to only maintain interactions with a maximum interaction range

Usage

```
trimHicRange(hic.obj, range = 1e+06)
```

```
trimHicRange(hic.obj, range = 1e+06)
```

Arguments

`hic.obj`	hic object (list) as imported.
`range`	maximum linear distance range to maintain interactions

Value

hic object with interactions trimmed for linear disance

hic object with interactions trimmed for linear disance

Examples

```
## Not run:
trimHicRange(hic, range=1000000)

## End(Not run)
```

```
## Not run:
trimHicRange(hic, range=1000000)

## End(Not run)
```

`vec_custom_ceil` *vec_custom_ceil*

Description

Vectorised custom_ceil CeilingFloor round to a custom base (nearest base)

Usage

```
vec_custom_ceil(x, base)
```

Arguments

`x`	Input number to round
`base`	Base to round to (base=1 for ordinary floor rounding)
`extra`	Character vector specifying which columns to extract interactions etc. from. Default c("value"). Optional length

Value

Vector x rounded to nearest base ceiling

Examples

```
vec_custom_ceil(c(2505, 1503, 499), 500)
```

`vec_custom_floor` *vec_custom_floor*

Description

Vectorised custom_floor Floor round to a custom base (nearest base)

Usage

```
vec_custom_floor(x, base)
```

Arguments

`x`	Input number to round
`base`	Base to round to (base=1 for ordinary floor rounding)
`extra`	Character vector specifying which columns to extract interactions etc. from. Default c("value"). Optional length

Value

Vector x rounded to nearest base floor

Examples

```
vec_custom_floor(c(2505, 1503, 499), 500)
```

vec_custom_round *vec_custom_round*

Description

Vectorised custom_round Round to a custom base (nearest base) round(x/base)*base

Usage

```
vec_custom_round(x, base)
```

Arguments

x	Input number to round
base	Base to round to (base=1 for ordinary rounding)
extra	Character vector specifying which columns to extract interactions etc. from. Default c("value"). Optional length

Value

Vector x rounded to nearest base.

Examples

```
vec_custom_round(c(2505, 1503, 499), 500)
```

virtual4C *triangularize*

Description

Virtual 4C current implementation. Map to floor and ceiling of nearby bins For downstream interactions map between pos and - bin.size For upstream interactions map between pos and + bin.size

Usage

```
virtual4C(
  idf,
  chr,
  pos,
  bin.size,
  window.size = 0,
  pred.start = 0,
  pred.end = 0
)
```

Arguments

`idf`	input dataframe in deepC/hic format (bin, pos)
`chr`	chromsome
`pos`	genomic positon of virtual 4C "viewpoint"
`bin.size`	Bin size in bp default = 10000
`window.size`	Window in which to calculate the virtual 4C. Distance from viewpoint. Set pred.start and pred.window to 0 if using this mode (default)
`pred.start`	Alternatively. leave window.size = 0 and set pred.end as end of region of interested.

Value

Data frame specifying polygon coordinates for HiC ploting.

Examples

```
## Not run:
virtual4C(idf, 'chr16', 1000000, bin.size = 5000, window.size = 2000000)

## End(Not run)
```

Index