Package 'sevenC'

April 18, 2025

Type Package

Title Computational Chromosome Conformation Capture by Correlation of ChIP-seq at CTCF motifs

Version 1.29.0

Description Chromatin looping is an essential feature of eukaryotic genomes and can bring regulatory sequences, such as enhancers or transcription factor binding sites, in the close physical proximity of regulated target genes. Here, we provide sevenC, an R package that uses protein binding signals from ChIP-seq and sequence motif information to predict chromatin looping events. Cross-linking of proteins that bind close to loop anchors result in ChIP-seq signals at both anchor loci. These signals are used at CTCF motif pairs together with their distance and orientation to each other to predict whether they interact or not.
The resulting chromatin loops might be used to associate enhancers or transcription factor binding sites (e.g., ChIP-seq peaks) to regulated target genes.

License GPL-3

Encoding UTF-8

LazyData TRUE

Imports rtracklayer (>= 1.34.1), BiocGenerics (>= 0.22.0), GenomeInfoDb (>= 1.12.2), GenomicRanges (>= 1.28.5), IRanges (>= 2.10.3), S4Vectors (>= 0.14.4), readr (>= 1.1.0), purrr (>= 0.2.2), data.table (>= 1.10.4), boot (>= 1.3-20), methods (>= 3.4.1)

Suggests testthat, BiocStyle, knitr, rmarkdown, GenomicInteractions, covr

Depends R (>= 3.5), InteractionSet (>= 1.2.0)

RoxygenNote 6.1.0.9000

URL https://github.com/ibn-salem/sevenC

biocViews DNA3DStructure, ChIPchip, Coverage, DataImport, Epigenetics, FunctionalGenomics, Classification, Regression, ChIPSeq, HiC, Annotation

VignetteBuilder knitr

BugReports https://github.com/ibn-salem/sevenC/issues

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

git_branch devel

git_last_commit 2fa3573

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-04-17

Author Jonas Ibn-Salem [aut, cre]

Maintainer Jonas Ibn-Salem <jonas.ibn-salem@tron-mainz.de>

Contents

addCor . . . 2
addCovCor . . . 3
addCovToGR . . . 5
addInteractionSupport . . . 6
addMotifScore . . . 7
addStrandCombination . . . 8
cutoffBest10 . . . 9
cutoffByTF . . . 9
getCisPairs . . . 10
getOutOfBound . . . 10
modelBest10Avg . . . 11
motif.hg19.CTCF . . . 12
motif.hg19.CTCF.chr22 . . . 13
motif.hg19.CTCF.chr22.cov . . . 13
noZeroVar . . . 14
parseLoopsRao . . . 14
parseLoopsTang . . . 15
predLogit . . . 16
predLoops . . . 16
prepareCisPairs . . . 17
sevenC . . . 18
slideMean . . . 19
TFspecificModels . . . 19

Index **21**

`addCor` *Add correlation of ChIP-seq coverage to motif pairs.*

Description

This function first adds ChIP-seq signals along all regions of motif location using the function `addCovToGR`. Than it calculates the correlation of coverage for each input pair using the function `addCovCor`. The Pearson correlation coefficient is added as new metadata column to the input interactions. Note, this function does not work on windows because reading of bigWig files is currently not supported on windows.

Usage

```
addCor(gi, bwFile, name = "chip", window = 1000, binSize = 1)
```

Arguments

gi	GInteractions object.
bwFile	File path or connection to BigWig file with ChIP-seq signals.
name	Character indicating the sample name.
window	Numeric scalar for window size around the center of ranges in gr.
binSize	Integer scalar as size of bins to which the coverage values are combined.

Value

An GInteractions object like gi with a new metadata column colname holding Pearson correlation coefficient of ChIP-seq signals for each anchor pair.

Examples

```
if (.Platform$OS.type != "windows") {

 # use example bigWig file of ChIP-seq signals on human chromosome 22
 exampleBigWig <- system.file("extdata",
 "GM12878_Stat1.chr22_1-30000000.bigWig", package = "sevenC")

 # use example CTCF moitf location on human chromosome 22
 motifGR <- sevenC::motif.hg19.CTCF.chr22

 # build candidate interactions
 gi <- prepareCisPairs(motifGR)


 # add ChIP-seq signals correlation
 gi <- addCor(gi, exampleBigWig)

 # use an alternative metadata column name for ChIP-seq correlation
 gi <- addCor(gi, exampleBigWig, name = "Stat1")


 # add ChIP-seq correlation for signals signals in windows of 500bp around
 # motif centers
 gi <- addCor(gi, exampleBigWig, window = 500)


 # add ChIP-seq correlation for signals in bins of 10 bp
 gi <- addCor(gi, exampleBigWig, binSize = 10)

}
```

---

addCovCor *Add correlation of anchor signals to pairs of close genomic regions.*

Description

This function adds a vector with correlation values for each input interaction. Only works for input interaction within the given maxDist distance. Note, this function does not work on windows because reading of bigWig files is currently not supported on windows.

Usage

```
addCovCor(gi, datacol = "chip", colname = "cor_chip", maxDist = NULL,
  use = "everything", method = "pearson")
```

Arguments

gi	A sorted GInteractions object.
datacol	a string matching an annotation column in regions(gi). This column is assumed to hold the same number of values for each interaction as a NumericList.
colname	A string that is used as columnname for the new column in gi.
maxDist	maximal distance of pairs in bp as numeric. If maxDist=NULL, the maximal distance is computed from input interactions gi by max(pairdist(gi)).
use	an optional character string giving a method for computing covariances in the presence of missing values. See cor for more details.
method	a character string indicating which correlation coefficient (or covariance) is to be computed. One of "pearson" (default), "kendall", or "spearman": can be abbreviated. See cor for more details.

Value

A GInteractions similar to gi just with an additional column added.

Examples

```
if (.Platform$OS.type != "windows") {

  # use internal motif data on chromosome 22
  motifGR <- sevenC::motif.hg19.CTCF.chr22

  # use example bigWig file
 exampleBigWig <- system.file("extdata",
     "GM12878_Stat1.chr22_1-30000000.bigWig", package = "sevenC")

  # add coverage from bigWig file
  motifGR <- addCovToGR(motifGR, exampleBigWig)

  # get all pairs within 1Mb
  gi <- getCisPairs(motifGR, 1e5)

  # compute correaltion of coverge for each pair
  gi <- addCovCor(gi)

  # addCovCor adds a new metadata column:
  mcols(gi)

}
```

addCovToGR *Add coverage to regions in* GRanges *object.*

Description

This function adds a NumericList of coverage (or any other signal in the input bigWig file) to each range in a GRanges object. The coverage is reported for a fixed-sized window around the region center. For regions with negative strand, the coverage vector is reversed. The coverage signal is added as new metadata column holding a NumericList object. Note, this function does not work on windows because reading of bigWig files is currently not supported on windows.

Usage

```
addCovToGR(gr, bwFile, window = 1000, binSize = 1, colname = "chip")
```

Arguments

gr	GRanges object with genomic regions. It should contain a valid seqinfo object with defined seqlengths.
bwFile	File path or connection to BigWig or wig file with coverage to parse from.
window	Numeric scalar for window size around the center of ranges in gr.
binSize	Integer scalar as size of bins to which the coverage values are combined.
colname	Character as name of the new column that is created in gr.

Value

GRanges as input but with an additional meta column containing the coverage values for each region as NumericList.

Examples

```
if (.Platform$OS.type != "windows") {

 # use example bigWig file of ChIP-seq signals on human chromosome 22
 exampleBigWig <- system.file("extdata",
 "GM12878_Stat1.chr22_1-30000000.bigWig", package = "sevenC")

 # use example CTCF moitf location on human chromosome 22
 motifGR <- sevenC::motif.hg19.CTCF.chr22

 # add ChIP-seq signals to motif regions
 motifGR <- addCovToGR(motifGR, exampleBigWig)

 # add ChIP-seq signals as column named "Stat1"
 motifGR <- addCovToGR(motifGR, exampleBigWig, colname = "Stat1")

 # add ChIP-seq signals in windows of 500bp around motif centers
 motifGR <- addCovToGR(motifGR, exampleBigWig, window = 500)

 # add ChIP-seq signals in bins of 10 bp
 motifGR <- addCovToGR(motifGR, exampleBigWig, binSize = 10)

}
```

addInteractionSupport *Add column to* GInteractions *with overlap support.*

Description

See overlap methods in InteractionSet package for more details on the overlap calculations: ?overlapsAny

Usage

```
addInteractionSupport(gi, subject, colname = "loop", ...)
```

Arguments

gi	GInteractions object
subject	another GInteractions object
colname	name of the new annotation column in gi.
...	additional arguments passed to overlapsAny.

Value

InteractionSet gi as input but with additional annotation column colname indicating whether each interaction is supported by subject or not.

Examples

```
# build example GRanges as anchors
anchorGR <- GRanges(
 rep("chr1", 4),
 IRanges(
   c(1, 5, 20, 14),
   c(4, 8, 23, 17)
 ),
 strand = c("+", "+", "+", "-"),
 score = c(5, 4, 6, 7)
)


# build example GIntreaction object
gi <- GInteractions(
 c(1, 2, 2),
 c(4, 3, 4),
 anchorGR,
 mode = "strict"
)

# build exapple support GInteractions object
exampleSupport <- GInteractions(
    GRanges("chr1", IRanges(1, 4)),
    GRanges("chr1", IRanges(15, 20))
)

# add support
```

```
gi <- addInteractionSupport(gi, subject = exampleSupport)

# Use colname argument to add support to differnt metadata column name
gi <- addInteractionSupport(gi, subject = exampleSupport, colname = "example")
```

---

addMotifScore *Add motif score of anchors.*

Description

If each anchor region (motif) has a score as annotation column, this function adds two new columns named "score_1" and "score_2" with the scores of the first and the second anchor region, respectively. Additionally, a column named "score_min" is added with holds for each interaction the minimum of "score_1" and "score_2".

Usage

```
addMotifScore(gi, scoreColname = "score")
```

Arguments

gi	GInteractions.
scoreColname	Character as name the metadata column in with motif score.

Value

The same GInteractions as gi but with three additional annotation columns.

Examples

```
# build example GRanges as anchors
anchorGR <- GRanges(
 rep("chr1", 4),
 IRanges(
   c(1, 5, 20, 14),
   c(4, 8, 23, 17)
 ),
 strand = c("+", "+", "+", "-"),
 score = c(5, 4, 6, 7)
)


# build example GIntreaction object
gi <- GInteractions(
 c(1, 2, 2),
 c(4, 3, 4),
 anchorGR,
 mode = "strict"
)

# add add motif score
gi <- addMotifScore(gi, scoreColname = "score")
```

addStrandCombination *Add combination of anchor strand orientation.*

Description

Each anchor region has a strand that is '+' or '-'. Therefore, the each interaction between two regions has one of the following strand combinations: "forward", "reverse", "convergent", or "divergent". Unstranded ranges, indicated by *, are treated as positive strand.

Usage

```
addStrandCombination(gi, colname = "strandOrientation")
```

Arguments

gi	GInteractions
colname	name of the new column that is created in gi.

Value

The same GInteractions as gi but with an additional column indicating the four possible combinations of strands "forward", "reverse", "convergent", or "divergent".

Examples

```
# build example GRanges as anchors
anchorGR <- GRanges(
 rep("chr1", 4),
 IRanges(
   c(1, 5, 20, 14),
   c(4, 8, 23, 17)
 ),
 strand = c("+", "+", "+", "-"),
 score = c(5, 4, 6, 7)
)


# build example GIntreaction object
gi <- GInteractions(
 c(1, 2, 2),
 c(4, 3, 4),
 anchorGR,
 mode = "strict"
)

# add combination of anchor strands as new metadata column
gi <- addStrandCombination(gi)

# build small matrix to check strand combination
cbind(
as.character(strand(anchors(gi, "first"))),
as.character(strand(anchors(gi, "second"))),
mcols(gi)[, "strandOrientation"]
```

)

`cutoffBest10` *Default optimal cutoff value of logistic regression.*

Description

This value is the average optimal cutoff value on the 10 best performing TF ChIP-seq data sets. It is used as default cutoff value on the logistic regression response score in `predLoops` function See `?'cutoffByTF'` for more details.

Usage

```
cutoffBest10
```

Format

An object of class `numeric` of length 1.

See Also

`cutoffByTF`, `modelBest10Avg`

`cutoffByTF` *Optimal cutoff values for logistic regression models.*

Description

This dataset contains optimal cutoff scores for the response value of logistic regression models. The cutoff is based on optimal F1-scores. A separate model was trained For each of 124 TF ChIP-seq datasets in human GM12878 cells. The model performance was calculated with Hi-C and ChIA-PET interactions using 10-fold cross-validation.

Usage

```
cutoffByTF
```

Format

An object of class `tbl_df` with 121 rows and 3 columns:

TF Transcription factor name

max_cutoff The optimal cutoff on the logistic regression response value

max_f1 The optimal f1-score associated to the `max_cutoff` value

See Also

`modelBest10Avg` and `TFspecificModels`

getCisPairs — *Build a GInteractions object with all pairs of input GRanges within a given distance.*

Description

Distance is calculated from the center of input regions.

Usage

```
getCisPairs(inGR, maxDist = 1e+06)
```

Arguments

inGR	GRanges object of genomic regions. The ranges should be sorted according to chr, strand, and start position. Use sort to sort it.
maxDist	maximal distance in base-pairs between pairs of ranges as single numeric value.

Value

A GInteractions object with all pairs within the given distance.

Examples

```
# build example GRanges as input
inGR <- GRanges(
rep("chr1", 5),
IRanges(
  c(10, 20, 30, 100, 1000),
  c(15, 25, 35, 105, 1005)
)
)

# get all pairs within 50 bp
gi <- getCisPairs(inGR, maxDist = 50)

# getCisPiars returns a StrictGInteractions object
class(gi)

# The input regions are accessibly via regions()
regions(gi)
```

getOutOfBound — *Get out of chromosomal bound ranges.*

Description

Get out of chromosomal bound ranges.

Usage

```
getOutOfBound(gr)
```

Arguments

gr	A GRanges object.

Value

A data.frame with rows for each range in gr that extends out of chromosomes. The first column holds the index of the range in gr, the second the size of the overlap to the left of the chromosome and the third the size of the overlap to the right of the chromosome.

---

modelBest10Avg *Default parameters for logistic regression model in sevenC.*

Description

This dataset contains term names and estimates for logistic regression model to predict chromatin looping interactions. The estimate represent an average of the 10 best performing models out of 124 transcription factor ChIP-seq data sets from ENCODE.

Usage

```
modelBest10Avg
```

Format

An object of class data.frame with 7 rows and 2 columns holding the term name and estimate.

(Intercept) The intercept of the logistic regression model.

dist The genomic distance between the centers of motifs in base pairs (bp).

strandOrientationdivergent Orientation of motif pairs. 1 if divergent 0 if not.

strandOrientationforward Orientation of motif pairs. 1 if forward 0 if not.

strandOrientationreverse Orientation of motif pairs. 1 if reverse 0 if not.

score_min Minimum of motif hit score between both motifs in pair. The motif score is defined as -log_10 of the p-value of the motif hit as reported by JASPAR motif tracks. The unit is -log_10(p) where p is the p-value of the motif hit.

cor Pearson correlation coefficient of ChIP-seq signals across +/- 500 bp around CTCF motif centers.

Details

Each of 124 transcription factor (TF) ChIP-seq data sets from ENCODE in GM12878 cells were used to train a logistic regression model. All CTCF motifs in motif.hg19.CTCF within a distance of 1 Mb were used as candidates. A given pair was labled as true loop interactions, if it has interaction support based on loops from Hi-C in human GM12878 cells from Rao et al. 2014 or ChIA-PET loops from Tang et al. 2015 in the same cell type. The 10 best performing models were selected based on the average area under the precision-recall-curve in 10-fold cross-validation. The parameters were than averaged across the 10 best performig models.

References

Suhas S.P. Rao, Miriam H. Huntley, Neva C. Durand, Elena K. Stamenova, Ivan D. Bochkov, James T. Robinson, Adrian L. Sanborn, Ido Machol, Arina D. Omer, Eric S. Lander, Erez Lieberman Aiden, A 3D Map of the Human Genome at Kilobase Resolution Reveals #' Principles of Chromatin Looping, Cell, Volume 159, Issue 7, 18 December 2014, Pages 1665-1680, ISSN 0092-8674, https://doi.org/10.1016/j.cell.2014.11.021.

Zhonghui Tang, Oscar Junhong Luo, Xingwang Li, Meizhen Zheng, Jacqueline Jufen Zhu, Przemyslaw Szalaj, Pawel Trzaskoma, Adriana Magalska, Jakub Wlodarczyk, Blazej Ruszczycki, Paul Michalski, Emaly Piecuch, Ping Wang, Danjuan Wang, Simon Zhongyuan Tian, May Penrad-Mobayed, Laurent M. Sachs, Xiaoan Ruan, Chia-Lin Wei, Edison T. Liu, Grzegorz M. Wilczynski, Dariusz Plewczynski, Guoliang Li, Yijun Ruan, CTCF-Mediated Human 3D Genome Architecture Reveals Chromatin Topology for Transcription, Cell, Volume 163, Issue 7, 17 December 2015, Pages 1611-1627, ISSN 0092-8674, https://doi.org/10.1016/j.cell.2015.11.024.

See Also

cutoffBest10 and TFspecificModels

`motif.hg19.CTCF` *CTCF motif locations in human genome hg19.*

Description

A dataset containing the motif hits of the CTCF recognition motif from JASPAR database (MA0139.1, http://jaspar.genereg.net/matrix/MA0139.1/) in human genome assembly hg19.

Usage

```
motif.hg19.CTCF
```

Format

GRanges object with 38774 ranges on positive and negative strand with 1 meta column:

score The significance socre of the motif hit, defined as -log_10(p-value).

Details

The dataset was downloaded from JASPAR 2018 motif tracks from the following URL: http://expdata.cmmt.ubc.ca/JASPAR/downloads/UCSC_tracks/2018/hg19/tsv/MA0139.1.tsv.gz

Motif locations were filtered to contain only motif hists with p-value $\leq 2.5 * 10^-6$. The p-value is the motif hit significance as repoted from the motif scanning alogrithim used during construction of the JASPAR motif tracks. More information on the JASPAR motif track pipeline can be found here: https://github.com/wassermanlab/JASPAR-UCSC-tracks.

Source

http://expdata.cmmt.ubc.ca/JASPAR/downloads/UCSC_tracks/2018/hg19/tsv/MA0139.1.tsv.gz

motif.hg19.CTCF.chr22 *CTCF motif locations on chromosome 22 in human genome hg19.*

Description

A dataset containing the motif hits of the CTCF recognition motif from JASPAR database (MA0139.1) in human genome assembly hg19. Only motifs with a p-value $\leq 2.5 * 10^-6$ on chromosome 22 are reported.

Usage

```
motif.hg19.CTCF.chr22
```

Format

An object of class GRanges of length 917.

Details

See '?motif.hg19.CTCF' for a more details and the full data set.

See Also

motif.hg19.CTCF

motif.hg19.CTCF.chr22.cov *CTCF motifs on human chromosome 22 with example coverage.*

Description

This dataset is the same as motif.hg19.CTCF.chr22 but with one additional metadata colum, named "chip", holding ChIP-seq signals for all motifs in a windows of 1000 bp around the motif center as NumericList. The data is from a ChIP-seq experiment for STAT1 in human GM12878 cells. The full bigWig file can be downloaded from ENCODE (Dunham et al. 2012) http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeSydhTfbs/wgEncodeSydhTfbsGm12878Stat1S bigWig. See motif.hg19.CTCF and motif.hg19.CTCF.chr22 for a more details and the motif data set.

Usage

```
motif.hg19.CTCF.chr22.cov
```

Format

An object of class GRanges of length 917.

See Also

motif.hg19.CTCF, motif.hg19.CTCF.chr22

noZeroVar *returns indices of columns with non-zero variance*

Description

returns indices of columns with non-zero variance

Usage

```
noZeroVar(dat)
```

Arguments

dat	data.frame or matrix

Value

column indices of columns with non-zero variance

parseLoopsRao *Parse chromatin loops from Rao et al. 2014 as strict GInteractions.*

Description

Parse chromatin loops from Rao et al. 2014 as strict GInteractions.

Usage

```
parseLoopsRao(inFile, ...)
```

Arguments

inFile	input file with loops
...	additional arguments, that will be passed to GRanges functions.

Value

GInteractions with loops from input file.

Examples

```
# use example loop file
exampleLoopFile <- system.file("extdata",
  "GM12878_HiCCUPS.chr22_1-30000000.loop.txt", package = "sevenC")

# read loops form example file:
gi <- parseLoopsRao(exampleLoopFile)


# read loops with custom seqinfo object:
customSeqInfo <- Seqinfo(seqnames = c("chr1", "chr22"),
```

```
   seqlengths = c(10^8, 10^8), isCircular = c(FALSE, FALSE),
   genome = "custom")
gi <- parseLoopsRao(exampleLoopFile, seqinfo = customSeqInfo)
```

`parseLoopsTang` *Parse chromatin interactions from Tang et al. 2015 as* `GInteractions`.

Description

Reads pairwise ChIA-PET interaction from an input file.

Usage

```
parseLoopsTang(inFile, ...)
```

Arguments

`inFile`	input file with loops
`...`	additional arguments, that will be passed to `GRanges` functions.

Details

It reads files with the following tab-delimited format:

chr12	48160351	48161634	chr12	48230665	48232848	27
chr7	77284664	77285815	chr7	77388242	77388928	7
chr4	128459961	128460166	chr4	128508304	128509082	4

This file format was used for ChIA-PET interaction data by Tang et al. 2015 http://dx.doi.org/10.1016/j.cell.2015.11.024. The last column of input file is added as annotation column with colname "score".

Value

An `GInteractions` with loops from input file.

Examples

```
exampleLoopTang2015File <- system.file("extdata",
   "ChIA-PET_GM12878_Tang2015.chr22_1-30000000.clusters.txt",
   package = "sevenC")

gi <- parseLoopsTang(exampleLoopTang2015File)

# read loops with custom seqinfo object:
customSeqInfo <- Seqinfo(seqnames = c("chr1", "chr22"),
   seqlengths = c(10^8, 10^8), isCircular = c(FALSE, FALSE),
   genome = "custom")
gi <- parseLoopsTang(exampleLoopTang2015File, seqinfo = customSeqInfo)
```

predLogit *Predict interaction probability using logistic regression model.*

Description

Predict interaction probability using logistic regression model.

Usage

```
predLogit(data, formula, betas)
```

Arguments

data	A data.frame like object with predictor variables.
formula	A formula. All predictor variables should be available in the data frame.
betas	A vector with parameter estimates for predictor variables. They should be in the same order as variables in formula.

Value

A numeric vector with interaction probabilities for each observation in df. NAs are produced for NAs in df.

predLoops *Predict looping interactions.*

Description

This function takes a GInteractions object with candidate looping interactions. It should be annotated with features in metadata columns. A logistic regression model is applied to predict looping interaction probabilities.

Usage

```
predLoops(gi, formula = NULL, betas = NULL, colname = "pred",
  cutoff = get("cutoffBest10"))
```

Arguments

gi	A GInteractions object with coverage correlation and genomic features in metadata columns. See prepareCisPairs and addCor to build it.
formula	A formula. All predictor variables should be available in the in metadata columns of gi. If NULL, the following default formula is used: ~ dist + strandOrientation + score_min + chip.
betas	A vector with parameter estimates for predictor variables. They should be in the same order as variables in formula. Per default estimates of modelBest10Avg are used. See ?modelBest10Avg for more detailed information on each parameter.

colname — A character as column name of new metadata column in gi for predictions.

cutoff — Numeric cutoff on prediction score. Only interactions with interaction probability >= cutoff are reported. If NULL, all input interactions are reported. Default is cutoffBest10, an optimal cutoff based on F1-score on 10 best performing transcription factor ChIP-seq data sets. See ?cutoffBest10 for more details.

Value

A GInteractions as gi with an additional metadata column holding the predicted looping probability.

See Also

prepareCisPairs, addCor

Examples

```
# use example CTCF moitf location on human chromosome 22 with chip coverage
motifGR <- sevenC::motif.hg19.CTCF.chr22.cov

# build candidate interactions
gi <- prepareCisPairs(motifGR)

# add ChIP-seq signals correlation
gi <- addCovCor(gi)

# predict chromatin looping interactions
loops <- predLoops(gi)

# add prediction score for all candidates without filter
gi <- predLoops(gi, cutof = NULL)

# add prediction score using custom column name
gi <- predLoops(gi, cutof = NULL, colname = "my_colname")

# Filter loop predictions on custom cutoff
loops <- predLoops(gi, cutoff = 0.4)

# predict chromatin looping interactions using custom model parameters
myParams <- c(-4, -5, -2, -1, -1, 5, 3)
loops <- predLoops(gi, betas = myParams)

# predict chromatin loops using custom model formula and params
myFormula <- ~ dist + score_min
# define parameters for intercept, dist and motif_min
myParams <- c(-5, -4, 6)
loops <- predLoops(gi, formula = myFormula, betas = myParams)
```

---

prepareCisPairs *Prepares motif pairs as GInteractions and add genomic features.*

Description

Prepares motif pairs as GInteractions and add genomic features.

Usage

```
prepareCisPairs(motifs, maxDist = 1e+06, scoreColname = "score")
```

Arguments

motifs	GRanges object with motif locations.
maxDist	maximal distance in base-pairs between pairs of ranges as single numeric value.
scoreColname	Character as name the metadata column in with motif score.

Value

An GInteractions object with motif pairs and annotations of distance, strand orientation, and motif scores.

Examples

```
# build example GRanges as anchors
anchorGR <- GRanges(
 rep("chr1", 4),
 IRanges(
   c(1, 5, 20, 14),
   c(4, 8, 23, 17)
 ),
 strand = c("+", "+", "+", "-"),
 score = c(5, 4, 6, 7)
)



# prepare candidates
gi <- prepareCisPairs(anchorGR)


# prepare candidates using a mimial distance of 10 bp
gi <- prepareCisPairs(anchorGR, maxDist = 10)

# prepare candidates using an alternative score value in anchors
anchorGR$myScore <- rnorm(length(anchorGR))
gi <- prepareCisPairs(anchorGR, scoreColname = "myScore")
```

---

sevenC *Computational chromosome conformation capture by correlation of ChIP-seq at CTCF motifs (7C)*

Description

Chromatin looping is an essential feature of eukaryotic genomes and can bring regulatory sequences, such as enhancers or transcription factor binding sites, in the close physical proximity of regulated target genes. Here, we provide sevenC, an R package that uses protein binding signals from ChIP-seq and sequence motif information to predict chromatin looping events. Cross-linking of proteins that bind close to loop anchors result in ChIP-seq signals at both anchor loci. These signals are used at CTCF motif pairs together with their distance and orientation to each other to predict whether they interact or not. The resulting chromatin loops might be used to associate enhancers or transcription factor binding sites (e.g., ChIP-seq peaks) to regulated target genes.

`slideMean` *Sliding mean over x of intervals of size k*

Description

Source: http://stats.stackexchange.com/questions/3051/mean-of-a-sliding-window-in-r

Usage

```
slideMean(x, k)
```

Arguments

x	numeric vector
k	interval size

Value

numeric vector of length `length(x) / k`.

`TFspecificModels` *TF specific parameters for logistic regression in sevenC*

Description

sevenC was trained on 124 TF ChIP-seq data sets from ENCODE. Specific parameters are provided in this data set.

Usage

```
TFspecificModels
```

Format

A `data.frame` with 868 rows and 7 columns.

TF TF name used in ChIP-seq experiment.
file_accession File accession ID from ENCODE project
term Model term name. See `modelBest10Avg` for more detials.
estimate_mean Mean parameter estimate in 10-fold cross-validation
estimate_median Median parameter estimate in 10-fold cross-validation
estimate_sd Standard deviation of parameter estimate in 10-fold cross-validation

See Also

modelBest10Avg and cutoffByTF

Index

∗ **datasets**
  cutoffBest10, 9
  cutoffByTF, 9
  modelBest10Avg, 11
  motif.hg19.CTCF, 12
  motif.hg19.CTCF.chr22, 13
  motif.hg19.CTCF.chr22.cov, 13
  TFspecificModels, 19

addCor, 2, *16, 17*
addCovCor, *2*, 3
addCovToGR, *2*, 5
addInteractionSupport, 6
addMotifScore, 7
addStrandCombination, 8

cor, *4*
cutoffBest10, 9, *12*, *17*
cutoffByTF, *9*, 9, *20*

formula, *16*

getCisPairs, 10
getOutOfBound, 10
GInteractions, *3, 4*, *6–8*, *10*, *14–18*
GRanges, *5*, *10*, *12*, *14, 15*, *18*

InteractionSet, *6*

modelBest10Avg, *9*, 11, *16*, *19, 20*
motif.hg19.CTCF, *11*, 12, *13*
motif.hg19.CTCF.chr22, *13*, 13
motif.hg19.CTCF.chr22.cov, 13

noZeroVar, 14
NumericList, *5*, *13*

overlapsAny, *6*

parseLoopsRao, 14
parseLoopsTang, 15
predLogit, 16
predLoops, *9*, 16
prepareCisPairs, *16, 17*, 17

sevenC, 18
sevenC-package *(*sevenC*)*, 18
slideMean, 19
sort, *10*

TFspecificModels, *9*, *12*, 19