Package 'nearBynding'

July 18, 2025

Type Package

Title Discern RNA structure proximal to protein binding

Version 1.19.0

Description Provides a pipeline to discern RNA structure at and proximal to the site of protein binding within regions of the transcriptome defined by the user. CLIP protein-binding data can be input as either aligned BAM or peak-called bedGraph files. RNA structure can either be predicted internally from sequence or users have the option to input their own RNA structure data. RNA structure binding profiles can be visually and quantitatively compared across multiple formats.

License Artistic-2.0

biocViews Visualization, MotifDiscovery, DataRepresentation, StructuralPrediction, Clustering, MultipleComparison

Encoding UTF-8

LazyData true

Depends R (>= 4.0)

Imports R.utils, matrixStats, plyranges, transport, Rsamtools, S4Vectors, grDevices, graphics, rtracklayer, dplyr, GenomeInfoDb, methods, GenomicRanges, utils, stats, magrittr, TxDb.Hsapiens.UCSC.hg19.knownGene, TxDb.Hsapiens.UCSC.hg38.knownGene, ggplot2, gplots, BiocGenerics, rlang

Suggests knitr, rmarkdown

SystemRequirements bedtools (>= 2.28.0), Stereogene (>= v2.22), CapR (>= 1.1.1)

VignetteBuilder knitr

Collate 'assessGrouping.R' 'bindingContextDistance.R' 'bindingContextDistanceCapR.R' 'CleanBAMtoBG.R' 'CleanBEDtoBG.R' 'ExtractTranscriptomeSequence.R' 'GenomeMappingToChainFile.R' 'get_outfiles.R' 'liftOverToExomicBG.R' 'processCapRout.R' 'runCapR.R' 'runStereogene.R' 'runStereogeneOnCapR.R' 'visualizeCapRStereogene.R' 'visualizeStereogene.R' 'write_config.R' 'write_fasta.R' 'getChainChrSize.R' 'utilities.R' 'symmetryCapR.R' 'symmetryContext.R'

RoxygenNote 7.1.1

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

git_branch devel

git_last_commit 43ea797

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-07-17

Author Veronica Busa [cre]

Maintainer Veronica Busa <vbusa1@jhmi.edu>

Contents

`assessGrouping` *assessGrouping*

Description

Assess grouping of samples assigned to the same category relative to random.

Usage

```
assessGrouping(
  distances,
  annotations,
  measurement = "mean",
  output = "KS.pvalue",
  ctrl_iterations = 10000
)
```

Arguments

distances	Data frame object with at least three columns where the first three columns are sample 1 name, sample 2 name, and the distance between them.
annotations	Data frame object with at least two columns where the first two columns are sample name and the category of the sample for grouping. Sample names must match sample 1 and sample 2 names in distances data frame.
measurement	The measurement for comparison between cases and controls and statistical analysis ("mean", "max", or "min). Default "mean"
output	A string denoting what information will be returned: either a list of test and control measurement distances ("measurements"), the p-value of the Kolmogorov-Smirnov test comparing test and control distributions ("KS.pvalue"), or a ggplot object plotting the test and control distributions ("plot"). Default "KS.pvalue"
ctrl_iterations	The number of iterations to test for the control distribution; an integer. Default 10000.

Value

output = "KS.pvalue"	the p-value of the Kolmogorov-Smirnov test comparing test and control distributions
output = "plot"	a ggplot object plotting the test and control distributions
output = "measurements"	a list or test and control measurement distances

Examples

```
## create random distance data frame
dist<-expand.grid(letters, letters)
dist$distance<-rnorm(nrow(dist))
annot<-data.frame(sample<-letters, category<- rep(1:13, 2))
## get KS p-value
assessGrouping(dist, annot)
## get plot of test vs control distributions
assessGrouping(dist, annot,
               output = "plot")
```

bindingContextDistance *bindingContextDistance*

Description

Calculate the Wasserstein distance between two replicates' or two proteins' binding contexts for CapR-generated RNA contexts.

Usage

```
bindingContextDistance(
  dir_stereogene_output = ".",
  RNA_context,
  protein_file,
  protein_file_input = NULL,
  dir_stereogene_output_2 = NULL,
  RNA_context_2 = NULL,
  protein_file_2 = NULL,
  protein_file_input_2 = NULL,
  range = c(-200, 200)
)
```

Arguments

dir_stereogene_output	Directory of Stereogene output for first protein. Default current directory.
RNA_context	Name of the RNA context file input to Stereogene. File names must exclude extensions such as ".bedGraph". Requred
protein_file	A vector of at least one protein file name to be averaged for calculation of distance. File names must exclude extensions such as ".bedGraph". All files in the list should be experimental/biological replicates. Required.
protein_file_input	A protein file name of background input to be subtracted from protein_file signal. File name must exclude extension. Only one input file is permitted. Optional.
dir_stereogene_output_2	Directory of Stereogene output for second protein. Default dir_stereogene_output.
RNA_context_2	Name of the RNA context file input to Stereogene. File names must exclude extensions such as ".bedGraph". Default same as RNA_context.
protein_file_2	Similar to protein_file. A second vector of at least one protein file name to be averaged for calculation of distance. File names must exclude extensions such as ".bedGraph". All files in the list should be experimental/biological replicates. Default same as protein_file
protein_file_input_2	Similar to protein_file_input. A second protein file name of background input to be subtracted from protein_file_2 signal. File name must exclude extension. Only one input file is permitted. Optional.

range — A vector of two integers denoting the range upstream and downstream of the center of protein binding to consider in the comparison. Ranges that are too small miss the holistic binding context, while large ranges amplify distal noise in the binding data. Cannot exceed wSize/2 from write_config. Default c(-200, 200)

Value

Wasserstein distance between the two protein file sets provided for the RNA structure context specified, minus the input binding signal if applicable

Note

Either RNA_context_2 or protein_file_2 must be input. Otherwise, the distance would be calculated between the same file and equal 0.

Wasserstein distance calculations are reciprocal, so it does not matter which protein is first or second so long as replicates and input files correspond to one another.

Examples

```
## pull example files
get_outfiles()
## distance between stem and hairpin contexts
bindingContextDistance(RNA_context = "chr4and5_3UTR_stem_liftOver",
                       protein_file = "chr4and5_liftOver",
                       RNA_context_2 = "chr4and5_3UTR_hairpin_liftOver")

## distance between internal and hairpin contexts
bindingContextDistance(RNA_context = "chr4and5_3UTR_internal_liftOver",
                       protein_file = "chr4and5_liftOver",
                       RNA_context_2 = "chr4and5_3UTR_hairpin_liftOver")
```

---

bindingContextDistanceCapR

bindingContextDistanceCapR

Description

Calculate the Wasserstein distance between two replicates' or two proteins' binding contexts.

Usage

```
bindingContextDistanceCapR(
  dir_stereogene_output = ".",
  CapR_prefix = "",
  protein_file,
  protein_file_input = NULL,
  dir_stereogene_output_2 = NULL,
  CapR_prefix_2 = "",
  protein_file_2,
  protein_file_input_2 = NULL,
```

```
  context = "all",
  range = c(-200, 200)
)
```

Arguments

dir_stereogene_output	Directory of Stereogene output for first protein. Default current directory.
CapR_prefix	The prefix common to CapR output files of protein_file, if applicable. Equivalent to output_prefix from runStereogeneOnCapR. Default ""
protein_file	A vector of strings with at least one protein file name to be averaged for calculation of distance. File names must exclude extensions such as ".bedGraph". All files in the list should be experimental/biological replicates. Required.
protein_file_input	A protein file name of background input to be subtracted from protein_file signal. File name must exclude extension. Only one input file is permitted. Optional.
dir_stereogene_output_2	Directory of Stereogene output for second protein. Default current directory.
CapR_prefix_2	The prefix common to CapR output files of protein_file_2, if applicable.Equivalent to output_prefix from r unStereogeneOnCapR. Default ""
protein_file_2	Similar to protein_file. A second vector of at least one protein file name to be averaged for calculation of distance. File names must exclude extensions such as ".bedGraph". All files in the list should be experimental/biological replicates. Required.
protein_file_input_2	Similar to protein_file_input. A second protein file name of background input to be subtracted from protein_file_2 signal. File name must exclude extension. Only one input file is permitted. Optional.
context	The RNA structure context being compared for the two protein file sets. Acceptable contexts include "all", which sums the distance of all six contexts, or any of the contexts individually ("bulge", "hairpin", "stem", "exterior", "multibranch", or "internal"). Default "all"
range	A vector of two integers denoting the range upstream and downstream of the center of protein binding to consider in the comparison. Ranges that are too small miss the holistic binding context, while large ranges amplify distal noise in the binding data. Cannot exceed wSize/2 from write_config. Default c(-200, 200)

Value

Wasserstein distance between the two protein file sets provided for the RNA structure context specified, minus the input binding signal if applicable

Note

Wasserstein distance calculations are reciprocal, so it does not matter which protein is first or second so long as replicates and input files correspond to one another.

Examples

```
## load example StereoGene output
get_outfiles()

## This boring example compares a protein's binding with itself for all
## contexts, therefore the distance is 0
bindingContextDistanceCapR(CapR_prefix = "chr4and5_3UTR",
                           protein_file = "chr4and5_liftOver",
                           CapR_prefix_2 = "chr4and5_3UTR",
                           protein_file_2 = "chr4and5_liftOver")
```

CleanBAMtoBG *CleanBAMtoBG*

Description

Writes a script to convert a BAM file to a clean bedGraph file.

Usage

```
CleanBAMtoBG(in_bam, out_bedGraph = NA, unwanted_chromosomes = NULL)
```

Arguments

in_bam	Name of sorted BAM file to be converted to a bedGraph file. Required.
out_bedGraph	Name of bedGraph output file, including full directory path. Default in_bam prefix.
unwanted_chromosomes	A vector of unwanted chromosomes that are present in the BAM file.

Value

deposits bedGraph from BAM in same directory

Examples

```
bam <- system.file("extdata/chr4and5.bam", package="nearBynding")
#sort BAM first
sorted_bam<-Rsamtools::sortBam(bam, "chr4and5_sorted")
CleanBAMtoBG(in_bam = sorted_bam)


    ## If BAM has unwanted chromosome "EBV"
    ## this file is from ENCODE database
    CleanBAMtoBG(in_bam = "ENCFF288LEG.bam",
                 unwanted_chromosomes = "EBV")
```

CleanBEDtoBG *CleanBEDtoBG*

Description

Writes a script to convert a BED file to a clean bedGraph file.

Usage

```
CleanBEDtoBG(
  in_bed,
  out_bedGraph = NA,
  unwanted_chromosomes = NULL,
  alignment = "hg19"
)
```

Arguments

in_bed	Name of sorted BAM file to be converted to a bedGraph file. Required.
out_bedGraph	Name of bedGraph output file, including full directory path; a string. Default in_bam prefix.
unwanted_chromosomes	A vector of unwanted chromosomes that are present in the BAM file.
alignment	The human genome alignment used, either "hg19" or "hg38". Default "hg19"

Value

deposits bedGraph from BED in same directory

Examples

```
bam <- system.file("extdata/chr4and5.bam", package="nearBynding")
out_bed <- "bamto.bed"
## convert BAM to BED
if(suppressWarnings(system2("bedtools", "--version",
stdout = NULL, stderr = NULL)) == 0){
    system2("bedtools", paste0("bamtobed -i ", bam, " > ", out_bed))
}
CleanBEDtoBG(in_bed = out_bed,
    alignment = "hg38")
```

ExtractTranscriptomeSequence
ExtractTranscriptomeSequence

Description

Writes a FASTA file of transcript sequences from a list of transcripts.

Usage

```
ExtractTranscriptomeSequence(
  transcript_list,
  ref_genome,
  genome_gtf,
  RNA_fragment = "exon",
  exome_prefix = "exome"
)
```

Arguments

transcript_list	A vector of transcript names that represent the most expressed isoform of their respective genes and correspond to GTF annotation names. Required
ref_genome	The name of the reference genome FASTA from which exome sequences will be derived; a string. Required
genome_gtf	The name of the GTF/GFF file that contains all exome annotations; a string. Coordinates must match the file input for the ref_genome parameter. Required
RNA_fragment	A string of RNA component of interest. Options depend on the gtf file but often include "gene", "transcript", "exon", "CDS", "five_prime_utr", and/or "three_prime_utr". Default "exon" for the whole exome.
exome_prefix	A string to add to the prefix for all output files. Default "exome"

Value

writes FASTA file of transcriptome sequences into directory

Note

transcript_list, genome_gtf, and RNA_fragment arguments should be the same as GenomeMappingToChainFile function arguments

Examples

```
## load transcript list
load(system.file("extdata/transcript_list.Rda", package="nearBynding"))
##get GTF file
gtf<-system.file("extdata/Homo_sapiens.GRCh38.chr4&5.gtf",
                 package="nearBynding")
ExtractTranscriptomeSequence(transcript_list = transcript_list,
             ref_genome = "Homo_sapiens.GRCh38.dna.primary_assembly.fa",
             genome_gtf = gtf,
```

```
                RNA_fragment = "three_prime_utr",
                exome_prefix = "chr4and5_3UTR")
```

GenomeMappingToChainFile *GenomeMappingToChainFile*

Description

Writes a chain file mapped from a genome annotation file.

Usage

```
GenomeMappingToChainFile(
  genome_gtf,
  out_chain_name,
  RNA_fragment = "exon",
  transcript_list,
  chrom_suffix = "exome",
  verbose = FALSE,
  alignment = "hg19",
  check_overwrite = FALSE
)
```

Arguments

genome_gtf	The name of the GTF/GFF file that will be converted to an exome chain file. Required
out_chain_name	The name of the chain file to be created. Required
RNA_fragment	RNA component of interest. Options depend on the gtf file but often include "gene", "transcript", "exon", "CDS", "five_prime_utr", and/or "three_prime_utr". Default "exon" for the whole exome.
transcript_list	A vector of transcript names that represent the most expressed isoform of their respective genes and correspond to gtf annotation names. Isoforms cannot overlap. Required
chrom_suffix	The suffix to be appended to all chromosome names created in the chain file. Default "exome"
verbose	Output updates while the function is running. Default FALSE
alignment	The human genome alignment used, either "hg19" or "hg38". Default "hg19"
check_overwrite	Check for file wth the same out_chain_name before writing new file. Default FALSE.

Value

writes a chain file into directory

Examples

```
## load transcript list
load(system.file("extdata/transcript_list.Rda", package="nearBynding"))
## get GTF file
gtf<-system.file("extdata/Homo_sapiens.GRCh38.chr4&5.gtf",
                 package="nearBynding")

GenomeMappingToChainFile(genome_gtf = gtf,
                         out_chain_name = "test.chain",
                         RNA_fragment = "three_prime_utr",
                         transcript_list = transcript_list,
                         alignment = "hg38")
```

`getChainChrSize` *getChainChrSize*

Description

Output a table of mapped chromosome names and lengths from a chain file.

Usage

```
getChainChrSize(chain, out_chr)
```

Arguments

`chain`	The name of the chain file for which chromosome sizes should be determined and output; a string. Required.
`out_chr`	Name of the chromosome names and lengths table file; a string. Required.

Value

writes a two-column tab-delineated file without a header containing chromosome names and lengths for a given chain file

Examples

```
## first, make the chain file
load(system.file("extdata/transcript_list.Rda", package="nearBynding"))
gtf<-system.file("extdata/Homo_sapiens.GRCh38.chr4&5.gtf",
                 package="nearBynding")
GenomeMappingToChainFile(genome_gtf = gtf,
                         out_chain_name = "test.chain",
                         RNA_fragment = "three_prime_utr",
                         transcript_list = transcript_list,
                         alignment = "hg38")

getChainChrSize(chain = "test.chain",
                out_chr = "chr4and5_3UTR.size")
```

get_outfiles *get_outfiles*

Description

Copy files necessary to complete the vignette onto the local machine in cases where Stereogene, CapR, or bedtools are not available.

Usage

```
get_outfiles(dir = ".")
```

Arguments

dir	Directory into which files ought to be stored. Default current work directory.

Value

deposits six *.dist StereoGene output files into the selected directory

Examples

```
## pull example StereoGene output files
get_outfiles()
```

liftOverToExomicBG *liftOverToExomicBG*

Description

Lifts features such as CLIP-seq reads or RNA structure annotations from genome to transcriptome.

Usage

```
liftOverToExomicBG(input, chain, chrom_size, output_bg, format = "bedGraph")
```

Arguments

input	A single input file name or a vector of input file names in the format of c(forward_reads, reverse_reads) for strand-separated alignments. Files must be BED or bedGraph format. Required
chain	The name of the chain file to be used for liftOver. Format should be like chain files derived from getChainChrSize function. Required
chrom_size	Name of chromosome size file. File must be in two-column format without a header where first column is chromosome name and second column is chromosome length, as from liftOverToExomicBG. Required.
output_bg	The name of the lifted-over output bedGraph file. Required.
format	File type of input file(s). Recommended "BED" or "bedGraph". Default "bedGraph"

Value

writes lifted-over bedGraph file

Examples

```
## first, get chain file
load(system.file("extdata/transcript_list.Rda", package="nearBynding"))
gtf<-system.file("extdata/Homo_sapiens.GRCh38.chr4&5.gtf",
                 package="nearBynding")
GenomeMappingToChainFile(genome_gtf = gtf,
                         out_chain_name = "test.chain",
                         RNA_fragment = "three_prime_utr",
                         transcript_list = transcript_list,
                         alignment = "hg38")
## and chain file chromosome sizes
getChainChrSize(chain = "test.chain",
                out_chr = "chr4and5_3UTR.size")

## get bedGraph file
chr4and5_sorted.bedGraph<-system.file("extdata/chr4and5_sorted.bedGraph",
                                      package="nearBynding")

liftOverToExomicBG(input = chr4and5_sorted.bedGraph,
                   chain = "test.chain",
                   chrom_size = "chr4and5_3UTR.size",
                   output_bg = "chr4and5_liftOver.bedGraph")
```

`nearBynding` *Discern RNA structure proximal to protein binding*

Description

nearBynding is a package designed to discern annotated RNA structures at and proximal to the site of protein binding. It allows users to annotate RNA structure contexts via CapR or input their own annotations in BED/bedGraph format and it accomodates protein binding information from CLIP-seq experiments as either aligned CLIP-seq reads or peak-called intervals.

Details

Package:	nearBynding
Type:	Package
Title:	nearBynding package
Version:	1.3.3
Date:	June 1, 2021‘
License:	Artistic-2.0
LazyLoad:	yes
URL:	http://github.com/vbusa1/nearBynding

Author(s)

Veronica Busa <vbusa1@jhmi.edu>

References

StereoGene: Stavrovskaya, Elena D., Tejasvi Niranjan, Elana J. Fertig, Sarah J. Wheelan, Alexander V. Favorov, and An
CapR: Tsukasa Fukunaga, Haruka Ozaki, Goro Terai, Kiyoshi Asai, Wataru Iwasaki, and Hisanori Kiryu. "CapR:

See Also

See the nearBynding package vignette.

Examples

```
## Not run:

library(nearBynding)
library(Rsamtools)

# get transcript list
load(system.file("extdata/transcript_list.Rda", package="nearBynding"))
# get GTF file
gtf<-system.file("extdata/Homo_sapiens.GRCh38.chr4&5.gtf",
                 package="nearBynding")
# make chain file
GenomeMappingToChainFile(genome_gtf = gtf,
                         out_chain_name = "test.chain",
                         RNA_fragment = "three_prime_utr",
                         transcript_list = transcript_list,
                         alignment = "hg38")
# get size of chromosomes of chain file
getChainChrSize(chain = "test.chain",
                out_chr = "chr4and5_3UTR.size")

# get transcript sequences
ExtractTranscriptomeSequence(transcript_list = transcript_list,
                     ref_genome = "Homo_sapiens.GRCh38.dna.primary_assembly.fa",
                     genome_gtf = gtf,
                     RNA_fragment = "three_prime_utr",
                     exome_prefix = "chr4and5_3UTR")
# run CapR on extracted sequences
runCapR(in_file = "chr4and5_3UTR.fa")

# get BAM file of protein binding
bam <- system.file("extdata/chr4and5.bam", package="nearBynding")
# sort it and convert to bedGraph format
sorted_bam<-sortBam(bam, "chr4and5_sorted")
CleanBAMtoBG(in_bam = sorted_bam)

# lift over protein binding and RNA structure to chain
liftOverToExomicBG(input = "chr4and5_sorted.bedGraph",
                     chain = "test.chain",
                     chrom_size = "chr4and5_3UTR.size",
```

```
                    output_bg = "chr4and5_liftOver.bedGraph")
processCapRout(CapR_outfile = "chr4and5_3UTR.out",
               chain = "test.chain",
               output_prefix = "chr4and5_3UTR",
               chrom_size = "chr4and5_3UTR.size",
               genome_gtf = gtf,
               RNA_fragment = "three_prime_utr")

# input to StereoGene
runStereogeneOnCapR(protein_file = "chr4and5_liftOver.bedGraph",
                    chrom_size = "chr4and5_3UTR.size",
                    name_config = "chr4and5_3UTR.cfg",
                    input_prefix = "chr4and5_3UTR")

# visualize protein binding context
visualizeCapRStereogene(CapR_prefix = "chr4and5_3UTR",
                        protein_file = "chr4and5_liftOver",
                        heatmap = T,
                        out_file = "all_contexts_heatmap",
                        x_lim = c(-500, 500))

## End(Not run)
```

`processCapRout` *processCapRout*

Description

Creates context-separated bedGraph files of CapR output for genome and transcriptome alignments.

Usage

```
processCapRout(
  CapR_outfile,
  output_prefix,
  chrom_size,
  genome_gtf,
  RNA_fragment,
  chain
)
```

Arguments

`CapR_outfile`	Name of CapR output file. Required
`output_prefix`	Prefix string to be appended to all output files. Required.
`chrom_size`	Name of chromosome size file. File must be in two-column format without a header where first column is chromosome name and second column is chromosome length, as from getChainChrSize. Required.
`genome_gtf`	The name of the GTF/GFF file that contains all exome annotations. Required
`RNA_fragment`	RNA component of interest. Options depend on the gtf file but often include "gene", "transcript", "exon", "CDS", "five_prime_utr", and/or "three_prime_utr". Default "exon" for the whole exome.

chain — The name of the chain file to be used. Format should be like chain files derived from GRangesMappingToChainFile. Required

Value

writes bedGraph files of structure signal for each of the six CapR contexts 1) mapped to the genome and 2) lifted-over to the transcriptome

Examples

```
## make chain file
load(system.file("extdata/transcript_list.Rda", package="nearBynding"))
gtf<-system.file("extdata/Homo_sapiens.GRCh38.chr4&5.gtf",
                 package="nearBynding")
GenomeMappingToChainFile(genome_gtf = gtf,
                         out_chain_name = "test.chain",
                         RNA_fragment = "three_prime_utr",
                         transcript_list = transcript_list,
                         alignment = "hg38")
## get chromosome size file
getChainChrSize(chain = "test.chain",
                out_chr = "chr4and5_3UTR.size")

processCapRout(CapR_outfile = system.file("extdata/chr4and5_3UTR.out",
                                          package="nearBynding"),
               chain = "test.chain",
               output_prefix = "chr4and5_3UTR",
               chrom_size = "chr4and5_3UTR.size",
               genome_gtf = gtf,
               RNA_fragment = "three_prime_utr")
```

---

runCapR *runCapR*

Description

Runs CapR

Usage

```
runCapR(in_file, out_file = NA, max_dist = 100)
```

Arguments

in_file — An .fa file like from ExtractTranscriptomeSequence that is a list of fasta sequences to be folded. Required

out_file — Name of the CapR output file of nucleotide folding probabilities. Default is in_file prefix.out

max_dist — Integer of maximum distance between folded nucleotides in sequences. Recommeded between 50 and 100, with higher values yielding potentially more accurate results but dramatically increasing run time. Default 100.

Value

generates CapR outfile

Examples

```
## make dummy file
write_fasta(paste0(sample(c("A", "T", "G", "C"), 50, replace = TRUE),
                   collapse = ""),
            "test",
            "test.fa")
## run CapR
runCapR("test.fa")
```

---

runStereogene *runStereogene*

Description

Writes a StereoGene script in the working directory

Usage

```
runStereogene(track_files,
        name_config,
        pcorProfile = NULL,
        confounder = NULL,
        nShuffle = 1000,
        get_error = FALSE)
```

Arguments

track_files	Vector of at least two track or interval file names to be pairwise-correlated by StereoGene. Required.
name_config	Name of corresponding configuration file; a string. Required
pcorProfile	Name of track file name for partial correlation; a string. More information for this can be found in the StereoGene README. Optional
confounder	Confounder file name; a string. More information for this can be found in the StereoGene README. Optional
nShuffle	Permutations used to estimate error. Default 5000.
get_error	Whether to calculate the standard error of background permutations from nShuffle. FALSE will save calculation time. Default FALSE

Value

generates StereoGene output files in directory

Examples

```
runStereogene(track_files = c("chr4and5_3UTR_stem_liftOver.bedGraph",
                              "chr4and5_liftOver.bedGraph"),
              name_config = "chr4and5_3UTR.cfg")
```

runStereogeneOnCapR *runStereogeneOnCapR*

Description

Writes a configuration file and Stereogene script and runs Stereogene for all CapR tracks

Usage

```
runStereogeneOnCapR(
  dir_CapR_bg = ".",
  input_prefix,
  protein_file,
  output_prefix = input_prefix,
  name_config = "config.cfg",
  chrom_size,
  nShuffle = 100,
  get_error = FALSE,
  ...
)
```

Arguments

dir_CapR_bg	Directory of lifted-over CapR bedGraph files. Default current directory
input_prefix	Prefix string appended to input files; same as input_prefix argument in processCapRout. Required
protein_file	Name of protein file in bedGraph format. Required
output_prefix	Prefix string to be appended to all output files. Default to be same as input_prefix
name_config	Name of output config file. Default config.cfg
chrom_size	Name of chromosome size file. File must be in two-column format without a header where first column is chromosome name and second column is chromosome length, as from getChainChrSize. Required
...	includes all other parameters acceptable to write_config and write_stereogene
nShuffle	Permutations used to estimate error. Default 100.
get_error	Whether to calculate the standard error of background permutations from nShuffle. FALSE will save calculation time. Default FALSE

Value

generates StereoGene output files, including *.dist files

Examples

```
runStereogeneOnCapR(protein_file = "chr4and5_liftOver.bedGraph",
                    chrom_size = "chr4and5_3UTR.size",
                    name_config = "chr4and5_3UTR.cfg",
                    input_prefix = "chr4and5_3UTR")
```

symmetryCapR *symmetryCapR*

Description

Calculate the symmetry of a binding context.

Usage

```
symmetryCapR(
  dir_stereogene_output = ".",
  CapR_prefix = "",
  protein_file,
  protein_file_input = NULL,
  context = "all",
  range = c(-200, 200)
)
```

Arguments

dir_stereogene_output	Directory of Stereogene output for first protein. Default current directory.
CapR_prefix	The prefix common to CapR output files of protein_file, if applicable. Equivalent to output_prefix from runStereogeneOnCapR. Default ""
protein_file	A vector of strings with at least one protein file name to be averaged for calculation of distance. File names must exclude extensions such as ".bedGraph". All files in the list should be experimental/biological replicates. Required.
protein_file_input	A protein file name of background input to be subtracted from protein_file signal. File name must exclude extension. Only one input file is permitted. Optional.
context	The RNA structure context being interrogated. Acceptable contexts include "all", which sums the distance of all six contexts, or any of the contexts individually ("bulge", "hairpin", "stem", "exterior", "multibranch", or "internal"). Default "all"
range	A vector of two integers denoting the range upstream and downstream of the center of protein binding to consider in the comparison. Ranges that are too small miss the holistic binding context, while large ranges amplify distal noise in the binding data. Cannot exceed wSize/2 from write_config. Default c(-200, 200)

Value

Wasserstein distance between the two halves of the binding context, with lower values suggesting greater symmetry.

Examples

```
## load example StereoGene output
get_outfiles()

## This boring example compares a protein's binding with itself for all
## contexts, therefore the distance is 0
symmetryCapR(CapR_prefix = "chr4and5_3UTR",
                protein_file = "chr4and5_liftOver")
```

`symmetryContext` *symmetryContext*

Description

Calculate the symmetry of a binding context.

Usage

```
symmetryContext(
  dir_stereogene_output = ".",
  context_file,
  protein_file,
  protein_file_input = NULL,
  range = c(-200, 200)
)
```

Arguments

`dir_stereogene_output`	Directory of Stereogene output for protein. Default current directory.
`context_file`	Name of the RNA context file input to Stereogene. File names must exclude extensions such as ".bedGraph". Requred
`protein_file`	A vector of at least one protein file name to be averaged for calculation of distance. File names must exclude extensions such as ".bedGraph". All files in the list should be experimental/biological replicates. Required.
`protein_file_input`	A protein file name of background input to be subtracted from protein_file signal. File name must exclude extension. Only one input file is permitted. Optional.
`range`	A vector of two integers denoting the range upstream and downstream of the center of protein binding to consider in the comparison. Ranges that are too small miss the holistic binding context, while large ranges amplify distal noise in the binding data. Cannot exceed wSize/2 from write_config. Default c(-200, 200)

Value

Wasserstein distance between the two halves of the binding context, with lower values suggesting greater symmetry.

Examples

```
## load example StereoGene output
get_outfiles()

## This boring example compares a protein's binding with itself for all
## contexts, therefore the distance is 0
symmetryContext(context_file = "chr4and5_3UTR_stem_liftOver",
                        protein_file = "chr4and5_liftOver")
```

`visualizeCapRStereogene` *visualizeCapRStereogene*

Description

Creates a visual output of all CapR RNA structure contexts relative to protein binding.

Usage

```
visualizeCapRStereogene(
  dir_stereogene_output = ".",
  CapR_prefix,
  protein_file,
  protein_file_input = NULL,
  x_lim = c(-100, 100),
  y_lim = NULL,
  error = 1,
  nShuffle = 100,
  out_file = "out_file",
  legend = TRUE,
  heatmap = FALSE
)
```

Arguments

`dir_stereogene_output`	Directory of stereogene output. Default working directory.
`CapR_prefix`	The prefix string common to CapR output files of protein_file. Required.
`protein_file`	A vector of at least one protein file name to be averaged for visualization. File names must exclude extensions such as ".bedGraph". All files in the list should be experimental or biological replicates. Required.
`protein_file_input`	A protein file name of background input to be subtracted from protein_file signal. File name must exclude extension. Only one input file is permitted. Optional.
`x_lim`	A vector of two integers denoting the lower and upper x axis limits. Cannot exceed wSize/2 from write_config. Default (-100, 100)
`y_lim`	A vector of two numbers denoting the lower and upper y axis limits. Optional

error	A numeric value that determines the number of standard deviations to show in the error bar. Default 1
nShuffle	Relevant if multiple protein files are input and background error has been calculated. It is the number of iterations used to derive background signal error. Should be same for all protein files. Default 100.
out_file	Name of output file, excluding extension. ".pdf" or ".jpeg" will be added as relevant to the output file name. Default "out_file"
legend	Whether a legend should be included with the output graph. Default TRUE
heatmap	Whether the output graph should be in the form of a heatmap (TRUE) or of a line graph (FALSE). Default FALSE

Value

heatmap (JPEG) or line graph (PDF) image file

Examples

```
## pull example files
get_outfiles()
## heatmap
visualizeCapRStereogene(CapR_prefix = "chr4and5_3UTR",
                        protein_file = "chr4and5_liftOver",
                        heatmap = TRUE,
                        out_file = "all_contexts_heatmap",
                        x_lim = c(-500, 500))
## line graph
visualizeCapRStereogene(CapR_prefix = "chr4and5_3UTR",
                        protein_file = "chr4and5_liftOver",
                        x_lim = c(-500, 500),
                        out_file = "all_contexts_line",
                        y_lim = c(-18, 22))
```

`visualizeStereogene` *visualizeStereogene*

Description

Creates a visual output of a single RNA structure context relative to protein binding.

Usage

```
visualizeStereogene(
  dir_stereogene_output = ".",
  context_file,
  protein_file,
  protein_file_input = NULL,
  x_lim = c(-100, 100),
  y_lim = NULL,
  error = 3,
  nShuffle = 1000,
```

```
  out_file = "out_file",
  legend = TRUE,
  heatmap = FALSE
)
```

Arguments

dir_stereogene_output	Directory of stereogene output. Default working directory.
context_file	A single context file name for visualization with the protein_file(s). File names must exclude extensions such as ".bedGraph". Required.
protein_file	A vector of at least one protein file name to be averaged for visualization. File names must exclude extensions such as ".bedGraph". All files in the list should be experimental or biological replicates. Required.
protein_file_input	A protein file name of background input to be subtracted from protein_file signal. File name must exclude extension. Only one input file is permitted. Optional.
x_lim	A vector of two integers denoting the lower and upper x axis limits. Cannot exceed wSize/2 from write_config. Default (-100, 100)
y_lim	A vector of two numbers denoting the lower and upper y axis limits. Optional.
error	A numeric value that determines the number of standard deviations to show in the error bar. Default 3
nShuffle	Relevant if multiple protein files are input and background error has been calculated. It is the number of iterations used to derive background signal error. Should be same for all protein files. Default 1000.
out_file	Name of output file, excluding extension. ".pdf" or ".jpeg" will be added as relevant to the output file name. Default "out_file"
legend	Whether a legend should be included with the output graph. Default TRUE.
heatmap	Whether the output graph should be in the form of a heatmap (TRUE) or of a line graph (FALSE). Default FALSE

Value

heatmap (JPEG) or line graph (PDF) image file

Examples

```
## pull example files
get_outfiles()
## heatmap
visualizeStereogene(context_file = "chr4and5_3UTR_stem_liftOver",
                    protein_file = "chr4and5_liftOver",
                    out_file = "stem_heatmap",
                    x_lim = c(-500, 500))
## line graph
visualizeStereogene(context_file = "chr4and5_3UTR_stem_liftOver",
                    protein_file = "chr4and5_liftOver",
                    heatmap = TRUE,
                    out_file = "stem_line",
                    x_lim = c(-500, 500))
```

write_config *write_config*

Description

Writes a configuration file for use by Stereogenes in the working directory.

Usage

```
write_config(
  name_config = "config.cfg",
  chrom_size,
  Rscript = FALSE,
  silent = TRUE,
  na_noise = FALSE,
  bin = 1,
  threshold = 0,
  cross_width = 200,
  wSize = 10000,
  kernel_width = 1000,
  resPath = "."
)
```

Arguments

name_config	Name of output config file. Default config.cfg
chrom_size	Name of chromosome size file. File must be in two-column format without a header where first column is chromosome name and second column is chromosome length, as from getChainChrSize. Required
Rscript	Write R script for the result presentation. Equivalent to -r argument in StereoGene. Default FALSE
silent	Provides an output when Stereogene is run. Equivalent to -s or -silent argument in StereoGene. Default TRUE
na_noise	Use NA values as unknown and fill them with noise. Equivalent to -NA argument in StereoGene. Default FALSE
bin	Bin size for input averaging; an integer. Default 1
threshold	Threshold for input data to remove small values. An integer between 0 and 250. Default 0
cross_width	Width of cross-correlation plot output in Rscript; an integer. Default 200.
wSize	Window size; an integer. If windows are too small, cross correlations will have a lot of noise; if they are too large, there may be too few windows for robust statistical assessment. Default 10000
kernel_width	Kernel span in nucleotides; an integer. Equivalent to KernelSigma invStereoGene. Default 1000
resPath	Folder to store results. Default is current directory.

Value

writes a configuration file into directory

Note

Not all StereoGene parameters are included in this function so refer to the StereoGene manual and modify the output .cfg file manually if additional parameters are desired.

Examples

```
## Write a config file named "test.cfg" with chromosome size file "test.size"
write_config(name_config = "test.cfg",
             chrom_size = "test.size")
```

`write_fasta` *write_fasta*

Description

Writes a FASTA file from a vector of sequences

Usage

```
write_fasta(sequences, names, file.out)
```

Arguments

`sequences`	A vector of sequences
`names`	A vector of names corresponding to the sequences
`file.out`	Name of output FASTA file; a string

Value

writes FASTA file into directory

Examples

```
sequences<-c(paste0(sample(c("A", "T", "G", "C"), 20, replace = TRUE),
                    collapse = ""),
             paste0(sample(c("A", "T", "G", "C"), 20, replace = TRUE),
                    collapse = ""),
             paste0(sample(c("A", "T", "G", "C"), 20, replace = TRUE),
                    collapse = ""))
write_fasta(sequences,
            c("one", "two", "three"),
            "test.fa")
```

Index