Package 'DegCre'

June 4, 2026

Type Package

Title Probabilistic association of DEGs to CREs from differential data

Version 1.9.0

Date 2025-07-22

Maintainer Brian S. Roberts `<brianroberts1976@yahoo.com>`

Description DegCre generates associations between differentially expressed genes (DEGs) and cis-regulatory elements (CREs) based on non-parametric concordance between differential data. The user provides GRanges of DEG TSS and CRE regions with differential p-value and optionally log-fold changes and DegCre returns an annotated Hits object with associations and their calculated probabilities. Additionally, the package provides functionality for visualization and conversion to other formats.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.4)

URL https://github.com/brianSroberts/DegCre

Imports GenomicRanges, InteractionSet, plotgardener, S4Vectors, stats, graphics, grDevices, BiocGenerics, Seqinfo, IRanges, BiocParallel, qvalue, TxDb.Hsapiens.UCSC.hg38.knownGene, org.Hs.eg.db, utils

biocViews GeneExpression, GeneRegulation, ATACSeq, ChIPSeq, DNaseSeq, RNASeq

Suggests BSgenome, BSgenome.Hsapiens.UCSC.hg38, BiocStyle, magick, knitr, rmarkdown, TxDb.Mmusculus.UCSC.mm10.knownGene, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

BugReports https://github.com/brianSroberts/DegCre/issues

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

git_branch devel

git_last_commit 60bbe80

git_last_commit_date 2026-04-28

Repository Bioconductor 3.24

Date/Publication 2026-06-04

Author Brian S. Roberts [aut, cre] (ORCID:
<`https://orcid.org/0009-0001-2914-6826`>)

Contents

`adjustRawAssocProb` *Adjust raw association probability for DEG true positive probability*

Description

For a raw associations probabilities, this function multiplies this probability by the probability that the DEG is truly differentially expressed.

Usage

```
adjustRawAssocProb(rawAssocProbs, pAdjs, alphaVal = 0.01, method = "qvalue")
```

Arguments

`rawAssocProbs`	Raw association probabilities
`pAdjs`	Adjusted p-values of the DEG.
`alphaVal`	Chosen significance level (only used if `method` = "bonferroni")
`method`	Method for adjusting p-values. One of "bonferroni" or "qvalue". (Default: `qvalue`).

Details

Not exported. This function multiplies the raw association probabilities by the probability that the DEG is truly differential expressed. This "true DEG" probability is determined depending on the choice of `method`. For "bonferroni" the probability is 1 if the corresponding pAdj is less than or equal to `alphaVal` and 0 otherwise. For "qvalue" the probability is 1 - localFDR as calculated in lfdr.

Value

A numeric vector of adjusted association probilities

Author(s)

Brian S. Roberts

Examples

```
## Not run:
adjAssocProbs <- adjustRawAssocProb(myRawAssocProbs,pAdjs=myPadjs,
                                    alphaVal=0.01,method="qvalue")

## End(Not run)
```

`calcAssocProbOR` *Calculate Odds-ratio*

Description

Given a DegCre results list, this function calculates the odds-ratio of the association probability.

Usage

```
calcAssocProbOR(degCreResList, type = "adj")
```

Arguments

`degCreResList`	List of DegCre results.
`type`	A character of either `"raw"` or `"adj"`. (Default:`"adj"`).

Details

This function is similar to calcRawAssocProbOR and will mimic its function when `type = "raw"`. When `type = "raw"` the calculation operates on the `"rawAssocProb"` metadata. When `type = "adj"` the calculation operates on the `"assocProb"` metadata. The OR is calculated relative to the distance bin null association probability, which would happen if all CRE p-values were identical. Thus it is a measure of the increase in association probability due to CRE p-value information content over what would occur by random chance.

Value

A numeric of the association probability odds-ratios

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load test data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)
```

```
#Calculate odds-ratio.

calcOR <- calcAssocProbOR(degCreResListDexNR3C1)
```

`calcAUC` *Calculate Area Under the Curve (AUC)*

Description

Calculates the Area Under the Curve (AUC) for a given set of x and y values using the trapezoidal rule.

Usage

```
calcAUC(xVals, yVals)
```

Arguments

`xVals`	A numeric vector of x-values.
`yVals`	A numeric vector of corresponding y-values.

Details

Not exported. This function calculates the AUC for a given set of x and y values using the trapezoidal rule. It provides a measure of the area under the curve formed by the x and y values, which is often used to assess the performance of models or the shape of a curve.

Value

A numeric value representing the AUC.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
# Get AUC of quadratic curve.
x <- seq_len(10)
y <- x^2
auc <- calcAUC(x, y)

## End(Not run)
```

`calcBinomFDRperBin` *Calculate Binomial FDR per Distance Bin*

Description

Calculates the False Discovery Rate (FDR) for association probabilities within a given distance bin using a binomial distribution approach.

Usage

```
calcBinomFDRperBin(allDistBinsStatsMat, chunkI, alphaVal)
```

Arguments

`allDistBinsStatsMat`	A matrix containing statistics for all distance bins.
`chunkI`	An integer vector specifying the indices of the current distance bin.
`alphaVal`	Numeric from 0 to 1 of the DEG significance level.

Details

Not exported. This function calculates the FDR for association probabilities within a given distance bin. It uses a binomial distribution approach (via pbinom) to estimate the FDR based on the number of significant associations and the total number of associations in the bin. Additionally, it adjusts FDR values for associations with low probabilities or ties in the significance ranks. It is meant to run within runDegCre. It will not run well on unintended inputs.

Value

A numeric vector of FDR values for the specified distance bin.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
# Get FDR from binomila distribtuion.
binomFDR <- calcBinomFDRperBin(allDistBinsStatsMat = myStatsMatrix,
chunkI = myChunkIndices, alphaVal = 0.05)

## End(Not run)
```

`calcDependIndependEnrichStats`

Calculate Enrichment Statistics for Dependent and Independent Data

---

Description

This function calculates enrichment statistics for dependent and independent data based on the provided data frame and parameters.

Usage

```
calcDependIndependEnrichStats(
  hitsWithDistDf,
  subHitsIndex,
  dependPadj,
  independP,
  etpFun,
  alpha
)
```

Arguments

`hitsWithDistDf`	A DataFrame derived from a Hits object.
`subHitsIndex`	Indices of the row subsets of `hitsWithDistDf` to analyze.
`dependPadj`	Numeric vector of adjusted p-values for the dependent data.
`independP`	Numeric vector of p-values for the independent data.
`etpFun`	Function for converting sets of dependPadj to expected true postives. Passed from calcPadjsAndEtpFun.
`alpha`	Numeric significance level threshold for DEGs.

Details

Not exported. This function calculates enrichment statistics for dependent and independent data based on provided adjusted p-values for dependent data and p-values for independent data. It computes the associated probabilities, which represent the probability of observing a significant association for each set of data under the specified significance level threshold. the independent variable in DegCre calculations is the CreP and the dependent is the DEG adjusted p-values. It is meant to run within runDegCre. It will not run well on unintended inputs.

Value

A matrix containing calculated statistics for enrichment analysis, including independent p-values, associated probabilities, and the total number of observations.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
# Get stat results.
statsMatrix <- calcDependIndependEnrichStats(hitsWithDistDf = myHitsDf,
                                             subHitsIndex = mySubHits,
                                             dependPadj = myDependPadj,
                                             independP = myIndependP,
                                             etpFun = myEtpFun,
                                             alpha = 0.05)

## End(Not run)
```

`calcKStestStatMedian` *Calculate the Median KS Statistic by Distance Bin Size*

Description

Calculates the median Kolmogorov-Smirnov (KS) statistic for different distance bin sizes.

Usage

```
calcKStestStatMedian(testBinSizes, allSortHitIndices, allHitPs)
```

Arguments

`testBinSizes`	A vector of integer values specifying the distance bin sizes to be tested.
`allSortHitIndices`	A vector of sorted hit indices.
`allHitPs`	A numeric vector of p-values corresponding to the associations.

Details

Not exported. This function calculates the median Kolmogorov-Smirnov (KS) statistic for different distance bin sizes. It splits the associations into bins of specified sizes, calculates the KS statistic for each bin, and returns the median KS statistic for each bin size. The KS statistic measures the maximum difference between two cumulative distribution functions, providing a measure of the difference between CRE p-value distributions within each bin and the global distribution. It is meant to run within distBinHeuristic. It will not run well on unintended inputs.

Value

A matrix containing two columns:

distBinSize The tested distance bin sizes.

KSRMSE The median KS statistic for each distance bin size.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
# Example usage:
binSizes <- c(10000, 5000, 200)
hitIndices <- seq_len(1e5)
pValues <- runif(1e5)

resultMatrix <- calcKStestStatMedian(testBinSizes = binSizes,
                                      allSortHitIndices = hitIndices,
                                      allHitPs = pValues)

## End(Not run)
```

`calcPadjsAndEtpFun` *Convert p-values to adjusted p-values*

Description

Convert a set of p-values to adjusted pvalues with a specified method and return a function for converting set of pvalues to number of expected true positives.

Usage

```
calcPadjsAndEtpFun(pVal, method = "qvalue")
```

Arguments

`pVal`	Input p-values.
`method`	One of "bonferroni" or "qvalue". (Default: `qvalue`).

Details

Not exported. This function adjusted p-values by the specified method. It also returns a function that calculates the fraction of expected true positives from a subset of the adjusted p-values

Value

A list two slots:

pAdj Adjusted p-values.

etpFun A function for calculating the number of expected true positives from adjusted p-values.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
outPadjList <- calcPadjsAndEtpFun(myPVals,method="qvalue")

## End(Not run)
```

`calcRawAssocProbOR` *Calculate Raw Association Probability Odds Ratio (OR)*

Description

Given a DegCre results list, this function calculates the raw association probability odds ratio (OR) for each association.

Usage

```
calcRawAssocProbOR(degCreResList)
```

Arguments

`degCreResList`	List of DegCre results.

Details

This function calculates the raw association probability odds ratio (OR) for each association in a DegCre analysis output. The OR is calculated relative to the distance bin null association probability, which would happen if all CRE p-values were identical. Thus it is a measure of the increase in association probability due to CRE p-value information content over what would occur by random chance.

Value

A numeric vector of raw association probability odds ratios (OR) for each association.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load test data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)
```

```
#Calculate raw odds ratio.
ORvec <- calcRawAssocProbOR(degCreResListDexNR3C1)
```

`changeColorAlpha` *Change Color Transparency*

Description

Changes the transparency (alpha channel) of a color or vector of colors.

Usage

```
changeColorAlpha(colorVec, newAlpha = 80)
```

Arguments

`colorVec`	Character vector of hexadecimal or named colors to be modified.
`newAlpha`	Numeric value specifying the new alpha transparency level (0-255) (Default: 80).

Details

Not exported. This function takes a color or vector of colors in hexadecimal or named colors and modifies their transparency by changing the alpha channel value. It returns the modified color(s) with the updated transparency.

Value

A character vector of modified colors with adjusted transparency in hexadecimal.

Author(s)

Brian S. Roberts

Examples

```
# Change transparency of a color
newColor <- changeColorAlpha(colorVec = "#FF0000", newAlpha = 80)
```

`collapseDegCreToGene` *Collapse DegCre associations to a single TSS per gene*

Description

Given a DegCre results list, this function finds associations between the same CRE and multiple TSSs of the same gene and keeps the nearest TSS only.

Usage

```
collapseDegCreToGene(
  degCreResList,
  method = "nearest",
  geneColname = "GeneSymb"
)
```

Arguments

`degCreResList`	List of DegCre results.
`method`	Method for choosing between multiple TSS. Currently only supported for "nearest". (Default:`"nearest"`)
`geneColname`	The name of the metadata column in DegGR within `degCreResList` that has the gene name (Default:`"GeneSymb"`)

Details

Often, the DegGR input to DegCre will contain multiple TSS's for a given gene. DegCre will create associations to all of them. Often, downstream analyses need to only have on CRE to TSS association per gene. This function modifies the DegCre Hits to keep the shortest (minimum) genomic distance association.

Value

A `degCreResList` with only the shortest TSS to CRE association per gene.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load test data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]
```

```
#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)

#Convert to single TSS per association.

degCreResListUniqTSS <- collapseDegCreToGene(degCreResListDexNR3C1,
                                             method = "nearest",
                                             geneColname = "GeneSymb")
```

`convDegCreResListToCreGeneScoreGR`

Convert a degCreResList to a creGeneScoreGR

Description

Converts a degCreResList to a hit indexed GRanges with metadata columns of predicted gene target ("predictGene") and the association score ("score").

Usage

```
convDegCreResListToCreGeneScoreGR(
  degCreResList,
  scoreType = "assocProb",
  geneColname = "GeneSymb",
  onlyDEGs = TRUE,
  DEgAlpha = NULL,
  degPadjColname = "pAdj"
)
```

Arguments

`degCreResList`	A `degCreResList`
`scoreType`	Character of one of `assocProb`, `adjOR`, `rawAssocProb`. (Default:`assocProb`) See Details for a description of these options.
`geneColname`	The name of the metadata column in `DegGR` in `degCreResList` that contains the gene names. (Default:`"GeneSymb"`)
`onlyDEGs`	Logical. If `TRUE`, only those associations involving a gene with an adjusted p-value less than or equal to `DEgAlpha` are returned. (Default:`TRUE`)
`DEgAlpha`	Significance threshold for DEGs in associations to report. (Default:`NULL`.)
`degPadjColname`	The metadata column name in `DegGR` that contains the adjusted p-values. (Default:`pAdj`)

Details

This function simplifies a `degResList` to a GRanges with metadata of the predicted associated gene and a score of that prediction (`predictScore`). If `scoreType = "assocProb"`, the score is the DegCre association probability. By setting `scoreType = "adjOR"` the odds-ratio of the association probability is calculated with calcAssocProbOR and reported. This can be useful to emphasize associations that are greatly above background and will tend to weight longer distance associations more heavily. By setting `scoreType = "rawAssocProb"` the raw association probability is used. This is the probability of association without considering the probability that the involved DEG is truly differential. This score may be useful when comparing to CRISPRi validation in the absence of a perturbation.

Value

A GRanges of CRE regions with metadata columns of `predictGene` and `predictScore`.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load test data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)

#Convert to GR with assocProb as score.

creToGeneGR <- convDegCreResListToCreGeneScoreGR(degCreResListDexNR3C1,
                                                 scoreType="assocProb",
                                                 geneColname="GeneSymb",
                                                 onlyDEGs=TRUE,
                                                 DEgAlpha=NULL,
                                                 degPadjColname="pAdj")
```

`convertDegCreDataFrame`

Convert DegCre Results List to DataFrame

Description

Given a DegCre results list, this function converts it into a DataFrame for further analysis and export.

Usage

```
convertDegCreDataFrame(degCreResList, assocAlpha = 0.05)
```

Arguments

`degCreResList`	List of DegCre results.
`assocAlpha`	The significance threshold for associations to be included in the output (Default: `0.05`).

Details

This function takes a DegCre results list as input and extracts the significant associations based on the adjusted p-values `assocProbFDR` compared to the specified significance threshold `assocAlpha`. It then creates a DataFrame with the genomic coordinates of the significant associations from both the `DegGR` and `CreGR` components of the input list. These are marked as `Deg_` or `Cre_` with `chr`, `start`, `end`, and `strand`. The coordinates are followed by the metadata of the Hits DataFramed by runDegCre. These are then followed by all metadata columns in the input `DegGR` or `CreGR` proceeded by either `Deg_` or `Cre_` in the colname.

If no associations pass the significance threshold, the function returns `NA`.

Value

A DataFrame containing the significant associations that pass the specified significance threshold. It is roughly in BEDPE format.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load test data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]
```

```
#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)

#Create DataFrame.
outDf <-
  convertDegCreDataFrame(degCreResList=degCreResListDexNR3C1,
                                    assocAlpha = 0.05)

#Write out as text file.
degCreDfFile <- tempfile(pattern="myDegCreResults",fileext=".tsv")

write.table(outDf,file=degCreDfFile[1],sep="\t",row.names=FALSE,quote=FALSE)

unlink(degCreDfFile[1])
```

---

convertdegCreResListToGInteraction

Convert DegCre Results List to GInteractions Object

Description

Given a DegCre results list, this function converts it into a GInteractions object.

Usage

```
convertdegCreResListToGInteraction(degCreResList, assocAlpha = 0.05)
```

Arguments

`degCreResList`	List of DegCre results.
`assocAlpha`	The significance threshold for associations to be included in the output (Default: `0.05`).

Details

This function takes a DegCre results list as input and extracts the significant associations based on the `assocProbFDR` compared to the specified significance threshold `assocAlpha`. It then creates a GInteractions object metadata columns from the input list.

If no associations pass the significance threshold, the function returns NA' and prints a message.

Value

A GInteractions object containing the associations that pass the specified significance threshold.

The GInteractions object has same metadata columns as the Hits returned from runDegCre with additional columns. These additional columns are every metadata column in the input `DegGR` or `CreGR` proceeded by either `Deg_` or `Cre_` in the colname.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load test data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)

#Create GInteractions object.
gInteractions <-
 convertdegCreResListToGInteraction(degCreResList=degCreResListDexNR3C1,
                                    assocAlpha = 0.01)
```

`correctAssocProbs` *Correct Association Probabilities*

Description

This function corrects association probabilities based on distance bins and reference association probabilities.

Usage

```
correctAssocProbs(sortHitsDf, assocProbs, refAssocProbs = NULL)
```

Arguments

`sortHitsDf`	A DataFrame containing sorted hits data.
`assocProbs`	A numeric vector of association probabilities.
`refAssocProbs`	A numeric vector of reference association probabilities. (Default: `NULL`, uses `assocProbs` if `NULL`)

Details

Not exported. This function corrects association probabilities within the same distance bin based on reference association probabilities in lower distance bins. It calculates adjusted association probabilities and reference association probabilities for each distance bin and updates the original association probabilities accordingly. The principle is that for all associations involving a single CRE, those associations to significant DEGs that span the shortest distances should be weighted higher than those that span farther distances. It is meant to run within runDegCre. It will not run well on unintended inputs.

Value

A numeric vector of corrected association probabilities.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
# Distance bin correct association probabilities.
correctedProbs <- correctAssocProbs(sortHitsDf = mySortedHits,
                                    assocProbs = myAssocProbs,
                                    refAssocProbs = myRefAssocProbs)

## End(Not run)
```

`creGRToSignal` *Convert CreGR to Pseudo-Continuous Signal for Plotting*

Description

This function converts a GenomicRanges object containing CRE data to a pseudo-continuous signal track suitable for plotting in plotgardener. The signal is derived from p-values and, optionally, log-fold change values associated with CREs.

Usage

```
creGRToSignal(
  CreGR,
  plotRegionGR,
  useLogFC = TRUE,
  pValColName = "pVal",
  logFcColName = "logFC",
  creSignalBinRes = 100
)
```

Arguments

`CreGR`	A GRanges object representing CRE data.
`plotRegionGR`	A GRanges object specifying the region of interest for plotting.
`useLogFC`	Logical, indicating whether to consider log-fold change values (Default: `TRUE`).
`pValColName`	Character specifying the column name in CreGR containing p-values (Default: `pVal`).
`logFcColName`	Character specifying the column name in CreGR containing log-fold change values (Default: `logFC`).
`creSignalBinRes`	Numeric value specifying the bin resolution in base pairs for the pseudo-continuous signal (Default: `100`).

Details

Not exported. This function takes a GRanges object (`CreGR`) representing CRE data, extracts p-values, and, if specified, log-fold change values. It then converts these values into a signed -log p-value in the `signal` metadata column.

Value

A GRanges object with signal values in metadata column `score`, suitable for plotting in plotSignal.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
#Load example data.
data(DexNR3C1)

myCreGR <- DexNR3C1$CreGR
myPlotRegionGR <- GRanges(seqnames="chr4",
                          ranges=IRanges(start=3.6e6,end=3.8e6))

# Convert CRE data to a pseudo-continuous signal
creSignalGR <- creGRToSignal(CreGR=myCreGR, plotRegionGR=myPlotRegionGR)

## End(Not run)
```

`DegCre` *DegCre*

Description

Probabilistic association of DEGs to CREs from differential data.

Author(s)

Maintainer: Brian S. Roberts `<brianroberts1976@yahoo.com>` (ORCID)

See Also

Useful links:

- https://github.com/brianSroberts/DegCre
- Report bugs at https://github.com/brianSroberts/DegCre/issues

---

`degCrePRAUC` *Calculate PR AUC for DegCre results.*

Description

This function calculates the Precision-Recall Area Under the Curve (AUC) from a DegCre results list.

Usage

```
degCrePRAUC(
  degCreResList,
  makePlot = TRUE,
  nShuff = 100,
  alphaVal = degCreResList$alphaVal,
  nThresh = 200
)
```

Arguments

`degCreResList`	A list of DegCre results.
`makePlot`	Logical indicating whether to generate a plot of the Precision-Recall curve. (Default: `TRUE`)
`nShuff`	Integer number of shuffles for no-skill curve. (Default: `100`)
`alphaVal`	Numeric from 0 to 1 threshold alpha value of DEG significance. (Default: alpha value from `degCreResList`)
`nThresh`	Integer number of threshold values for the Precision-Recall curve. (Default: `200`)

Details

This function calculates the Precision-Recall curve and AUC based on the provided DegCre results. It also estimates the statistical significance of the AUC by shuffling the associations and calculating AUC for shuffled data. Note that the PR AUCs tend to be small (0.05-0.2). Under the calculation framework, a PR AUC of 1 could only be achieved from DegCre results in which every association involves a significant DEG and has an association probability of 1. This situation will never actually occur but serves as a theoretical optimum for comparison.

Value

Invisibly, a list containing:

actualTprPpvMat A matrix of actual True Positive Rate (TPR) and apparent Positive Predictive Value (PPV).

shuffTprQMat A matrix of shuffled TPR quantiles.

shuffPpvQMat A matrix of shuffled PPV quantiles.

AUC Numeric of the total Area Under the Curve (AUC) for the Precision-Recall curve.

deltaAUC Numeric of the difference in AUC between the actual curve and shuffled curves.

normDeltaAUC Numeric of the normalized difference in AUC.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load sample data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)

#Plot PR curve.

degCrePRAUC(degCreResList=degCreResListDexNR3C1)

#Get PR results with out plotting.

prAUCList <- degCrePRAUC(degCreResList=degCreResListDexNR3C1,
                         makePlot=FALSE)
```

`DexNR3C1` *DegCre input data for examples.*

Description

DegCre input data for examples.

Format

A named list with two slots: DegGR and CreGR.

DegGR GRanges of RNA-seq data. The coordinates reference TSS sites. It has the following mcols:

- **promGeneName** EPDNew promoter names
- **GeneSymb** Gene symbols
- **GeneID** Ensembl gene ids
- **baseMean** baseMean values from DESeq2
- **logFC** Log-2 fold-changes from DESeq2
- **pVal** P-values from DESeq2
- **pAdj** Adjusted p-values from DESeq2

CreGR GRanges of differntial CRE data. The coordinates reference signal regions. It has the following mcols:

- **logFC** Log-2 fold-changes from csaw
- **pVal** P-values from csaw
- **pAdj** Adjusted p-values from csaw

Details

This is a list with two slots: DegGR and CreGR. This data was derived from work by McDowell et al. in which they generated RNA-seq and ChIP-seq data by treating A549 cells with dexamethasone at several time points. Specifically this is RNA-seq and NR3C1 ChIP-seq at four hours versus control.

Author(s)

Brian S. Roberts

References

`https://genome.cshlp.org/content/28/9/1272`

`distBinHeuristic` *Determine Optimal Distance Bin Size*

Description

Analyzes the associations between DEG and CRE Hits to determine the optimal distance bin size for further analysis.

Usage

```
distBinHeuristic(
  degCreHits,
  CreP,
  fracMinKsMedianThresh = 0.2,
  smallestTestBinSize = 100,
  verbose = TRUE
)
```

Arguments

`degCreHits`	A Hits object containing the associations between DEG (Differentially Expressed Genes) and CRE (Cis-Regulatory Element) hits with distances.
`CreP`	A numeric vector of CRE p-values corresponding to the associations in `degCreHits`.
`fracMinKsMedianThresh`	Numeric value from 0 to 1 of the threshold for minimum Kolmogorov-Smirnov (KS) median range. Determines the range of KS statistics considered for optimal bin size. (Default: `0.2`)
`smallestTestBinSize`	Integer minimum number of associations in each test bin. (Default: `100`)
`verbose`	Logical indicating whether to display progress messages. (Default: `TRUE`)

Details

Not exported. This function analyzes the associations between DEG and CRE hits to determine the optimal distance bin size. It uses Kolmogorov-Smirnov (KS) statistics to assess the difference in distribution between CRE p-values for different distance bin sizes versus the global. The function selects the largest bin size that falls within the specified fraction of the KS median range. This function operates within runDegCre on controlled inputs. It will not run well on unintended inputs.

Value

A list containing:

pickedBinSize The optimal distance bin size selected based on KS statistics.

crePKsMat A matrix of distance bin sizes and their corresponding median KS statistics.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
# Example usage:

optimalBinSize <- distBinHeuristic(degCreHits = myDegCreHits,
                                   CreP = myCreP)

# Access the selected bin size:
selectedBinSize <- optimalBinSize$pickedBinSize

# Access the matrix of bin sizes and median KS statistics:
binStatsMatrix <- optimalBinSize$crePKsMat

# Plot the results:
plot(binStatsMatrix[, 1], binStatsMatrix[, 2], type = "l", xlab =
"Distance Bin Size", ylab = "Median KS Statistic")

## End(Not run)
```

`fastKS` *Perform Kolmogorov-Smirnov (KS) Test Without Calculating P-Value*

Description

This function performs a Kolmogorov-Smirnov (KS) test quickly without calculating the p-value. It measures the maximum difference between cumulative probability distributions of a test set and a reference set.

Usage

```
fastKS(testSet, testIndices, refCumProbs)
```

Arguments

`testSet`	Numeric vector representing the values of the test set.
`testIndices`	Indices of elements in the test set to consider.
`refCumProbs`	Numeric vector representing the pre-computed cumulative probabilities of the reference set.

Details

Not exported. The function compares the cumulative probability distributions of the specified test set to the pre-computed cumulative probabilities of the reference set. It returns the the KS statistic without calculating the p-value (for computational speed). It is meant to run within calcKStestStatMedian. It will not run well on unintended inputs.

Value

Numeric value of the KS test statistic.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
# Example usage of the function.
ks_stat <- fastKS(testSet = testData,
                          testIndices = indices,
                          refCumProbs = refCumProbs)

## End(Not run)
```

`getAssocDistHits` *Get Associations and Distances Between Genomic Regions*

Description

This function finds associations and distances between two sets of genomic regions.

Usage

```
getAssocDistHits(DegGR, CreGR, maxDist = 1e+06)
```

Arguments

`DegGR`	A GRanges object representing DEG TSSs.
`CreGR`	A GRanges object representing the CREs.
`maxDist`	Integer value representing the maximum distance allowed for associations. Regions further apart than this threshold will not be associated. (Default: `1e6`)

Details

This function identifies associations between genomic regions from two GenomicRanges objects (DegGR and CreGR) based on their spatial overlap within a specified maximum distance threshold using findOverlaps. It calculates the distances with distance between associated regions and stores them in the metadata of the Hits object. Large values `maxDist` will require more computational resources.

Value

A Hits object containing associations and their distances between the genomic regions represented by DegGR and CreGR.

Author(s)

Brian S. Roberts

Examples

```
#Load sample data.
data(DexNR3C1)

# Get hits with association distances.
hits <- getAssocDistHits(DegGR = DexNR3C1$DegGR,
                         CreGR = DexNR3C1$CreGR,
                         maxDist = 1e6)
```

`getDegCrePlotRegionFromGene` *Get Genomic Range for a Gene and Associated CREs Below an FDR Threshold*

Description

Given a list of DegCre results, (`degCreResList`), this function generates a GRanges object encompassing all associated CRE regions for a specific gene with an associated FDR below a specified threshold.

Usage

```
getDegCrePlotRegionFromGene(
  degCreResList,
  geneName,
  geneNameColName,
  assocAlpha = 0.05
)
```

Arguments

`degCreResList`	List of DegCre results.
`geneName`	Character of the name of the gene for which to retrieve associated CRE regions.
`geneNameColName`	Character specifying the column name for gene names in `DegGR`.
`assocAlpha`	Numeric value from 0 to 1 specifying the threshold for the association probability FDR (Default: `0.05`).

Details

Not exported. This function extracts the relevant components from the input `degCreResList` and identifies associations for the specified gene with an association probability FDR below `assocAlpha`. If associations are found, it computes the genomic range encompassing all associated CREs and returns it as a GenomicRanges object. If no associations meet the threshold, it returns `NA`. This function is meant to run within plotBrowserDegCre. It will not run well on unintended inputs.

Value

A GRanges object representing the genomic region encompassing all associated CREs for the specified gene, or `NA` if no associations below the FDR threshold are found.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
#Load example data.
data(DexNR3C1)

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=DexNR3C1$DegGR,
                         DegP=DexNR3C1$DegGR$pVal,
                         DegLfc=DexNR3C1$DegGR$logFC,
                         CreGR=DexNR3C1$CreGR,
                         CreP=DexNR3C1$CreGR$pVal,
                         CreLfc=DexNR3C1$CreGR$logFC)

#Get plot region.
plotRegionGR <-
 getDegCrePlotRegionFromGene(degCreResList=degCreResListDexNR3C1,
                             geneName = "ERRFI1",
                             geneNameColName = "GeneSymb",
                             assocAlpha = 0.05)

## End(Not run)
```

`getDistBinNullAssocProb`

Calculate Null Association Probability for Each Distance Bin

Description

Calculates the null association probability for each distance bin in the DegCre analysis.

Usage

```
getDistBinNullAssocProb(degCreResList)
```

Arguments

`degCreResList`	A list of DegCre results.

Details

This function takes the results of the DegCre analysis and computes the null association probability for each unique distance bin. The null association probability represents the expected proportion of differentially expressed genes (DEGs) in each distance bin under the null hypothesis.

Value

A matrix with these columns:

binAssocDist Numeric value representing the distance bin (TSS to CRE) in base pairs.

nullAssocProb Numeric value representing the null association probability of the bin.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load example data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                    DegP=subDegGR$pVal,
                                    DegLfc=subDegGR$logFC,
                                    CreGR=subCreGR,
                                    CreP=subCreGR$pVal,
                                    CreLfc=subCreGR$logFC)

# Calculate null association probabilities.
outNullMat <- getDistBinNullAssocProb(degCreResList = degCreResListDexNR3C1)
```

`getExpectAssocPerDEG` *Get Expected Associations per DEG*

Description

Calculates the expected associations per DEG (Differentially Expressed Gene).

Usage

```
getExpectAssocPerDEG(degCreResList, geneNameColName = NULL, assocAlpha = 0.05)
```

Arguments

`degCreResList`	A list of DegCre results.
`geneNameColName`	Character value of the name of the column in DegGR (that was inputted to runDegCre) that contains gene names. If NULL, the function will attempt to automatically find the gene name column. (Default: `NULL`)
`assocAlpha`	Numeric value from 0 to 1 specifying the significance threshold for associations. (Default: `0.05`)

Details

This function calculates the expected associations per DEG based on DegCre analysis results. It first filters significant associations based on the provided association significance threshold (`assocAlpha`) and then computes the expected associations per gene. The function returns a DataFrame with gene-level information, including expected associations, number of associations, and significance thresholds.

Value

A DataFrame with the all data in the input DegGR with these columns added:

geneName Character values of gene names extracted from `geneNameColName` column (or column found if `geneNameColName` = `NULL`) in DegGR.

expectAssocs Numeric values of the expected associations per gene.

nAssocs Integer value of the number of associations passing `assocAlpha` per gene.

assocAlpha Numeric value from 0 to 1 of input `assocAlpha`

degAlpha Numeric value from 0 to 1 of the significance threshold for DEGs. Obtained from `degCreResList`

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load example data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)


# Get expected associations per DEG
expectAssocsDf <- getExpectAssocPerDEG(degCreResList = degCreResListDexNR3C1,
                                       geneNameColName = "GeneSymb",
                                       assocAlpha = 0.05)

head(expectAssocsDf)
```

`getLabelYfromPlotgardenerObj`

Get Y Coordinate for Label Placement from a PlotGardener Plot Object

---

Description

Given a PlotGardener plot object, this function calculates the Y coordinate for label placement based on the vertical positioning specified by the 'just' parameter.

Usage

```
getLabelYfromPlotgardenerObj(plotgardenerObj, just = "center")
```

Arguments

`plotgardenerObj`	A plotgardener plot object.
`just`	Character specifying the vertical positioning of the label. Options include "top" (top-aligned), "bottom" (bottom-aligned), and "center" (center-aligned). (Default: `center`)

Details

Not exported. This function takes a plotgardener plot object as input and calculates the Y coordinate for label placement based on the vertical positioning specified by the 'just' parameter. The 'just' parameter determines whether the label should be placed at the top, bottom, or center of the plot object. The function returns the calculated Y coordinate.

Value

A numeric value representing the Y coordinate in inches for label placement.

Author(s)

[Author Name]

Examples

```
## Not run:
# Get y label.
labelY <- getLabelYfromPlotgardenerObj(plotObj=mySignalPlotObj,
                                       just = "center")

## End(Not run)
```

`makeDistBinChunkList` *Convert a Hits object to association distance list*

Description

Convert a Hits of DEG to CRES with distance to a list broken into distance bins

Usage

```
makeDistBinChunkList(sortHitsWithDistDf, pickedBinSize, verbose = FALSE)
```

Arguments

`sortHitsWithDistDf`	A DataFrame with queryHits and subjectHits from a Hits object. They must be sorted by increasing association distance.
`pickedBinSize`	The number of associations per bin.

Details

Not exported. This function splits by the associations by distance to create bins with equal number of elements. The last bin is merged to the second to last bin if it is smaller than 0.8*`pickedBinSize`.

Value

A list of DataFrame(s) with `pickedBinSize` associations, grouped by distance.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
outChunkList <- makeDistBinChunkList(inSortHitsWithDistDf,pickedBinSize=1000)

## End(Not run)
```

`makePlotGInter` *Create a GInteractions Object for Plotgardener Arches Plotting*

Description

This function generates a GInteractions object suitable for plotting arches in plotgardener.

Usage

```
makePlotGInter(
  DegGR,
  CreGR,
  plotRegionGR,
  keepPromCreHits,
  plotUnit,
  maskFoundGene,
  mergeGenePromotersDist
)
```

Arguments

`DegGR`	A GRanges object representing DEG TSSs.
`CreGR`	A GRanges object representing CRE regions.
`plotRegionGR`	A GRanges object specifying the region of interest for plotting.
`keepPromCreHits`	A Hits object containing filtered promoter-CRE associations.
`plotUnit`	Numeric value specifying the minimum plot unit width in base pairs.
`maskFoundGene`	Character specifying the index of column for gene name in `DegGR`.
`mergeGenePromotersDist`	Numeric value specifying the distance in base pairs within which promoters of the same gene are merged.

Details

This function takes genomic data, including differential expression data (`DegGR`), CRE data (`CreGR`), a region of interest (`plotRegionGR`), TSS-CRE associations (`keepPromCreHits`), and other parameters, and creates a GInteractions object suitable for plotting arches in plotgardener.

Not exported. The function processes the data to ensure that promoters and CREs are in a suitable format for plotting. It merges promoters within a specified distance (`mergeGenePromotersDist`) and adjusts the width of regions to meet the minimum plot unit width (`plotUnit`). It also handles cases where promoters and CREs may have been merged or contain duplicate regions, ensuring correct associations. This function is meant to run within plotBrowserDegCre. It will not run well on unintended inputs.

Value

A GInteractions object suitable for arches plotting in plotgardener.

Author(s)

Brian S. Roberts

Examples

```
## Not run:
# Create the GInteractions object for plotting
GInterX <- makePlotGInter(DegGR=myDegGR,
                          CreGR=myCreGR,
                          plotRegionGR=myPlotRegionGR,
                          keepPromCreHits=myKeepPromCreHits,
```

```
                              plotUnit=5000,
                              maskFoundGene=3,
                              mergeGenePromotersDist=2000)

## End(Not run)
```

`optimizeAlphaDegCre` *Run DegCre with DEG alpha optimization.*

Description

Runs DegCre across a set of DEG alpha thesholds to find optimal performance.

Usage

```
optimizeAlphaDegCre(
  DegGR,
  DegP,
  DegLfc = NULL,
  CreGR,
  CreP,
  CreLfc = NULL,
  reqEffectDirConcord = TRUE,
  padjMethod = "bonferroni",
  maxDist = 1e+06,
  verbose = FALSE,
  smallestTestBinSize = 100,
  fracMinKsMedianThresh = 0.2,
  testedAlphaVals = c(0.005, 0.01, 0.02, 0.03, 0.05, 0.1),
  minNDegs = 5
)
```

Arguments

`DegGR`	A GRanges object of gene TSSs. Multiple TSSs per gene are allowed.
`DegP`	A numeric vector of differential expression p-values for genes in `DegGR`.
`DegLfc`	A numeric vector of log fold-change values of differential expression for gene in `DegGR`. Required when `reqEffectDirConcord` = `TRUE`. (Default: `NULL`)
`CreGR`	A GRanges object of CRE regions.
`CreP`	A numeric vector differential signal p-values for regions in `CreGR`.
`CreLfc`	A numeric vector log fold-change values of differential signal for regions in `CreGR`. Required when `reqEffectDirConcord` = `TRUE`. (Default: `NULL`)
`reqEffectDirConcord`	A logical whether to require concordance between the effect direction between DEG and CRE differential values. (Default: `NULL`)
`padjMethod`	A character value indicating the method for p-value adjustment. Do not change from default under most circumstances. Can be any method name accepted by `p.adjust()` (Default: `bonferroni`)

`maxDist`	An integer value specifying the maximum distance for probability calculation of TSS to CRE associations. (Default: `1e6`)
`verbose`	A logical indicating whether to print messages of step completion and algorithm results. (Default: `NULL`)
`smallestTestBinSize`	An integer value specifying the size (number of elements) of the smallest distance bin to be considered in the optimization algorithm. (Default: `100`)
`fracMinKsMedianThresh`	A numeric value between 0 and 1 specifying the optimization criterion for the distance bin size algorithm (See Details). (Default: `0.2`)
`testedAlphaVals`	A numeric vector of DEG alpha values to test (Default: `c(0.005,0.01,0.02,0.03,0.05,0.1)`).
`minNDegs`	An integer specifying minimum number of DEGs that pass the lowest `testedAlphaVals`. (Default: `5`)

Details

This function runs runDegCre for each value in `testedAlphaVals`. The performance at each tested alpha is evaluated with degCrePRAUC. which generates a Precision-Recall curve based on the recovery rate of DEGs by associations. Various AUCs are calculated as performance metrics. Using the alpha with the highest value of `normDeltaAUC` is recommended (see Examples).

Value

A named list containing:

alphaPRMat A matrix of Precision-Recall Area Under the Curve (AUC) values.

degCreResListsByAlpha Named list of DegCre results lists indexed by the `testedAlphaVals`.

The columns of `alphaPRMat` are:

alphaVal Numeric vector of tested DEG alpha value.

AUC Numeric vector of Area under the curve of a Precision-Recall (PR) curve based on associations recovering significant DEGs.

deltaAUC Numeric vector of PR AUC minus the AUC of the no-skill line.

normDeltaAUC Numeric vector of the value of `deltaAUC` divided by one minus the no-skill AUC.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load sample data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]
```

```
# Run DegCre over range of alpha values:
alphaOptList <- optimizeAlphaDegCre(DegGR = subDegGR,
                                    DegP = subDegGR$pVal,
                                    DegLfc = subDegGR$logFC,
                                    CreGR = subCreGR,
                                    CreP = subCreGR$pVal,
                                    CreLfc = subCreGR$logFC)

bestAlphaId <- which.max(alphaOptList$alphaPRMat[,4])
bestDegCreResList <- alphaOptList$degCreResListsByAlpha[[bestAlphaId]]
```

`plotBrowserDegCre` *Make Browser Plots from DegCre Results*

Description

Creates browser plots of specified genomic regions or gene regions based on the provided DegCre analysis results.

Usage

```
plotBrowserDegCre(
  degCreResList,
  assocAlpha = 0.05,
  browserWinPad = 1000,
  geneName = NULL,
  plotRegionGR = NULL,
  CreSignalName = "CRE",
  assembly = "hg38",
  plotWidth = grDevices::dev.size("in")[1],
  plotHeight = grDevices::dev.size("in")[2],
  plotXbegin = 0.9,
  mergeGenePromotersDist = 1000,
  sigPlotMaxY = 4,
  assocColorRange = NULL,
  lowAssocColor = "#88CCEE",
  hiAssocColor = "#CC6677",
  signalColor = "#DDCC77",
  geneLabelFontsize = 8,
  axisFontSize = 6,
  panelTitleFontSize = 7,
  geneNameColName = NULL,
  geneHighlightDf = NULL,
  dePrioritizeSmallRNA = TRUE,
  useLogFC = TRUE,
  creSignalBinRes = 100
)
```

Arguments

`degCreResList`	List of DegCre results.
`assocAlpha`	Numeric value from 0 to 1 of significance threshold for associations. (Default: `0.05`)
`browserWinPad`	Numeric value of the padding size (in base pairs) to extend the plotting region. (Default: `1000`)
`geneName`	Character of name of the gene of interest. If specified, the function will plot the region associated with this gene. (Default: `NULL`)
`plotRegionGR`	GRanges of length 1 specifying the region to plot. If provided, geneName is ignored. (Default: `NULL`)
`CreSignalName`	Character of name of the differential CRE signal track. For plot labeling purposes only (Default: `CRE`)
`assembly`	Character of genome assembly name, e.g., "hg38". Must be one of accepted inputs to `assembly` argument to plotGenomeLabel. (Default: `hg38`)
`plotWidth`	Numeric value of width of the browser plot in inches. (Default: `dev.size("in")[1]`)
`plotHeight`	Numeric value of height of the browser plot in inches. (Default: `dev.size("in")[2]`)
`plotXbegin`	Numeric value of the width of the left margin (where track annotations will be) in inches. (Default: `0.9`)
`mergeGenePromotersDist`	Maximum distance (in base pairs) for merging promoters of the same gene in plot. (Default: `1000`)
`sigPlotMaxY`	Numeric value of maximum value for the CRE differential signal plot (Y-axis). (Default: `4`)
`assocColorRange`	Numeric vector of values from 0 to 1 of length 2. These values specify the lower and upper values of DegCre association probabilities for color saturation for arch color. (Default: `NULL`. If `NULL` will be set to 0 and maximum association probability in input data.)
`lowAssocColor`	Character color for low saturation point of association probabilities in arches plot. (Default: `#88CCEE`)
`hiAssocColor`	Character color for high saturation point of association probabilities in arches plot. (Default: `#CC6677`)
`signalColor`	Character color for the CRE differential signal plot. (Default: `#DDCC77`)
`geneLabelFontsize`	Numeric of font size (as implemented in plotgardener) for gene labels. (Default: `8`)
`axisFontSize`	Numeric of font size for axis labels and tick marks. (Default: `6`)
`panelTitleFontSize`	Numeric of font size for panel titles. (Default: `7`)
`geneNameColName`	Character of name of the column in DegGR metadata that was inputted to runDegCre that contains gene names to query by `geneName`. (Default: `NULL`. If omitted, the column name will be guessed, with warnings if not.)
`geneHighlightDf`	DataFrame specifying genes to highlight in the plot as accepted by plotGenes argument `geneHighlights`.

`dePrioritizeSmallRNA`	Logical, indicating whether small RNA genes should be deprioritized in plotting. (Default: `TRUE`)
`useLogFC`	Logical, indicating whether to use log-fold change values for the CRE differential signal. (Default: `TRUE`)
`creSignalBinRes`	Bin resolution in base pairs for the CRE signal track. Only used for initial calculation and will likely differ from display resolution. (Default: `100`)

Details

This function uses plotgardener functionality to generate browser plots for visualizing DegCre analysis results in specified regions. The user can input genomic regions or gene names. The output plot consists of an arches plot made with plotPairsArches of DegCre associations colored by association probability. Below is a signal track plot made via plotSignal of the data in `CreGR` that was inputted to runDegCre. This plot displays the signed negative log p-value, meaning the $-log_{10}(p_{CRE})$ multiplied by the sign of the the CRE log fold-change. Beneath this panel genomic coordinates via plotGenomeLabel and gene models via plotGenes are displayed.

Value

Invisibly, a named list containing:

plotRegionGR GRanges of the plotted region.

creSignalPlotGR GRanges of the CRE signal (signed negative log CRE p-value) across the plotted region.

assocGinter GInteractions of the DegCre associations in the plotted region.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load example data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)

#Make browser plot from specified gene name.
browserOuts <- plotBrowserDegCre(degCreResList=degCreResListDexNR3C1,
```

```
                                 geneName="ERRFI1",
                                 geneNameColName="GeneSymb",
                                 CreSignalName="NR3C1")
dev.off()

#Make plot of specified region.
zoomGR <- GenomicRanges::GRanges(seqnames="chr1",
                                 ranges=IRanges(start=7900e3,end=8400e3))

zoomedBrowserOuts <- plotBrowserDegCre(degCreResList=degCreResListDexNR3C1,
                                       plotRegionGR=zoomGR,
                                       geneNameColName="GeneSymb",
                                       CreSignalName="NR3C1")
dev.off()
```

`plotDegCreAssocProbVsDist` *Plot DegCre Association Probability vs. Binned Genomic Distance*

Description

Plots the DegCre association probability against binned genomic distance and highlights the quantile range.

Usage

```
plotDegCreAssocProbVsDist(
  degCreResList,
  assocProbFDRThresh = 0.05,
  plotQRange = c(0.25, 0.75),
  hiYLim = NULL,
  loYLim = NULL,
  qRangeFillColor = "#88CCEE",
  nullLineColor = "#CC6677"
)
```

Arguments

`degCreResList`	A list of DegCre results.
`assocProbFDRThresh`	Numeric value from 0 to 1 specifying the FDR threshold for association probability. (Default: `0.05`)
`plotQRange`	Numeric vector of quantile range for plotting (e.g., `c(0.25, 0.75)` for interquartile range). (Default: `c(0.25, 0.75)`)
`hiYLim`	Numeric value specifying the upper limit of the y-axis. (Default: `NULL`)
`loYLim`	Numeric value specifying the lower limit of the y-axis. (Default: `NULL`)
`qRangeFillColor`	Color for filling the quantile range polygon. (Default: `#88CCEE`)
`nullLineColor`	Color for the null association probability line. (Default: `#CC6677`)

Details

This function takes the results of the DegCre analysis, including genomic distances and association probabilities, and creates a plot of association probabilities against binned genomic distances. It highlights the quantile range (e.g., interquartile range) and includes a line for null association probabilities. The top panel shows the number of associations passing `assocProbFDRThresh`. The bottom panel shows the median FDR-passing association probability as a black line, with the specified quantile range (defaults to interquartile) plotted as `qRangeFillColor` region. The `nullLineColor` colored line is the null association probability, that is the association probability for a bin with uniform CRE p-values.

Value

Invisibly, a matrix with these columns:

binMidDist Numeric value of the midpoint distance of the bin (TSS to CRE) in kb.

q_<`plotQRange[1]` x `100`> Numeric value of lower bound of the highlight region.

q_50 Numeric value of plotted line.

q_<`plotQRange[2]` x `100`> Numeric value of upper bound of the highlight region.

nullAssocProb Numeric of null association probability of the bin.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load example data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)

#Plot association probability versus binned genomic distance.

outProbVsDistMat <-
 plotDegCreAssocProbVsDist(degCreResList=degCreResListDexNR3C1)
```

`plotDegCreBinHeuristic` *Plot DegCre Bin Algorithm Statistics*

Description

Plots the DegCre distance bin optimization statistic against different bin sizes, highlighting the optimal bin size.

Usage

```
plotDegCreBinHeuristic(degCreResList)
```

Arguments

`degCreResList`	A list of DegCre results.

Details

This function takes a DegCre results list and plots the bin heuristic statistics against different bin sizes. It also highlights the optimal bin size chosen based on the analysis. The y-axis of the plot is the median KS statistic of all bins versus the global CRE p-value distribution.

Value

Invisibly, the picked optimal bin size.

Author(s)

Brian S. Roberts

See Also

distBinHeuristic for calculating the DEG-CRE bin heuristic.

Examples

```
#Load required packages.
library(GenomicRanges)

#Load example data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
```

```
                                        CreP=subCreGR$pVal,
                                        CreLfc=subCreGR$logFC)

#Plot distance bin median KS statistic curve.

plotDegCreBinHeuristic(degCreResList=degCreResListDexNR3C1)
```

`plotExpectedAssocsPerDeg`

Plot Histogram of Expected Associations per DEG

Description

Plots a histogram of the expected number of associations per DEG (Differentially Expressed Gene) based on DegCre analysis.

Usage

```
plotExpectedAssocsPerDeg(
  expectAssocPerDegDf,
  barOutlineColor = "#88CCEE",
  barFillColor = NULL,
  extraText = FALSE
)
```

Arguments

`expectAssocPerDegDf`	DataFrame output of getExpectAssocPerDEG.
`barOutlineColor`	Color for the outline of the histogram bars. (Default: `#88CCEE`)
`barFillColor`	Fill color for the histogram bars. If `NULL`, it will be derived from `barOutlineColor` with adjusted transparency.
`extraText`	Logical, indicating whether additional text information (Details) should be added to the plot.

Details

This function generates a histogram of the expected number of associations per DEG and optionally adds additional text information to the plot, such as DEG FDR, association FDR, and the fraction of DEGs with at least one association. Plot displays a dashed line a value indicating the median expected associations per DEG.

Value

Invisibly, the median expected associations per DEG.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load example data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#Generate DegCre results.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)

# Generate data frame of expected associations per DEG
expectAssocPerDegDf <-
 getExpectAssocPerDEG(degCreResList = degCreResListDexNR3C1,
                      geneNameColName = "GeneSymb",
                      assocAlpha = 0.05)

# Plot histogram of expected associations per DEG
medianExpAssocs <- plotExpectedAssocsPerDeg(expectAssocPerDegDf,
                                            barOutlineColor = "blue",
                                            extraText = TRUE)
```

`runDegCre` *Generate DegCre associations*

Description

Create DEG to CRE associations from differential data.

Usage

```
runDegCre(
  DegGR,
  DegP,
  DegLfc = NULL,
  CreGR,
  CreP,
  CreLfc = NULL,
  reqEffectDirConcord = TRUE,
  padjMethod = "qvalue",
  maxDist = 1e+06,
  verbose = TRUE,
  smallestTestBinSize = 100,
```

```
  fracMinKsMedianThresh = 0.2,
  alphaVal = 0.01,
  binNOverride = NULL
)
```

Arguments

`DegGR`	A GRanges object of gene TSSs. Multiple TSSs per gene are allowed.
`DegP`	A numeric vector of differential expression p-values for genes in `DegGR`.
`DegLfc`	A numeric vector of log fold-change values of differential expression for gene in `DegGR`. Required when `reqEffectDirConcord = TRUE`. (Default: `NULL`)
`CreGR`	A GRanges object of CRE regions.
`CreP`	A numeric vector differential signal p-values for regions in `CreGR`.
`CreLfc`	A numeric vector log fold-change values of differential signal for regions in `CreGR`. Required when `reqEffectDirConcord = TRUE`. (Default: `NULL`)
`reqEffectDirConcord`	A logical whether to require concordance between the effect direction between DEG and CRE differential values. (Default: `TRUE`)
`padjMethod`	A character value indicating the method for p-value adjustment. Do not change from default under most circumstances. Can be any method name accepted by p.adjust (Default: `qvalue`)
`maxDist`	An integer value specifying the maximum distance for probability calculation of TSS to CRE associations. (Default: `1e6`)
`verbose`	A logical indicating whether to print messages of step completion and algorithm results. (Default: `TRUE`)
`smallestTestBinSize`	An integer value specifying the size (number of elements) of the smallest distance bin to be considered in the optimization algorithm. (Default: `100`)
`fracMinKsMedianThresh`	A numeric value between 0 and 1 specifying the optimization criterion for the distance bin size algorithm (See Details). (Default: `0.2`)
`alphaVal`	A numeric value between 0 and 1 specifying the alpha value for DEG significance. (Default: `0.01`)
`binNOverride`	An integer value specifying the number of elements per distance bin. When specified, overrides distance bin size optimization (Not recommended). (Default: `NULL`)

Details

The DegCre algorithm considers experimental data from a perturbation experiment and produces associations between cis-regulatory elements (CREs) and differentially expressed genes (DEGs). The user provides differential expression data such as RNA-seq, and differential regulatory signal data such as ATAC-seq, DNase Hypersensitivity, and ChIP-seq. For RNA-seq analysis, we suggest methods such as DESeq2 or edgeR. For the analysis of differential regulatory data we recommend csaw. As an example experiment, we use data from McDowell et al. (PMID = 30097539) in which A549 cells were treated with dexamethasone and control. RNA-seq and ChIP-seq data were collected at various time points.

A complete description of the mathematical basis of the DegCre core algorithms is provided in DegCre bioRxiv. DegCre takes two inputs. The first is a GRanges of p-values and optionally log fold-changes associated with DEG TSSs. The second input is a GRanges of differential signal p-values and optionally log fold-changes for CRE regions. DegCre generates a Hits object of all associations between DEG TSSs and CREs within `maxDist`. Associations are then binned by TSS-to-CRE distance according to an algorithm that balances resolution (many bins with few members) versus minimization of the deviance of each bin's CRE p-value distribution from the global distribution, seleting an optimal bin size.

Next, DegCre applies a non-parametric algorithm to find concordance between and CRE differential effects within bins and derives an association probability. For all association probabilities involving one given CRE, the probabilities are adjusted to favor associations across shorter distances. An FDR of the association probability is then estimated. Results are returned in list containing a Hits object and both input GRanges.

Value

A named list containing:

degCreHits A Hits object with metadata. The queryHits of `degCreHits` reference `DegGR`. The subjectHits of `degCreHits` reference `CreGR`

binHeurOutputs List of outputs from the distance binning algorithm.

alphaVal Numeric alpha value used for DEG significance threshold.

DegGR GRanges of input `DegGR` with added metadata columns "pVal", "pAdj",and possibly "logFC" if `reqEffectDirConcord==TRUE`. Will overwrite existing metadata with same colnames.

CreGR GRanges of input `CreGR` with added metadata columns "pVal", "pAdj",and possibly "logFC" if `reqEffectDirConcord==TRUE`. Will overwrite existing metadata with same colnames.

The degCreHits Hits object metadata has these columns:

assocDist Integer of distance in base pairs between the TSS and CRE for the association.

assocProb Numeric from 0 to 1 of association probability.

assocProbFDR Numeric from 0 to 1 of False discovery rate of the association probability exceeding distance only null.

rawAssocProb Numeric from 0 to 1 of association probability not adjusted for DEG significance or shorter associations involving this CRE.

CreP Numeric of differential p-value of the CRE.

DegP Numeric of differential p-value of the DEG.

DegPadj Numeric of differential adjusted p-value of the DEG.

binAssocDist Integer of the maximum association distance cutoff for the bin containing the association.

numObs Integer number of associations in the distance bin containing the association.

distBinId Integer that uniquely identifies the distance containing the association.

Author(s)

Brian S. Roberts

Examples

```
#Load required packages.
library(GenomicRanges)

#Load sample data.
data(DexNR3C1)

subDegGR <-
 DexNR3C1$DegGR[which(Seqinfo::seqnames(DexNR3C1$DegGR)=="chr1")]
subCreGR <-
 DexNR3C1$CreGR[which(Seqinfo::seqnames(DexNR3C1$CreGR)=="chr1")]

#With defaults.
degCreResListDexNR3C1 <- runDegCre(DegGR=subDegGR,
                                   DegP=subDegGR$pVal,
                                   DegLfc=subDegGR$logFC,
                                   CreGR=subCreGR,
                                   CreP=subCreGR$pVal,
                                   CreLfc=subCreGR$logFC)

#With custom settings.
modDegCreResList <- runDegCre(DegGR=subDegGR,
                           DegP=subDegGR$pVal,
                           CreGR=subCreGR,
                           CreP=subCreGR$pVal,
                           reqEffectDirConcord=FALSE,
                           maxDist=1e5,
                           alphaVal=0.001)
```

Index