Package 'corto'

December 7, 2023

Type Package

Title Inference of Gene Regulatory Networks

Version 1.2.4

Maintainer Federico M. Giorgi <federico.giorgi@gmail.com>

Description We present 'corto' (Correlation Tool), a simple package to infer gene regulatory networks and visualize master regulators from gene expression data using DPI (Data Processing Inequality) and bootstrapping to recover edges. An initial step is performed to calculate all significant edges between a list of source nodes (centroids) and target genes. Then all triplets containing two centroids and one target are tested in a DPI step which removes edges. A bootstrapping process then calculates the robustness of the network, eventually re-adding edges previously removed by DPI. The algorithm has been optimized to run outside a computing cluster, using a fast correlation implementation. The package finally provides functions to calculate network enrichment analysis from RNA-Seq and ATAC-Seq signatures as described in the article by Giorgi lab (2020) <doi:10.1093/bioinformatics/btaa223>.

License LGPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 3.6)

NeedsCompilation no

Imports dplyr, gplots, knitr, methods, rmarkdown, parallel, pbapply, plotrix, stats, utils

VignetteBuilder knitr

Author Federico M. Giorgi [aut, cre], Daniele Mercatelli [ctb], Gonzalo Lopez-Garcia [ctb]

Repository CRAN

Date/Publication 2023-12-06 23:20:03 UTC

R topics documented:

`barplot2` *barplot2 - Bar plot with upper error bars*

Description

barplot2 - Bar plot with upper error bars

Usage

```
barplot2(values, errors, lower = FALSE, flat = TRUE, ...)
```

Arguments

`values`	A matrix of values
`errors`	A matrix of values for upper error bar
`lower`	Boolean, whether the lower error bar should be plotted, default FALSE
`flat`	Boolean, whether the head of bars should be flat, default TRUE
`...`	Arguments to be passed to the core _barplot_ function

Value

A plot

Examples

```
values<-matrix(rnorm(10*4,mean=10),nrow=4,ncol=10)
errors<-matrix(runif(10*4),nrow=4,ncol=10)
colnames(values)<-colnames(errors)<-LETTERS[1:10]
barplot2(values,errors,main="Bar plot with error bars")
```

`corto` *Calculate a regulon from a data matrix*

Description

This function applies Correlation and DPI to generate a robust regulon object based on the input data matrix and the selected centroids.

Usage

```
corto(
  inmat,
  centroids,
  nbootstraps = 100,
  p = 1e-30,
  nthreads = 1,
  verbose = FALSE,
  cnvmat = NULL,
  boot_threshold = 0
)
```

Arguments

`inmat`	Input matrix, with features (e.g. genes) as rows and samples as columns
`centroids`	A character vector indicating which features (e.g. genes) to consider as centroids (a.k.a. Master Regulators) for DPI
`nbootstraps`	Number of bootstraps to be performed. Default is 100
`p`	The p-value threshold for correlation significance (by default 1E-30)
`nthreads`	The number of threads to use for bootstrapping. Default is 1
`verbose`	Logical. Whether to print progress messages. Default is FALSE
`cnvmat`	An optional matrix with copy-number variation data. If specified, the program will calculate linear regression between the gene expression data in the input matrix (exp) and the cnv data, and target profiles will be transformed to the residuals of each linear model exp~cnv. Default is NULL
`boot_threshold`	The fraction of bootstraps in which the edge should appear to be included in the final network. It can be any number between 0.0 and 1.0. Default is 0.0.

Value

A list (object of class regulon), where each element is a centroid

- tfmode: a named vector containing correlation coefficients between features and the centroid
- likelihood: a numeric vector indicating the likelihood of interaction

Examples

```
# Load data matrix inmat (from TCGA mesothelioma project)
load(system.file("extdata","inmat.rda",package="corto",mustWork=TRUE))
# Load centroids
load(system.file("extdata","centroids.rda",package="corto",mustWork=TRUE))
# Run corto
regulon <- corto(inmat,centroids=centroids,nthreads=2,nbootstraps=10,verbose=TRUE)

# In a second example, a CNV matrix is provided. The analysis will be run only
# for the features (rows) and samples (columns) present in both matrices
load(system.file("extdata","cnvmat.rda",package="corto",mustWork=TRUE))
regulon <- corto(inmat,centroids=centroids,nthreads=2,nbootstraps=6,verbose=TRUE,cnvmat=cnvmat,
p=1e-8)
```

---

fcor *A fast correlation function*

Description

A fast correlation function

Usage

```
fcor(inmat, centroids, r)
```

Arguments

inmat	An input matrix with features as rows and samples as columns
centroids	A character vector indicating the centroids
r	A numeric correlation threshold

Value

A matrix describing which edges were significant in the input matrix matrix according to the r correlation threshold provided

fisherp *Fisher integration of p-values*

Description

This function applies the Fisher integration of pvalues

Usage

```
fisherp(ps)
```

Arguments

ps	a vector of p-values

Value

p.val an integrated p-value

Examples

```
ps<-c(0.01,0.05,0.03,0.2)
fisherp(ps)
```

gsea *GSEA*

Description

This function performs Gene Set Enrichment Analysis

Usage

```
gsea(
  reflist,
  set,
  method = c("permutation", "pareto"),
  np = 1000,
  w = 1,
  gsea_null = NULL
)
```

Arguments

reflist	named vector of reference scores
set	element set
method	one of 'permutation' or 'pareto'
np	Number of permutations (Default: 1000)
w	exponent used to raise the supplied scores. Default is 1 (original scores unchanged)
gsea_null	a GSEA null distribution (Optional)

Value

A GSEA object. Basically a list of s components:

ES The enrichment score

NES The normalized enrichment socre

ledge The items in the leading edge

p.value The permutation-based p-value

Examples

```
reflist<-setNames(-sort(rnorm(1000)),paste0('gene',1:1000))
set<-paste0('gene',sample(1:200,50))
obj<-gsea(reflist,set,method='pareto',np=1000)
obj$p.value
```

---

gsea2 *2-way GSEA GSEA Gene set enrichment analysis of two complementary gene sets using gsea*

Description

2-way GSEA GSEA Gene set enrichment analysis of two complementary gene sets using gsea

Usage

```
gsea2(
  reflist,
  set1,
  set2,
  method = c("permutation", "pareto"),
  np = 1000,
  w = 1,
  gsea_null = NULL
)
```

Arguments

reflist	named vector of reference scores
set1	element set 1
set2	element set 1
method	one of 'permutation' or 'pareto'
np	Number of permutations (Default: 1000)
w	exponent used to raise the supplied scores. Default is 1 (original scores unchanged)
gsea_null	a GSEA null distribution (Optional)

Value

A list of 2 GSEA objects. Each of which is a list of components:

ES The enrichment score

NES The normalized enrichment socre

ledge The items in the leading edge

p.value The permutation-based p-value

Examples

```
reflist<-setNames(-sort(rnorm(1000)),paste0('gene',1:1000))
set1<-paste0('gene',sample(1:200,50))
set2<-paste0('gene',sample(801:1000,50))
obj<-gsea2(reflist,set1,set2,method='pareto',np=1000)
obj$p.value
```

---

kmgformat *kmgformat - Nice Formatting of Numbers*

Description

This function will convert thousand numbers to K, millions to M, billions to G, trillions to T, quadrillions to P

Usage

```
kmgformat(input, roundParam = 1)
```

Arguments

input	A vector of values
roundParam	How many decimal digits you want

Value

A character vector of formatted numebr names

Examples

```
# Thousands
set.seed(1)
a<-runif(1000,0,1e4)
plot(a,yaxt='n')
kmg<-kmgformat(pretty(a))
axis(2,at=pretty(a),labels=kmg)

# Millions to Billions
set.seed(1)
a<-runif(1000,0,1e9)
plot(a,yaxt='n',pch=20,col="black")
kmg<-kmgformat(pretty(a))
axis(2,at=pretty(a),labels=kmg)
```

---

mra *Perform Master Regulator Analysis (mra).*

Description

The analysis is performed between two groups of samples in the form of expression matrices, with genes/features as rows and samples as columns.

Usage

```
mra(
  expmat1,
  expmat2 = NULL,
  regulon,
  minsize = 10,
  nperm = NULL,
  nthreads = 2,
  verbose = FALSE,
  atacseq = NULL
)
```

Arguments

expmat1	A numeric expression matrix, with genes/features as rows and samples as columns. If only expmat1 is provided (without expmat2), the function will perform a sample-by-sample master regulator analysis, with the mean of the dataset as a reference. If expmat2 is provided, expmat1 will be considered the "treatment" sample set. If a named vector is provided, with names as genes/features and values as signature values (e.g. T-test statistics), signature master regulator analysis is performed.

expmat2	A numeric expression matrix, with genes/features as rows and samples as columns. If provided, it will be considered as the "control" or "reference" sample set for expmat1.
regulon	A _regulon_ object, output of the _corto_ function.
minsize	A minimum network size for each centroid/TF to be analyzed. Default is 10.
nperm	The number of times the input data will be permuted to generate null signatures. Default is 1000 if expmat2 is provided, and 10 if expmat2 is not provided (single sample mra).
nthreads	The number of threads to use for generating null signatures. Default is 1
verbose	Boolean, whether to print full messages on progress analysis. Default is FALSE
atacseq	An optional 3 column matrix derived from an ATAC-Seq analysis, indicating 1) gene symbol, 2) -log10(FDR)*sing(log2FC) of an ATAC-Seq design, 3) distance from TSS. If provided, the output will contain an _atacseq_ field.

Value

A list summarizing the master regulator analysis

- nes: the normalized enrichment score: positive if the centroid/TF network is upregulated in expmat1 vs expmat2 (or in expmat1 vs the mean of the dataset), negative if downregulated. A vector in multisample mode, a matrix in sample-by-sample mode.
- pvalue: the pvalue of the enrichment.
- sig: the calculated signature (useful for plotting).
- regulon: the original regulon used in the analysis (but filtered for _minsize_)
- atac: Optionally present if atacseq data is provided. For each centroid/TF a number ranging from 0 to 1 will indicate the fraction of changes in activity due to promoter effects rather than distal effects.

---

mraplot *Plot a master regulator analysis*

Description

Plotting function for master regulator analysis performed by the _mra_ function

Usage

```
mraplot(
  mraobj,
  mrs = 5,
  title = "corto - Master Regulator Analysis",
  pthr = 0.01
)
```

Arguments

mraobj	The input object, output of the function mra
mrs	Either a numeric value indicating how many MRs to show, sorted by significance, or a character vector specifying which TFs to show. Default is 5
title	Title of the plot (optional, default is "corto - Master Regulator Analysis")
pthr	The p-value at which the MR is considered significant. Default is 0.01

Value

A plot is generated

---

p2r *p2r Convert a P-value to the corresponding Correlation Coefficient*

Description

p2r Convert a P-value to the corresponding Correlation Coefficient

Usage

```
p2r(p, n)
```

Arguments

p	the p-value
n	the number of samples

Value

a correlation coefficient

Examples

```
p2r(p=0.08,n=20)
```

p2z *p2z*

Description

This function gives a gaussian Z-score corresponding to the provided p-value Careful: sign is not provided

Usage

```
p2z(p)
```

Arguments

p	a p-value

Value

z a Z score

Examples

```
p<-0.05
p2z(p)
```

plot_gsea *Plot GSEA results*

Description

This function generates a GSEA plot from a gsea object

Usage

```
plot_gsea(
  gsea.obj,
  twoColors = c("red", "blue"),
  plotNames = FALSE,
  colBarcode = "black",
  title = "Running Enrichment Score",
  bottomTitle = "List Values",
  bottomYlabel = "Signature values",
  ext_nes = NULL,
  ext_pvalue = NULL,
  ext_es = NULL,
  omit_middle = FALSE
)
```

Arguments

gsea.obj	GSEA object produced by the gsea function
twoColors	the two colors to use for positive[1] and negative[2] enrichment scores
plotNames	Logical. Should the set names be plotted?
colBarcode	The color of the barcode
title	String to be plotted above the Running Enrichment Score
bottomTitle	String for the title of the bottom part of the plot
bottomYlabel	String for the Y label of the bottom plot
ext_nes	Provide a NES from an external calculation
ext_pvalue	Provide a pvalue from an external calculation
ext_es	Provide an ES from an external calculation
omit_middle	If TRUE, will not plot the running score (FALSE by default)

Value

Nothing, a plot is generated in the default output device

Examples

```
reflist<-setNames(-sort(rnorm(1000)),paste0('gene',1:1000))
set<-paste0('gene',sample(1:200,50))
obj<-gsea(reflist,set,method='pareto',np=1000)
plot_gsea(obj)
```

---

plot_gsea2 *Plot 2-way GSEA results*

Description

This function generates a GSEA plot from a gsea object

Usage

```
plot_gsea2(
  gsea.obj,
  twoColors = c("red", "blue"),
  plotNames = FALSE,
  title = "Running Enrichment Score",
  bottomTitle = "List Values",
  bottomYlabel = "Signature values",
  legside1 = NULL,
  legside2 = NULL
)
```

Arguments

gsea.obj	GSEA object produced by the gsea function
twoColors	the two colors to use for positive[1] and negative[2] enrichment scores, and of the barcodes
plotNames	Logical. Should the set names be plotted?
title	String to be plotted above the Running Enrichment Score
bottomTitle	String for the title of the bottom part of the plot
bottomYlabel	String for the Y label of the bottom plot (FALSE by default)
legside1	String specifying the position of the first NES legend, for example "topright", "bottomleft". Default is NULL, letting the function automatically place it
legside2	String specifying the position of the second NES legend, for example "topright", "bottomleft". Default is NULL, letting the function automatically place it

Value

Nothing, a plot is generated in the default output device

Examples

```
reflist<-setNames(-sort(rnorm(1000)),paste0('gene',1:1000))
set1<-paste0('gene',sample(1:200,50))
set2<-paste0('gene',sample(801:1000,50))
obj<-gsea2(reflist,set1,set2,method='pareto',np=1000)
plot_gsea2(obj)
```

---

r2p *r2p Convert Correlation Coefficient to P-value*

Description

r2p Convert Correlation Coefficient to P-value

Usage

```
r2p(r, n)
```

Arguments

r	the correlation coefficient
n	the number of samples

Value

a numeric p-value

Examples

```
r2p(r=0.4,n=20) # 0.08
```

scatter *scatter - XY scatter plot with extra information*

Description

This function will plot two variables (based on their common names), calculate their Coefficient of Correlation (CC), plot a linear regression line and color the background if the correlation is positive (red), negative (blue) or non-significant (white)

Usage

```
scatter(
  x,
  y,
  method = "pearson",
  threshold = 0.01,
  showLine = TRUE,
  grid = TRUE,
  bgcol = FALSE,
  pch = 20,
  subtitle = NULL,
  extendXlim = FALSE,
  ci = FALSE,
  ...
)
```

Arguments

x	The first named vector
y	The second named vector
method	a character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman": can be abbreviated.
threshold	a numeric value indicating the significance threshold (p-value) of the correlation, in order to show a colored background. Default is 0.01.
showLine	a boolean indicating if a linear regression line should be plotted. Default is TRUE
grid	a boolean indicating whether to show a plot grid. Default is TRUE
bgcol	Boolean. Should a background coloring associated to significance and sign of correlation be used? Default is TRUE, and it will color the background in red if the correlation coefficient is positive, in blue if negative, in white if not significant (accordin to the _threshold_ parameter)
pch	the _pch_ parameter indicating the points shape. Default is 20
subtitle	NULL by default, in which case the function will print as a subtitle the correlation coefficient (CC) and its pvalue. Otherwise, a user-provided string, bypassing the predefined subtitle

extendXlim	logical. If TRUE, the x-axis limits are extended by a fraction (useful for labeling points on the margins of the plot area). Default is FALSE
ci	logical. If TRUE, confidence intervals of linear regression are shown at 95 percent confidence.
...	Arguments to be passed to the core _plot_ function (if a new plot is created)

Value

A plot

Examples

```
x<-setNames(rnorm(200),paste0("var",1:200))
y<-setNames(rnorm(210),paste0("var",11:220))
scatter(x,y,xlab="Variable x",ylab="Variable y",main="Scatter plot by corto package",ci=TRUE)
```

---

scinot *scinot - Convert a number to a scientific notation expression*

Description

This function will convert any numeric vector

Usage

```
scinot(v, digits = 3)
```

Arguments

v	The input numeric object. It can be a single value or a vector
digits	An integer indicating how many significant digits to show. Default is 3.

Value

An object of class _expression_.

Examples

```
# Usage on single value
scinot(0.00000543)
# Demonstration on a vector
numbers<-c(3.456e-12,0.00901,5670000,-3.16e18,0.000004522,rnorm(5,sd=0.0000001))
plot(0,xlim=c(0,10),ylim=c(0,10),type="n")
text(c(2,6),c(10,10),labels=c("Before","After"),font=2)
for(i in 10:1){
    text(c(2,6),c(i-1,i-1),labels=c(numbers[i],scinot(numbers)[i]))
}
```

slice *Slice*

Description

This function prints a slice of a matrix

Usage

```
slice(matrix)
```

Arguments

matrix	A matrix

Value

A visualization of the first 5 rows and columns of the input matrix

Examples

```
set.seed(1)
example<-matrix(rnorm(1000),nrow=100,ncol=10)
slice(example)
```

ssgsea *ssGSEA*

Description

This function performs single sample GSEA

Usage

```
ssgsea(inmat, groups, scale = TRUE, minsize = 10)
```

Arguments

inmat	A numeric matrix, with rownames/rows as genes or features, and colnames/columns as sample names
groups	a named list. Names are names of the groups (e.g. pathways) and elements are character vectors indicating gene or feature names (that should match, at least partially, with the rownames of inmat)
scale	Boolean. Wheter the matrix should be row-scaled.
minsize	Numeric. Include only groups with at least this many elements Default is 10

Value

A matrix of Normalized Enrichment Scores (NES), which can be converted to p-values using the function _corto::z2p_

Examples

```
# A random matrix
set.seed(1)
inmat<-matrix(rnorm(200*50),nrow=200,ncol=50)
rownames(inmat)<-paste0("gene",1:nrow(inmat))
# A random list of groups
groups<-list()
for(i in 1:10){
    somegenes<-sample(rownames(inmat),30)
    groups[[paste0("pathway_",i)]]<-somegenes
}
# Run ssGSEA
nesmat<-ssgsea(inmat,groups)
```

---

stouffer *Stouffer integration of Z scores*

Description

This function gives a gaussian Z-score corresponding to the provided p-value Careful: sign is not provided

Usage

```
stouffer(x)
```

Arguments

x	a vector of Z scores

Value

Z an integrated Z score

Examples

```
zs<-c(1,3,5,2,3)
stouffer(zs)
```

textrepel *textrepel - Plot text with non-overlapping labels*

Description

This function plots text with x and y coordinates, forcing overlapping labels to not overlap

Usage

```
textrepel(
  x,
  y,
  labels = NULL,
  padding = " ",
  rstep = 0.1,
  tstep = 0.1,
  vertical = FALSE,
  textSize = 1,
  showLines = TRUE,
  lineColor = "#00000066",
  lineWidth = 2,
  showPoints = TRUE,
  pointColor = "#00000033",
  pointSize = 2,
  pointPch = 16,
  add = FALSE,
  ...
)
```

Arguments

x	A numeric vector of x coordinates
y	A numeric vector of y coordinates (must have the same length of x)
labels	A vector of labels associated with x and y (must have the same length of x)
padding	A character object specifying left and right padding for words. Default is a single whitespace " "
rstep	Decimal numeric specifying the lateral step length for label distancing. Default is 0.1
tstep	Decimal numeric specifying the theta step length for label distancing. Default is 0.1
vertical	Boolean. If FALSE (default), the labels are plotted horizontally. If TRUE, vertically
textSize	Numeric. Size of text. Default is 1
showLines	Boolean. Whether to show lines connecting displaced labels to their original plot. Default is TRUE

lineColor	String indicating the color of the connecting line
lineWidth	Numeric indicating the width of the connecting line
showPoints	Boolean. Whether to show points over original x-y coordinates
pointColor	String indicating the color of the point
pointSize	Numeric indicating the size of the point
pointPch	Integer applying to shape of points. Default is 16 (filled circle)
add	Boolean. If FALSE (default), a new plot is generated. If TRUE, the textrepel labels are plotted over the existing plot
...	Arguments to be passed to the core _plot_ function

Value

A plot

Examples

```
# Simple example, generating a new plot, taking care of some overlapping labels
set.seed(1)
x<-rnorm(100)
y<-abs(x)+rnorm(100)
names(x)<-names(y)<-paste0("OBJ",1:length(x))
labels<-names(x)
textrepel(x,y,labels)
# More advanced example, adding textrepel over an existing plot
set.seed(1)
x<-rnorm(1000)
y<-abs(x)+rnorm(1000)
names(x)<-names(y)<-paste0("GENE",1:length(x))
labels<-names(x)
plot(x,y,pch=16,col="#00000066",xlim=1.3*c(min(x),max(x)))
subset1<-which(x<(-2.2))
textrepel(x[subset1],y[subset1],labels[subset1],add=TRUE,pointCol="cornflowerblue")
subset2<-which(x>(+2.2))
textrepel(x[subset2],y[subset2],labels[subset2],add=TRUE,pointCol="salmon")
```

---

val2col *val2col - Convert a numeric vector into colors*

Description

val2col - Convert a numeric vector into colors

Usage

```
val2col(
  z,
  col1 = "navy",
  col2 = "white",
  col3 = "red3",
  nbreaks = 1000,
  center = TRUE,
  rank = FALSE
)
```

Arguments

z	a vector of numbers
col1	a color name for the min value, default 'navy'
col2	a color name for the middle value, default 'white'
col3	a color name for the max value, default 'red3'
nbreaks	Number of colors to be generated. Default is 30.
center	boolean, should the data be centered? Default is TRUE
rank	boolean, should the data be ranked? Default is FALSE

Value

a vector of colors

Examples

```
a<-rnorm(1000)
cols<-val2col(a)
plot(a,col=cols,pch=16)
```

---

wstouffer *Weighted Stouffer integration of Z scores*

Description

This function gives a gaussian Z-score corresponding to the provided p-value Careful: sign is not provided

Usage

```
wstouffer(x, w)
```

Arguments

x	a vector of Z scores
w	weight for each Z score

Value

Z an integrated Z score

Examples

```
zs<-c(1,-3,5,2,3)
ws<-c(1,10,1,2,1)
wstouffer(zs,ws)
```

---

z2p *z2p*

Description

This function gives a gaussian p-value corresponding to the provided Z-score

Usage

```
z2p(z)
```

Arguments

z	a Z score

Value

a p-value

Examples

```
z<-1.96
z2p(z)
```

Index