

Package ‘corto’

October 12, 2022

Type Package

Title Inference of Gene Regulatory Networks

Version 1.1.11

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](https://doi.org/10.1093/bioinformatics/btaa223)>.

License LGPL-3

Encoding UTF-8

RoxygenNote 7.2.0

Depends R (>= 3.6)

NeedsCompilation no

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

VignetteBuilder knitr

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

Repository CRAN

Date/Publication 2022-05-23 15:30:20 UTC

R topics documented:

barplot2	2
corto	3
fcor	4
fisherp	5
gsea	5
gsea2	6
kmgformat	7
mra	8
mrplot	9
p2r	10
p2z	11
plot_gsea	11
plot_gsea2	12
r2p	13
scatter	14
slice	15
ssgsea	16
stouffer	17
val2col	17
wstouffer	18
z2p	19
Index	20

barplot2	<i>barplot2 - Bar plot with error bars</i>
----------	--

Description

barplot2 - Bar plot with error bars

Usage

```
barplot2(values, errors, ...)
```

Arguments

values	A matrix of values
errors	A matrix of values for upper error bar
...	Arguments to be passed to the core <code>_barplot_</code> 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

<code>inmat</code>	Input matrix, with features (e.g. genes) as rows and samples as columns
<code>centroids</code>	A character vector indicating which features (e.g. genes) to consider as centroids (a.k.a. Master Regulators) for DPI
<code>nbootstraps</code>	Number of bootstraps to be performed. Default is 100
<code>p</code>	The p-value threshold for correlation significance (by default 1E-30)
<code>nthreads</code>	The number of threads to use for bootstrapping. Default is 1
<code>verbose</code>	Logical. Whether to print progress messages. Default is FALSE
<code>cnvmat</code>	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 (<code>exp</code>) and the <code>cnv</code> data, and target profiles will be transformed to the residuals of each linear model <code>exp~cnv</code> . Default is NULL
<code>boot_threshold</code>	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

<code>inmat</code>	An input matrix with features as rows and samples as columns
<code>centroids</code>	A character vector indicating the centroids
<code>r</code>	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	<i>Fisher integration of p-values</i>
---------	---------------------------------------

Description

This function applies the Fisher integration of p-values

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	<i>GSEA</i>
------	-------------

Description

This function performs Gene Set Enrichment Analysis

Usage

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

Arguments

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

Value

A GSEA object. Basically a list of s components:

ES The enrichment score

NES The normalized enrichment score

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
```

<code>gsea2</code>	<i>2-way GSEA GSEA Gene set enrichment analysis of two complementary gene sets using gsea</i>
--------------------	---

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 score

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 <code>_regulon_</code> object, output of the <code>_corto_</code> 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) $-\log_{10}(\text{FDR}) \cdot \text{sing}(\log_2\text{FC})$ of an ATAC-Seq design, 3) distance from TSS. If provided, the output will contain an <code>_atacseq_</code> 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"
)
```

Arguments

<code>gsea.obj</code>	GSEA object produced by the <code>gsea</code> function
<code>twoColors</code>	the two colors to use for positive[1] and negative[2] enrichment scores, and of the barcodes
<code>plotNames</code>	Logical. Should the set names be plotted?
<code>title</code>	String to be plotted above the Running Enrichment Score
<code>bottomTitle</code>	String for the title of the bottom part of the plot
<code>bottomYlabel</code>	String for the Y label of the bottom plot (FALSE by default)

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

<code>r</code>	the correlation coefficient
<code>n</code>	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,
  ...
)
```

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 (according to the <code>_threshold_</code> parameter)
pch	the <code>_pch_</code> 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

`...` Arguments to be passed to the core `_plot_` function

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")
```

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

<code>inmat</code>	A numeric matrix, with rownames/rows as genes or features, and colnames/columns as sample names
<code>groups</code>	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 <code>inmat</code>)
<code>scale</code>	Boolean. Whether the matrix should be row-scaled.
<code>minsize</code>	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	<i>Stouffer integration of Z scores</i>
----------	---

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)
```

val2col	<i>val2col - Convert a numeric vector into colors</i>
---------	---

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

barplot2, 2

corto, 3

fcor, 4

fisherp, 5

gsea, 5

gsea2, 6

kmgformat, 7

mra, 8

mrplot, 9

p2r, 10

p2z, 11

plot_gsea, 11

plot_gsea2, 12

r2p, 13

scatter, 14

slice, 15

ssgsea, 16

stouffer, 17

val2col, 17

wstouffer, 18

z2p, 19