Package ‘ccdf’

Type Package
Title Distribution-Free Single-Cell Differential Expression Analysis
Version 1.1.4
Imports pbapply, parallel, matrixStats, CompQuadForm, RcppNumerical, doParallel, foreach, ggplot2, randomForest, rpart, statmod, viridisLite, survey, cowplot
Maintainer Marine Gauthier <marine.gauthier@u-bordeaux.fr>
License GPL (>= 3)
Encoding UTF-8
RoxygenNote 7.1.1
Depends R (>= 3.6)
BugReports https://github.com/mgauth/ccdf/issues
NeedsCompilation no
Author Marine Gauthier [aut, cre], Denis Agniel [aut], Boris P. Hejblum [aut]
Repository CRAN
Date/Publication 2021-09-24 08:00:05 UTC

R topics documented:

CCDF ................................................................. 2
ccdf_testing ....................................................... 3
perm_cont ......................................................... 5
plot_CCDF ......................................................... 6
plot_pvals ......................................................... 7
CCDF

Function to compute (un)conditional cumulative distribution function (CDF), used by plot_CCDF function.

Description

Function to compute (un)conditional cumulative distribution function (CDF), used by plot_CCDF function.

Usage

CCDF(
  Y, 
  X, 
  Z = NULL, 
  method = c("linear regression", "logistic regression", "RF"), 
  fast = TRUE, 
  space_y = FALSE, 
  number_y = length(Y) 
)

Arguments

Y  a numeric vector of size n containing the preprocessed expressions from n samples (or cells).
X  a data frame containing numeric or factor vector(s) of size n containing the variable(s) to be tested (the condition(s) to be tested).
Z  a data frame containing numeric or factor vector(s) of size n containing the covariate(s).
method  a character string indicating which method to use to compute the CCDF, either 'linear regression', 'logistic regression' and 'permutations' or 'RF' for Random Forests. Default is 'linear regression' since it is the method used in the test.
fast  a logical flag indicating whether the fast implementation of logistic regression should be used. Only if 'dist_permutations' is specified. Default is TRUE.
space_y  a logical flag indicating whether the y thresholds are spaced. When space_y is TRUE, a regular sequence between the minimum and the maximum of the observations is used. Default is FALSE.
number_y  an integer value indicating the number of y thresholds (and therefore the number of regressions) to perform the test. Default is length(Y).
Value

A list with the following elements:

- `cdf`: a vector of the cumulative distribution function of a given gene.
- `ccdf`: a vector of the conditional cumulative distribution function of a given gene, computed given X. Only if Z is NULL.
- `ccdf_nox`: a vector of the conditional cumulative distribution function of a given gene, computed given Z only (i.e. X is ignored.). Only if Z is not NULL.
- `ccdf_x`: a vector of the conditional cumulative distribution function of a given gene, computed given X and Z. Only if Z is not NULL.
- `y_sort`: a vector of the sorted expression points at which the CDF and the CCDFs are calculated.
- `x_sort`: a vector of the variables associated with `y_sort`.
- `z_sort`: a vector of the covariates associated with `y_sort`. Only if Z is not NULL.

Examples

```r
X <- as.factor(rbinom(n=100, size = 1, prob = 0.5))
Y <- ((X==1)*rnorm(n = 50,0,1)) + ((X==0)*rnorm(n = 50,0.5,1))
res <- CCDF(Y, data.frame(X=X), method="linear regression")
```

ccdf_testing

Main function to perform complex hypothesis testing using (un)conditional independence test

Description

Main function to perform complex hypothesis testing using (un)conditional independence test

Usage

```r
ccdf_testing(
  exprmat = NULL,
  variable2test = NULL,
  covariate = NULL,
  distance = c("L2", "L1", "L_sup"),
  test = c("asymptotic", "permutations", "dist_permutations"),
  method = c("linear regression", "logistic regression", "RF"),
  fast = TRUE,
  n_perm = 100,
  n_perm_adaptive = c(100, 150, 250, 500),
  thresholds = c(0.1, 0.05, 0.01),
  parallel = TRUE,
  n_cpus = NULL,
)```

adaptive = FALSE,
space_y = FALSE,
number_y = ncol(exprmat)
)

Arguments

exprmat a data frame of size $G \times n$ containing the preprocessed expressions from $n$ samples (or cells) for $G$ genes. Default is NULL.

variable2test a data frame of numeric or factor vector(s) of size $n$ containing the variable(s) to be tested (the condition(s))

covariate a data frame of numeric or factor vector(s) of size $n$ containing the covariate(s)

distance a character string indicating which distance to use to compute the test, either 'L2', 'L1' or 'L_sup', when method is 'dist_permutations'. Default is 'L2'.

test a character string indicating which method to use to compute the test, either 'asymptotic', 'permutations' or 'dist_permutations'. 'dist_permutations' allows to compute the distance between the CDF and the CCDF or two CCDFs. Default is 'asymptotic'.

method a character string indicating which method to use to compute the CCDF, either 'linear regression', 'logistic regression' and 'permutations' or 'RF' for Random Forests. Default is 'linear regression' since it is the method used in the test.

fast a logical flag indicating whether the fast implementation of logistic regression should be used. Only if 'dist_permutations' is specified. Default is TRUE.

n_perm the number of permutations. Default is 100.

n_perm_adaptive a vector of the increasing numbers of adaptive permutations when adaptive is TRUE. length(n_perm_adaptive) should be equal to length(thresholds)+1. Default is c(0.1,0.05,0.01).

thresholds a vector of the decreasing thresholds to compute adaptive permutations when adaptive is TRUE. length(thresholds) should be equal to length(n_perm_adaptive)-1. Default is c(100,150,250,500).

parallel a logical flag indicating whether parallel computation should be enabled. Default is TRUE.

n_cpus an integer indicating the number of cores to be used when parallel is TRUE. Default is parallel::detectCores() - 1.

adaptive a logical flag indicating whether adaptive permutations should be performed. Default is FALSE.

space_y a logical flag indicating whether the y thresholds are spaced. When space_y is TRUE, a regular sequence between the minimum and the maximum of the observations is used. Default is FALSE.

number_y an integer value indicating the number of y thresholds (and therefore the number of regressions) to perform the test. Default is ncol(exprmat).
perm_cont

Value

A list with the following elements:

- **which_test**: a character string carrying forward the value of the 'which_test' argument indicating which test was performed (either 'asymptotic','permutations','dist_permutations').
- **n_perm**: an integer carrying forward the value of the 'n_perm' argument or 'n_perm_adaptive' indicating the number of permutations performed (NA if asymptotic test was performed).
- **pval**: computed p-values. A data frame with one raw for each gene, and with 2 columns: the first one 'raw_pval' contains the raw p-values, the second one 'adj_pval' contains the FDR adjusted p-values using Benjamini-Hochberg correction.

References


Examples

```r
X <- as.factor(rbinom(n=100, size = 1, prob = 0.5))
Y <- t(replicate(10, ((X==1)*rnorm(n = 50,0,1)) + ((X==0)*rnorm(n = 50,0.5,1))))
res_asymp <- ccdf_testing(exprmat=data.frame(Y=Y),
variable2test=data.frame(X=X), test="asymptotic",
n_cpus=1)$pvals # asymptotic test
```

---

perm_cont  

Permutation procedure when Z is continuous

Description

Permutation procedure when Z is continuous

Usage

```r
perm_cont(Y, X, Z)
```

Arguments

- **Y**: a numeric vector of size n containing the preprocessed expressions from n samples (or cells).
- **X**: a numeric or factor vector of size n containing the variable to be tested (the condition to be tested).
- **Z**: a numeric vector of size n containing the covariate. Multiple variables are not allowed.
Value

X_star a vector of permuted X.

Examples

if(interactive()){
  X <- rbinom(n=100, size = 1, prob = 0.5)
  Z <- rnorm(100,0,1)
  Y <- ((X==1)*rnorm(n = 50,0,1)) + ((X==0)*rnorm(n = 50,0.5,1))
  res <- perm_cont(Y,X,Z)
}

plot_CCDF

Function to plot the CCDF according to the type of X et Z

Description

Function to plot the CCDF according to the type of X et Z

Usage

plot_CCDF(
  Y,
  X,
  Z = NULL,
  method = "linear regression",
  fast = TRUE,
  space_y = FALSE,
  number_y = length(Y)
)

Arguments

Y a numeric vector of size n containing the preprocessed expressions from n samples (or cells).
X a numeric or factor vector of size n containing the variable to be tested (the condition to be tested).
Z a numeric or factor vector of size n containing the covariate. Multiple variables are not allowed.
method a character string indicating which method to use to compute the CCDF, either 'linear regression', 'logistic regression' and 'permutations' or 'RF' for Random Forests. Default is 'linear regression' since it is the method used in the test.
fast a logical flag indicating whether the fast implementation of logistic regression should be used. Only if 'dist_permutations' is specified. Default is TRUE.
plot_pvals

space_y
a logical flag indicating whether the y thresholds are spaced. When space_y is TRUE, a regular sequence between the minimum and the maximum of the observations is used. Default is FALSE.

number_y
an integer value indicating the number of y thresholds (and therefore the number of regressions) to perform the test. Default is length(Y).

Value
a ggplot object

Examples

X <- as.factor(rbinom(n=100, size = 1, prob = 0.5))
Y <- ((X==1)*rnorm(n = 50,0,1)) + ((X==0)*rnorm(n = 50,0,5,1))
plot_CCDF(data.frame(Y=Y),data.frame(X=X),method="linear regression")

plot_pvals
Plot of gene-wise p-values

Description
This function prints the sorted exact p-values along with the Benjamini-Hochberg limit and the 5

Usage
plot_pvals(pvals)

Arguments
pvals
a vector of length n containing the raw p-values for each gene

Value
a plot of sorted gene-wise p-values
a ggplot object

Examples

plot_pvals(runif(100,0,1))
test_asymp

Asymptotic test

description

Asymptotic test

usage

test_asymp(Y, X, Z = NULL, space_y = FALSE, number_y = length(unique(Y)))

arguments

Y a numeric vector of size \( n \) containing the preprocessed expression for a given gene from \( n \) samples (or cells).

X a data frame of numeric or factor vector(s) of size \( n \) containing the variable(s) to be tested (the condition(s))

Z a data frame of numeric or factor vector(s) of size \( n \) containing the covariate(s)

space_y a logical flag indicating whether the y thresholds are spaced. When \( \text{space}_y \) is TRUE, a regular sequence between the minimum and the maximum of the observations is used. Default is FALSE.

number_y an integer value indicating the number of y thresholds (and therefore the number of regressions) to perform the test. Default is length(Y).

value

A data frame with the following elements:

- \text{raw}_pval contains the raw p-values for a given gene.
- \text{Stat} contains the test statistic for a given gene.

examples

\begin{verbatim}
X <- as.factor(rbinom(n=100, size = 1, prob = 0.5))
Y <- ((X==1)*rnorm(n = 50,0,1)) + ((X==0)*rnorm(n = 50,0.5,1))
res_asymp <- test_asymp(Y, data.frame(X=X))
\end{verbatim}
**Description**

Permutation test

**Usage**

```r
test_perm(
  Y,
  X,
  Z = NULL,
  n_perm = 100,
  parallel = FALSE,
  n_cpus = NULL,
  space_y = FALSE,
  number_y = length(Y)
)
```

**Arguments**

- `Y`: a numeric vector of size `n` containing the preprocessed expression for a given gene from `n` samples (or cells).
- `X`: a data frame of numeric or factor vector(s) of size `n` containing the variable(s) to be tested (the condition(s)). Multiple variables are not allowed.
- `Z`: a data frame of numeric or factor vector(s) of size `n` containing the covariate(s). Multiple variables are not allowed.
- `n_perm`: the number of permutations. Default is `100`.
- `parallel`: a logical flag indicating whether parallel computation should be enabled. Default is `TRUE`.
- `n_cpus`: an integer indicating the number of cores to be used when `parallel` is `TRUE`. Default is `parallel::detectCores() - 1`.
- `space_y`: a logical flag indicating whether the `y` thresholds are spaced. When `space_y` is `TRUE`, a regular sequence between the minimum and the maximum of the observations is used. Default is `FALSE`.
- `number_y`: an integer value indicating the number of `y` thresholds (and therefore the number of regressions) to perform the test. Default is `length(Y)`.

**Value**

A data frame with the following elements:

- `score` contains the test statistic for a given gene.
- `raw_pval` contains the raw p-values for a given gene computed from `n_perm` permutations.
Examples

if(interactive()){
  X <- as.factor(rbinom(n=100, size = 1, prob = 0.5))
  Y <- ((X==1)*rnorm(n = 50,0,1)) + ((X==0)*rnorm(n = 50,0.5,1))
  res_perm <- test_perm(Y, data.frame(X=X), n_perm=10)
}
Index

CCDF, 2
ccdf_testing, 3

ggplot, 7

perm_cont, 5
plot_CCDF, 6
plot_pvals, 7

test_asymp, 8
test_perm, 9