Package 'INDEED'

April 16, 2019

Title An Implementation of Integrated Differential Expression and Differential Network Analysis for Biomarker Candidate Selection

Version 1.0.1

Author Zhenzhi Li <zzrickli@gmail.com>

Maintainer Zhenzhi Li <zzrickli@gmail.com>, Yiming Zuo <yimingzuo@gmail.com>

Description An Implementation of Integrated Differential Expression and Differential Network Analysis of Omic Data. The differential network is obtained based on partial correlation or correlation.

License Artistic-2.0

URL http://github.com/ressomlab/INDEED

BugReports http://github.com/ressomlab/INDEED/issues

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Depends R (>= 3.5), glasso (>= 1.8)

Imports utils (>= 3.3.1), stats (>= 3.3.1), graphics (>= 3.3.1), devtools (>= 1.13.0)

Suggests knitr (>= 1.19), rmarkdown (>= 1.8), testthat (>= 2.0.0)

VignetteBuilder knitr

biocViews ImmunoOncology, Software, ResearchField, BiologicalQuestion, StatisticalMethod, DifferentialExpression, MassSpectrometry, Metabolomics

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

git_branch RELEASE_3_8

git_last_commit d8c8869

git_last_commit_date 2019-01-04

Date/Publication 2019-04-15

R topics documented:

`choose_rho` *Draw error curve*

Description

Draw error curve using cross-validation.

Usage

```
choose_rho(data, n_fold, rho)
```

Arguments

`data`	a matrix.
`n_fold`	specify n to n-fold cross_validation.
`rho`	multiple regularization parameter values to be evalueated in terms of errors.

Value

a list of errors and their corresponding $log(rho)$

compute_cor *Compute the correlation*

Description

Compute either pearson or spearman correlation coefficient.

Usage

```
compute_cor(data_group_2, data_group_1, type_of_cor)
```

Arguments

data_group_2	a n*p matrix.
data_group_1	a n*p matrix
type_of_cor	if NULL, pearson correlation coefficient will be calculated. Otherwise, a character string "spearman" to calculate spearman correlation coefficient.

Value

A list of correlation matrix for both group 1 and group 2

compute_dns *Calculate differential network score*

Description

Calculate differential network score.

Usage

```
compute_dns(binary_link, z_score)
```

Arguments

binary_link	binary correlation matrix with 1 indicating positive correlation and -1 indicating negative correlation for each biomolecular pair.
z_score	converted from p-value.

Value

An activity score associated with each biomarker candidate

compute_par *Compute the partial correlation*

Description

Compute the partial correlation coefficient.

Usage

```
compute_par(pre_inv)
```

Arguments

pre_inv	an inverse covariance matrix.

Value

An $nbyn$ partial correlation matrix

INDEED *INDEED: A package for biomarker candidate prioritization.*

Description

The INDEED package provides a important functions below: pre_partial

pre_partial function

pre_partical function preprocess data for partical correlation analysis, the result contains list of preprocessed data and rho values and error plot for user to choose desired rho value

partial_cor function

partial_cor function performs partical correlation analysis user input preprocessed list from pre_partical step and the rho choosing method or rho of their choice and number of permutations (default 1000), p-value is optional the result of score table and differential network will be returned

non_partial_cor function

non_partial_cor function performs correlation analysis user input data,class label,p-value, sample id, number of permutations, and method(default pearson) p value is optional the result of score table and differential network will be returned

loglik_ave *Create log likelihood error function*

Description

Calculate log likelihood error function.

Usage

```
loglik_ave(data, theta)
```

Arguments

data	a matrix or data.frame.
theta	a precision matrix.

Value

log likelihood error function

Met_Group_GU *Group label.*

Description

A dataset containing group information group 1: 0; group 2: 1.

Usage

```
Met_Group_GU
```

Format

A data frame with 1 row and 120 (subjects) columns.

Met_GU *GU CIRR and GU HCC combined .*

Description

A dataset containing the expression levels for each of the 120 subjects (HCC: 60; CIRR: 60) in terms of 39 metabolites.

Usage

```
Met_GU
```

Format

A data frame with 39 variables (rows) and 120 subjects (columns).

Met_name_GU *KEGG ID*

Description

A dataset containing the KEGG ID for each metabolite.

Usage

```
Met_name_GU
```

Format

A data frame with 39 KEGG ID as rows and 1 column:

non_partial_cor *Non-partial correlaton analysis*

Description

A method that integrates differential expression (DE) analysis and differential network (DN) analysis to select biomarker candidates for survival time prediction. non_partial_cor is a one step function for user to perform analysis, no pre-processing step required

Usage

```
non_partial_cor(data = NULL, class_label = NULL, id = NULL,
  method = "pearson", p_val = NULL, permutation = 1000)
```

Arguments

data,	input matrix of expression from all metabolites from all samples
class_label,	a binary array with 0: group 1; 1: group 2.
id,	an array of biomolecule ID to label.
method	a character string indicating which correlation coefficient is to be computed. One of "pearson" (default) or "spearman".
p_val	optional, a dataframe contains p values for each metabolite/molecule
permutation,	a positive integer of desired number of permutations, default 1000

Value

a list of processed data for next step and rho

Examples

```
non_partial_cor(data=Met_GU,class_label = Met_Group_GU,id=Met_name_GU,
   method="spearman")
```

partial_cor *Data preprocessing for partial correlaton analysis*

Description

A method that integrates differential expression (DE) analysis and differential network (DN) analysis to select biomarker candidates for survival time prediction. partial_cor is the second step of partial correlation calculation after the output result from pre_partial function

Usage

```
partial_cor(data_list = NULL, rho_group1 = NULL, rho_group2 = NULL,
  permutation = 1000, p_val = NULL)
```

Arguments

data_list,	list of pre-processed data from pre_partial function
rho_group1	rule to choose rho for group 1, "min": minimum rho, "ste" one standard error from minimum, or user can input rho of their choice, default: minimum
rho_group2	rule to choose rho for group 1, "min": minimum rho, "ste" one standard error from minimum, or user can input rho of their choice, default: minimum
permutation,	a positive integer of desired number of permutations, default 1000
p_val	optional, a dataframe contains p values for each metabolite/molecule

Value

a list containing a score dataframe and a differential network dataframe

Examples

```
preprocess<- pre_partial(data=Met_GU,class_label = Met_Group_GU,id=Met_name_GU)
   partial_cor(data_list=preprocess,rho_group1='min',
   rho_group2="min",permutation = 1000,p_val=pvalue_M_GU)
```

permutation_cor *Permutations to build differential network using correlation*

Description

A permutation test that randomly permutes the sample labels in distinct biological groups for each biomolecule. The difference in each paired biomolecule is considered significant if it falls into the 2.5 distribution curve.

Usage

```
permutation_cor(m, p, n_group_1, n_group_2, data_group_1, data_group_2,
  type_of_cor)
```

Arguments

m	number of permutations.
p	number of biomarker candidates.
n_group_1	number of subjects in group 1.
n_group_2	number of subjects in group 2.
data_group_1	a $n * p$ matrix or data.frame containing group 1 data.
data_group_2	a $n * p$ matrix of data.frame containing group 2 data.
type_of_cor	if NULL, pearson correlation coefficient will be calculated. Otherwise, a character string "spearman" to calculate spearman correlation coefficient.

Value

A multi-dimensional matrix that contains the permutation results

---

permutation_pc *Permutations to build differential network using partial correlation*

Description

A permutation test that randomly permutes the sample labels in distinct biological groups for each biomolecule. The difference in paired partial correlation is considered significant if it falls into the 2.5 distribution curve.

Usage

```
permutation_pc(m, p, n_group_1, n_group_2, data_group_1, data_group_2,
  rho_group_1_opt, rho_group_2_opt)
```

Arguments

m	number of permutations.
p	number of biomarker candidates.
n_group_1	number of subjects in group 1.
n_group_2	number of subjects in group 2.
data_group_1	a $n * p$ matrix or data.frame containing group 1 data.
data_group_2	a $n * p$ matrix of data.frame containing group 2 data.
rho_group_1_opt	optimal tuning parameter to sparse the differential network for group 1
rho_group_2_opt	optimal tuning parameter to sparse the differential network for group 2

Value

A multi-dimensional matrix that contains the permutation results

permutation_thres — *Calculate the positive and negative threshold based on the permutation result*

Description

Calculate the positive and negative threshold based on the permutation result.

Usage

```
permutation_thres(thres_left, thres_right, p, diff_p)
```

Arguments

thres_left	2.5 percent left tails.
thres_right	2.5 percent right tails.
p	number of biomarker candidates.
diff_p	permutation results.

Value

A list of positive and negative threshold

pre_partial — *Data preprocessing for partial correlaton analysis*

Description

A method that integrates differential expression (DE) analysis and differential network (DN) analysis to select biomarker candidates for survival time prediction. pre_partial is the pre-processing step for INDEED partial differential analysis

Usage

```
pre_partial(data = NULL, class_label = NULL, id = NULL)
```

Arguments

data	input matrix of expression from all metabolites from all samples
class_label	a binary array with 0: group 1; 1: group 2.
id	an array of biomolecule ID to label.

Value

a list of processed data for next step and rho, error curve for group 1 and 2

Examples

```
pre_partial(data=Met_GU,class_label = Met_Group_GU,id=Met_name_GU)
```

pvalue_logit *Obtain p-values using logistic regression*

Description

Calculate p-values using logistic regression.

Usage

```
pvalue_logit(x, class_label, Met_name)
```

Arguments

x	a data frame consists of data from group 1 and group 2.
class_label	a binary array indicating 0: group 1; 1: group 2.
Met_name	an array of ID.

Value

p-values

pvalue_M_GU *P-values obtained by differential expression (DE) analysis.*

Description

A dataset containing the p-values of each metabolite obtained through DE.

Usage

```
pvalue_M_GU
```

Format

A data frame with 39 rows and 3 variables:

KEGG.ID KEGG.ID

p.value p-values

Index