Package ‘INDEED’

April 1, 2024

Title Interactive Visualization of Integrated Differential Expression and Differential Network Analysis for Biomarker Candidate Selection Package

Version 2.16.0

Author Yiming Zuo <yimingzuo@gmail.com>, Kian Ghaffari <kg.ghaffari@gmail.com>, Zhenzhi Li <zzrickli@gmail.com>

Maintainer Ressom group <hwr@georgetown.edu>, Yiming Zuo <yimingzuo@gmail.com>

Description An R package for integrated differential expression and differential network analysis based on omic data for cancer biomarker discovery. Both correlation and partial correlation can be used to generate differential network to aid the traditional differential expression analysis to identify changes between biomolecules on both their expression and pairwise association levels. A detailed description of the methodology has been published in Methods journal (PMID: 27592383). An interactive visualization feature allows for the exploration and selection of candidate biomarkers.

License Artistic-2.0

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

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

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

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

Imports devtools (>= 1.13.0), graphics (>= 3.3.1), stats (>= 3.3.1), utils (>= 3.3.1), igraph (>= 1.2.4), visNetwork(>= 2.0.6)

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
choose_rho

**Description**

This function draws error curve using cross-validation.

**Usage**

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

**Arguments**

- `data` This is a matrix.
- `n_fold` This parameter specifies the n number in n-fold cross-validation.
- `rho` This is the regularization parameter values to be evaluated in terms their errors.
compute_cor

Value
A list of errors and their corresponding $\log(\rho)$.

Description
This function computes either the pearson or spearman correlation coefficient.

Usage
compute_cor(data_group_1, data_group_2, type_of_cor)

Arguments
- data_group_1: This is a n*p matrix.
- data_group_2: This is a n*p matrix.
- type_of_cor: If this is NULL, pearson correlation coefficient will be calculated as default. Otherwise, a character string "spearman" will calculate the spearman correlation coefficient.

Value
A list of correlation matrices for both group 1 and group 2.

calculate_dns

Calculate the differential network score

Description
This function calculates differential network score by using the binary link and z-scores.

Usage
calculate_dns(binary_link, z_score)

Arguments
- binary_link: This is the binary correlation matrix with 1 indicating positive correlation and -1 indicating negative correlation for each biomolecular pair.
- z_score: This is converted from the given or calculated p-value.

Value
An activity score associated with each biomarker candidate.
**compute_par** *Compute the partial correlation*

**Description**

This function computes the partial correlation coefficient.

**Usage**

```r
compute_par(pre_inv)
```

**Arguments**

- `pre_inv` This is an inverse covariance matrix.

**Value**

A $p \times p$ partial correlation matrix.

---

**indeed**

**indeed: A network-based method for cancer biomarker discovery.**

**Description**

The INDEED R package provides important functions as shown below: `non_partial_cor()`, `select_rho_partial()`, `partial_cor()`, and `network_display()`.

**non_partial_cor function**

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

**select_rho_partial function**

`select_rho_partial` function preprocesses data for partial correlation analysis, the result contains list of preprocessed data and rho values and error plot for user to choose desired rho value for graphical lasso.

**partial_cor function**

`partial_cor` function performs partial correlation analysis based on user input preprocessed list from `select_rho_partial` step and the rho choosing method or values of their choice and number of permutations (default 1000), p-value is optional, the result of score table and differential network will be returned.
**network_display function**

A function to assist in the network visualization of the result from INDEED functions non_partial_cor() and partial_cor().

---

**loglik_ave**

*Create log likelihood error*

**Description**

This function calculates the log likelihood error.

**Usage**

`loglik_ave(data, theta)`

**Arguments**

- `data` This is a matrix.
- `theta` This is a precision matrix.

**Value**

log likelihood error

---

**Met_Group_GU**

*Group label.*

**Description**

A dataset containing group information (CIRR group: 0 and HCC group: 1).

**Usage**

Met_Group_GU

**Format**

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

*GU cirrhosis (CIRR) and GU Hepatocellular carcinoma (HCC) data.*

**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:

---

**network_display**

*Interactive Network Visualization*

**Description**

An interactive tool to assist in the visualization of the results from INDEED functions `non_partial_corr()` or `partial_corr()`. The size and the color of each node can be adjusted by users to represent either the Node_Degree, Activity_Score, Z_Score, or P_Value. The color of the edge is based on the binary value of either 1 corresponding to a positive correlation depicted as green or a negative correlation of -1 depicted as red. The user also has the option of having the width of each edge be proportional to its weight value. The layout of the network can also be customized by choosing from the options: 'nice', 'sphere', 'grid', 'star', and 'circle'. Nodes can be moved and zoomed in on. Each node and edge will display extra information when clicked on. Secondary interactions will be highlighted as well when a node is clicked on.
non_partial_cor

Usage

network_display(results = NULL, nodesize = "P_Value", nodecolor = "Activity_Score", edgewidth = "NO", layout = "nice")

Arguments

results This is the result from calling either non_partial_cor() or partial_corr().

nodesize This parameter determines what the size of each node will represent. The options are 'Node_Degree', 'Activity_Score', 'P_Value' and 'Z_Score'. The title of the resulting network will identify which parameter was selected to represent the node size. The default is P_Value.

nodecolor This parameter determines what color each node will be based on a yellow to blue color gradient. The options are 'Node_Degree', 'Activity_Score', 'P_Value', and 'Z_Score'. A color bar will be created based on which parameter is chosen. The default is Activity_Score.

edgewidth This is a 'YES' or 'NO' option as to if the edgewidth should be representative of the weight value corresponding to the correlation change between two nodes. The default is NO.

layout User can choose from a handful of network visualization templates including: 'nice', 'sphere', 'grid', 'star', and 'circle'. The default is nice.

Value

An interactive dipiction of the network resulting from INDEED functions non_partial_cor() or partial_corr().

Examples

result = non_partial_cor(data = Met_GU, class_label = Met_Group_GU, id = Met_name_GU, method = "spearman", permutation_thres = 0.05, permutation = 1000)
network_display(results = result, nodesize = "P_Value", nodecolor = "Activity_Score", edgewidth = "NO", layout = "nice")

non_partial_cor  Non-partial correlation analysis

Description

A method that integrates differential expression (DE) analysis and differential network (DN) analysis to select biomarker candidates for cancer studies. non_partial_cor is a one step function for user to perform the analysis based on typical correlation analysis, no pre-processing step required.

Usage

non_partial_cor(data = NULL, class_label = NULL, id = NULL, method = "pearson", p_val = NULL, permutation = 1000, permutation_thres = 0.05)
Arguments

- **data**: This is a matrix of expression from all biomolecules and all samples.
- **class_label**: This is a binary array with 0 for group 1 and 1 for group 2.
- **id**: This is an array of biomolecule IDs.
- **method**: This is a character string indicating which correlation coefficient is to be computed. The options are either "pearson" as the default or "spearman".
- **p_val**: This is optional, it is a dataframe containing p-value for each biomolecule.
- **permutation**: This is a positive integer representing the desired number of permutations, default is 1000.
- **permutation_thres**: This is a threshold for permutation. The default is 0.05 to make 95 percent confidence.

Value

A list containing a score table with "ID", "P_value", "Node_Degree", "Activity_Score" and a differential network table with "Node1", "Node2", the binary link value and the weight link value.

Examples

```r
non_partial_cor(data = Met_GU, class_label = Met_Group_GU, id = Met_name_GU,
                method = "pearson", permutation = 1000, permutation_thres = 0.05)
```

---

**partial_cor**

Partial correlation analysis

Description

A method that integrates differential expression (DE) analysis and differential network (DN) analysis to select biomarker candidates for cancer studies. partial_cor is the second step of partial correlation calculation after getting the result from select_rho_partial function.

Usage

```r
partial_cor(data_list = NULL, rho_group1 = NULL, rho_group2 = NULL,
            permutation = 1000, p_val = NULL, permutation_thres = 0.05)
```

Arguments

- **data_list**: This is a list of pre-processed data outputed by the select_rho_partial function.
- **rho_group1**: This is the rule for choosing rho for group 1, "min": minimum rho, "ste": one standard error from minimum, or user can input rho of their choice, the default is minimum.
- **rho_group2**: This is the rule for choosing rho for group 2, "min": minimum rho, "ste": one standard error from minimum, or user can input rho of their choice, the default is minimum.
permutation_cor

permutation This is a positive integer of the desired number of permutations. The default is 1000 permutations.
p_val This is optional. It is a data frame that contains p-values for each biomolecule.

permutation_thres This is the threshold for permutation. The default is 0.05 to make 95 percent confidence.

Value
A list containing a score table with "ID", "P_value", "Node_Degree", "Activity_Score" and a differential network table with "Node1", "Node2", the binary link value and the weight link value.

Examples
# step 1: select_rho_partial
preprocess<- select_rho_partial(data = Met_GU, class_label = Met_Group_GU, id = Met_name_GU,
                                 error_curve = "YES")
# step 2: partial_cor
partial_cor(data_list = preprocess, rho_group1 = 'min', rho_group2 = "min", permutation = 1000,
             p_val = pvalue_M_GU, permutation_thres = 0.05)

permutation_cor Permutations to build a differential network based on correlation analysis

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 statistically significant if it falls into the 2.5 empirical distribution curve.

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

Arguments
m This is the number of permutations desired.
p This is the number of biomarker candidates present.
n_group_1 This is the number of subjects in group 1.
n_group_2 This is the number of subjects in group 2.
data_group_1 This is a n * p matrix containing group 1 data.
data_group_2 THis is a n * p matrix containing group 2 data.
type_of_cor If this is NULL, pearson correlation coefficient will be calculated as default. Otherwise, a character string "spearman" will calculate the spearman correlation coefficient.
permuation_pc

Value

A multi-dimensional matrix that contains the permutation result.

---

permuation_pc  
Permutations to build differential network based on partial correlation analysis

---

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 statistically significant if it falls into the 2.5 empirical distribution curve.

Usage

permuation_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  This is the number of permutations desired.
p  This is the number of biomarker candidates present.
n_group_1  This is the number of subjects in group 1.
n_group_2  This is the number of subjects in group 2.
data_group_1  This is a $n \times p$ matrix containing group 1 data.
data_group_2  This is a $n \times p$ matrix containing group 2 data.
rho_group_1_opt  This is an optimal tuning parameter to obtain a sparse differential network for group 1.
rho_group_2_opt  This is an optimal tuning parameter to obtain a sparse differential network for group 2.

Value

A multi-dimensional matrix that contains the permutation result.
permutation_thres  

**Calculate the positive and negative thresholds based on the permutation result**

**Description**

This function calculates the positive and negative thresholds based on the permutation result.

**Usage**

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

**Arguments**

- `thres_left` This is the threshold representing 2.5 percent of the left tail of the empirical distribution curve.
- `thres_right` This is the threshold representing 2.5 percent of the right tail of the empirical distribution curve.
- `p` This is the number of biomarker candidates present.
- `diff_p` This is the permutation result from either permutation_cor or permutation_pc.

**Value**

A list of positive and negative thresholds.

---

pvalue_logit  

**Obtain p-values using logistic regression**

**Description**

This function calculates p-values using logistic regression in cases that p-values are not provided.

**Usage**

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

**Arguments**

- `x` This is a data frame consists of data from group 1 and group 2.
- `class_label` This is a binary array indicating 0 for group 1 and 1 for group 2.
- `Met_name` This is an array of IDs.

**Value**

p-values
pvalue_M_GU  

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

Description

A dataset containing the p-value for each metabolite obtained through DE analysis.

Usage

pvalue_M_GU

Format

A data frame with 39 rows and 2 variables:

- **KEGG.ID**  
  KEGG ID
- **p.value**  
  p-value

scale_range  

Scale list of numbers

Description

This function is used to help spread out data values across 0 to 1. This is so that it will be easier to distinguish values later incorporated into the network_display function.

Usage

scale_range(x)

Arguments

- **x**  
  This is a list of numbers taken form on the columns outputted from calling non_partial_corr or patial_corr functions.

Value

Scaled version of data that fits between 0 to 1.
select_rho_partial

Data preprocessing for partial correlation analysis

Description

A method that integrates differential expression (DE) analysis and differential network (DN) analysis to select biomarker candidates for cancer studies. select_rho_partial is the pre-processing step for INDEED partial differential analysis.

Usage

```r
select_rho_partial(data = NULL, class_label = NULL, id = NULL, error_curve = "YES")
```

Arguments

- `data`: This is a matrix of expression from all biomolecules and all samples.
- `class_label`: This is a binary array with 0 for group 1 and 1 for group 2.
- `id`: This is an array of biomolecule IDs.
- `error_curve`: This is an option on whether a error curve plot will be provided to the user, user can choose "YES" or "NO". The default is YES.

Value

A list of processed data for the next step, and generates an error curve to select rho for graphical lasso.

Examples

```r
select_rho_partial(data = Met_GU, class_label = Met_Group_GU, id = Met_name_GU, error_curve = "YES")
```
Index

* datasets
  Met_Group_GU, 5
  Met_GU, 6
  Met_name_GU, 6
  pvalue_M_GU, 12

choose_rho, 2
compute_cor, 3
compute_dns, 3
compute_par, 4

INDEED, 4
INDEED-package (INDEED), 4

loglik_ave, 5

Met_Group_GU, 5
Met_GU, 6
Met_name_GU, 6

network_display, 6
non_partial_cor, 7

partial_cor, 8
permutation_cor, 9
permutation_pc, 10
permutation_thres, 11
pvalue_logit, 11
pvalue_M_GU, 12

scale_range, 12
select_rho_partial, 13