Package ‘miRLAB’

March 7, 2024

Type Package

Title Dry lab for exploring miRNA-mRNA relationships

Version 1.32.0

Date 2021.10.07

Author Thuc Duy Le, Junpeng Zhang, Mo Chen, Vu Viet Hoang Pham

Maintainer Thuc Duy Le <Thuc.Le@unisa.edu.au>

Imports methods, stats, utils, RCurl, httr, stringr, Hmisc, energy, entropy, gplots, glmnet, impute, limma, pcalg, TCGAbiolinks, dplyr, SummarizedExperiment, etc, InvariantCausalPrediction, Category, GOstats, org.Hs.eg.db

Description Provide tools exploring miRNA-mRNA relationships, including popular miRNA target prediction methods, ensemble methods that integrate individual methods, functions to get data from online resources, functions to validate the results, and functions to conduct enrichment analyses.

License GPL (>=2)

Suggests knitr,BiocGenerics, AnnotationDbi,RUnit,rmarkdown

VignetteBuilder knitr

biocViews miRNA, GeneExpression, NetworkInference, Network

RoxygenNote 7.1.1

NeedsCompilation no

URL https://github.com/pvvhoang/miRLAB

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

git_branch RELEASE_3_18

git_last_commit 10514c4

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-03-06
**R topics documented:**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>miRLAB-package</td>
<td>2</td>
</tr>
<tr>
<td>Borda</td>
<td>3</td>
</tr>
<tr>
<td>BordaTopk</td>
<td>4</td>
</tr>
<tr>
<td>bRank</td>
<td>5</td>
</tr>
<tr>
<td>convert</td>
<td>6</td>
</tr>
<tr>
<td>Dcov</td>
<td>6</td>
</tr>
<tr>
<td>DiffExpAnalysis</td>
<td>7</td>
</tr>
<tr>
<td>Elastic</td>
<td>8</td>
</tr>
<tr>
<td>experiment</td>
<td>9</td>
</tr>
<tr>
<td>Extopk</td>
<td>10</td>
</tr>
<tr>
<td>filterAndCompare</td>
<td>10</td>
</tr>
<tr>
<td>getData</td>
<td>11</td>
</tr>
<tr>
<td>GOBPenrichment</td>
<td>11</td>
</tr>
<tr>
<td>Hoeffding</td>
<td>12</td>
</tr>
<tr>
<td>ICCPam50</td>
<td>13</td>
</tr>
<tr>
<td>IDA</td>
<td>13</td>
</tr>
<tr>
<td>identifymiRTargetsByEnsemble</td>
<td>15</td>
</tr>
<tr>
<td>identifymiRTargetsByICPPam50</td>
<td>15</td>
</tr>
<tr>
<td>ImputeNormData</td>
<td>16</td>
</tr>
<tr>
<td>KEGGenrichment</td>
<td>17</td>
</tr>
<tr>
<td>Kendall</td>
<td>17</td>
</tr>
<tr>
<td>Lasso</td>
<td>18</td>
</tr>
<tr>
<td>MI</td>
<td>19</td>
</tr>
<tr>
<td>Pearson</td>
<td>20</td>
</tr>
<tr>
<td>RDC</td>
<td>21</td>
</tr>
<tr>
<td>Read</td>
<td>21</td>
</tr>
<tr>
<td>ReadExtResult</td>
<td>22</td>
</tr>
<tr>
<td>readHeader</td>
<td>23</td>
</tr>
<tr>
<td>Spearman</td>
<td>23</td>
</tr>
<tr>
<td>Standardise</td>
<td>24</td>
</tr>
<tr>
<td>ValidateAll</td>
<td>25</td>
</tr>
<tr>
<td>Validation</td>
<td>26</td>
</tr>
<tr>
<td>ValidationT</td>
<td>26</td>
</tr>
<tr>
<td>Zscore</td>
<td>27</td>
</tr>
</tbody>
</table>

**Index**

| miRLAB-package                          | 29 |

**Description**

Provide tools exploring miRNA-mRNA relationships, including popular miRNA target prediction methods using expression data, ensemble methods that integrate individual methods, functions to get data from online resources, functions to validate the results, and functions to conduct enrichment analyses.
Details

Package: miRLAB
Type: Package
Version: 0.99
Date: 2015-04-23
License: GPL(>=2)

Author(s)

Thuc Duy Le, Junpeng Zhang
Maintainer: Thuc Duy Le <Thuc.Le@unisa.edu.au>

References

miRLAB: An R based dry lab for exploring miRNA-mRNA relationships

---

Borda

Ensemble method for miRNA target prediction using Borda count election

Description

Use the Borda count election method to integrate the rankings from different miRNA target prediction methods

Usage

Borda(listCEmatrices)

Arguments

listCEmatrices a list of matrices that include the correlation coefficients/causal effects/scores resulting from different target prediction methods

Value

a matrix of ranking scores (averaging all the rankings from different methods). Columns are miRNAs and rows are miRNAs
References


Examples

```r
dataset = system.file("extdata", "ToyEMT.csv", package = "miRLAB")
ps = Pearson(dataset, cause = 1:3, effect = 4:18)
ida = IDA(dataset, cause = 1:3, effect = 4:18)
borda = Borda(list(ps, ida))
```

---

**BordaTopk**

*Ensemble method for miRNA target prediction using Borda count election with topK targets*

Description

Use the Borda count election method to integrate the rankings from different miRNA target prediction methods, but only topK targets of each miRNA are included in the calculation. The targets outside the topK will be assigned a large and fixed rank, e.g. number of genes in the dataset.

Usage

```r
BordaTopk(listCEmatrices, topk)
```

Arguments

- `listCEmatrices`: a list of matrices that include the correlation/causal effects/scores resulting from a target prediction method
- `topk`: number of targets of a miRNA to be included in the calculation (Borda count election)

Value

A matrix of ranking scores (averaging all the rankings from different methods). Columns are miRNAs and rows are mRNAs

References

**bRank**

### Examples

```r
dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
ps=Pearson(dataset, cause=1:3, effect=4:18)
ida=IDA(dataset, cause=1:3, effect=4:18)
borda=BordaTopk(list(ps, ida), topk=10)
```

### Description

**Extract topk predicted targets of a miRNA**

Rank all the targets of a miRNA and extract the topk targets

### Usage

```r
bRank(CEmatrix, causeIndex, topk, downreg = TRUE)
```

### Arguments

- `CEmatrix`: the matrix of correlation/causal effect/score results with columns are miRNAs and rows are mRNAs
- `causeIndex`: the column index of the miRNA that we would like to extract
- `topk`: the number of targets being extracted
- `downreg`: if TRUE the negative correlation/causal effect/score will be on the top of the ranking. This is to favour the negative regulations.

### Value

A matrix with 3 columns, where the first column contains the miRNA, the second column contains the mRNAs and the last column contains the correlations/causal effects/scores

### Examples

```r
dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
ps=Pearson(dataset, cause=1:3, effect=4:18)
miR200aTop10 = bRank(ps, 3, 10, TRUE)
```
convert  

*Convert miRNA symbols from a miRBase version to another*

**Description**

This function converts the miRNAs in the input file from the "source" miRBase version to the "target" version. If users do not know the miRBase version of the input file, please set the source version to 0. The function will match the miRNAs in the input file to all miRBase versions to find the most likely miRBase version. Currently, we have versions 16-21.

**Usage**

`convert(miRNAListFile, sourceV, targetV)`

**Arguments**

- `miRNAListFile`: the input file containing a list of miRNA symbols in csv format
- `sourceV`: the miRBase version of the input miRNAs, e.g. 16. If users do not know the version, use 0.
- `targetV`: the miRBase version that we want to convert into, e.g. 21.

**Value**

A csv file in the working directory containing the converted miRNA symbols.

**Examples**

```r
miRs = system.file("extdata", "ToymiRs.csv", package="miRLAB")
convert(miRs, 17, 21)
```

---

Dcov  

*miRNA target prediction with the Distance correlation method*

**Description**

Calculate the Distance correlation of each pair of miRNA-mRNA, and return a matrix of correlation coefficients with columns are miRNAs and rows are mRNAs.

**Usage**

`Dcov(datacsv, cause, effect, targetbinding = NA)`
**Arguments**

- `data_csv`: the input dataset in csv format
- `cause`: the column range that specifies the causes (miRNAs), e.g. 1:35
- `effect`: the column range that specifies the effects (mRNAs), e.g. 36:2000
- `target_binding`: the putative target, e.g. "TargetScan.csv". If target_binding is not specified, only expression data is used. If target_binding is specified, the prediction results using expression data will be intersected with the interactions in the target binding file.

**Value**

A matrix that includes the Distance correlation values. Columns are miRNAs, rows are mRNAs.

**References**


**Examples**

```r
dataset = system.file("extdata", "ToyEMT.csv", package="miRLAB")
results = Dcov(dataset, 1:3, 4:18)
```

---

**DiffExpAnalysis**  
**Differentially expressed analysis**

**Description**

Find the top miRNAs and mRNAs that are differently expressed between different conditions, e.g. cancer vs normal

**Usage**

```r
DiffExpAnalysis(miR1, miR2, mR1, mR2, topkmiR, topkmR, p.miR, p.mR)
```

**Arguments**

- `miR1`: the miRNA dataset for condition 1, e.g. cancer
- `miR2`: the miRNA dataset for condition 1, e.g. normal
- `mR1`: the mRNA dataset for condition 1, e.g. cancer
- `mR2`: the mRNA dataset for condition 2, e.g. normal
- `topkmiR`: the maximum number of miRNAs that we would like to extract, e.g. top 50 miRNAs.
- `topkmR`: the maximum number of mRNAs that we would like to extract, e.g. top 2000 mRNAs.
Elastic

\[ p.m\text{miR} \]

cutoff value for adjusted p-values when conducting differentially expressed analysis for miRNAs.

\[ p.m\text{R} \]

cutoff value for adjusted p-values when conducting differentially expressed analysis for mRNAs.

Value

the dataset that includes differentially expressed miRNAs and mRNAs. columns are miRNAs and mRNAs and rows are samples

References


Elastic

\textit{miRNA target prediction with the Elastic-net regression coefficient method}

Description

Calculate the Elastic-net regression coefficient of each pair of miRNA-mRNA, and return a matrix of correlation coefficients with columns are miRNAs and rows are mRNAs.

Usage

\begin{verbatim}
Elastic(datacsv, cause, effect, targetbinding = NA)
\end{verbatim}

Arguments

\begin{itemize}
\item \texttt{datacsv} the input dataset in csv format
\item \texttt{cause} the column range that specifies the causes (miRNAs), e.g. 1:35
\item \texttt{effect} the column range that specifies the effects (mRNAs), e.g. 36:2000
\item \texttt{targetbinding} the putative target, e.g. "TargetScan.csv". If targetbinding is not specified, only expression data is used. If targetbinding is specified, the prediction results using expression data with be intersected with the interactions in the target binding file.
\end{itemize}

Value

A matrix that includes the Elastic-net regression coefficients. Columns are miRNAs, rows are mRNAs.
experiment

References


Examples

```r
dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
results=Elastic(dataset, 1:3, 4:18)
```

---

experiment

Function for validate the results from all 12 methods.

Description

Function for validate the results from all 12 methods.

Usage

```r
experiment(allmethods, topk, Expgroundtruth, LFC, downreg)
```

Arguments

- **allmethods**: A list of results (matrix with columns are miRNA and rows are mRNAs).
- **topk**: Top k targets of each miRNA that will be extracted for validation
- **Expgroundtruth**: The ground truth in .csv file for validation
- **LFC**: log fold-change for validating the results using transfection experiments
- **downreg**: If set to TRUE the negative effects will have higher ranks than the positives.

Value

The validation results for all 12 methods
### Extopk

**Extract top k miRNA-mRNA interactions**

**Description**

Rank the miRNA-mRNA interactions based on absolute values of the correlations/scores/causal effects, and return the topk interactions.

**Usage**

```r
Extopk(cormat, topk)
```

**Arguments**

- `cormat`: the correlation matrix that need to be extracted with columns are miRNAs and rows are mRNAs
- `topk`: the number of interactions that need to be extracted.

**Value**

topk interactions

**Examples**

```r
dataset = system.file("extdata", "ToyEMT.csv", package="miRLAB")
EMTresults = Pearson(dataset, 1:3, 4:18)
top10 = Extopk(EMTresults, 10)
```

---

### filterAndCompare

**Filter and compare the validation results from 12 methods Keep the miRNAs that have at least noVal confirmed targets and compare the validation results from all methods.**

**Description**

Filter and compare the validation results from 12 methods Keep the miRNAs that have at least noVal confirmed targets and compare the validation results from all methods.

**Usage**

```r
filterAndCompare(allresults, noVal)
```

**Arguments**

- `allresults`: the results from all methods generated from experiment function. This is a list.
- `noVal`: Number of confirmed targets in each method (threshold) to filter. Records (miRNA) with less than this will be removed.
**getData**

**Value**

the validation results of all methods

**Examples**

```python
print("result=filterAndCompare(allresults, 2)")
```

**Description**

gData from GDC

**Usage**

gData(cancerName)

**Arguments**

cancerName  The name of cancer in string format

**Value**

dataset in matrix format

---

**GOBPenrichment**  *Functional enrichment analysis*

**Description**

GO BP enrichment analysis for a gene list

**Usage**

```python
GOBPenrichment(Genes, Cutoff)
```

**Arguments**

Genes  a list of gene symbols

Cutoff  the significant level, e.g. 0.05

**Value**

a list of GO terms for the genes
References


Examples

```r
print("result = GOBPenrichment(genelist, 0.05)")
```

---

### Hoeffding

**miRNA target prediction with the Hoeffding correlation coefficient method**

#### Description

Calculate the Hoeffding correlation coefficient of each pair of miRNA-mRNA, and return a matrix of correlation coefficients with columns are miRNAs and rows are mRNAs.

#### Usage

```r
Hoeffding(datacsv, cause, effect, targetbinding = NA)
```

#### Arguments

- `datacsv`: the input dataset in csv format
- `cause`: the column range that specifies the causes (miRNAs), e.g. 1:35
- `effect`: the column range that specifies the effects (mRNAs), e.g. 36:2000
- `targetbinding`: the putative target, e.g. "TargetScan.csv". If targetbinding is not specified, only expression data is used. If targetbinding is specified, the prediction results using expression data with be intersected with the interactions in the target binding file.

#### Value

A matrix that includes the Hoeffding correlation coefficients. Columns are miRNAs, rows are mRNAs.

#### References


#### Examples

```r
dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
results=Hoeffding(dataset, 1:3, 4:18)
```
ICPPam50  
Identify miRNA targets by ICP and PAM50

Description

This function identifies miRNA targets by ICP and PAM50.

Usage

ICPPam50(d, nmiR, nmR, fiftymRNAsData)

Arguments

d  A matrix of expression of miRNAs and mRNAs with columns being miRNA or mRNA names and rows being samples
nmiR  Number of miRNAs
nmR  Number of mRNAs
fiftymRNAsData  A matrix of expression of 50 mRNAs in PAM50 with columns being mRNA names and rows being samples

Value

The matrix of causal effects of miRNAs and mRNAs with columns being miRNAs and rows being mRNAs

References


IDA  
miRNA target prediction with the IDA method

Description

Calculate the causal effect of each pair of miRNA-mRNA, and return a matrix of causal effects with columns are miRNAs and rows are mRNAs.
**Usage**

```r
IDA(
  datacsv,
  cause,
  effect,
  pcmethod = "original",
  alpha = 0.05,
  targetbinding = NA
)
```

**Arguments**

- `datacsv` the input dataset in csv format
- `cause` the column range that specifies the causes (miRNAs), e.g. 1:35
- `effect` the column range that specifies the effects (mRNAs), e.g. 36:2000
- `pcmethod` choose different versions of the PC algorithm, including "original" (default) "stable", and "stable.fast"
- `alpha` significance level for the conditional independence test, e.g. 0.05.
- `targetbinding` the putative target, e.g. "TargetScan.csv". If targetbinding is not specified, only expression data is used. If targetbinding is specified, the prediction results using expression data with be intersected with the interactions in the target binding file.

**Value**

A matrix that includes the causal effects. Columns are miRNAs, rows are mRNAs.

**References**


**Examples**

```r
dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
results=IDA(dataset, 1:3, 4:18)
```
**identifymiRTargetsByEnsemble**

*Identify the top miRNA targets by an ensemble method with ICP-PAM50, Pearson and Lasso*

**Description**

This function identifies the top miRNA targets by an ensemble method with ICP-PAM50, Pearson and Lasso.

**Usage**

```
identifymiRTargetsByEnsemble(d, nmiR, nmR, fiftymRNAsData, top = 1, topk = 500)
```

**Arguments**

- `d`: A matrix of expression of miRNAs and mRNAs with columns being miRNA or mRNA names and rows being samples
- `nmiR`: Number of miRNAs
- `nmR`: Number of mRNAs
- `fiftymRNAsData`: A matrix of expression of 50 mRNAs in PAM50 with columns being mRNA names and rows being samples
- `top`: 1 if getting the top of all miRNAs and 2 if getting the top of each miRNA
- `topk`: Number of the top to get

**Value**

The top k miRNA targets

**References**


**identifymiRTargetsByICPPam50**

*Identify the top miRNA targets by ICP and PAM50*

**Description**

This function identifies the top miRNA targets by ICP and PAM50.

**Usage**

```
identifymiRTargetsByICPPam50(d, nmiR, nmR, fiftymRNAsData, top = 1, topk = 500)
```
Arguments

- **d**: A matrix of expression of miRNAs and mRNAs with columns being miRNA or mRNA names and rows being samples
- **nmiR**: Number of miRNAs
- **nmR**: Number of mRNAs
- **fiftymRNAsData**: A matrix of expression of 50 mRNAs in PAM50 with columns being mRNA names and rows being samples
- **top**: 1 if getting the top of all miRNAs and 2 if getting the top of each miRNA
- **topk**: Number of the top to get

Value

The top k miRNA targets

References


Filter, impute, and normalise data.

Description

Remove the genes (rows) that have more than r% of missing data; use the impute package to fill in missing data, and finally normalise the data.

Usage

```r
ImputeNormData(dataset, r)
```

Arguments

- **dataset**: The input dataset in csv format. e.g. "EMT.csv"
- **r**: The rate threshold to filter the records (genes). Genes with more than r% missing data will be removed.

Value

The processed dataset.

References

KEGGenrichment

Examples

dataset = system.file("extdata", "ToyEMT.csv", package="miRLAB")
impdata = ImputeNormData(dataset, 0.1)

---------------------------------------------------------------------

KEGGenrichment

Functional enrichment analysis KEGG enrichment analysis for a gene list

Description

Functional enrichment analysis KEGG enrichment analysis for a gene list

Usage

KEGGenrichment(Genes, Cutoff)

Arguments

Genes a list of gene symbols
Cutoff the significant level, e.g. 0.05

Value

a list of pathways for the genes

References


Examples

print("result = KEGGenrichment(genelist, 0.05)")

---------------------------------------------------------------------

Kendall

miRNA target prediction with the Kendall correlation coefficient method

Description

Calculate the Kendall correlation coefficient of each pair of miRNA-mRNA, and return a matrix of correlation coefficients with columns are miRNAs and rows are mRNAs.

Usage

Kendall(datacsv, cause, effect, targetbinding = NA)
Lasso

Arguments

- **datacsv**: the input dataset in csv format
- **cause**: the column range that specifies the causes (miRNAs), e.g. 1:35
- **effect**: the column range that specifies the effects (mRNAs), e.g. 36:2000
- **targetbinding**: the putative target, e.g. "TargetScan.csv". If targetbinding is not specified, only expression data is used. If targetbinding is specified, the prediction results using expression data will be intersected with the interactions in the target binding file.

Value

A matrix that includes the Kendall correlation coefficients. Columns are miRNAs, rows are mRNAs.

References


Examples

```r
dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
results=Kendall(dataset, 1:3, 4:18)
```

---

**Lasso**

*miRNA target prediction with the Lasso method*

Description

Calculate the Lasso regression coefficient of each pair of miRNA-mRNA, and return a matrix of coefficients with columns are miRNAs and rows are mRNAs.

Usage

```r
Lasso(datacsv, cause, effect, targetbinding = NA)
```

Arguments

- **datacsv**: the input dataset in csv format
- **cause**: the column range that specifies the causes (miRNAs), e.g. 1:35
- **effect**: the column range that specifies the effects (mRNAs), e.g. 36:2000
- **targetbinding**: the putative target, e.g. "TargetScan.csv". If targetbinding is not specified, only expression data is used. If targetbinding is specified, the prediction results using expression data will be intersected with the interactions in the target binding file.
**MI**

**Value**
A matrix that includes the Lasso regression coefficients. Columns are miRNAs, rows are mRNAs.

**References**

**Examples**
dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
results=Lasso(dataset, 1:3, 4:18)

---

**MI**

*miRNA target prediction with mutual information method*

**Description**
Calculate the mutual information of each pair of miRNA-mRNA, and return a matrix of mutual information values with columns are miRNAs and rows are mRNAs.

**Usage**

MI(datacsv, cause, effect, targetbinding = NA)

**Arguments**
- **datacsv**: the input dataset in csv format
- **cause**: the column range that specifies the causes (miRNAs), e.g. 1:35
- **effect**: the column range that specifies the effects (mRNAs), e.g. 36:2000
- **targetbinding**: the putative target, e.g. "TargetScan.csv". If targetbinding is not specified, only expression data is used. If targetbinding is specified, the prediction results using expression data with be intersected with the interactions in the target binding file.

**Value**
A matrix that includes the mutual information values. Columns are miRNAs, rows are mRNAs.

**References**

**Examples**

dataset = system.file("extdata", "ToyEMT.csv", package="miRLAB")
results = MI(dataset, 1:3, 4:18)

---

**Description**

Calculate the Pearson correlation coefficient of each pair of miRNA-mRNA, and return a matrix of correlation coefficients with columns are miRNAs and rows are mRNAs.

**Usage**

`Pearson(datacsv, cause, effect, targetbinding = NA)`

**Arguments**

- `datacsv`: the input dataset in csv format
- `cause`: the column range that specifies the causes (miRNAs), e.g. 1:35
- `effect`: the column range that specifies the effects (mRNAs), e.g. 36:2000
- `targetbinding`: the putative target, e.g. "TargetScan.csv". If `targetbinding` is not specified, only expression data is used. If `targetbinding` is specified, the prediction results using expression data with be intersected with the interactions in the target binding file.

**Value**

A matrix that includes the Pearson correlation coefficients. Columns are miRNAs, rows are mRNAs.

**References**


**Examples**

dataset = system.file("extdata", "ToyEMT.csv", package="miRLAB")
results = Pearson(dataset, 1:3, 4:18)
RDC

miRNA target prediction with the Randomized Dependence Coefficient method

Description

Calculate the Randomized Dependence coefficient of each pair of miRNA-mRNA, and return a matrix of coefficients with columns are miRNAs and rows are mRNAs.

Usage

RDC(datacsv, cause, effect, targetbinding = NA)

Arguments

datacsv the input dataset in csv format
cause the column range that specifies the causes (miRNAs), e.g. 1:35
effect the column range that specifies the effects (mRNAs), e.g. 36:2000
targetbinding the putative target, e.g. "TargetScan.csv". If targetbinding is not specified, only expression data is used. If targetbinding is specified, the prediction results using expression data with be intersected with the interactions in the target binding file.

Value

A matrix that includes the correlation coefficients. Columns are miRNAs, rows are mRNAs.

Examples

dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
results=RDC(dataset, 1:3, 4:18)

Read

Read dataset from csv file

Description

Read dataset from csv file

Usage

Read(dataset)

Arguments

dataset The input dataset in csv format
Value

dataset in matrix format

Examples

dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
data=Read(dataset)

Description

Read the results predicted by external methods (methods that are not in this package and may not be implemented in R). Consequently, we can compare the results predicted by the external methods and results predicted by the methods in the miRLAB package.

Usage

ReadExtResult(datacsv, cause, effect, ExtCEcsv)

Arguments

datacsv the input dataset in csv format
cause the column range that specifies the causes (miRNAs), e.g. 1:35
effect the column range that specifies the effects (mRNAs), e.g. 36:2000
ExtCEcsv score matrix predicted by an external matrix with columns are miRNAs and rows are mRNAs.

Value

a matrix of scores predicted by an external matrix and ready for further validation and comparison tasks.

Examples

print("Genemir=ReadExtResult(dataset, cause=1:3, effect=4:18, 'genemirresults.csv')")
**readHeader**

*Read the header of the dataset*

**Description**
Read the header of the dataset

**Usage**
readHeader(dataset)

**Arguments**
dataset the character string of the names of the dataset in csv format, e.g. "ToyEMT.csv"

**Value**
the header of the dataset

**Examples**
dataset = system.file("extdata", "ToyEMT.csv", package="miRLAB")
header = readHeader(dataset)

---

**Spearman**

*miRNA target prediction with the Spearman correlation coefficient method*

**Description**
Calculate the Spearman correlation coefficient of each pair of miRNA-mRNA, and return a matrix of correlation coefficients with columns are miRNAs and rows are mRNAs.

**Usage**
Spearman(datacsv, cause, effect, targetbinding = NA)

**Arguments**
datacsv the input dataset in csv format
cause the column range that specifies the causes (miRNAs), e.g. 1:35
effect the column range that specifies the effects (mRNAs), e.g. 36:2000
targetbinding the putative target, e.g. "TargetScan.csv". If targetbinding is not specified, only expression data is used. If targetbinding is specified, the prediction results using expression data will be intersected with the interactions in the target binding file.
Standardise

Value

A matrix that includes the Spearman correlation coefficients. Columns are miRNAs, rows are mRNAs.

References


Examples

```r
dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
results=Spearman(dataset, 1:3, 4:18)
```

**Standardise**

Standardise the dataset to have mean=0 and std=1 in each column.

Description

Standardise the dataset to have mean=0 and std=1 in each column.

Usage

`Standardise(dataset)`

Arguments

- `dataset` The input dataset in csv format. e.g. "ToyEMT.csv". The first column is the sample name.

Value

The standardised dataset.

Examples

```r
## Not run:
dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
stddata=Standardise(dataset)

## End(Not run)
```
ValidateAll

Validate the targets of all miRNA using both experimentally confirmed and transfection data

Description

Given the predicted target of all miRNA, the function returns a list of targets of each miRNA that are confirmed based on the experimentally validated interactions or curated transfection data. Users need to download the file logFC.imputed.rda from nugget.unisa.edu.au/Thuc/miRLAB/ and place it in the working directory (this file is obtained from the TargetScoreData package)

Usage

ValidateAll(CEmatrix, topk, groundtruth, LFC, downreg = TRUE)

Arguments

- **CEmatrix**: the matrix of correlation/causal effects/scores with columns are miRNAs and rows are mRNAs
- **topk**: the number of targets of each miRNA that are being validated.
- **groundtruth**: the csv file containing the ground truth.
- **LFC**: the log fold change threshold for the transfection data. The targets that have the absolute value of log fold change greater than the LFC will be regarded as the confirmed targets.
- **downreg**: if TRUE the negative correlation/causal effect/score values will be ranked on the top of the ranking. This is to favour the down regulations.

Value

a list of matrices that contains the confirmed interactions by both provided ground truth and built-in transfection data.

Examples

```R
print("ps=Pearson(dataset, cause=1:3, effect=4:18)")
print("results=ValidateAll(ps, 10, groundtruth, LFC=0.5, downreg=TRUE)")
```
Validation

Validate the targets of a miRNA

Description

Given the predicted target of a miRNA, the function returns a list of targets that are experimentally confirmed based on the provided ground truth. Users can provide their own ground truth or use the built-in ground truth which is the union of Tarbase, miRTarbase, miRecords, and miRWalk.

Usage

Validation(topkList, datacsv)

Arguments

topkList  a matrix with 3 columns. The first column is the miRNA name, the second contains the target mRNAs, and the third contains the correlation values/ causal effects/ scores

datacsv  the ground truth for the validation. The ground truth is a matrix with 2 columns, where the first column is the miRNA and the second is the mRNA.

Value

a matrix in the same format of the input matrix put only contains the confirmed interactions.

Examples

dataset=system.file("extdata", "ToyEMT.csv", package="miRLAB")
ps=Pearson(dataset, cause=1:3, effect=4:18)
miR200aTop10=bRank(ps, 3, 10, TRUE)
groundtruth=system.file("extdata", "Toygroundtruth.csv", package="miRLAB")
miR200aTop10Confirmed = Validation(miR200aTop10, groundtruth)

ValidationT

Validate the targets of a miRNA using transfection data

Description

Given the predicted target of a miRNA, the function returns a list of targets that are confirmed based on the curated transfection data. Users need to download the file logFC.imputed.rda from nugget.unisa.edu.au/Thuc/miRLAB/ and place it in the working directory (this file is obtained from the TargetScoreData package)

Usage

ValidationT(topkList, LFC)
Zscore

Arguments

topkList: a matrix with 3 columns. The first column is the miRNA name, the second contains the target mRNAs, and the third contains the correlation values/causal effects/scores.

LFC: the log fold change threshold. The targets that have the absolute value of log fold change greater than the LFC will be regarded as the confirmed targets.

Value

A matrix in the same format of the input matrix put only contains the confirmed interactions.

References


Examples

```r
print("ps=Pearson(dataset, cause=1:35, effect=36:1189)")
print("miR200aTop100=bRank(ps, 11, 100, TRUE)")
print("miR200aTop100Confirmed = ValidationT(miR200aTop100, 1.0)")
```

Zscore

miRNA target prediction with the Z-score method

Description

Calculate the Z-score value of each pair of miRNA-mRNA, and return a matrix of values with columns are miRNAs and rows are mRNAs.

Usage

```
Zscore(datacsv, cause, effect, targetbinding = NA)
```

Arguments

datacsv: the input dataset in csv format

cause: the column range that specifies the causes (miRNAs), e.g. 1:35

effect: the column range that specifies the effects (mRNAs), e.g. 36:2000

targetbinding: the putative target, e.g. "TargetScan.csv". If targetbinding is not specified, only expression data is used. If targetbinding is specified, the prediction results using expression data with be intersected with the interactions in the target binding file.
Value

A matrix that includes the Z-score values. Columns are miRNAs, rows are mRNAs.

References


Examples

dataset = system.file("extdata", "ToyEMT.csv", package="miRLAB")
results = Zscore(dataset, 1:3, 4:18)
Index

* package
  miRLAB-package, 2
  *
    miRLAB-package, 2

Borda, 3
BordaTopk, 4
bRank, 5

convert, 6

Dcov, 6
DiffExpAnalysis, 7

Elastic, 8
experiment, 9
Extopk, 10

filterAndCompare, 10

getData, 11
GOBPenrichment, 11

Hoeffding, 12

ICPPam50, 13
IDA, 13
identifymiRTargetsByEnsemble, 15
identifymiRTargetsByICPPam50, 15
ImputeNormData, 16

KEGGenrichment, 17
Kendall, 17

Lasso, 18

MI, 19
miRLAB (miRLAB-package), 2
miRLAB-package, 2

Pearson, 20

RDC, 21
Read, 21
ReadExtResult, 22
readHeader, 23

Spearman, 23
Standardise, 24

ValidateAll, 25
Validation, 26
ValidationT, 26

Zscore, 27