

Package ‘decode’

July 22, 2025

Title Differential Co-Expression and Differential Expression Analysis

Version 1.2

Description Integrated differential expression (DE) and differential co-expression (DC) analysis on gene expression data based on DECODE (DifferEntial CO-expression and Differential Expression) algorithm.

Depends R (>= 3.1.2)

License GPL-3

LazyData false

VignetteBuilder knitr

Suggests knitr

NeedsCompilation no

Author Thomas Lui [aut, cre]

Maintainer Thomas Lui <tlui27@yahoo.com>

Repository CRAN

Date/Publication 2015-07-14 20:40:34

Contents

getAssoGeneSetPValue	2
getBestAssociatedGeneSet	2
getBonferroniPValue	3
getDE_DC_OptimalThreshold	3
getFDR	4
getPartitionIndex	4
getPathway	5
openFileToWrite	5
runDecode	6
sumResult_MinGain	7

Index	8
--------------	----------

getAssoGeneSetPValue *Calculate the p-value between selected genes and functional gene set*

Description

Calculate the p-value between selected genes and functional gene set

Usage

```
getAssoGeneSetPValue(geneList, geneSet, multipleTestCount, MaxGene)
```

Arguments

geneList	Selected genes
geneSet	Functional gene set
multipleTestCount	Number of multiple testing
MaxGene	Number of genes in expression data

Value

The adjusted p-value for the associated gene set

getBestAssociatedGeneSet
 Get best associated functional gene sets for partitions of gene i

Description

Get best associated functional gene sets for partitions of gene i

Usage

```
getBestAssociatedGeneSet(pathway, all8Partitions, onePartition, MaxGene,  
                          minSupport)
```

Arguments

pathway	All functional gene sets
all8Partitions	All eight possible partitions for gene i
onePartition	The partition to be associated with the functional gene set
MaxGene	Number of genes in expression data
minSupport	Minimum support for functional gene set

Value

The adjusted p-values for the best associated gene set of the input partition

getBonferroniPValue	<i>Adjust p-value by Bonferroni correction</i>
---------------------	--

Description

Adjust p-value by Bonferroni correction

Usage

```
getBonferroniPValue(pValues)
```

Arguments

pValues	Unadjusted p-values
---------	---------------------

Value

Adjusted p-values

getDE_DC_OptimalThreshold	<i>Perform chi-square optimization</i>
---------------------------	--

Description

Perform chi-square optimization

Usage

```
getDE_DC_OptimalThreshold(t_result, MaxGene, d_r, minSupport)
```

Arguments

t_result	The t-statistics
MaxGene	Number of genes in expression data
d_r	DC measures
minSupport	The minimum expected frequency in contingency table

Value

The optimal threshold information

getFDR	<i>Adjust p-value by Benjamini and Hochberg method</i>
--------	--

Description

Adjust p-value by Benjamini and Hochberg method

Usage

```
getFDR(pValues)
```

Arguments

pValues	Unadjusted p-values
---------	---------------------

Value

Adjusted p-values

getPartitionIndex	<i>Get gene index of 8 partitions for gene i</i>
-------------------	--

Description

Get gene index of 8 partitions for gene i

Usage

```
getPartitionIndex(gene_i, t_result, optimalCutOff, abs_r)
```

Arguments

gene_i	Gene i index
t_result	t-statistics
optimalCutOff	Optimal thresholds
abs_r	Matrix consisting of absolute values of all differential co-expression measures

Value

The selected genes for each partition in index

getPathway	<i>read functional gene sets</i>
------------	----------------------------------

Description

read functional gene sets

Usage

```
getPathway(inputFile, geneName, minSupport)
```

Arguments

inputFile	Input file name
geneName	Gene name lists
minSupport	Minimum support

Value

Functional gene set

openFileToWrite	<i>Open file to write result</i>
-----------------	----------------------------------

Description

Open file to write result

Usage

```
openFileToWrite(filename)
```

Arguments

filename	file name Output: Results in text file
----------	--

runDecode

*Differential Co-Expression and Differential Expression Analysis***Description**

Given a set of gene expression data and functional gene set data, the program will return a table summary for the selected gene sets with high differential co-expression and high differential expression (HDC-HDE). User need to specify the input paths for the gene expression data and functional gene set data.

Usage

```
runDecode(geneSetInputFile, geneExpressionFile)
```

Arguments

geneSetInputFile

Path for functional gene set data

geneExpressionFile

Path for gene expression data

Input: (1) gene expression data

(2) functional gene set data

Output: Table summary for the selected HDC-HDE gene sets, 'out_summary.txt'

Data format for gene expression data (Columns are tab-separated):

Column 1: Official gene symbol

Column 2: Probe ID

Starting from column 3: Expression for different samples

Row 1 (starting from column 3): Sample class ("1" indicates control group; "2" indicates case group)

Row 2: Sample id

Starting from row 3: Expression for different genes

Example:

```
geneName probeID 2 2 2 1 1 1
```

```
- - Case1 Case2 Case3 Control1 Control2 Control3
```

```
7A5 ILMN_1762337 5.12621 5.19419 5.06645 5.40649 5.51259 5.387
```

```
A1BG ILMN_2055271 5.63504 5.68533 5.66251 5.37466 5.43955 5.50973
```

```
A1CF ILMN_2383229 5.41543 5.58543 5.43239 5.49634 5.62685 5.36962
```

```
A26C3 ILMN_1653355 5.56713 5.5547 5.59547 5.46895 5.49622 5.50094
```

```
A2BP1 ILMN_1814316 5.23016 5.33808 5.31413 5.30586 5.40108 5.31855
```

```
A2M ILMN_1745607 7.65332 6.56431 8.20163 9.19837 9.04295 10.1448
```

```
A2ML1 ILMN_2136495 5.53532 5.93801 5.33728 5.36676 5.79942 5.13974
```

```
A3GALT2 ILMN_1668111 5.18578 5.35207 5.30554 5.26107 5.26536 5.28932
```

Data format for functional gene set data (Columns are tab-separated):

Column 1: Functional gene set name
Column 2: Other description such as gene set id
Starting from column 3: Official gene symbols for the functional gene set
Example:
B cell activation GO\GO:0042113 AKAP17A ZAP70 PFDN1 ...
apoptotic signaling pathway GO\GO:0097190 ITPR1 PTH DNAJC10 HINT1 ...

Details

The main program for DECODE algorithm

To run an example using expression data with 1400 genes.

```
runDecode("\extdata\geneSet.txt", "\extdata\Expression_data_1400genes.txt")
```

or

```
runDecode("/extdata/geneSet.txt", "/extdata/Expression_data_1400genes.txt")
```

The sample data with 1400 genes takes 16 minutes to complete. (Computer used: An Intel Core i7-4600 processor, 2.69 GHz, 8 GB RAM)

Examples

```
## Not run:  
path = system.file('extdata', package='decode')  
geneSetInputFile = file.path(path, "geneSet.txt")  
geneExpressionFile = file.path(path, "Expression_data_50genes.txt")  
runDecode(geneSetInputFile, geneExpressionFile)  
  
## End(Not run)
```

sumResult_MinGain

Summarize the functional gene set results into text file

Description

Summarize the functional gene set results into text file

Usage

```
sumResult_MinGain()
```

Index

`getAssoGeneSetPValue`, [2](#)
`getBestAssociatedGeneSet`, [2](#)
`getBonferroniPValue`, [3](#)
`getDE_DC_OptimalThreshold`, [3](#)
`getFDR`, [4](#)
`getPartitionIndex`, [4](#)
`getPathway`, [5](#)

`openFileToWrite`, [5](#)

`runDecode`, [6](#)

`sumResult_MinGain`, [7](#)