Package 'DEComplexDisease'

April 15, 2019

Type Package

Title A tool for differential expression analysis and DEGs based investigation to complex diseases by bi-clustering analysis

Version 1.2.0

Author Guofeng Meng

Maintainer Guofeng Meng <menggf@gmail.com>

Description

It is designed to find the differential expressed genes (DEGs) for complex disease, which is characterized by the heterogeneous genomic expression profiles. Different from the established DEG analysis tools, it does not assume the patients of complex diseases to share the common DEGs. By applying a bi-clustering algorithm, DECD finds the DEGs shared by as many patients. In this way, DECD describes the DEGs of complex disease in a novel syntax, e.g. a gene list composed of 200 genes are differentially expressed in 30% percent of studied complex disease. Applying the DECD analysis results, users are possible to find the patients affected by the same mechanism based on the shared signatures.

License GPL-3

LazyData TRUE

Depends R (>= 3.3.3)

Imports Rcpp (>= 0.12.7), DESeq2, edgeR, SummarizedExperiment, ComplexHeatmap, grid, parallel, BiocParallel, grDevices, graphics, stats, methods, utils

Suggests knitr

LinkingTo Rcpp

biocViews DNASeq, WholeGenome, FunctionalGenomics, DifferentialExpression,GeneExpression, Clustering

RoxygenNote 6.0.1

VignetteBuilder knitr

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

git_branch RELEASE_3_8

git_last_commit 9785cef

git_last_commit_date 2018-10-30

Date/Publication 2019-04-15

R topics documented:

`ann.er` *ER status annotation*

Description

ER status annotation

Format

a data.frame

Value

a data.frame

`bi.deg` *Transformation an expression matrix to binary differential expression matrix*

Description

Transform the RNA-seq counts or normalized expression matrix into binary differential expression matrix of -1, 0 and 1, which indicates the down-regulation, no change and up-regulation.

Usage

```
bi.deg(exp, cl, method = c("edger", "deseq2", "normalized")[1],
  cutoff = 0.05, cores = 1)
```

Arguments

`exp`	a matrix or data frame for expression data. The expression value can be counts or normalized expression data
`cl`	a vector of 0 and 1. It has equal length with the column number of exp. 1 indicates the corresponding samples are patients and 0 is control or normal
`method`	defines the methods applied for DE analysis. The possible value is 'edger', 'deseq2', 'normalized'. 'edger' or 'deseq2' is used for RNA-seq count data; 'normalized' is used for normalized RNA-seq or microarray data
`cutoff`	the p-value cutoff for DEGs
`cores`	the thread number

Details

For each sample in 'exp', 'cl' defines the patients and normal. The normal samples are used to construct the expression references with negative binomial distribution (e.g. method='edger' or method='deseq2') or a normal distribution (method='normalized').

When counts data are used, the DEG analysis is performed using the functions implemented by 'DESeq2' or 'edgeR'. The dispersion and mu values are estimated.

Value

A deg class object with value of 1, 0 and -1.

Author(s)

Guofeng Meng

Examples

```
deg <- bi.deg(exp,cl=cl, method='edger', cutoff=0.05) # exp is the RNA-seq counts matrix
```

cl — *Binary patient annotation (part)*

Description

A binary vectory for disease status (only part)

Format

a vector

Value

A vector

cluster.mod — *seed-based module analysis (part)*

Description

The clustered module analysis results (only part)

Format

a list

Value

A list

cluster.module — *Predict the DEGs modules shared by patients*

Description

This function uses the output of `bi.deg` as input to predict the patient-DEG lists (or modules) shared by patients.

Usage

```
cluster.module(res.module, vote.seed = FALSE, model.method = NULL,
  cores = 1, max.show.n = 1, seed = 1)
```

Arguments

res.module	a 'seed.module' object. It should be the output of seed.module
vote.seed	boolean, generate a generic seed or not
model.method	the method to find the breakpoint of bi-clustering. It is accepted value including 'slope.clustering', 'max.square', 'min.slope', 'min.similarity'. If it is NULL, its value will be get from res.module[['decd.input']][['module.method']]
cores	the thread number
max.show.n	the number of sub-modules to report
seed	a seed for random generator

Details

The function is to cluster the modules predicted by seed.module, which is very useful when there are too many modules in 'res.module'.

This functon perform a k-mean based clustering to cluster the predicted modules. The patients within the same cluster are ranked based on their connecting degrees so that to find the representative patient(s). if 'vote.seed' is false, the bi-clustering analysis results of the representative patient will be used as the final results of the module. Otherwise, a generic seed will be generated by a voting method and the final results is predicted by bi-clustering analysis using the new seed.

Value

A cluster.module object. It has two keys with prefix of 'decd':

'decd.input', the input information, including binary DEG matrix, used.genes and other parameter setting.

'decd.clustering', the clustering and the representative patient information.

Other keys has a prefix of 'M', which indicates clustered modules. Each module have several keys:

'curve', the patient-gene number during bi-clustering analysis;

'max.genes', the patient and genes when 'min.patients' is observed in bi-clustering analysis;

'max.patients', the patient and genes when 'min.genes' is reached in bi-clustering analysis';

'model', the patient and genes at the breakpoint of the 'curve';

'genes.removed', the ordered genes that are removed from module during bi-clustering analysis;

'patients.added', the ordered patients that are added to module during bi-clustering analysis

Author(s)

Guofeng Meng

Examples

```
cluster.mod <- cluster.module(seed.mod, model.method='slope.clustering')
cluster.mod2 <- cluster.module(seed.mod, model.method='slope.clustering', vote.seed=TRUE)
```

deg *DEG matrix of breast cancer (part)*

Description

A matrix of bianry DEG matrix for breast cancer patients (only part)

Format

a matrix

Value

A matrix

deg.spc *The patient-specific DEGs (part)*

Description

The patient-specific DEGs (only part)

Format

a list

Value

A list

deg.specific *Predict the patient-specific DEGs using bi-clustering analysis*

Description

This function is use the output of `bi.deg` as input to predict the differentially expressed genes (DEGs) for each patient by cross-validation of multiple patients.

Usage

```
deg.specific(deg, test.patients = NULL, min.genes = 50, min.patients = 5,
  overlap = 0.85, cores = 1)
```

Arguments

deg	a 'deg' object. This should be the output of bi.deg
test.patients	the patients to test. Only the patients in 'test.patients' are used as seed for bi-clustering analysis
min.genes	the minimum number of genes
min.patients	the minimum number of patients. It includes the patients as seeds (see details).
overlap	the minimum similarity for selected DEGs from two or more patients
cores	the thread number

Details

The DEGs from bi.deg are mixed with noises, e.g. the DEGs not associated with disease. This is especially true when the differential expression analysis tests are done using one variable again references. The assumption behind his analysis is that the disease associated DEGs will be observed in other patients. This function implements a bi-clustering algorithm to find the DEGs shared by 'min.patients' in the binary DEG matrix. In this process, each patient is used as seed and its DEGs are gradually excluded to find if there is a DEG list which is observed in 'min.patients' when the similarity is greater is 'overlap'.

'test.patients' option is used to find the cross-validated DEGs for some interest patients. Otherwise, all the patients will be used as seeds. 'overlap' is the threshold to determine the minimum similarity between neighbor and seed patients.

Value

A 'deg.specific' or 'deg.specific.test' object. It has a key of 'decd.input', which stores the binary DEG matrix, genes, patients and used parameter setting. Other keys are the patients IDs and they store the cross-validated DEGs.

Author(s)

Guofeng Meng

Examples

```
# the DEGs has at least 100 genes and validated by 5 other patients
deg.spc <- deg.specific(deg, min.genes=50, min.patients=5, overlap=0.85)
```

---

exp *expression matrix of breast cancer (part)*

Description

A matrix of breast cancer patients from TCGA (only part)

Format

a matrix

Value

A matrix

`module.compare` *Compare and plot the overlap among predicted modules*

Description

Plot the overlap among predicted DEG modules

Usage

```
module.compare(res.module1, res.module2, used.mods1 = NULL,
  used.mods2 = NULL, type = c("model", "max.patients", "max.genes")[1],
  max.n1 = 30, max.n2 = max.n1, show.overlap = TRUE, cex = 10)
```

Arguments

`res.module1`	a 'seed.module' or 'cluster.module' object returned by `seed.module` or `cluster.module`
`res.module2`	a 'seed.module' or 'cluster.module' object returned by `seed.module` or `cluster.module`
`used.mods1`	the modules to display
`used.mods2`	the modules to display
`type`	the module type to display
`max.n1`	the maximum number of modules to display. If 'used.mods1' is set, this option will be ignored.
`max.n2`	the maximum number of modules to display. If 'used.mods2' is set, this option will be ignored.
`show.overlap`	boolean, display the overlap number
`cex`	the font cex to display the overlap number

Details

This function is to compare the modules from different studies, e.g. the different diseases or the different data for the same disease.

Value

The heatmap plot for gene overlaps.

Author(s)

Guofeng Meng

Examples

```
module.compare(res.mod1,res.mod2, type='model', max.n1=5)
```

module.curve *Display the patient-gene numbers during bi-clustering analysis*

Description

Plot the curve of patient-gene number during bi-clustering analysis

Usage

```
module.curve(res.module, mod = names(res.module)[1])
```

Arguments

res.module	a 'seed.module' or 'cluster.module' object returned by seed.module or cluster.module
mod	the module to plot

Details

This function is used to display the patient and gene number during the bi-clustering analysis. It can be used for users to select the better gene or patient number for breakpoints.

Value

The plot for gene and patient number.

Author(s)

Guofeng Meng

Examples

```
module.curve(cluster.mod, 'M1')
```

module.exact *Extract the patients and genes from module*

Description

Return the patients and genes from module under user defined setting

Usage

```
module.extract(res.module, mod, n.patients = NULL, n.genes = NULL)
```

Arguments

res.module	a 'seed.module' or 'cluster.module' object returned by seed.module or cluster.module
mod	a module name to extract genes and patients
n.patients	the patient number to return
n.genes	the gene number to return

Details

This function is used to return the patients and genes at user defined breakpoints.

Users can set ‘n.patients‘ or ‘n.genes‘ to define the break points of bi-clustering. But it is not allowed to set both value.

Value

A list for genes or patients.

Author(s)

Guofeng Meng

Examples

```
# extract the genes and patients when 200 patients are observed in M1.
page=module.extract(cluster.mod, 'M1', n.patients=100)
head(page$patients)
head(page$geness)
# find genes and patients in patient-seeded module
module.extract(seed.mod, names(seed.mod)[1], n.patients=50)
```

`module.modeling` *Predict and report the proper gene-patient number of bi-clustering analysis*

Description

This function attempts to find the breakpoint in patient-gene number curve generated during bi-clustering analysis, which may indicate inclusion/exclusion of molecular mechanism for selected patients

Usage

```
module.modeling(res.module, keep.gene.num = NULL,
  model.method = c("slope.clustering", "max.square", "min.slope",
  "min.similarity")[1], cores = 1, overlap = NULL, para = NULL)
```

Arguments

`res.module`	a ’seed.module’ or ’cluster.module’ object
`keep.gene.num`	a integer value or a vector (see details).
`model.method`	the modeling methods (see details). The possible values are ’max.square’, ’slope.clustering’, ’min.slope’ and ’min.similarity’
`cores`	the thread number
`overlap`	the minimum similarity for selected DEGs from two or more patients
`para`	a list, with two keys ’deg’ and ’overlap’. It should not be NULL if res.module is a list.

Details

This function will be automatic used by seed.module and cluster.module during its module discovery steps. User can explicitly use it to refine the modelling results. After checking the 'curve' plot, users can change the break points to modify the modelling results by setting 'keep.gene.num', which is the number of DEGs to keep. If users only need to change part of the modules, just give the the 'keep.gene.num' for the selected modules.

'keep.gene.num' can a integer value or a vector. If it is a integer number, all the modules will have the same 'keep.gene.num'. If it is a vector, its elements should use module name as their names. Otherwise, only the first element will be used and all the modules will be set. When 'keep.gene.num' is vector, it is not necessary to have the same length as modules. It is possible to only changes some of the modules. And the left modules will use the default setting.

Another way to modify the modelling results is to change the 'model.method'. In this version, 'model.method' has four possible values: 'slope.clustering', 'max.square', 'min.slope' and 'min.similarity', which indicate the different four different modelling methods: 'slope.clustering' has maximum slope changes, which may indicate the inclusion/exclusion of molecular mechanism. 'max.square' is the gene-patients number that has the maximum product; 'min.slope' is the point with minimum slope in gene-patient number curve; 'min.similarity' is based on the similarity scores and the point with minimum similarity scores is choosed.

Within this package, users have two ways to refine the modeling results. One way is to run seed.module or cluster.module by setting the 'res.module' and 'model.method'. Another way is to run module.modeling.

Value

a 'deg.modules' object and its modules has a 'model' to be added or refined.

Author(s)

Guofeng Meng

Examples

```
x=c(100,200)
names(x)<-c('M1','M3')
new.seed.mod=module.modeling(seed.mod, keep.gene.num = x)
#here, only 'M1' and 'M3' are modified
new.seed.mod=module.modeling(seed.mod, keep.gene.num = 100)
# here, all the modules are modified
new.cluster.mod=module.modeling(cluster.mod, model.method='min.similarity')
# here, change the modeling method
```

---

module.overlap *Plot the overlap among predicted DEG modules*

Description

Plot the overlap among predicted DEG modules

Usage

```
module.overlap(res.module, show.mods = NULL, type = c("model",
  "max.patients", "max.genes")[1], max.n = 30, show.overlap = TRUE,
  cex = 10)
```

Arguments

res.module	a 'seed.module' or 'cluster.module' object returned by seed.module or cluster.module
show.mods	the modules to display
type	the module type to display
max.n	the maximum number of modules to plot. If 'show.mods' is set, this option will be ignored.
show.overlap	boolen, display the overlap number
cex	the font cex to display the overlap number, default 10.

Details

The DEG modules may have partial overlaps for either genes or patients. This function plots the overlaps for genes and patients from DEG modules in two heatmaps. It can be used to find DEG patterns.

Value

The heatmap plot for gene and patient overlaps.

Author(s)

Guofeng Meng

Examples

```
data(seed.mod)
module.overlap(seed.mod, max.n=5,type='model')
```

---

module.screen *Screen the feature patients or genes in predicted modules*

Description

Screen feature patients or genes given by users among the predicted modules

Usage

```
module.screen(res.module, feature.patients = NULL, feature.genes = NULL,
  show.mods = NULL, show.n = 4, method = c("ratio", "fisher.test")[1],
  cores = 1)
```

Arguments

res.module	a 'seed.module' or 'cluster.module' object returned by seed.module or cluster.module
feature.patients	the patients to screen
feature.genes	the genes to screen
show.mods	the modules to display
show.n	the number of modules to display
method	the method to find the most associated modules
cores	the thread number

Details

This function is used to find the modules associated with the 'feature.patients' or 'feature genes'.

In current version, two methods can be used: 'ratio' and 'fisher.test'. 'ratio' is to rank the modules based on ratio between observed overlaps and the expected overlaps that estimated using all the samples. 'fisher.test' is to use Fisher's exact test to check the significance of association.

Value

A plot for gene or patient overlaps with the feature genes or patients.

Author(s)

Guofeng Meng

Examples

```
# screen the modules for feature patients.
module.screen(seed.mod, feature.patients=sample(colnames(deg),15))
```

Plot.cluster.module *Plot the DEGs modules*

Description

Plot the DEGs modules

Usage

```
## S3 method for class 'cluster.module'
Plot(res.module, ann = NULL, deg = NULL,
  col.order = NULL, show.mods = NULL, overlap = NULL,
  dissimilarity = NULL, max.n = min(length(res.module), 30),
  type = c("model", "max.patients", "max.genes")[1], label.col = "blue",
  ...)
```

Arguments

res.module	a 'cluster.module' object returned by cluster.module
ann	a data.frame for the patient annotation
deg	a 'deg' to display. It is returned by bi.deg
col.order	the order of column in heatmap
show.mods	a vector, the modules to display
overlap	the similarity cutoff to display as carrying the module
dissimilarity	the similarity cutoff to display as not carrying the module
max.n	the maximum number of modules to display
type	the module type to display
label.col	the color to label
...	other setting of 'oncoPrint'

Details

This function is to dispaly the relationship of the predicted DEG modules and the patients.

'deg' can be set to display the modules from different datasets, e.g. the modules predicted from disease A and display them in the binary DEG matrix of disease B.

The output is a heatmap Plot where the modules with maximum observations are showed.

Value

A heatmap plot

Author(s)

Guofeng Meng

Examples

```
Plot(cluster.mod, ann.er, max.n=5)
Plot(cluster.mod, ann.er, deg=deg, max.n=5)
```

Plot.deg *Plot the DEGs before or after cross-validation*

Description

Plot the binary differential expression matrix transformed by bi.deg

Usage

```
## S3 method for class 'deg'
Plot(input, ann = NULL, col.order = NULL, show.genes = NULL,
  max.n = 30, up.col = "red", down.col = "blue", ...)

Plot(...)
```

Arguments

input	a 'deg' object returned by bi.deg
ann	a data.frame for the patient annotation
col.order	the order of column in heatmap
show.genes	the gene ids to plot
max.n	the maximum number of genes to plot
up.col	the color for up-regulated genes
down.col	the color for down-regulated genes
...	other setting of 'oncoPrint'

Details

This function applied the function of oncoPrint from 'ComplexHeatmap' to dispaly ownership of the DEGs. The output is a heatmap plots where the genes with maximum observations are showed.

Value

A heatmap plot

Author(s)

Guofeng Meng

Examples

```
Plot(deg,ann.er, max.n=5)
Plot(deg.spc, ann.er, max.n=5)
```

Plot.deg.specific *Plot the DEGs before or after cross-validation*

Description

Plot the cross-validated DEGs predicted by deg.specific.

Usage

```
## S3 method for class 'deg.specific'
Plot(input, ann = NULL, col.order = NULL,
  show.genes = NULL, max.n = 30, up.col = "red", down.col = "blue", ...)
```

Arguments

input	a 'deg.specific' object returned by deg.specific
ann	a data.frame for the patient annotation
col.order	the order of column in heatmap
show.genes	the gene ids to plot
max.n	the maximum number of genes to plot
up.col	the color for up-regulated genes
down.col	the color for down-regulated genes
...	other setting of 'oncoPrint'

Details

This function applied the function of oncoPrint from 'ComplexHeatmap' to dispaly ownership of the DEGs. The output is a heatmap plots where the genes with maximum observations are showed.

Value

A heatmap plot

Author(s)

Guofeng Meng

Examples

```
Plot(deg,ann.er, max.n=5)
Plot(deg.spc, ann.er, max.n=5)
```

Plot.deg.specific.test

Plot the DEGs before or after cross-validation

Description

Plot the cross-validated DEGs predicted by deg.specific.

Usage

```
## S3 method for class 'deg.specific.test'
Plot(input, ann = NULL, col.order = NULL,
  show.genes = NULL, max.n = 30, up.col = "red", down.col = "blue", ...)
```

Arguments

input	a 'deg.specific' object returned by deg.specific
ann	a data.frame for the patient annotation
col.order	the order of column in heatmap
show.genes	the gene ids to plot
max.n	the maximum number of genes to plot
up.col	the color for up-regulated genes
down.col	the color for down-regulated genes
...	other setting of 'oncoPrint'

Details

This function applied the function of oncoPrint from 'ComplexHeatmap' to dispaly ownership of the DEGs. The output is a heatmap plots where the genes with maximum observations are showed.

Value

A heatmap plot

Author(s)

Guofeng Meng

Examples

```
Plot(deg,ann.er, max.n=5)
Plot(deg.spc, ann.er, max.n=5)
```

Plot.seed.module *Plot the DEGs modules shared by patients*

Description

Plot the DEGs modules

Usage

```
## S3 method for class 'seed.module'
Plot(res.module, ann = NULL, deg = NULL,
  col.order = NULL, show.mods = NULL, overlap = NULL,
  dissimilarity = NULL, max.n = min(length(res.module), 30),
  type = c("model", "max.patients", "max.genes")[1], label.col = "blue",
  ...)
```

Arguments

res.module	a 'seed.module' object returned by seed.module
ann	a data.frame for the patient annotation
deg	a 'deg' to display. It is returned by bi.deg
col.order	the order of column in heatmap
show.mods	a vector, the modules to display
overlap	the similarity cutoff to display as carrying the module
dissimilarity	the similarity cutoff to display as not carrying the module
max.n	the maximum number of modules to display
type	the module type to display
label.col	the color to label
...	other setting of 'oncoPrint'

Details

This function is to display the relationship of the predicted DEG modules and the patients.

'deg' can be set to display the modules from different datasets, e.g. the modules predicted from disease A and display them in the binary DEG matrix of disease B.

The output is a heatmap Plot where the modules with maximum observations are showed.

Value

A heatmap plot

Author(s)

Guofeng Meng

Examples

```
Plot(seed.mod, ann.er, max.n=5)
Plot(seed.mod, ann.er, deg=deg, max.n=5)
```

---

res.mod1 *clustered module 1*

Description

The module analysis results

Format

a list

Value

A list

res.mod2 *clustered module 1*

Description

The module analysis results

Format

a list

Value

A list

seed.mod *seed-based module analysis (part)*

Description

The seed-based module analysis results (only part)

Format

a list

Value

A list

seed.module *Predict the DEGs modules shared by patients*

Description

This function uses the output of `bi.deg` as input to predict the patient-DEG lists (or modules) shared by patients.

Usage

```
seed.module(deg, res.deg = NULL, test.patients = NULL, min.genes = 100,
  min.patients = 25, overlap = 0.85, model.method = c("slope.clustering",
  "max.square", "min.slope", "min.similarity")[1], cores = 1)
```

Arguments

deg	a binary matrix. This should be the output of bi.deg or a binary matrix
res.deg	a 'deg.specific' object. It should be the output of deg.specific. It is optional.
test.patients	the patient IDs used as test seed to find the modules
min.genes	the minimum number of genes in the modules
min.patients	the minimum number of patients in the modules
overlap	the minimum similarity for selected DEGs from two or more patients
model.method	the method to find the breakpoint of bi-clustering. It is accepted value including 'slope.clustering', 'max.square', 'min.slope', 'min.similarity'
cores	the thread number

Details

The function is to find the DEGs lists shared by patients. Like deg.specific, it carries out the bi-clustering analysis to the output of bi.deg. The difference is that this function has more complex setting and steps to predict DEGs modules shared by patients.

No matter whatever is the parameter setting, 'deg.module' will firstly try to find a modules shared by all the patients, where the finally patient number may less than the 'min.patients' and gene number may be less than 'min.genes'. If such module exists, it will named as 'M0' and the module genes of 'M0' will be filtered and they will not be included in other modules. Then, the patients of 'deg' will used as seed to do bi-clustering analysis.

The bi-clustering analysis is started by a DEG seed, composing of the DEGs of a patient. If 'res.deg' is set, only the patients with cross-validated DEGs will be used as seeds and the seed will be initilized with the cross-validated DEGs. Otherwise, all the patients will be used and all the DEGs are used as seed. The DEGs of patients will be gradually removed to ckeck if the left seed are observed in 'min.patients' when keeping the similarity is not less than 'overlap'. 'deg.module' will record the track of gene-patient number in the bi-clustering analysis, which is stored in 'curve' for each patient.

During the bi-clustering analysis, 'deg.module' will record the bi-clustering results at three scenarios:

'max.genes' records the patient and genes information when the seed is observed in 'min.patient'.

'max.patients' stores the patient and gene information when 'min.genes' are observed, which is also the terminated point of bi-clustering analysis.

'model' stores the gene/patient information when the gene-patients number 'curve' fits the criteria of 'model.method'.

The detailed information of the bi-clustering analysis results for all used patients is stored in 'decd.specific' of output list.

In this version, 'model.method' has four possible values: 'slope.clustering', 'max.square', 'min.slope' and 'min.similarity', which indicate the different four different modelling methods:

'slope.clustering' has maximum slope changes, which may indicate the inclusion/exclusion of molecular mechanism;

'max.square' is the gene-patients number that has the maximum product;

'min.slope' has the minimum slope in gene-patient number curve;

'min.similarity' is based on the similarity scores and the point with minimum similarity scores is choosed.

Value

A seed.module object. It has one key with prefix of 'decd':

'decd.input', the input information, including binary DEG matrix, test.patients and other parameter setting.

It may have one key of 'M0':

'M0', a modules shared by all the patients. In many cases, M0 is NULL when M0 is not predicted.

Other keys are patient IDs, which are the modules predicted with DEG seed of the patient. Each one have several keys:

'curve', the patient-gene number during bi-clustering analysis;

'max.genes', the patient and genes when 'min.patients' is observed in bi-clustering analysis;

'max.patients', the patient and genes when 'min.genes' is reached in bi-clustering analysis';

'model', the patient and genes at the breakpoint of the 'curve';

'genes.removed', the ordered genes that are removed from module during bi-clustering analysis;

'patients.added', the ordered patients that are added to module during bi-clustering analysis

Author(s)

Guofeng Meng

Examples

```
seed.mod <- seed.module(deg, min.genes=30, min.patient=10, overlap=0.85,
                        model.method='slope.clustering')
seed.mod2 <- seed.module(deg, model.method='min.similarity')
```

`summarize.cluster.module`

Summarize the DEG modules

Description

Summarize the DEG modules

Usage

```
## S3 method for class 'cluster.module'
summarize(res.module, max.n = 10, ...)
```

Arguments

`res.module`	a 'cluster.module' object returned by `cluster.module`
`max.n`	the number of modules to display
`...`	other setting

Details

This function summarize the DEG modules

Value

A data.frame.

Author(s)

Guofeng Meng

Examples

```
report.module <- summarize(cluster.mod)
```

`summarize.deg.specific` *Summarize the patient-specific DEGs*

Description

Summarize the binary differential expression matrix transformed by `bi.deg` or the cross-validated DEGs predicted by `deg.specific`.

Usage

```
## S3 method for class 'deg.specific'
summarize(res.deg, max.n = 10, ...)
```

Arguments

`res.deg`	a list returned by `deg.specific`
`max.n`	the number of DEGs to display
`...`	other setting

Details

This function summarizes the gene number before and after cross-validation.

Value

A data.frame.

Author(s)

Guofeng Meng

Examples

```
report.deg <- summarize(deg.spc)
```

summarize.seed.module *Summarize the DEG modules*

Description

Summarize the DEG modules

Usage

```
## S3 method for class 'seed.module'
summarize(res.module, max.n = 10, ...)

summarize(...)
```

Arguments

res.module	a 'seed.module' object returned by seed.module
max.n	the number of modules to display
...	other setting

Details

This function summarize the DEG modules

Value

A data.frame.

Author(s)

Guofeng Meng

Examples

```
report.module <- summarize(seed.mod)
```

Index