

# GNET2

March 26, 2019

**Type** Package

**Title** GNET2: Constructing module pathway networks from expression data

**Version** 0.1.0

**Author** Chen Chen

**Maintainer** The package maintainer <ccm3x@mail.missouri.edu>

**Description** Cluster genes to functional groups with E-M process.  
Iteratively perform TF assigning and Gene assigning, until the assignment of genes did not change, or max number of iterations reached

**License** Apache License 2.0

**Encoding** UTF-8

**LazyData** true

**LinkingTo** Rcpp

**Depends** R (>= 3.1)

**Imports** ggplot2, reshape2, grid, scales, DiagrammeR

**RoxygenNote** 6.1.1

**Suggests** knitr,  
rmarkdown

**VignetteBuilder** knitr

## R topics documented:

build_module . . . . .	2
build_moduleR . . . . .	2
calc_likelihood_score . . . . .	3
get_correlation_list . . . . .	4
gnet . . . . .	4
kneepointDetection . . . . .	5
plot_gene_group . . . . .	6
plot_group_correlation . . . . .	6
plot_tree . . . . .	7
<b>Index</b>	<b>8</b>

---

build_module	<i>Fit a regression tree based on Gaussian Likelihood score.</i>
--------------	--

---

### Description

Fit a regression tree based on Gaussian Likelihood score.

### Usage

```
build_module(X, Y, max_partition_level, cor_cutoff, min_divide_size)
```

### Arguments

X	A n by p matrix as input.
Y	A n by q matrix as response.
max_partition_level	Maximum partition level in the tree.
cor_cutoff	Cutoff for within group Pearson correlation coefficient, if all data belong to a node have average correlation greater or equal to this, the node would not split anymore.
min_divide_size	Minimum number of data belong to a node allowed for further split of the node.

### Value

A matrix for sample informatrion for each partition level. First column is feature index used by the node and second is the value used to split, the rest of the columns are the split of sample: 0 means less or equal, 1 means greater and -1 means the sample does not belong to this node.

### Examples

```
build_module(X = matrix(rnorm(50*100),50,100), Y = matrix(rnorm(50*200),50,200))
```

---

build_moduleR	<i>R version of build regression tree based on Gaussian Likelihood score. Runs slower than the build_module() which is C++ version but gives flexibility of customization of R functions.</i>
---------------	---

---

### Description

R version of build regression tree based on Gaussian Likelihood score. Runs slower than the build\_module() which is C++ version but gives flexibility of customization of R functions.

### Usage

```
build_moduleR(X, Y, max_partition_level, cor_cutoff, min_divide_size)
```

**Arguments**

max_partition_level	Maximum partition level in the tree.
cor_cutoff	Cutoff for within group Pearson correlation coefficient, if all data belong to a node have average correlation greater or equal to this, the node would not split anymore.
min_divide_size	Minimum number of data belong to a node allowed for further split of the node.

**Value**

A matrix for sample informatrion for each partition level. First column is feature index used by the node and second is the value used to split, the rest of the columns are the split of sample: 0 means less or equal, 1 means greater and -1 means the sample does not belong to this node.

**Examples**

```
build_moduleR(X = matrix(rnorm(50*100),50,100), Y = matrix(rnorm(50*200),50,200))
```

---

calc\_likelihood\_score *Calculate Gaussian Likelihood score.*

---

**Description**

Calculate Gaussian Likelihood score.

**Usage**

```
calc_likelihood_score(x, labels)
```

**Arguments**

x	A n by p matrix.
labels	A vector of length n, indicating the group of rows.

**Value**

The sum of log likelihood score of each group on each column.

**Examples**

```
calc_likelihood_score(x = matrix(rnorm(50*100),50,100), labels = c(rep(1,20),rep(2,30)))
```

---

get_correlation_list	<i>Calculate Pearson correlation coefficient within each group.</i>
----------------------	---

---

### Description

Calculate Pearson correlation coefficient within each group.

### Usage

```
get_correlation_list(x, labels)
```

### Arguments

x	A n by p matrix.
labels	A vector of length n, indicating the group of rows.

### Value

An array of Pearson correlation coefficient for each row, rows belong to the same group have same values.

### Examples

```
get_correlation_list(x = matrix(rnorm(50*100),50,100), labels = c(rep(1,20),rep(2,30)))
```

---

gnet	<i>Build regulation modules by iteratively perform TF assigning and Gene assigning, until the assignment of genes did not change, or max number of iterations reached</i>
------	---

---

### Description

Build regulation modules by iteratively perform TF assigning and Gene assigning, until the assignment of genes did not change, or max number of iterations reached

### Usage

```
gnet(rnaseq_data, reg_names, init_group_num = 8,
     max_partition_level = 3, cor_cutoff = 0.9, min_divide_size = 3,
     min_group_size = 2, max_iter = 5)
```

### Arguments

reg_names	A list of name of potential upstream regulators such as transcription factors.
init_group_num	Initial number of function clusters used by the algorithm.
max_partition_level	max_partition_level Maximum partition level in the tree.

cor_cutoff	Cutoff for within group Pearson correlation coefficient, if all data belong to a node have average correlation greater or equal to this, the node would not split anymore.
min_divide_size	Minimum number of data belong to a node allowed for further split of the node.
max_iter	Maximum number of iterations allowed if not converged.
gene_data	A p by n matrix of expression data of p genes and n samples, for example log2 RPKM from RNA-Seq.
min_divide_size	Minimum number of genes allowed in a group.

**Value**

A list of expression data of genes, expression data of regulators, within group score, table of tree structure and final assigned group of each gene.

**Examples**

```
set.seed(1)
gene_num = 1000
tf_num = 100
rnaseq_data <- matrix(rnorm(gene_num*12),gene_num,12)
tf_list <- paste0('TF',1:tf_num)
rownames(rnaseq_data) <- c(tf_list,paste0('gene',1:(nrow(rnaseq_data)-length(tf_list))))
colnames(rnaseq_data) <- paste0('sample_',1:ncol(rnaseq_data))
gnet_result = gnet(rnaseq_data,tf_list,init_group_num,max_partition_level,
cor_cutoff,min_divide_size,min_group_size,max_iter,min_group_num)
```

---

kneepointDetection	<i>Detect the knee point of the array.</i>
--------------------	--

---

**Description**

Detect the knee point of the array.

**Usage**

```
kneepointDetection(vect)
```

**Arguments**

vect	A list of sorted numbers.
------	---------------------------

**Value**

The index of the data point which is the knee.

**Examples**

```
kneepointDetection(sort(rnorm(100),T))
```

---

plot_gene_group	<i>Plot the regulators module and heatmap of the expression inferred downstream genes for each sample. It can be interpreted as two parts: the bars at the top shows how samples are splited by the regression tree and the heatmap at the bottom shows how downstream genes are regulated by each subgroup determined by the regulators.</i>
-----------------	---

---

### Description

Plot the regulators module and heatmap of the expression inferred downstream genes for each sample. It can be interpreted as two parts: the bars at the top shows how samples are splited by the regression tree and the heatmap at the bottom shows how downstream genes are regulated by each subgroup determined by the regulators.

### Usage

```
plot_gene_group(gnet_result, group_idx)
```

### Arguments

gnet_result	Results returned by gnet().
group_idx	Index of the module..

---

plot_group_correlation	<i>Plot the correlation of each group and auto detected knee point. It can be used to determined which clustered are kept for further analysis.</i>
------------------------	---

---

### Description

Plot the correlation of each group and auto detected knee point. It can be used to determined which clustered are kept for further analysis.

### Usage

```
plot_group_correlation(gnet_result)
```

### Arguments

avg_cor_list	The average within group Pearson correlation coefficient of each group from gnet().
--------------	---

### Value

A list of indices of the data point with correlation higher than the knee point.

---

`plot_tree`*Plot the regression tree given the index of a module.*

---

**Description**

Plot the regression tree given the index of a module.

**Usage**

```
plot_tree(gnet_result, group_idx)
```

**Arguments**

<code>gnet_result</code>	Results returned by <code>gnet()</code> .
<code>group_idx</code>	Index of the module..

# Index

`build_module`, [2](#)  
`build_moduleR`, [2](#)  
  
`calc_likelihood_score`, [3](#)  
  
`get_correlation_list`, [4](#)  
`gnet`, [4](#)  
  
`kneepointDetection`, [5](#)  
  
`plot_gene_group`, [6](#)  
`plot_group_correlation`, [6](#)  
`plot_tree`, [7](#)