

Package ‘DNLC’

December 11, 2016

Type Package

Title Differential Network Local Consistency Analysis

Version 1.0.0

Date 2016-11-23

Maintainer Yusheng Ding <yushengding93@gmail.com>

Description Using Local Moran's I for detection of differential network local consistency.

License GPL (>= 2)

Imports igraph, spdep, fdrtool, GOstats, locfdr, mvtnorm, caTools

LazyData true

Depends R (>= 2.10)

NeedsCompilation no

Author Yao Lu [aut],
Yusheng Ding [aut, cre],
Linqing Liu [aut],
Tianwei Yu [aut]

Repository CRAN

Date/Publication 2016-12-11 12:26:26

R topics documented:

| | |
|------------------------------------|---|
| cal_lmi_data | 2 |
| DNLC_statistics | 3 |
| gene_fdrtest | 4 |
| init_simulation_gene_net | 4 |
| significant_genes | 5 |

| | |
|--------------|----------|
| Index | 7 |
|--------------|----------|

| | |
|--------------|--|
| cal_lmi_data | <i>Calculate local moran's I matrix for a network and a gene expression matrix</i> |
|--------------|--|

Description

cal_lmi_data() will calculate the local moran's I data for a input igraph object and gene expression data matrix.

Usage

```
cal_lmi_data(gene_expr, gene_graph)
```

Arguments

| | |
|------------|--|
| gene_expr | Expression for genes. Each row is a gene, and each column is a sample. |
| gene_graph | The graph of gene network. |

Details

cal_lmi_data() will calculate the local moran's I matrix for a input igraph object and gene expression data matrix. The function will return a matrix in the same dimension of the input gene expression matrix. Every gene x's lmi data takes a row.

Value

A table of local moran's I data. Row name is gene id. Each row stands for the local moran's I data of gene x. Each column stands for a sample.

Examples

```
## Not run:  
simulation <- init_simulation_gene_net()  
lmi_data = cal_lmi_data( simulation$gene_expr, simulation$gene_graph)  
t_data = DNLC_statistics(simulation$gene_graph, simulation$gene_expr,  
  simulation$patient_matrix, lmi_data = lmi_data)  
fdr_result <- gene_fdrtest(t_data)  
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)  
## End(Not run)
```

| | |
|-----------------|---|
| DNLC_statistics | <i>calculate t statistics for gene graph using DNLC method.</i> |
|-----------------|---|

Description

a function to calculate t statistics for genes in the graph.

Usage

```
DNLC_statistics(gene_graph, gene_expr = "x", clinic_data = "y",  
               confounder_matrix = NULL, lmi_data = NULL)
```

Arguments

| | |
|-------------------|---------------------------------------|
| gene_graph | graph of gene |
| gene_expr | expr of gene |
| clinic_data | patient data |
| confounder_matrix | other message describe clinic message |
| lmi_data | lmi data for each gene. |

Details

This function first calculates the matrix of local moran's I, and then conducts testing for the association of each gene's local moran's I with the clinical outcome variable. Clinical confounder variables such as age, gender etc can be included.

Value

| | |
|-------------|----------------------|
| all_gene_id | gene ids in graph |
| t_data | t-data for each gene |

Examples

```
## Not run:  
simulation <- init_simulation_gene_net()  
t_data = DNLC_statistics(simulation$gene_graph, simulation$gene_expr,  
                        simulation$patient_matrix, lmi_data = simulation$lmi_matrix)  
fdr_result <- gene_fdrtest(t_data)  
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)  
  
## End(Not run)
```

| | |
|--------------|--|
| gene_fdrtest | <i>Use local false discovery rate for the detection of genes with significant LMI change</i> |
|--------------|--|

Description

This function use locfdr function to calculate fdr_result

Usage

```
gene_fdrtest(gene.data)
```

Arguments

gene.data gene_id_all: gene id t_data: t statistic for each gene

Value

return fdr_result for t_data

fdr\$name all gene id

fdr\$fdr fdr value for gene

Examples

```
## Not run:
simulation <- init_simulation_gene_net()
t_data = DNLC_statistics(simulation$gene_graph, simulation$gene_expr,
  simulation$patient_matrix, lmi_data=simulation$lmi_matrix)
fdr_result <- gene_fdrtest(t_data)
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)

## End(Not run)
```

| | |
|--------------------------|---|
| init_simulation_gene_net | <i>Create a random network for simulation</i> |
|--------------------------|---|

Description

This function will create a network for DNLC. This function will change correlation of chosen genes and its one hop neighbor between treatment groups to simulate LMI changes.

Usage

```
init_simulation_gene_net(base_correlation = 0.4,
  change_correlation = 0.8, sample_size = 100, num_gene = 5000, change_gene_num=5)
```

Arguments

| | |
|--------------------|---|
| base_correlation | base correlation of network |
| change_correlation | change correlation for selected genes |
| sample_size | multi size of patient data |
| num_gene | gene number in the network |
| change_gene_num | number of genes around which the correlation structure is to be changed |

Value

| | |
|----------------|--------------------------------|
| lmi_matrix | matrix of local moran's I data |
| patient_matrix | matrix of patient data |
| neigh_list | id of changed gene. |
| gene_graph | igraph object of gene network |
| gene_expr | gene expression data matrix |

Examples

```
## Not run:
simulation <- init_simulation_gene_net()
t_data = DNLC_statistics(simulation$gene_graph, simulation$gene_expr,
  simulation$patient_matrix, lmi_data = simulation$lmi_matrix)
fdr_result <- gene_fdrtest(t_data)
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)
## End(Not run)
```

significant_genes *Selecting significant genes according to fdr result*

Description

Choose the significant genes according to fdr result

Usage

```
significant_genes(fdr_obj, thres)
```

Arguments

| | |
|---------|--|
| fdr_obj | fdr result come from function gene_fdrtest |
| thres | threshold to identify significant genes |

Value

ID of significant genes

Examples

```
## Not run:
simulation <- init_simulation_gene_net()
t_data = DNLC_statistics(simulation$gene_graph, simulation$gene_expr,
  simulation$patient_matrix, lmi_data = simulation$lmi_matrix)
fdr_result <- gene_fdrtest(t_data)
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)

## End(Not run)
```

Index

*Topic \textasciitildekw1

- cal_lmi_data, 2
- DNLC_statistics, 3
- gene_fdrtest, 4
- init_simulation_gene_net, 4
- significant_genes, 5

*Topic \textasciitildekw2

- cal_lmi_data, 2
- DNLC_statistics, 3
- gene_fdrtest, 4
- init_simulation_gene_net, 4
- significant_genes, 5

cal_lmi_data, 2

DNLC_statistics, 3

gene_fdrtest, 4

init_simulation_gene_net, 4

significant_genes, 5