Package 'NLCD'

November 25, 2016

Type Package

Title Network Local Consistency Detection

Version 1.0.0

Date 2016-11-23

Author Yao Lu, Yusheng Ding, Lingqing Liu, Tianwei Yu

Maintainer Yusheng Ding <yushengding93@gmail.com>

Description Using Local Moran's I for Network local consistency Detection

License GPL >= 2

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

LazyData true

R topics documented:

`cal_lmi_data` *calculate lmi for a network*

Description

cal_lmi_data() will calculate the lmi data for a input network.

Usage

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

Arguments

gene_expr	expression for gene
gene_graph	The graph of gene network.

Details

this function will create a table of lmi data, and each column in this table stand for one lmi of one gene expression.

Value

a table for lmi data. Each line of this table stand for one lmi of one gene expression.

Examples

```
data("gene_graph")
data("gene_expr")
data("patient_raw_data")
lmi_data <- cal_lmi_data(gene_expr, gene_graph)
patient_data<-solve_patient_data(patient_raw_data)
t_data <- top3_bottom3_t_data(gene_graph, patient_data, lmi_data)
fdr_result <- gene_fdrtest(t_data)
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)
## The function is currently defined as
function (gene_expr, gene_graph)
{
    dis_mat <- n_hop_matrix(gene_graph, hop_n = 2)
    listw_obj <- spdep::mat2listw(dis_mat, style = "W")
    gene_expr <- t(apply(gene_expr, 1, function(gene_expr) (gene_expr -
        min(gene_expr))/(max(gene_expr) - min(gene_expr))))
    lmi_data <- double()
    for (i in seq(1, ncol(gene_expr), 1)) {
        lmi <- spdep::localmoran(gene_expr[, i], listw = listw_obj)
        lmi_data <- cbind(lmi_data, lmi[, 1])
    }
```

```
  return(lmi_data)
}
```

---

gene_expr *gene expression*

Description

a data set for gene expression.

Usage

```
data("gene_expr")
```

Format

A data frame with 7965 observations on the following 161 variables.

GSM258912 a numeric vector

GSM258913 a numeric vector

GSM258914 a numeric vector

GSM258915 a numeric vector

GSM258916 a numeric vector

GSM258917 a numeric vector

GSM258918 a numeric vector

GSM258919 a numeric vector

GSM258920 a numeric vector

GSM258921 a numeric vector

GSM258922 a numeric vector

GSM258923 a numeric vector

GSM258924 a numeric vector

GSM258925 a numeric vector

GSM258926 a numeric vector

GSM258927 a numeric vector

GSM258928 a numeric vector

GSM258929 a numeric vector

GSM258930 a numeric vector

GSM258931 a numeric vector

GSM258932 a numeric vector

GSM258933 a numeric vector

GSM258934 a numeric vector

GSM258935 a numeric vector
GSM258936 a numeric vector
GSM258937 a numeric vector
GSM258938 a numeric vector
GSM258939 a numeric vector
GSM258940 a numeric vector
GSM258941 a numeric vector
GSM258942 a numeric vector
GSM258943 a numeric vector
GSM258944 a numeric vector
GSM258945 a numeric vector
GSM258946 a numeric vector
GSM258947 a numeric vector
GSM258948 a numeric vector
GSM258949 a numeric vector
GSM258950 a numeric vector
GSM258951 a numeric vector
GSM258952 a numeric vector
GSM258953 a numeric vector
GSM258954 a numeric vector
GSM258955 a numeric vector
GSM258956 a numeric vector
GSM258957 a numeric vector
GSM258958 a numeric vector
GSM258959 a numeric vector
GSM258960 a numeric vector
GSM258961 a numeric vector
GSM258962 a numeric vector
GSM258963 a numeric vector
GSM258964 a numeric vector
GSM258965 a numeric vector
GSM258966 a numeric vector
GSM258967 a numeric vector
GSM258968 a numeric vector
GSM258969 a numeric vector
GSM258970 a numeric vector
GSM258971 a numeric vector

GSM258972 a numeric vector
GSM258973 a numeric vector
GSM258974 a numeric vector
GSM258975 a numeric vector
GSM258976 a numeric vector
GSM258977 a numeric vector
GSM258978 a numeric vector
GSM258979 a numeric vector
GSM258980 a numeric vector
GSM258981 a numeric vector
GSM258982 a numeric vector
GSM258983 a numeric vector
GSM258984 a numeric vector
GSM258985 a numeric vector
GSM258986 a numeric vector
GSM258987 a numeric vector
GSM258988 a numeric vector
GSM258989 a numeric vector
GSM258990 a numeric vector
GSM258991 a numeric vector
GSM258992 a numeric vector
GSM258993 a numeric vector
GSM258994 a numeric vector
GSM258995 a numeric vector
GSM258996 a numeric vector
GSM258997 a numeric vector
GSM258998 a numeric vector
GSM258999 a numeric vector
GSM259000 a numeric vector
GSM259001 a numeric vector
GSM259002 a numeric vector
GSM259003 a numeric vector
GSM259004 a numeric vector
GSM259005 a numeric vector
GSM259006 a numeric vector
GSM259007 a numeric vector
GSM259008 a numeric vector

GSM259009 a numeric vector
GSM259010 a numeric vector
GSM259011 a numeric vector
GSM259012 a numeric vector
GSM259013 a numeric vector
GSM259014 a numeric vector
GSM259015 a numeric vector
GSM259016 a numeric vector
GSM259017 a numeric vector
GSM259018 a numeric vector
GSM259019 a numeric vector
GSM259020 a numeric vector
GSM259021 a numeric vector
GSM259022 a numeric vector
GSM259023 a numeric vector
GSM259024 a numeric vector
GSM259025 a numeric vector
GSM259026 a numeric vector
GSM259027 a numeric vector
GSM259028 a numeric vector
GSM259029 a numeric vector
GSM259030 a numeric vector
GSM259031 a numeric vector
GSM259032 a numeric vector
GSM259033 a numeric vector
GSM259034 a numeric vector
GSM259035 a numeric vector
GSM259036 a numeric vector
GSM259037 a numeric vector
GSM259038 a numeric vector
GSM259039 a numeric vector
GSM259040 a numeric vector
GSM259041 a numeric vector
GSM259042 a numeric vector
GSM259043 a numeric vector
GSM259044 a numeric vector
GSM259045 a numeric vector

GSM259046 a numeric vector

GSM259047 a numeric vector

GSM259048 a numeric vector

GSM259049 a numeric vector

GSM259050 a numeric vector

GSM259051 a numeric vector

GSM259052 a numeric vector

GSM259053 a numeric vector

GSM259054 a numeric vector

GSM259055 a numeric vector

GSM259056 a numeric vector

GSM259057 a numeric vector

GSM259058 a numeric vector

GSM259059 a numeric vector

GSM259060 a numeric vector

GSM259061 a numeric vector

GSM259062 a numeric vector

GSM259063 a numeric vector

GSM259064 a numeric vector

GSM259065 a numeric vector

GSM259066 a numeric vector

GSM259067 a numeric vector

GSM259068 a numeric vector

GSM259069 a numeric vector

GSM259070 a numeric vector

GSM259071 a numeric vector

GSM259072 a numeric vector

Examples

```
data(gene_expr)
```

gene_fdrtest *Using locfdr to calulate fdr_result for a t.test result*

Description

This function use locfdr function to calculate fdr_result

Usage

```
gene_fdrtest(t_data)
```

Arguments

gene.corr	gene_id_all: all gene id t_data: t_data for each gene

Value

return fdr_result for t_data

fdr$name	all gene id
fdr$fdr	fdr value for gene

Examples

```
data("gene_graph")
data("gene_expr")
data("patient_raw_data")
lmi_data <- cal_lmi_data(gene_expr, gene_graph)
patient_data<-solve_patient_data(patient_raw_data)
t_data <- top3_bottom3_t_data(gene_graph, patient_data, lmi_data)
fdr_result <- gene_fdrtest(t_data)
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)

## The function is currently defined as
function (gene.corr)
{
    name <- gene.corr$gene_id_all
    coef_val <- gene.corr$t_data
    z.cent <- scale(coef_val)
    fdr_result <- data.frame(fdr = locfdr(z.cent, nulltype = 0,
        plot = 4)$fdr, name)
    return(fdr_result)
  }
```

gene_graph *gene graph*

Description

a data set of gene graph.

Usage

```
data("gene_graph")
```

Format

The format is: List of 10 $: num 7965 $: logi FALSE $: num [1:29996] 1 3 5 7 8 9 10 11 13 14 ... $: num [1:29996] 0 2 4 6 6 6 6 6 12 12 ... $: num [1:29996] 0 1 2 3 4 ... $: num [1:29996] 0 3323 1565 18196 18218 ... $: num [1:7966] 0 0 1 1 2 2 3 3 4 5 ... $: num [1:7966] 0 1 111 112 145 146 379 384 509 547 ... $:List of 4 ..$: num [1:3] 1 0 1 ..$: Named list() ..$:List of 1 .. ..$ name: chr [1:7965] "27004" "4223" "151050" "10133" ... ..$: list() $:<environment: 0x362b2b0> - attr(*, "class")= chr "igraph"

Examples

```
data(gene_graph)
```

---

init_simulation_gene_net
Creating a simulation network for further calculation.

Description

This function will create a network for NLCD. This function will change correlation of chosen gene's and its one hop neighbor.

Usage

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

Arguments

base_correlation	base correlation of network
change_correlation	change correlation of network
sample_size	multi size of patient data
num_gene	gene num in the network

Value

lmi_matrix	matrix of lmi data
patient_matrix	matrix of patients' status
gene_code_list	code of gene.0 for normal gene; 1 for changed gene; 2 for chosen gene
neigh_list	id of gene which is changed.

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (base_correlation = 0.4, change_correlation = 0.8, sample_size = 100,
    num_gene = 5000)
{
    n_hop_matrix <- function(gene_graph, hop_n) {
        dis_mat <- igraph::shortest.paths(gene_graph)
        dis_mat[dis_mat > hop_n] <- 0
        dis_mat[dis_mat == hop_n] <- 0.5
        return(dis_mat)
    }
    netdensity = 1
    gene_graph <-  igraph::barabasi.game(num_gene, m = netdensity)
    deg_data <- igraph::degree(gene_graph)
    simu_sel_gene_list = numeric()
    neigh_list_3 <- numeric()
    simu_gene_num = 5
    times = 100
    while (length(simu_sel_gene_list) < simu_gene_num) {
        sel_gene <- sample(which(deg_data <= 10 & deg_data >=
            5), 1)
        if (sel_gene %in% neigh_list_3) {
            times = times - 1
            if (times < 0) {
                return(NULL)
            }
            (next)()
        }
        simu_sel_gene_list <- c(simu_sel_gene_list, sel_gene)
        neigh_list_3 <- c(neigh_list_3, unlist(igraph::neighborhood(gene_graph,
            order = 3, nodes = sel_gene, mode = "all")))
    }
    neigh_list <- numeric()
    for (sel_gene in simu_sel_gene_list) {
        neigh_list <- c(neigh_list, unlist(igraph::neighborhood(gene_graph,
            order = 1, nodes = sel_gene, mode = "all")))
    }
    gene_code_list <- rnorm(num_gene, mean = 0, sd = 0)
    for (gid in neigh_list) {
```

```
        gene_code_list[gid] <- 1
    }
    for (gid in simu_sel_gene_list) {
        gene_code_list[gid] <- 2
    }
    s <- igraph::shortest.paths(gene_graph)
    s1 <- base_correlation^(s)
    x <- t(mvtnorm::rmvnorm(n = sample_size, mean = rep(0, num_gene),
        sigma = s1, method = "chol"))
    x <- t(apply(x, 1, function(x) (x - min(x))/(max(x) - min(x))))
    y <- c(rnorm(sample_size, mean = 0, sd = 0), rnorm(sample_size,
        mean = 1, sd = 0))
    dis_mat <- n_hop_matrix(gene_graph = gene_graph, hop_n = 2)
    listw_obj <- spdep::mat2listw(dis_mat, style = "W")
    origin_lmi_data <- double()
    for (i in seq(1, ncol(x), 1)) {
        lmi <- spdep::localmoran(x[, i], listw = listw_obj)
        origin_lmi_data <- cbind(origin_lmi_data, lmi[, 1])
    }
    for (i in neigh_list) {
        for (j in neigh_list) {
            s1[i, j] <- change_correlation^s[i, j]
        }
    }
    x <- t(mvtnorm::rmvnorm(n = sample_size, mean = rep(0, num_gene),
        sigma = s1, method = "chol"))
    x <- t(apply(x, 1, function(x) (x - min(x))/(max(x) - min(x))))
    y <- c(rnorm(sample_size, mean = 0, sd = 0), rnorm(sample_size,
        mean = 1, sd = 0))
    lmi_data <- double()
    for (i in seq(1, ncol(x), 1)) {
        lmi <- spdep::localmoran(x[, i], listw = listw_obj)
        lmi_data <- cbind(lmi_data, lmi[, 1])
    }
    lmi_matrix <- cbind(lmi_data, origin_lmi_data)
    return(list(lmi_matrix = lmi_matrix, patient_matrix = y,
        gene_code_list = gene_code_list, neigh_list = neigh_list))
}
```

NLCD *Network Local Consistency Detection*

Description

Using Local Moran's I for Network local consistency Detection

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.
sing Local Moran's I for Network local consistency Detection

Author(s)

Yao Lu, Yusheng Ding, Lingqing Liu, Tianwei Yu

Maintainer: Yusheng Ding <yushengding93@gmail.com>

Examples

```
data("gene_graph")
data("gene_expr")
data("patient_raw_data")
lmi_data <- cal_lmi_data(gene_expr, gene_graph)
patient_data<-solve_patient_data(patient_raw_data)
t_data <- top3_bottom3_t_data(gene_graph, patient_data, lmi_data)
fdr_result <- gene_fdrtest(t_data)
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)
```

---

n_hop_matrix *Turning a gene graph to a matrix.*

Usage

```
n_hop_matrix(gene_graph, hop_n)
```

Arguments

gene_graph	The graph of the gene network.
hop_n	the hop should be described in the matrix

Value

a matrix describe the connection of the graph. 0.5 stand for the shortest path between two node equal to hop_n

See Also

cal_lmi_data() init_simulation_gene_net()

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (gene_graph, hop_n)
{
    dis_mat <- igraph::shortest.paths(gene_graph)
    dis_mat[dis_mat > hop_n] <- 0
    dis_mat[dis_mat == hop_n] <- 0.5
    return(dis_mat)
  }
```

---

patient_raw_data *patient raw data*

Description

a data set of patient situation

Usage

```
data("patient_raw_data")
```

Format

A data frame with 161 observations on the following 7 variables.

`cel` a factor with levels `JD0001-ALL-v5-U133A` `JD0002-ALL-v5-U133A` `JD0003-ALL-v5-U133A` `JD0004-ALL-v5-U133A` `JD0005-ALL-v5-U133A` `JD0006-ALL-v5-U133A` `JD0008-ALL-v5-U133A` `JD0009-ALL-v5-U133A` `JD0011-ALL-v5-U133A` `JD0012-ALL-v5-U133A` `JD0013-ALL-v5-U133A` `JD0015-ALL-v5-U133A` `JD0016-ALL-v5-U133A` `JD0017-ALL-v5-U133A` `JD0018-ALL-v5-U133A` `JD0019-ALL-v5-U133A` `JD0020-ALL-v5-U133A` `JD0022-ALL-v5-U133A` `JD0023-ALL-v5-U133A` `JD0025-ALL-v5-U133A` `JD0030-ALL-v5-U133A` `JD0031-ALL-v5-U133A` `JD0032-ALL-v5-U133A` `JD0036-ALL-v5-U133A` `JD0037-ALL-v5-U133A` `JD0038-ALL-v5-U133A` `JD0039-ALL-v5-U133A` `JD0041-ALL-v5-U133A` `JD0042-ALL-v5-U133A` `JD0043-ALL-v5-U133A` `JD0044-ALL-v5-U133A` `JD0047-ALL-v5-U133A` `JD0048-ALL-v5-U133A` `JD0050-ALL-v5-U133A` `JD0055-ALL-v5-U133A` `JD0056-ALL-v5-U133A` `JD0057-ALL-v5-U133A` `JD0058-ALL-v5-U133A` `JD0059-ALL-v5-U133A` `JD0062-ALL-v5-U133A` `JD0064-ALL-v5-U133A` `JD0065-ALL-v5-U133A` `JD0066-ALL-v5-U133A` `JD0067-ALL-v5-U133A` `JD0073-ALL-v5-U133A` `JD0074-ALL-v5-U133A` `JD0076-ALL-v5-U133A` `JD0079-ALL-v5-U133A` `JD0080-ALL-v5-U133A` `JD0083-ALL-v5-U133A` `JD0084-ALL-v5-U133A` `JD0085-ALL-v5-U133A` `JD0088-ALL-v5-U133A` `JD0095-ALL-v5-U133A` `JD0096-ALL-v5-U133A` `JD0097-ALL-v5-U133A` `JD0098-ALL-v5-U133A` `JD0099-ALL-v5-U133A` `JD0100-ALL-v5-U133A` `JD0101-ALL-v5-U133A` `JD0102-ALL-v5-U133A` `JD0104-ALL-v5-U133A` `JD0105-ALL-v5-U133A` `JD0106-ALL-v5-U133A` `JD0107-ALL-v5-U133A` `JD0108-ALL-v5-U133A` `JD0109-ALL-v5-U133A` `JD0110-ALL-v5-U133A` `JD0111-ALL-v5-U133A` `JD0117-ALL-v5-U133A` `JD0118-ALL-v5-U133A` `JD0119-ALL-v5-U133A` `JD0120-ALL-v5-U133A` `JD0121-ALL-v5-U133A` `JD0122-ALL-v5-U133A`

JD0123-ALL-v5-U133A JD0124-ALL-v5-U133A JD0125-ALL-v5-U133A JD0126-ALL-v5-U133A
JD0127-ALL-v5-U133A JD0130-ALL-v5-U133A JD0131-ALL-v5-U133A JD0133-ALL-v5-U133A
JD0135-ALL-v5-U133A JD0137-ALL-v5-U133A JD0138-ALL-v5-U133A JD0139-ALL-v5-U133A
JD0140-ALL-v5-U133A JD0144-ALL-v5-U133A JD0149-ALL-v5-U133A JD0151-ALL-v5-U133A
JD0163-ALL-v5-U133A JD0165-ALL-v5-U133A JD0166-ALL-v5-U133A JD0167-ALL-v5-U133A
JD0173-ALL-v5-U133A JD0188-ALL-v5-U133A JD0196-ALL-v5-U133A JD0202-ALL-v5-U133A
JD0203-ALL-v5-U133A JD0212-ALL-v5-U133A JD0219-ALL-v5-U133A JD0221-ALL-v5-U133A
JD0225-ALL-v5-U133A JD0226-ALL-v5-U133A JD0227-ALL-v5-U133A JD0228-ALL-v5-U133A
JD0233-ALL-v5-U133A JD0236-ALL-v5-U133A JD0245-ALL-v5-U133A JD0246-ALL-v5-U133A
JD0247-ALL-v5-U133A JD0249-ALL-v5-U133A JD0252-ALL-v5-U133A JD0255-ALL-v5-U133A
JD0256-ALL-v5-U133A JD0258-ALL-v5-U133A JD0264- ALL-v5-U133A JD0265- ALL-v5-U133A
JD0266- ALL-v5-U133A JD0267- ALL-v5-U133A JD0268- ALL-v5-U133A JD0269- ALL-v5-U133A
JD-ALD004-v5-U133A JD-ALD009-v5-U133A JD-ALD054-v5-U133A JD-ALD066-v5-U133A JD-ALD071-v5-U133A
JD-ALD078-v5-U133A JD-ALD083-v5-U133A JD-ALD108-v5-U133A JD-ALD113-v5-U133A JD-ALD115-v5-U133A
JD-ALD163-v5-U133A JD-ALD196-v5-U133A JD-ALD258-v5-U133A JD-ALD262-v5-U133A JD-ALD283-v5-U133A
JD-ALD360-v5-U133A JD-ALD374-v5-U133A JD-ALD375-v5-U133A JD-ALD428-v5-U133A JD-ALD431-v5-U133A
JD-ALD436-v5-U133A JD-ALD437-v5-U133A JD-ALD485-v5-U133A JD-ALD488-v5-U133A JD-ALD491-v5-U133A
JD-ALD493-v5-U133A JD-ALD494-v5-U133A JD-ALD510-v5-U133A JD-ALD515-v5-U133A JD-ALD537-v5-U133A
JD-ALD540-v5-U133A JD-ALD553-v5-U133A JD-ALD557-v5-U133A JD-ALD610-v5-U133A JD-ALD611-v5-U133A
JD-ALD612-v5-U133A JD-ALD613-v5-U133A JD-ALD619-v5-U133A

`GEO.id` a numeric vector

`WBCCountDay0` a numeric vector

`WBCCountDay3` a numeric vector

`WBCCountDay0Log` a numeric vector

`WBCCountDay3Log` a numeric vector

`residuals.WBCCountDay3Log.WBCCountDay0Log.` a numeric vector

Examples

```
data(patient_raw_data)
```

`significant_genes` *Selecting significant genes according to fdr result*

Usage

```
significant_genes(fdr_obj = fdr_result, thres = 0.2)
```

Arguments

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

Value

ID of significant genes

Examples

```
data("gene_graph")
data("gene_expr")
data("patient_raw_data")
lmi_data <- cal_lmi_data(gene_expr, gene_graph)
patient_data<-solve_patient_data(patient_raw_data)
t_data <- top3_bottom3_t_data(gene_graph, patient_data, lmi_data)
fdr_result <- gene_fdrtest(t_data)
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)

## The function is currently defined as
function (fdr_obj, thres)
{
    sel.entrez <- fdr_obj$name[which(fdr_obj$fdr < thres)]
    return(sel.entrez)
  }
```

`simulate_NLCD` *simulate NLCD*

Description

simulate NLCD

Usage

```
simulate_NLCD(base_correlation = 0.4,
      change_correlation = 0.8, sample_size = c(50, 100), num_gene = 5000)
```

Arguments

`base_correlation`	base correlation of network
`change_correlation`	change correlation of network
`sample_size`	multi size of patient data
`num_gene`	gene num in the network

Value

picture for fdr_result

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (base_correlation = 0.4, change_correlation = 0.8, sample_size = c(50,
    100), num_gene = 5000)
{
    simulation <- init_simulation_gene_net(base_correlation = base_correlation,
        change_correlation = change_correlation, sample_size = max(sample_size),
        num_gene = num_gene)
    over = 0
    while (is.null(simulation)) {
        over = over + 1
        simulation <- init_simulation_gene_net(base_correlation = base_correlation,
            change_correlation = change_correlation, sample_size = max(sample_size),
            num_gene = num_gene)
    }
    lmi_data = simulation$lmi_matrix
    patient_matrix = simulation$patient_matrix
    gene_code_list = simulation$gene_code_list
    neigh_list = simulation$neigh_list
    labels = gene_code_list
    simu_sel_gene = which(gene_code_list == 2)
    labels[which(labels == 2)] = 1
    gene_ids <- seq(1, num_gene, 1)
    sample_size = sort(sample_size, decreasing = TRUE)
    left = seq(1, max(sample_size))
    right = seq(1, max(sample_size)) + max(sample_size)
    for (sz in sample_size) {
        left = sample(left, sz)
        right = sample(right, sz)
        pl = cbind(left, right)
        fdr_result <- sim_cal_fdr_result(gene_ids, patient_matrix[pl],
            lmi_data[, pl])
    }
  }
```

`sim_cal_fdr_result` *calucating fdr result for simulation network*

Description

calucating fdr result for simulation network. The reason why we create this function is the simulation data structure is different from real data set.

Usage

```
sim_cal_fdr_result(gene_ids, patient_matrix, lmi_data)
```

Arguments

gene_ids	id for all genes
patient_matrix	patients' data
lmi_data	lmi data for each gene

Value

fdr result for gene

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (gene_ids, patient_matrix, lmi_data)
{
    gene_fdrtest <- function(gene.corr) {
        name <- gene.corr$gene_id_all
        coef_val <- gene.corr$t_data
        z.cent <- coef_val - median(coef_val)
        fdr_result <- data.frame(fdr = fdrtool::fdrtool(z.cent, statistic = "normal"),
            name)
        return(fdr_result)
    }
    t_gene_test <- function(clinic_data, gene_id, lmi_data) {
        gene.corr <- data.frame(clinic = clinic_data, lmi = lmi_data[which(gene_ids ==
            gene_id), ])
        result <- t.test(gene.corr$lmi[which(gene.corr$clinic ==
            0)], gene.corr$lmi[which(gene.corr$clinic == 1)])
        return(result$statistic)
    }
    top3_bottom3_t_data <- function(gene_id_all, patient_matrix,
        lmi_data) {
        t_data = numeric()
        for (single_gene in gene_id_all) {
            t_ts <- t_gene_test(clinic_data = patient_matrix,
                gene_id = single_gene, lmi_data = lmi_data)
            t_data <- append(t_data, t_ts)
        }
        return(data.frame(gene_id_all = gene_id_all, t_data = t_data))
    }
    p <- top3_bottom3_t_data(gene_ids, patient_matrix, lmi_data)
    fdr_result <- gene_fdrtest(p)
    return(fdr_result)
  }
```

`solve_patient_data` *Changing the sturcture of raw patient data*

Description

Changing the sturcture of raw patient data

Usage

```
solve_patient_data(patient_raw_data)
```

Arguments

`patient_raw_data`	Unprocessed patient data

Value

data of patient which can be used in t.test calculation process

Examples

```
data("gene_graph")
data("gene_expr")
data("patient_raw_data")
lmi_data <- cal_lmi_data(gene_expr, gene_graph)
patient_data<-solve_patient_data(patient_raw_data)
t_data <- top3_bottom3_t_data(gene_graph, patient_data, lmi_data)
fdr_result <- gene_fdrtest(t_data)
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)

## The function is currently defined as
function (patient_raw_data)
{
    patient_data <- patient_raw_data$WBCCountDay3Log - patient_raw_data$WBCCountDay0Log
    return(patient_data)
  }
```

top3_bottom3_t_data *Using t.test to calculate the t_data*

Description

using t.test to calculate the t_data.

Usage

```
top3_bottom3_t_data(gene_graph, patient_data, lmi_data)
```

Arguments

gene_graph	The graph of gene network.
patient_data	The matrix of patient data
lmi_data	The data of local moran's I

Value

gene_id_all: id of all genes t_data: matirx of t.test result

Examples

```
data("gene_graph")
data("gene_expr")
data("patient_raw_data")
lmi_data <- cal_lmi_data(gene_expr, gene_graph)
patient_data<-solve_patient_data(patient_raw_data)
t_data <- top3_bottom3_t_data(gene_graph, patient_data, lmi_data)
fdr_result <- gene_fdrtest(t_data)
sig_genes <- significant_genes(fdr_obj = fdr_result, thres = 0.2)

## The function is currently defined as
function (gene_graph, patient_data, lmi_data)
{
    gene_ids <- igraph::V(gene_graph)$name
    t_data = numeric()
    for (gene_id in gene_ids) {
        sel = which(gene_ids == gene_id)
        lmi = lmi_data[sel, ]
        gene.corr <- data.frame(clinic = patient_data, lmi = lmi)
        gg <- gene.corr[order(gene.corr$clinic), ]
        t.result <- t.test(gg[1:53, 2], gg[106:161, 2])
        t_data <- append(t_data, t.result$statistic)
    }
    return(data.frame(gene_id_all = gene_ids, t_data = t_data))
  }
```

Index