Package 'DiffGraph'

September 7, 2017

Type Package

Title DiffGraph: An R Package for Identifying Gene Network Rewiring using Differential Graphical Models

Version 1.1.0

Author Xiao-Fei Zhang

Maintainer Xiao-Fei Zhang <zhangxf@mail.ccnu.edu.cn>

Description We develop DiffGraph, an R package that integrates four influential differential graphical models for identifying gene network rewiring between two different conditions from gene expression data. The input and output of different models are packaged in the same format, making it convenient for users to compare different models using a wide range of datasets and carry out follow-up analysis. Furthermore, the inferred differential networks are visualized in both non-interactive and interactive manners. The package is useful for identifying gene network rewiring from an input dataset, comparing the predictions of different well-established methods, and visualizing the results.

License GPL (>= 2)

Encoding UTF-8

NeedsCompilation no

Depends R (>= 2.10), igraph, MASS, Matrix

R topics documented:

`DiffGraph-package` *DiffGraph: An R Package for Identifying Gene Network Rewiring using Differential Graphical Models*

Description

DiffGraph is an R package for identifying gene network rewiring between two different conditions from gene expression data.

Details

DiffGraph integrates four influential differential graphical models which identify gene network rewiring through estimating the difference of condition specific precision matrices: fused graphical lasso (FGL), lasso penalized D-Trace loss (Dtrace), perturbed-node joint graphical lasso (PNJGL) and prior information-induced differential network analysis (pDNA). DiffGraph also provides three approaches to compute the condition specific covariance matrices (Pearson, Spearman and Kendall). To fit normal data using Gaussian graphical models, we suggest using Pearson correlation. To fit non-normal data using nonparanormal graphical models, Spearman's rho or Kendall's tau correlation is suggested. DiffGraph outputs the estimated differential networks as igraph graphs, which can be visualized in both non-interactive and interactive manners using the igraph R package.

Author(s)

Xiao-Fei Zhang

Maintainer: Xiao-Fei Zhang <zhangxf@mail.ccnu.edu.cn>

References

Xiao-Fei Zhang, Le Ou-Yang, Shuo Yang, Xiaohua Hu and Hong Yan (2017), DiffGraph: An R package for identifying gene network rewiring using differential graphical models.

See Also

`FGL`, `Dtrace`, `PNJGL`, `pDNA`, `TCGA.BRCA`, `TCGA.GBM`

`Dtrace` *Lasso penalized D-Trace loss*

Description

This function is implemented to identify gene network rewiring using the Dtrace model.

Usage

```
Dtrace(X, lambda, covType = "pearson", tol = 1e-05, maxiter = 500,
    rho = 0.1, rho.incr = 1.05, rho.max = 1e+05)
```

Arguments

X	A list of input matrices. They can be data matrices or covariance matrices. If every matrix in the X is a symmetric matrix, the matrices are assumed to be covariance matrices among genes. If the input are data matrices, the rows represent the observations, and the columns represents the genes.
lambda	A nonnegative number. The hyperparameter controls the sparsity level of the estimated differential network.
covType	A parameter to decide which approach we choose to compute the sample covariance matrices. If covType = "pearson", it means that we compute (when input X represents data directly) the sample covariance matrices using Pearson correlation and that Dtrace is implemented based on Gaussian graphical models. If data is normal, we suggest covType = "pearson". If covType = "kendall" or "spearman", it means that we compute (when input X represents data directly) the sample covariance matrices using kendall's tau correlation or Spearman's rho correlation and that Dtrace is implemented based on nonparanormal graphical models. If data is non-normal, we suggest covType = "kendall" or "spearman".
tol	The tolerance parameter for convergence criteria.
maxiter	The maximum number of iterations for the ADMM algorithm.
rho	The penalty parameter in the ADMM algorithm.
rho.incr	The increase step parameter for varying penalty parameter rho.
rho.max	The maximum value of rho.

Details

Dtrace estimates the precision matrix difference directly without estimating the condition specific precision matrices. It contains a D-trace loss and a lasso penalty:

$$\min_{\Delta=\Delta^T} \left\{ \frac{1}{2} \left\langle \Delta^2, \frac{S^{(1)}S^{(2)} + S^{(2)}S^{(1)}}{2} \right\rangle - \left\langle \Delta, S^{(1)} - S^{(2)} \right\rangle \right\} + \lambda \left\| \Delta \right\|_1 ,$$

where the first term is the D-trace loss function, the second term is the lasso penalty function, and λ is a nonnegative tuning parameter. Unlike FGL, Dtrace does not require assumption on the condition specific precision matrices. In addition, it has an advantage of using much smaller sample size to achieve competitive performance. The estimated difference of condition specific precision matrices $\hat{\Delta}$ is used to construct the differential network.

We solve the optimization problem using the ADMM algorithm. We accelerate the ADMM iterations by adaptively changing ρ in iterations.

Value

Delta	The estimated difference of condition specific precision matrices.
Delta.graph.full	The estimated differential network over all nodes.
Delta.graph.connected	The estimated differential network over only the connected nodes.

Author(s)

Xiao-Fei Zhang

Maintainer: Xiao-Fei Zhang <zhangxf@mail.ccnu.edu.cn>

References

Xiao-Fei Zhang, Le Ou-Yang, Shuo Yang, Xiaohua Hu and Hong Yan (2017), DiffGraph: An R package for identifying gene network rewiring using differential graphical models.

Huili Yuan, Ruibin Xi, Chong Chen and Minghua Deng (2015). Differential network analysis via the lasso penalized d-trace loss. arXiv preprint arXiv:1511.09188.

Dechao Tian, Quanquan Gu and Jian Ma (2016). Identifying gene regulatory network rewiring using latent differential graphical models. Nucleic Acids Research, 44(17), e140

See Also

FGL, PNJGL, pDNA, TCGA.BRCA,TCGA.GBM

Examples

```
# Identify differential network between breast cancer subtypes
data(TCGA.BRCA)
X = TCGA.BRCA$X[1,]
dtrace.results= Dtrace(X, 0.45, covType = "spearman")
net.dtrace = dtrace.results$Delta.graph.connected
# Visualize the estimated differential network in an interactive manner.
tkid <- tkplot(net.dtrace, vertex.size= degree(net.dtrace)*1.5, layout =layout_with_fr,
    vertex.color="red", vertex.label.cex=0.8, edge.width =1.5, edge.color="orange")
# Visualize the estimated differential network in a non-interactive manner.
# grab the coordinates from tkplot
l.dtrace <- tkplot.getcoords(tkid)
plot(net.dtrace, layout=l.dtrace, vertex.size= degree(net.dtrace)*1.5, vertex.color="red",
    vertex.label.cex=0.9, edge.width =1.5, edge.color="orange")



# Identify differential network between glioblastoma subtypes
data(TCGA.GBM)
X = TCGA.GBM$X[1,]
dtrace.results= Dtrace(X, 0.32, covType = "spearman")
net.dtrace = dtrace.results$Delta.graph.connected
# Visualize the estimated differential network in an interactive manner.
tkid <- tkplot(net.dtrace, vertex.size= degree(net.dtrace)*1.5, layout =layout_with_fr,
    vertex.color="red", vertex.label.cex=0.8, edge.width =1.5, edge.color="orange")
# Visualize the estimated differential network in a non-interactive manner.
# grab the coordinates from tkplot
l.dtrace <- tkplot.getcoords(tkid)
plot(net.dtrace, layout=l.dtrace,  vertex.size= degree(net.dtrace)*1.5,  vertex.color="red",
    vertex.label.cex=0.9, edge.width =1.5, edge.color="orange")
```

FGL *Fused graphical lasso*

Description

This function is implemented to identify gene network rewiring using the FGL model.

Usage

```
FGL(X, lambda1, lambda2, covType = "pearson", weights = "equal",
    penalize.diagonal = FALSE, tol = 1e-05, maxiter = 500, rho = 0.1,
    rho.incr = 1.05, rho.max = 1e+05)
```

Arguments

X	A list of input matrices. They can be data matrices or covariance matrices. If every matrix in the X is a symmetric matrix, the matrices are assumed to be covariance matrices among genes. If the input are data matrices, the rows represent the observations, and the columns represents the genes.
lambda1	A nonnegative number. The hyperparameter controls the sparsity level of the estimated condition specific networks.
lambda2	A nonnegative number. The hyperparameter controls the sparsity level of the estimated differential network.
covType	A parameter to decide which approach we choose to compute the sample covariance matrices. If covType = "pearson", it means that we compute (when input X represents data directly) the sample covariance matrices using Pearson correlation and that FGL is implemented based on Gaussian graphical models. If data is normal, we suggest covType = "pearson". If covType = "kendall" or "spearman", it means that we compute (when input X represents data directly) the sample covariance matrices using kendall's tau correlation or Spearman's rho correlation and that FGL is implemented based on nonparanormal graphical models. If data is non-normal, we suggest covType = "kendall" or "spearman".
weights	Determines the putative sample size of each condition's data. Allowed values: a vector with length two; "equal", giving each condition weight 1; "sample.size", giving each condition weight corresponding to its sample size.
penalize.diagonal	Determines whether the sparsity penalty is applied to the diagonal of condition specific precision matrices.
tol	The tolerance parameter for convergence criteria.
maxiter	The maximum number of iterations for the ADMM algorithm.
rho	The penalty parameter in the ADMM algorithm.
rho.incr	The increase step parameter for varying penalty parameter rho.
rho.max	The maximum value of rho.

Details

FGL is an extension of graphical lasso to infer two condition specific networks from two data sets. It is based on a penalized log likelihood approach:

$$\min_{\Theta^{(1)},\Theta^{(2)}} \sum_{c=1}^{2} n_c \left(\mathrm{tr}\left(S^{(c)}\Theta^{(c)}\right) - \log\det\left(\Theta^{(c)}\right)\right) + \lambda_1 \left(\left\|\Theta^{(1)}\right\|_1 + \left\|\Theta^{(2)}\right\|_1 \right) + \lambda_2 \sum_{i,j} \left|\Theta_{ij}^{(1)} - \Theta_{ij}^{(2)}\right|,$$

where the fist term is the negative log-likelihood (up to a constant), the second term applies a lasso penalty to the condition specific precision matrices, the third term applies a lasso penalty to the difference of precision matrices, and λ_1 and λ_2 are two non-negative tuning parameters. FGL results in sparse estimates $\hat{\Theta}^{(1)}$ and $\hat{\Theta}^{(2)}$ when the tuning parameter λ_1 is large. After obtaining $\hat{\Theta}^{(1)}$ and $\hat{\Theta}^{(2)}$, the difference of precision matrices can be obtained by $\hat{\Delta} = \hat{\Theta}^{(2)} - \hat{\Theta}^{(1)}$. The fused lasso penalty encourages the two condition specific precision matrices to have similar edge values, which can integrate information across different conditions and yield a sparse estimate of precision matrix difference.

We solve the optimization problem using the ADMM algorithm. We accelerate the ADMM iterations by adaptively changing ρ in iterations.

Value

`Delta` The estimateed difference of condition specific precision matrices.

`Delta.graph.full`
The estimated differential network over all nodes.

`Delta.graph.connected`
The estimated differential network over only the connected nodes.

`Theta` A list of the estimated condition specific precision matrices.

`Theta.graph.full`
A list of the estimated condition specific networks over all nodes.

`Theta.graph.connected`
A list of the estimated condition specific networks over only the connected nodes.

Author(s)

Xiao-Fei Zhang

Maintainer: Xiao-Fei Zhang <zhangxf@mail.ccnu.edu.cn>

References

Xiao-Fei Zhang, Le Ou-Yang, Shuo Yang, Xiaohua Hu and Hong Yan (2017), DiffGraph: An R package for identifying gene network rewiring using differential graphical models.

Patrick Danaher, Pei Wang and Daniela M. Witten (2014). The joint graphical lasso for inverse covariance estimation across multiple classes. Journal of the Royal Statistical Society, Series B (Statistical Methodology), 76(2), 373-397

See Also

`Dtrace`, `PNJGL`, `pDNA`, `TCGA.BRCA`, `TCGA.GBM`

Examples

```
##Identify differential network between breast cancer subtypes
data(TCGA.BRCA)
X = TCGA.BRCA$X[1,]
fgl.results= FGL(X, 0.05, 0.2, covType = "spearman")
net.fgl = fgl.results$Delta.graph.connected
# Visualize the estimated differential network in an interactive manner
tkid <- tkplot(net.fgl, vertex.size= degree(net.fgl)*1.5, layout =layout_with_fr,
    vertex.color="red", vertex.label.cex=0.9, edge.width =1.5, edge.color="orange")
# grab the coordinates from tkplot
l.fgl <- tkplot.getcoords(tkid)
# Visualize the estimated differential network in a non-interactive manner.
plot(net.fgl, layout=l.fgl, vertex.size= degree(net.fgl)*1.5,  vertex.color="red",
    vertex.label.cex =0.9, edge.width =1.5, edge.color="orange")


## Identify differential network between glioblastoma subtypes
data(TCGA.GBM)
X = TCGA.GBM$X[1,]
fgl.results= FGL(X, 0.1, 0.16, covType = "spearman")
net.fgl = fgl.results$Delta.graph.connected
# Visualize the estimated differential network in an interactive manner
tkid <- tkplot(net.fgl, vertex.size= degree(net.fgl)*1.5, layout =layout_with_fr,
    vertex.color="red", vertex.label.cex=0.9, edge.width =1.5, edge.color="orange")
# grab the coordinates from tkplot
l.fgl <- tkplot.getcoords(tkid)
# Visualize the estimated differential network in a non-interactive manner.
plot(net.fgl, layout=l.fgl, vertex.size= degree(net.fgl)*1.5, vertex.color="red",
     vertex.label.cex =0.9, edge.width =1.5, edge.color="orange")
```

---

pDNA *Prior information-induced differential network analysis*

Description

This function is implemented to identify gene network rewiring using the pDNA model.

Usage

```
pDNA(X, lambda, covType = "pearson", tol = 1e-05, maxiter = 500, rho = 0.1,
    rho.incr = 1.05, rho.max = 1e+10)
```

Arguments

X	A list of input matrices. They can be data matrices or covariance matrices. If every matrix in the X is a symmetric matrix, the matrices are assumed to be covariance matrices among genes. If the input are data matrices, the rows represent the observations, and the columns represents the genes.
lambda	A nonnegative number. The hyperparameter controls the sparsity level of the estimated differential networks.

covType	A parameter to decide which approach we choose to compute the sample covariance matrices. If covType = "pearson", it means that we compute (when input X represents data directly) the sample covariance matrices using Pearson correlation and that pDNA is implemented based on Gaussian graphical models. If data is normal, we suggest covType = "pearson". If covType = "kendall" or "spearman", it means that we compute (when input X represents data directly) the sample covariance matrices using kendall's tau correlation or Spearman's rho correlation and that pDNA is implemented based on nonparanormal graphical models. If data is non-normal, we suggest covType = "kendall" or "spearman".
tol	The tolerance parameter for convergence criteria.
maxiter	The maximum number of iterations for the ADMM algorithm.
rho	The penalty parameter in the ADMM algorithm.
rho.incr	The increase step parameter for varying penalty parameter rho.
rho.max	The maximum value of rho.

Details

pDNA is developed to deal with the sceneries where gene expression levels are measured using multiple data types. For each subject, we assume that the gene expression levels are collected from K different data types. Let $S^{(kc)}$ be the sample covariance matrix for the Kth data types and the gth condition for $k = 1, \ldots, K$ and $g = 1, 2$, and let $\Delta^{(k)}$ be the precision matrix difference for the kth data types. pDNA estimates $\Delta^{(1)}, \ldots, \Delta^{(K)}$ by solving the following optimization problem:

$$\begin{aligned} \min_{\{\Delta^k\},\{V^k\}} \quad & \sum_{k=1}^{K} \left\{ \frac{1}{2} \left\langle \Delta^2, \frac{S^{(k1)}S^{(k2)}+S^{(k2)}S^{(k1)}}{2} \right\rangle - \left\langle \Delta, S^{(k1)} - S^{(k2)} \right\rangle \right\} + \lambda \sum_{j=1}^{p} \sqrt{\sum_{i=1}^{p} \sqrt{\sum_{k=1}^{K} \left| V_{ij}^{(k)} \right|}} \\ \text{subject to} \quad & \Delta^{(k)} = V^{(k)} + \left(V^{(k)}\right)^T, \ k = 1, \ldots, K, \end{aligned}$$

where the first term is the D-trace lasso used to infer precision matrix difference directly, the second term is a hierarchical group bridge penalty. The hierarchical group bridge penalty is developed to (i) integrate information across different data types, and (ii) capture important genes that drive the changes of network in a similar manner to PNJGL.

Value

Delta	List of the estimateed differences of condition specific precision matrices for different data types.
Delta.graph.full	List of the estimated differential network over all nodes for different data types.
Delta.graph.connected	List of the estimated differential network over only the connected nodes for different data types.
Delta.weight	The difference of condition specific precision matrices averaged over different data types.
Delta.graph.weight.full	The weighted differential network over all nodes, where the weight of a pair of genes is the frequency of data types in which they are identified as a differential edge.

`Delta.graph.connected`
: The weighted differential network over only the connected nodes, where the weight of a pair of genes is the frequency of data types in which they are identified as a differential edge.

Author(s)

Xiao-Fei Zhang

Maintainer: Xiao-Fei Zhang <zhangxf@mail.ccnu.edu.cn>

References

Xiao-Fei Zhang, Le Ou-Yang, Shuo Yang, Xiaohua Hu and Hong Yan (2017), DiffGraph: An R package for identifying gene network rewiring using differential graphical models.

Xiao-Fei Zhang, Le Ou-Yang and Hong Yan (2017). Incorporating prior information into differential network analysis using non-paranormal graphical models. Bioinformatics, 33(16), 2436-2445

See Also

`Dtrace`, `FGL`, `PNJGL`, `TCGA.BRCA`,`TCGA.GBM`

Examples

```
#
# # Identify differential network between breast cancer subtypes
# data(TCGA.BRCA)
# pdna.results= pDNA(TCGA.BRCA$X, 0.7, covType = "spearman")
# net.pdna = pdna.results$Delta.graph.weight.connected
# # Visualize the estimated differential network in an interactive manner.
# tkid <- tkplot(net.pdna, vertex.size= degree(net.pdna)*1.5, layout =layout_with_fr,
#                vertex.color="red", vertex.label.cex =0.9, edge.width =1.5, edge.color="orange")
# # Visualize the estimated differential network in a non-interactive manner.
# # grab the coordinates from tkplot
# l.pdna <- tkplot.getcoords(tkid)
# plot(net.pdna, layout=l.pdna,  vertex.size= degree(net.pdna)*1.5,  vertex.color="red",
#      vertex.label.cex=0.9,  edge.width =1.5, edge.color="orange")
#
#
# # Identify differential network between glioblastoma subtypes
# data(TCGA.GBM)
# pdna.results= pDNA(TCGA.GBM$X, 0.5, covType = "spearman")
# net.pdna = pdna.results$Delta.graph.weight.connected
# # Visualize the estimated differential network in an interactive manner.
# tkid <- tkplot(net.pdna, vertex.size= degree(net.pdna)*1.5, layout =layout_with_fr,
#                vertex.color="red", vertex.label.cex=0.9, edge.width =1.5, edge.color="orange")
# # Visualize the estimated differential network in a non-interactive manner.
# # grab the coordinates from tkplot
# l.pdna <- tkplot.getcoords(tkid)
# plot(net.pdna, layout=l.pdna,  vertex.size= degree(net.pdna)*1.5,  vertex.color="red",
#      vertex.label.cex =0.9, edge.width =1.5, edge.color="orange")
```

PNJGL *Perturbed-node joint graphical lasso*

Description

This function is implemented to identify gene network rewiring using the PNJGL model.

Usage

```
PNJGL(X, lambda1, lambda2, covType = "pearson", weights = "equal",
    penalize.diagonal = FALSE, tol = 1e-05, maxiter = 500, rho = 0.1,
    rho.incr = 1.05, rho.max = 1e+05)
```

Arguments

X	A list of input matrices. They can be data matrices or covariance matrices. If every matrix in the X is a symmetric matrix, the matrices are assumed to be covariance matrices among genes. If the input are data matrices, the rows represent the observations, and the columns represents the genes.
lambda1	A nonnegative number. The hyperparameter controls the sparsity level of the estimated condition specific networks.
lambda2	A nonnegative number. The hyperparameter controls the similarities between the estimated condition specific networks.
covType	A parameter to decide which approach we choose to compute the sample covariance matrices. If covType = "pearson", it means that we compute (when input X represents data directly) the sample covariance matrices using Pearson correlation and that PNJGL is implemented based on Gaussian graphical models. If data is normal, we suggest covType = "pearson". If covType = "kendall" or "spearman", it means that we compute (when input X represents data directly) the sample covariance matrices using kendall's tau correlation or Spearman's rho correlation and that PNJGL is implemented based on nonparanormal graphical models. If data is non-normal, we suggest covType = "kendall" or "spearman".
weights	Determines the putative sample size of each condition's data. Allowed values: a vector with length two; "equal", giving each condition weight 1; "sample.size", giving each condition weight corresponding to its sample size.
penalize.diagonal	Determines whether the sparsity penalty is applied to the diagonal of condition specific precision matrices.
tol	The tolerance parameter for convergence criteria.
maxiter	The maximum number of iterations for the ADMM algorithm.
rho	The penalty parameter in the ADMM algorithm.
rho.incr	The increase step parameter for varying penalty parameter rho.
rho.max	The maximum value of rho.

Details

PNJGL is similar to FGL. A major difference between the two models is that FGL assumes that network rewiring arises from individual edges, while PNJGL assumes that network rewiring arises from certain nodes perturbed across conditions. To identify the hub nodes driving gene network rewiring, Mohan et al proposed the PNJGL model:

$$\begin{array}{ll} \min\limits_{\Theta^{(1)},\Theta^{(2)}} & \sum_{c=1}^{2} n_c \left(\mathrm{tr}\left(S^{(c)}\Theta^{(c)}\right) - \log\det\left(\Theta^{(c)}\right)\right) + \lambda_1 \left(\left\|\Theta^{(1)}\right\|_1 + \left\|\Theta^{(2)}\right\|_1\right) + \lambda_2 \sum_{j=1}^{p} \|V_j\|_2 \\ \text{subject to} & \Theta^{(1)} - \Theta^{(2)} = V + V^T, \end{array}$$

where a row-column overlap norm penalty is applied to the precision matrix difference. Due to the row-column overlap norm penalty, the support of $\hat{\Delta} = \hat{\Theta}^{(2)} - \hat{\Theta}^{(1)}$ will be a union of several rows and the corresponding columns, which can be interpreted as a set of nodes driving the changes of network.

We solve the optimization problem using the ADMM algorithm. We accelerate the ADMM iterations by adaptively changing ρ in iterations.

Value

`Delta` The estimateed difference of condition specific precision matrices.

`Delta.graph.full`
The estimated differential network over all nodes.

`Delta.graph.connected`
The estimated differential network over only the connected nodes.

`Theta` A list of the estimated condition specific precision matrices.

`Theta.graph.full`
A list of the estimated condition specific networks over all nodes.

`Theta.graph.connected`
A list of the estimated condition specific networks over only the connected nodes.

Author(s)

Xiao-Fei Zhang

Maintainer: Xiao-Fei Zhang <zhangxf@mail.ccnu.edu.cn>

References

Xiao-Fei Zhang, Le Ou-Yang, Shuo Yang, Xiaohua Hu and Hong Yan (2017), DiffGraph: An R package for identifying gene network rewiring using differential graphical models.

Karthik Mohan, Palma London, Daniela Witten and Su-In Lee (2014). Node-based learning of multiple Gaussian graphicalmodels. Journal of Machine Learning Research, 15(1), 445-488.

See Also

`Dtrace`, `FGL`, `pDNA`, `TCGA.BRCA`, `TCGA.GBM`

Examples

```
# Identify differential network between breast cancer subtypes
data(TCGA.BRCA)
X = TCGA.BRCA$X[1,]
pnjgl.results= PNJGL(X, 0.3, 1.2, covType = "spearman")
net.pnjgl = pnjgl.results$Delta.graph.connected
# Visualize the estimated differential network in an interactive manner.
tkid <- tkplot(net.pnjgl, vertex.size= degree(net.pnjgl)*.3, layout =layout_with_fr,
    vertex.color="red", vertex.label.cex=0.9, edge.width =1.5, edge.color="orange")
# grab the coordinates from tkplot
l.pnjgl <- tkplot.getcoords(tkid)
plot(net.pnjgl, layout=l.pnjgl,  vertex.size= degree(net.pnjgl)*.3,  vertex.color="red",
    vertex.label.cex =0.9, edge.width =1.5, edge.color="orange")



# Identify differential network between glioblastoma subtypes
data(TCGA.GBM)
X = TCGA.GBM$X[1,]
pnjgl.results= PNJGL(X, 0.2, 1.5, covType = "spearman")
net.pnjgl = pnjgl.results$Delta.graph.connected
# Visualize the estimated differential network in an interactive manner.
tkid <- tkplot(net.pnjgl, vertex.size= degree(net.pnjgl)*.3, layout =layout_with_fr,
    vertex.color="red", vertex.label.cex=0.9, edge.width =1.5, edge.color="orange")
# grab the coordinates from tkplot
l.pnjgl <- tkplot.getcoords(tkid)
plot(net.pnjgl, layout=l.pnjgl,  vertex.size= degree(net.pnjgl)*.3,  vertex.color="red",
    vertex.label.cex =0.9,edge.width =1.5, edge.color="orange")
```

TCGA.BRCA *TCGA breast cancer data*

Description

The TCGA breast cancer gene expression datasets used in our case study. The datasets are obtained from the TCGA database. They are collected from three platforms: mRNA expression (Agilent G450 microarray), mRNA expression (RNA sequencing) and copy number variants (Affymetrix genome-wide human SNP Array 6.0). They include expression profiles for 207 luminal A cancers and 86 basal-like cancers. The data only include expression measurement of genes that overlap with the breast cancer pathway collected from the Kyoto Encyclopedia of Genes and Genomes database.

Usage

```
data("TCGA.BRCA")
```

Author(s)

Xiao-Fei Zhang

References

Xiao-Fei Zhang, Le Ou-Yang, Shuo Yang, Xiaohua Hu and Hong Yan (2017), DiffGraph: An R package for identifying gene network rewiring using differential graphical models.

The Cancer Genome Atlas Research Network (2012), Comprehensive molecular portraits of human breast tumors. Nature. 490(7418), 61-70. (http://cancergenome.nih.gov/)

See Also

TCGA.GBM

Examples

```
data(TCGA.BRCA)
## maybe str(TCGA.BRCA) ...
```

TCGA.GBM *TCGA glioblastoma data*

Description

The TCGA glioblastoma gene expression data used in our case study. The data are obtained from the TCGA database. They are collected from three platforms: Agilent 244K Custom Gene Expression G450, Affymetrix HT Human Genome U133 Array Plate Set and Affymetrix Human Exon 1.0 ST Array. They include expression measurments for 81 proneural tumors and 124 mesenchymal tumors. The data only include expression measurements of genes that overlap with the RTK/PI3K, p53, Rb signaling pathways which consists of the most frequently altered genes in glioblastoma.

Usage

```
data("TCGA.GBM")
```

Author(s)

Xiao-Fei Zhang

Source

The Cancer Genome Atlas Research Network (2012), Comprehensive molecular portraits of human breast tumors. Nature. 490(7418), 61-70. (http://cancergenome.nih.gov/)

References

Xiao-Fei Zhang, Le Ou-Yang, Shuo Yang, Hong Qin, Xiaohua Hu and Hong Yan (2017), JGNI: A graphical model for joint gene network inference across multiple subpopulations and data types.

Roel GW Verhaak, Katherine A Hoadley, Elizabeth Purdom, Victoria Wang, Yuan Qi, Matthew D Wilkerson, C Ryan Miller, Li Ding, Todd Golub, Jill P Mesirov, et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh1, egfr, and nf1. Cancer cell, 17(1):98-110, 2010.

See Also

TCGA.BRCA

Examples

```
data(TCGA.GBM)
## maybe str(TCGA.GBM) ...
```

Index