Package ‘fssemR’

October 13, 2022

Title  Fused Sparse Structural Equation Models to Jointly Infer Gene Regulatory Network

Version  0.1.8

Author  Xin Zhou, Xiaodong Cai

Maintainer  Xin Zhou <xxz220@miami.edu>

Description  An optimizer of Fused-Sparse Structural Equation Models, which is the state of the art jointly fused sparse maximum likelihood function for structural equation models proposed by Xin Zhou and Xiaodong Cai (2018 <doi:10.1101/466623>).

License  GPL (>= 3)

Encoding  UTF-8

Depends  methods

Imports  Rcpp, Matrix, stats, igraph, mvtnorm, qtl, stringr, glmnet, MASS, qpdf

Suggests  plotly, knitr, rmarkdown, network, ggnetwork

LinkingTo  Rcpp, RcppEigen

RoxygenNote  7.1.2

URL  https://github.com/Ivis4ml/fssemR

NeedsCompilation  yes

Repository  CRAN

VignetteBuilder  knitr

Date/Publication  2022-02-11 13:00:02 UTC

R topics documented:

  cv.multiFSSEMiPALM .................................................. 2
  cv.multiFSSEMiPALM2 ................................................ 4
  cv.multiNFSESEMiPALM2 ............................................. 5
  cv.multiRegression ................................................. 6
  cwiseGradient4FSSEM .............................................. 6
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDR</td>
<td>7</td>
</tr>
<tr>
<td>flinvB</td>
<td>7</td>
</tr>
<tr>
<td>floneB</td>
<td>8</td>
</tr>
<tr>
<td>fssemR</td>
<td>8</td>
</tr>
<tr>
<td>implFSSEM</td>
<td>9</td>
</tr>
<tr>
<td>initLambdaiPALM</td>
<td>10</td>
</tr>
<tr>
<td>initLambdaiPALM2</td>
<td>10</td>
</tr>
<tr>
<td>initLambdaiPALM3</td>
<td>11</td>
</tr>
<tr>
<td>initRhoiPALM</td>
<td>12</td>
</tr>
<tr>
<td>initRhoiPALM2</td>
<td>12</td>
</tr>
<tr>
<td>initRhoiPALM3</td>
<td>13</td>
</tr>
<tr>
<td>inverseB</td>
<td>14</td>
</tr>
<tr>
<td>invoneB</td>
<td>14</td>
</tr>
<tr>
<td>logLikFSSEM</td>
<td>15</td>
</tr>
<tr>
<td>logLikNFSSEM</td>
<td>15</td>
</tr>
<tr>
<td>multiFSSEMiPALM</td>
<td>16</td>
</tr>
<tr>
<td>multiFSSEMiPALM2</td>
<td>18</td>
</tr>
<tr>
<td>multiNFSSEMiPALM2</td>
<td>20</td>
</tr>
<tr>
<td>multiRegression</td>
<td>22</td>
</tr>
<tr>
<td>obj.multiRegression</td>
<td>23</td>
</tr>
<tr>
<td>opt.multiFSSEMiPALM</td>
<td>24</td>
</tr>
<tr>
<td>opt.multiFSSEMiPALM2</td>
<td>25</td>
</tr>
<tr>
<td>opt.multiNFSSEMiPALM2</td>
<td>27</td>
</tr>
<tr>
<td>pninvB</td>
<td>28</td>
</tr>
<tr>
<td>pnoneB</td>
<td>28</td>
</tr>
<tr>
<td>proc.centerFSSEM</td>
<td>29</td>
</tr>
<tr>
<td>proc.centerFSSEM2</td>
<td>29</td>
</tr>
<tr>
<td>randomFSSEMData</td>
<td>30</td>
</tr>
<tr>
<td>randomFSSEMData2</td>
<td>31</td>
</tr>
<tr>
<td>randomFSSEMData4Cor</td>
<td>32</td>
</tr>
<tr>
<td>TPR</td>
<td>33</td>
</tr>
<tr>
<td>transx</td>
<td>33</td>
</tr>
</tbody>
</table>

**Index**

---

**cv.multiFSSEMiPALM**

**Description**

cv.multiFSSEMiPALM
cv.multiFSSEMiPALM

Usage

```r
cv.multiFSSEMiPALM(
  Xs,
  Ys,
  Bs,
  Fs,
  Sk,
  sigma2,
  nlambda = 20,
  nrho = 20,
  nfold = 5,
  p,
  q,
  wt = TRUE,
  plot = FALSE
)
```

Arguments

- **Xs**: eQTL matrices
- **Ys**: Gene expression matrices
- **Bs**: initialized GRN-matrices
- **Fs**: initialized eQTL effect matrices
- **Sk**: eQTL index of genes
- **sigma2**: initialized noise variance
- **nlambda**: number of hyper-parameter of lasso term in CV
- **nrho**: number of hyper-parameter of fused-lasso term in CV
- **nfold**: CVfold number. Default 5/10
- **p**: number of genes
- **q**: number of eQTLs
- **wt**: use adaptive lasso or not. Default TRUE.
- **plot**: plot contour of cvmean or not. Default FALSE.

Value

- list of cross-validation result
cv.multiFSSEMiPALM2

Description

   cv.multiFSSEMiPALM2

Usage

   cv.multiFSSEMiPALM2(
      Xs,           # eQTL matrices
      Ys,           # Gene expression matrices
      Bs,           # initialized GRN-matrices
      Fs,           # initialized eQTL effect matrices
      Sk,           # eQTL index of genes
      sigma2,       # initialized noise variance
      nlambda = 20,  # number of hyper-parameter of lasso term in CV
      nrho = 20,    # number of hyper-parameter of fused-lasso term in CV
      nfold = 5,    # CVfold number. Default 5/10
      p,            # number of genes
      q,            # number of eQTLs
      wt = TRUE,    # use adaptive lasso or not. Default TRUE.
      plot = FALSE  # plot contour of cvmean or not. Default FALSE.
   )

Arguments

   Xs           # eQTL matrices
   Ys           # Gene expression matrices
   Bs           # initialized GRN-matrices
   Fs           # initialized eQTL effect matrices
   Sk           # eQTL index of genes
   sigma2       # initialized noise variance
   nlambda      # number of hyper-parameter of lasso term in CV
   nrho         # number of hyper-parameter of fused-lasso term in CV
   nfold        # CVfold number. Default 5/10
   p            # number of genes
   q            # number of eQTLs
   wt           # use adaptive lasso or not. Default TRUE.
   plot         # plot contour of cvmean or not. Default FALSE.

Value

   list of cross-validation result
cv.multiNFSSEMiPALM2

**Description**

cv.multiNFSSEMiPALM2

**Usage**

cv.multiNFSSEMiPALM2(
    Xs,
    Ys,
    Bs,
    Fs,
    Sk,
    sigma2,
    nlambda = 20,
    nrho = 20,
    nfold = 5,
    p,
    q,
    wt = TRUE,
    plot = FALSE
)

**Arguments**

- **Xs**: eQTL matrices
- **Ys**: Gene expression matrices
- **Bs**: initialized GRN-matrices
- **Fs**: initialized eQTL effect matrices
- **Sk**: eQTL index of genes
- **sigma2**: initialized noise variance
- **nlambda**: number of hyper-parameter of lasso term in CV
- **nrho**: number of hyper-parameter of fused-lasso term in CV
- **nfold**: CVfold number. Default 5/10
- **p**: number of genes
- **q**: number of eQTLs
- **wt**: use adaptive lasso or not. Default TRUE.
- **plot**: plot contour of cvmean or not. Default FALSE.

**Value**

list of cross-validation result for NFSSEM
### cv.multiRegression

description

**Usage**

```
cv.multiRegression(Xs, Ys, Sk, ngamma = 20, nfold = 5, n, p, k)
```

**Arguments**

- **Xs**: eQTL matrices
- **Ys**: Gene expression matrices
- **Sk**: eQTL index of genes
- **ngamma**: number of hyper-parameter in CV
- **nfold**: CV fold number. Default 5/10
- **n**: number of observations
- **p**: number of genes
- **k**: number of eQTLs

**Value**

```
gamma_min optimal gamma to minimize cross-validation error
```

---

### cwiseGradient4FSSEM

description

**Usage**

```
cwiseGradient4FSSEM(n, c, Y, R, Y2norm, sigma2)
```

**Arguments**

- **n**: number of observations
- **c**: cofactor vector
- **Y**: Matrix of gene expression
- **R**: Residual matrix
- **Y2norm**: Column of YtY
- **sigma2**: noise variance
**FDR**

Value

function whose argument is column vector bi

---

**Description**

False discovery rate for network prediction

**Usage**

\[
\text{FDR}(X, B, \text{PREC} = 0)
\]

**Arguments**

- **X**: list of predicted network matrices
- **B**: list of true network matrices
- **PREC**: precision threshold for FDR test. Default 0.

---

**flinvB**

**Description**

inversed difference of two B matrices. For adaptive fused lasso penalty

**Usage**

\[
\text{flinvB}(Bs)
\]

**Arguments**

- **Bs**: list of network matrices

**Value**

inversed difference matrices
**Description**

if you do not want adaptive fused lasso penalty, floneB replace flinvB

**Usage**

floneB(Bs)

**Arguments**

Bs list of network matrices

**Value**

matrix whose entries are all 1

---

**Description**

An optimizer of Fused-Sparse Structural Equation Models, which is the state of the art jointly fused sparse maximum likelihood function for structural equation models proposed by Xin Zhou and Xiaodong Cai (2018 <doi:10.1101/466623>)

**Author(s)**

Xin Zhou <xxz220@miami.edu>

**Examples**

```r
seed = as.numeric(Sys.time())
N = 100 # sample size
Ng = 5 # gene number
Nk = 5 * 3 # eQTL number
Ns = 1 # sparse ratio
sigma2 = 0.01 # sigma2
set.seed(seed)
library(fssemR)
data = randomFSSEMdata(n = N, p = Ng, k = Nk, sparse = Ns, df = 0.3, sigma2 = sigma2, u = 5, type = "DG", nhub = 1, dag = TRUE)
gamma = cv.multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, ngamma = 20, nfold = 5, N, Ng, Nk)
fit = multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, gamma, N, Ng, Nk,)
```
trans = FALSE)
Xs = data$Data$X
Ys = data$Data$Y
Sk = data$Data$Sk

## cross-validation
## cvfitc <- cv.multiFSSEMiPALM(Xs = Xs, Ys = Ys, Bs = fit$Bs, Fs = fit$Fs, Sk = Sk,
## sigma2 = fit$sigma2, nlambda = 10, nrho = 10,
## nfold = 5, p = Ng, q = Nk, wt = TRUE)

fitm <- opt.multiFSSEMiPALM(Xs = Xs, Ys = Ys, Bs = fit$Bs, Fs = fit$Fs, Sk = Sk,
                         sigma2 = fit$sigma2, nlambda = 10, nrho = 10,
                         p = Ng, q = Nk, wt = TRUE)

fitc0 <- fitm$fit

(TPR(fitc0$Bs[[1]], data$Vars$B[[1]]) + TPR(fitc0$Bs[[2]], data$Vars$B[[2]])) / 2
(FDR(fitc0$Bs[[1]], data$Vars$B[[1]]) + FDR(fitc0$Bs[[2]], data$Vars$B[[2]])) / 2
TPR(fitc0$Bs[[1]] - fitc0$Bs[[2]], data$Vars$B[[1]] - data$Vars$B[[2]])
FDR(fitc0$Bs[[1]] - fitc0$Bs[[2]], data$Vars$B[[1]] - data$Vars$B[[2]])
Description

initLambdaiPALM

Usage

initLambdaiPALM(Xs, Ys, Bs, Fs, Sk, sigma2, Wl, Wf, p, k)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xs</td>
<td>eQTL matrices</td>
</tr>
<tr>
<td>Ys</td>
<td>Gene expression matrices</td>
</tr>
<tr>
<td>Bs</td>
<td>initialized GRN-matrices</td>
</tr>
<tr>
<td>Fs</td>
<td>initialized eQTL effect matrices</td>
</tr>
<tr>
<td>Sk</td>
<td>eQTL index of genes</td>
</tr>
<tr>
<td>sigma2</td>
<td>initialized noise variance</td>
</tr>
<tr>
<td>Wl</td>
<td>weight matrices for adaptive lasso terms</td>
</tr>
<tr>
<td>Wf</td>
<td>weight matrix for adaptive fused lasso term</td>
</tr>
<tr>
<td>p</td>
<td>number of genes</td>
</tr>
<tr>
<td>k</td>
<td>number of eQTL</td>
</tr>
</tbody>
</table>

Value

lambda_max

Description

initLambdaiPALM2

Usage

initLambdaiPALM2(Xs, Ys, Bs, Fs, Sk, sigma2, Wl, Wf, p, k)
Arguments

- **Xs**: eQTL matrices
- **Ys**: Gene expression matrices
- **Bs**: initialized GRN-matrices
- **Fs**: initialized eQTL effect matrices
- **Sk**: eQTL index of genes
- **sigma2**: initialized noise variance
- **Wl**: weight matrices for adaptive lasso terms
- **Wf**: weight matrix for adaptive fused lasso term
- **p**: number of genes
- **k**: number of eQTL

Value

- **lambda_max**

---

**Description**

*initLambdaiPALM3*

**Usage**

*initLambdaiPALM3(Xs, Ys, Bs, Fs, Sk, sigma2, Wl, Wf, p, k)*

**Arguments**

- **Xs**: eQTL matrices
- **Ys**: Gene expression matrices
- **Bs**: initialized GRN-matrices
- **Fs**: initialized eQTL effect matrices
- **Sk**: eQTL index of genes
- **sigma2**: initialized noise variance
- **Wl**: weight matrices for adaptive lasso terms
- **Wf**: weight matrix for adaptive fused lasso term
- **p**: number of genes
- **k**: number of eQTL

**Value**

- **lambda_max**
initRhoiPALM

Description
initRhoiPALM

Usage
initRhoiPALM(Xs, Ys, Bs, Fs, Sk, sigma2, Wl, Wf, lambda, n, p)

Arguments
Xs  eQTL matrices
Ys  Gene expression matrices
Bs  initialized GRN-matrices
Fs  initialized eQTL effect matrices
Sk  eQTL index of genes
sigma2  initialized noise variance
Wl  weight matrices for adaptive lasso terms
Wf  weight matrix for adaptive fused lasso term
lambda  lambda w.r.t. rho_max
n  number of observations
p  number of genes

Value
rho_max

initRhoiPALM2

Description
initRhoiPALM2

Usage
initRhoiPALM2(Xs, Ys, Bs, Fs, Sk, sigma2, Wl, Wf, lambda, n, p)
Arguments

Xs  eQTL matrices
Ys  Gene expression matrices
Bs  initialized GRN-matrices
Fs  initialized eQTL effect matrices
Sk  eQTL index of genes
sigma2  initialized noise variance
Wl  weight matrices for adaptive lasso terms
Wf  weight matrix for adaptive fused lasso term
lambda  lambda w.r.t. rho_max
n  number of observations
p  number of genes

Value

rho_max

Description

initRhoiPALM3

Usage

initRhoiPALM3(Xs, Ys, Bs, Fs, Sk, sigma2, Wl, Wf, lambda, n, p)

Arguments

Xs  eQTL matrices
Ys  Gene expression matrices
Bs  initialized GRN-matrices
Fs  initialized eQTL effect matrices
Sk  eQTL index of genes
sigma2  initialized noise variance
Wl  weight matrices for adaptive lasso terms
Wf  weight matrix for adaptive perturbation group lasso term
lambda  lambda w.r.t. rho_max
n  number of observations
p  number of genes

Value

rho_max
**Description**

inverse matrices of B network for adaptive FSSEM

**Usage**

`inverseB(Bs)`

**Arguments**

- **Bs** : list of network matrices

**Value**

list of inversed B matrices

---

**Description**

if you do not want to get inversed B matrices, `invoneB` gives you a matrix with constant 1 instead in FSSEM

**Usage**

`invoneB(Bs)`

**Arguments**

- **Bs** : list of network matrices

**Value**

list of invone B matrices
**logLikFSSEM**

<table>
<thead>
<tr>
<th>Description</th>
<th>logLikFSSEM</th>
</tr>
</thead>
</table>

**Usage**

logLikFSSEM(Bs, Wl, Wf, lambda, rho, sigma2, Dets, n, p)

**Arguments**

- **Bs**: Network matrices
- **Wl**: Weights for lasso term
- **Wf**: Weights for fused term
- **lambda**: Hyperparameter of lasso term
- **rho**: Hyperparameter of fused lasso term
- **sigma2**: noise variance
- **Dets**: determinants of I-B matrices
- **n**: number of observations
- **p**: number of genes

**Value**

objective value of FSSEM with specified hyper-parameters

---

**logLikNFSSEM**

<table>
<thead>
<tr>
<th>Description</th>
<th>logLikNFSSEM</th>
</tr>
</thead>
</table>

**Usage**

logLikNFSSEM(Bs, Wl, Wf, lambda, rho, sigma2, Dets, n, p)
multiFSSEMipALM

Arguments

Bs      Network matrices
Wl      Weights for lasso term
Wf      Weights for group perturb lasso term
lambda  Hyperparameter of lasso term
rho     Hyperparameter of group fused lasso term
sigma2  noise variance
Dets    determinants of I-B matrices
n       number of observations
p       number of genes

Value

objective value of NFSSEM with specified hyper-paramters

Usage

multiFSSEMipALM(
  Xs, Ys, Bs, Fs, Sk, sigma2, lambda, rho, Wl, Wf, p,
  maxit = 100, inert = inert_opt("linear"),
  threshold = 1e-06, verbose = TRUE,
  sparse = TRUE, trans = FALSE,
  B2norm = NULL, strict = FALSE
)
Argument

- \textbf{Xs}  
  eQTL matrices
- \textbf{Ys}  
  Gene expression matrices
- \textbf{Bs}  
  initialized GRN-matrices
- \textbf{Fs}  
  initialized eQTL effect matrices
- \textbf{Sk}  
  eQTL index of genes
- \textbf{sigma2}  
  initialized noise variance from ridge regression
- \textbf{lambda}  
  Hyperparameter of lasso term in FSSEM
- \textbf{rho}  
  Hyperparameter of fused-lasso term in FSSEM
- \textbf{Wl}  
  weight matrices for adaptive lasso terms
- \textbf{Wf}  
  weight matrix for adaptive fused lasso term
- \textbf{p}  
  number of genes
- \textbf{maxit}  
  maximum iteration number. Default 100
- \textbf{inert}  
  inertial function for iPALM. Default as k-1/k+2
- \textbf{threshold}  
  convergence threshold. Default 1e-6
- \textbf{verbose}  
  Default TRUE
- \textbf{sparse}  
  Sparse Matrix or not
- \textbf{trans}  
  Fs matrix is transposed to k x p or not. If Fs from ridge regression, trans = TRUE, else, trans = FALSE
- \textbf{B2norm}  
  B2norm matrices generated from ridge regression. Default NULL.
- \textbf{strict}  
  Converge strictly or not. Default False

Value

- \textbf{fit}  
  List of FSSEM model
- \textbf{Bs}  
  coefficient matrices of gene regulatory networks
- \textbf{Fs}  
  coefficient matrices of eQTL-gene effect
- \textbf{mu}  
  Bias vector
- \textbf{sigma2}  
  estimate of covariance in SEM

Examples

```r
seed = 1234
N = 100  # sample size
Ng = 5   # gene number
Nk = 5 * 3  # eQTL number
Ns = 1  # sparse ratio
sigma2 = 0.01  # sigma2
set.seed(seed)
library(fssemR)
data = randomFSSEMdata(n = N, p = Ng, k = Nk, sparse = Ns, df = 0.3, sigma2 = sigma2,
u = 5, type = "DG", nhub = 1, dag = TRUE)
```
## If we assume that different condition has different genetics perturbations (eQTLs)
## gamma = cv.multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, ngamma = 20, nfold = 5, N, Ng, Nk)
gamma = 0.6784248 ## optimal gamma computed by cv.multiRegression
fit = multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, gamma, N, Ng, Nk, trans = FALSE)
Xs = data$Data$X
Ys = data$Data$Y
Sk = data$Data$Sk
cvfitc <- cv.multiFSSEMiPALM(Xs = Xs, Ys = Ys, Bs = fit$Bs, Fs = fit$Fs, Sk = Sk, 
  sigma2 = fit$sigma2, nlambda = 5, nrho = 5, nfold = 5, p = Ng, q = Nk, wt = TRUE)
fitc0 <- multiFSSEMiPALM(Xs = Xs, Ys = Ys, Bs = fit$Bs, Fs = fit$Fs, Sk = Sk, 
  sigma2 = fit$sigma2, lambda = cvfitc$lambda, rho = cvfitc$rho, 
  Wl = inverseB(fit$Bs), Wf = flinvB(fit$Bs), 
  p = Ng, maxit = 100, threshold = 1e-5, sparse = TRUE, 
  verbose = TRUE, trans = TRUE, strict = TRUE)

(TPR(fitc0$Bs[[1]], data$Vars$B[[1]]) + TPR(fitc0$Bs[[2]], data$Vars$B[[2]])) / 2
(FDR(fitc0$Bs[[1]], data$Vars$B[[1]]) + FDR(fitc0$Bs[[2]], data$Vars$B[[2]])) / 2
TPR(fitc0$Bs[[1]] - fitc0$Bs[[2]], data$Vars$B[[1]] - data$Vars$B[[2]])
FDR(fitc0$Bs[[1]] - fitc0$Bs[[2]], data$Vars$B[[1]] - data$Vars$B[[2]])

---

### Description
Implementing FSSELM algorithm for network inference. If Xs is identify for different conditions, multiFSSEMiPALM will be use, otherwise, please use multiFSSEMiPALM2 for general cases

### Usage
multiFSSEMiPALM2(
  Xs,
  Ys,
  Bs,
  Fs,
  Sk,
  sigma2,
  lambda,
  rho,
  Wl,
  Wf,
  p,
maxit = 100,
inert = inert_opt("linear"),
threshold = 1e-06,
verbose = TRUE,
sparse = TRUE,
trans = FALSE,
B2norm = NULL,
strict = FALSE)
)

Arguments

Xs eQTL matrices
Ys Gene expression matrices
Bs initialized GRN-matrices
Fs initialized eQTL effect matrices
Sk eQTL index of genes
sigma2 initialized noise variance from ridge regression
lambda Hyperparameter of lasso term in FSSEM
rho Hyperparameter of fused-lasso term in FSSEM
Wl weight matrices for adaptive lasso terms
Wf weight matrix for adaptive fused lasso term
p number of genes
maxit maximum iteration number. Default 100
inert inertial function for iPALM. Default as k-1/k+2
threshold convergence threshold. Default 1e-6
verbose Default TRUE
sparse Sparse Matrix or not
trans Fs matrix is transposed to k x p or not. If Fs from ridge regression, trans = TRUE, else, trans = FALSE
B2norm B2norm matrices generated from ridge regression. Default NULL.
strict Converge strictly or not. Default False

Value

fit List of FSSEM model

Bs coefficient matrices of gene regulatory networks
Fs coefficient matrices of eQTL-gene effect
mu Bias vector
sigma2 estimate of covariance in SEM
Examples

\begin{verbatim}
seed = 1234
N = 100 # sample size
Ng = 5 # gene number
Nk = 5 * 3 # eQTL number
Ns = 1 # sparse ratio
sigma2 = 0.01 # sigma2
set.seed(seed)
library(fssemR)
data = randomFSSEMeMiPALM2(n = N, p = Ng, k = Nk, sparse = Ns, df = 0.3, sigma2 = sigma2,
u = 5, type = "DG", nhub = 1, dag = TRUE)
## If we assume that different condition has different genetics perturbations (eQTLs)
data$Data$X = list(data$Data$X, data$Data$X)
## gamma = cv.multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, ngamma = 20, nfold = 5,
##    N, Ng, Nk)
gamma = 0.6784248 ## optimal gamma computed by cv.multiRegression
fit = multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, gamma, N, Ng, Nk,
    trans = FALSE)
Xs = data$Data$X
Ys = data$Data$Y
Sk = data$Data$Sk
cvfitc <- cv.multiFSSEMiPALM2(Xs = Xs, Ys = Ys, Bs = fit$Bs, Fs = fit$Fs, Sk = Sk,
sigma2 = fit$sigma2, nlambda = 5, nrho = 5,
nfold = 5, p = Ng, q = Nk, wt = TRUE)
fitc0 <- multiFSSEMiPALM2(Xs = Xs, Ys = Ys, Bs = fit$Bs, Fs = fit$Fs, Sk = Sk,
sigma2 = fit$sigma2, lambda = cvfitc$lambda, rho = cvfitc$rho,
    Wl = inverseB(fit$Bs), Wf = flinvB(fit$Bs),
p = Ng, maxit = 100, threshold = 1e-5, sparse = TRUE,
    verbose = TRUE, trans = TRUE, strict = TRUE)

(TPR(fitc0$Bs[[1]], data$Vars$B[[1]]) + TPR(fitc0$Bs[[2]], data$Vars$B[[2]])) / 2
(FDR(fitc0$Bs[[1]], data$Vars$B[[1]]) + FDR(fitc0$Bs[[2]], data$Vars$B[[2]])) / 2
TPR(fitc0$Bs[[1]] - fitc0$Bs[[2]], data$Vars$B[[1]] - data$Vars$B[[2]])
FDR(fitc0$Bs[[1]] - fitc0$Bs[[2]], data$Vars$B[[1]] - data$Vars$B[[2]])
\end{verbatim}

Description

Implementing NFSSEM algorithm for network inference. If Xs is identify for different conditions, multiNFSSEMiPALM will be use, otherwise, please use multiNFSSEMiPALM2 for general cases

Usage

multiNFSSEMiPALM2(}
multiNFSSEMiPALM2

Xs,
Ys,
Bs,
Fs,
Sk,
sigma2,
lambda,
rho,
Wl,
Wf,
p,
maxit = 100,
inert = inert_opt("linear"),
threshold = 1e-06,
verbose = TRUE,
 sparse = TRUE,
trans = FALSE,
B2norm = NULL,
strict = FALSE
)

Arguments

Xs eQTL matrices
Ys Gene expression matrices
Bs initialized GRN-matrices
Fs initialized eQTL effect matrices
Sk eQTL index of genes
sigma2 initialized noise variance from ridge regression
lambda Hyperparameter of lasso term in NFSSEM
rho Hyperparameter of fused-lasso term in NFSSEM
Wl weight matrices for adaptive lasso terms
Wf weight matrix for columnwise l2 norm adaptive group lasso
p number of genes
maxit maximum iteration number. Default 100
inert inertial function for iPALM. Default as k-1/k+2
threshold convergence threshold. Default 1e-6
verbose Default TRUE
sparse Sparse Matrix or not
trans Fs matrix is transposed to k x p or not. If Fs from ridge regression, trans = TRUE, else, trans = FALSE
B2norm B2norm matrices generated from ridge regression. Default NULL.
strict Converge strictly or not. Default False
multiRegression

Value

fit List of NFSSEM model

Bs  coefficient matrices of gene regulatory networks
Fs  coefficient matrices of eQTL-gene effect
mu  Bias vector
sigma2  estimate of covariance in SEM

Description

Ridge regression on multiple conditions, initialization of FSSEM algorithm

Usage

multiRegression(Xs, Ys, Sk, gamma, n, p, k, trans = FALSE)

Arguments

Xs  eQTL matrices. eQTL matrix can be matrix/list of multiple conditions
Ys  Gene expression matrices
Sk  eQTL index of genes
gamma  Hyperparameter for ridge regression
n  number of observations
p  number of genes
k  number of eQTLs
trans  if rows for sample, trans = TRUE, otherwise, trans = FALSE. Default FALSE

Value

fit List of SEM model

Bs  coefficient matrices of gene regulatory networks
fs  eQTL’s coefficients w.r.t each gene
Fs  coefficient matrices of eQTL-gene effect
mu  Bias vector
sigma2  estimate of covariance in SEM
**Examples**

```r
set.seed(seed)
data = randomFSSEMdata(n = N, p = Ng, k = Nk, sparse = Ns, df = 0.3, sigma2 = sigma2, u = 5, type = "DG", nhub = 1, dag = TRUE)

## If we assume that different condition has different genetics perturbations (eQTLs)
data$Data$X = list(data$Data$X, data$Data$X)
gamma = cv.multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, ngamma = 20, nfold = 5, N, Ng, Nk)
fit = multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, gamma, N, Ng, Nk, trans = FALSE)
```

---

**Description**

obj.multiRegression

**Usage**

```r
obj.multiRegression(Xs, Ys, fit, trans = F)
```

**Arguments**

- **Xs**: eQTL matrices
- **Ys**: gene expression matrices
- **fit**: regression fit result object
- **trans**: if rows for sample, trans = TRUE, otherwise, trans = FALSE. Default FALSE

**Value**

error squared norm of \( \| (I-B)Y - FX \|_2^2 \)
Description

optimize multiFSSEMiPALM's parameters by minimize BIC, when feature size is large (> 300), BIC methods will be much faster than Cross-validation

Usage

opt.multiFSSEMiPALM(
  Xs,
  Ys,
  Bs,
  Fs,
  Sk,
  sigma2,
  nlambda = 20,
  nrho = 20,
  p,
  q,
  wt = TRUE
)

Arguments

<table>
<thead>
<tr>
<th>Xs</th>
<th>eQTL matrices</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ys</td>
<td>Gene expression matrices</td>
</tr>
<tr>
<td>Bs</td>
<td>initialized GRN-matrices</td>
</tr>
<tr>
<td>Fs</td>
<td>initialized eQTL effect matrices</td>
</tr>
<tr>
<td>Sk</td>
<td>eQTL index of genes</td>
</tr>
<tr>
<td>sigma2</td>
<td>initialized noise variance</td>
</tr>
<tr>
<td>nlambda</td>
<td>number of hyper-parameter of lasso term in CV</td>
</tr>
<tr>
<td>nrho</td>
<td>number of hyper-parameter of fused-lasso term in CV</td>
</tr>
<tr>
<td>p</td>
<td>number of genes</td>
</tr>
<tr>
<td>q</td>
<td>number of eQTLs</td>
</tr>
<tr>
<td>wt</td>
<td>use adaptive lasso or not. Default TRUE.</td>
</tr>
</tbody>
</table>

Value

list of model selection result
opt.multiFSSEMiPALM2

Examples

seed = 1234
N = 100 # sample size
Ng = 5 # gene number
Nk = 5 * 3 # eQTL number
Ns = 1 # sparse ratio
sigma2 = 0.01 # sigma2
set.seed(seed)
library(fssemR)
data = randomFSSEMdata(n = N, p = Ng, k = Nk, sparse = Ns, df = 0.3, sigma2 = sigma2,
                      u = 5, type = "DG", nhub = 1, dag = TRUE)
## If we assume that different condition has different genetics perturbations (eQTLs)
data$Data$X = list(data$Data$X, data$Data$X)
## gamma = cv.multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, ngamma = 20, nfold = 5,
## gamma = 0.6784248 ## optimal gamma computed by cv.multiRegression
fit = multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, gamma, N, Ng, Nk,
                    trans = FALSE)
Xs = data$Data$X
Ys = data$Data$Y
Sk = data$Data$Sk
fitm <- opt.multiFSSEMiPALM(Xs = Xs, Ys = Ys, Bs = fit$Bs, Fs = fit$Fs, Sk = Sk,
                          sigma2 = fit$sigma2, nlambda = 10, nrho = 10,
                          p = Ng, q = Nk, wt = TRUE)
fitc0 <- fitm$fit

Description

optimize multiFSSEMiPALM's parameters by minimize BIC, when feature size is large (> 300),
BIC methods will be much faster than Cross-validation

Usage

opt.multiFSSEMiPALM2(Xs, Ys, Bs, Fs, Sk,
sigma2,
nlambda = 20,
nrho = 20,
p,
q,
wt = TRUE
)

Arguments

Xs   eQTL matrices
Ys   Gene expression matrices
Bs   initialized GRN-matrices
Fs   initialized eQTL effect matrices
Sk   eQTL index of genes
sigma2 initialized noise variance
nlambda number of hyper-parameter of lasso term in CV
nrho number of hyper-parameter of fused-lasso term in CV
p number of genes
q number of eQTLs
wt use adaptive lasso or not. Default TRUE.

Value

list of model selection result

Examples

seed = 1234
N = 100   # sample size
Ng = 5    # gene number
Nk = 5 * 3 # eQTL number
Ns = 1    # sparse ratio
sigma2 = 0.01  # sigma2
set.seed(seed)
library(fssemR)
data = randomFSSEMdata(n = N, p = Ng, k = Nk, sparse = Ns, df = 0.3, sigma2 = sigma2,
         u = 5, type = "DG", nhub = 1, dag = TRUE)
## If we assume that different condition has different genetics perturbations (eQTLs)
data$Data$X = list(data$Data$X, data$Data$X)
## gamma = cv.multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, ngamma = 20, nfold = 5,
## N, Ng, Nk)
gamma = 0.6784248  ## optimal gamma computed by cv.multiRegression
fit = multiRegression(data$Data$X, data$Data$Y, data$Data$Sk, gamma, N, Ng, Nk,
                      trans = FALSE)
Xs = data$Data$X
Ys = data$Data$Y
Sk = data$Data$Sk
fitm <- opt.multiNFSSEMIPALM2(Xs = Xs, Ys = Ys, Bs = fit$Bs, Fs = fit$Fs, Sk = Sk, 
sigma2 = fit$sigma2, nlambda = 10, nrho = 10, 
p = Ng, q = Nk, wt = TRUE)

fitc0 <- fitm$fit

(TPR(fitc0$Bs[[1]], data$Vars$B[[1]]) + TPR(fitc0$Bs[[2]], data$Vars$B[[2]])) / 2
(FDR(fitc0$Bs[[1]], data$Vars$B[[1]]) + FDR(fitc0$Bs[[2]], data$Vars$B[[2]])) / 2
(TPR(fitc0$Bs[[1]] - fitc0$Bs[[2]], data$Vars$B[[1]] - data$Vars$B[[2]]))
(FDR(fitc0$Bs[[1]] - fitc0$Bs[[2]], data$Vars$B[[1]] - data$Vars$B[[2]]))

---

opt.multiNFSSEMIPALM2  opt.multiNFSSEMIPALM2

**Description**

optimize multiNFSSEMIPALM's parameters by minimize BIC, when feature size is large (> 300),
BIC methods will be much faster than Cross-validation

**Usage**

opt.multiNFSSEMIPALM2(
  Xs,
  Ys,
  Bs,
  Fs,
  Sk,
  sigma2,
  nlambda = 20,
  nrho = 20,
  p,
  q,
  wt = TRUE
)

**Arguments**

- **Xs**: eQTL matrices
- **Ys**: Gene expression matrices
- **Bs**: initialized GRN-matrices
- **Fs**: initialized eQTL effect matrices
- **Sk**: eQTL index of genes
- **sigma2**: initialized noise variance
- **nlambda**: number of hyper-parameter of lasso term in CV
- **nrho**: number of hyper-parameter of fused-lasso term in CV
\[ p \] number of genes
\[ q \] number of eQTLs
\[ wt \] use adaptive lasso or not. Default TRUE.

**Value**

list of model selection result

---

\[ pninvB(Bs) \]

**Description**

inversed column l2 norm for perturbed group lasso penalty

**Usage**

\[ pninvB(Bs) \]

**Arguments**

\[ Bs \] list of network matrices

**Value**

inversed l2 norm of \( B_2 - B_1 \)

---

\[ pnoneB(Bs) \]

**Description**

if you do not want adaptive group lasso penalty, \( pnoneB \) replace \( pninvB \)

**Usage**

\[ pnoneB(Bs) \]

**Arguments**

\[ Bs \] list of network matrices

**Value**

inversed l2 norm of \( B_2 - B_1 \) with all entries is 1
proc.centerFSSEM

Description
proc.centerFSSEM

Usage
proc.centerFSSEM(Xs, Ys)

Arguments
Xs  eQTL matrices
Ys  list of gene expression matrices

Value
centered Xs and Ys and mean vectors

---

proc.centerFSSEM2

Description
proc.centerFSSEM2

Usage
proc.centerFSSEM2(Xs, Ys)

Arguments
Xs  list of eQTL matrices
Ys  list of gene expression matrices

Value
centered Xs and Ys and mean vectors
randomFSSEMdata

Description

randomFSSEMdata

Usage

randomFSSEMdata(
  n,
  p,
  k,
  sparse = 0.1,
  df = 0.2,
  sigma2 = 0.01,
  u = 5,
  type = c("DG", "ER"),
  dag = TRUE,
  coef = c(0.2, 0.4),
  nhub = 2
)

Arguments

n number of observations
p number of genes
k number of eQTLs
sparse ratio of edges / gene_number
df ratio of differential edges among two network
sigma2 noise variance of error
u variance of bias in SEM model.
type type of generated network, can be selected as DG, ER, Scale-free network
dag network is directed-acyclic or not. Default TRUE
coeff Range of absolute value of coefficients in simulated network matrices. Default (0.2, 0.4), or (0.5, 1)
nhub If you select to generate ER network, nhub is the number of pre-defined hub node number. Default 2

Value

list of generated data

Data List of observed, Xs, Ys, Sk
Vars List of model, Bs, Fs, mu, n, p, k
randomFSSEMdata2

Description
randomFSSEMdata2

Usage
randomFSSEMdata2(n, p, k, sparse = 0.1, df = 0.2, sigma2 = 0.01, u = 5, type = c("DG", "ER"), dag = TRUE, coef = c(0.2, 0.4), nhub = 2)

Arguments
n  number of observations. Vector for unbalance observations
p  number of genes
k  number of eQTLs
sparse  ratio of edges / gene_number
df  ratio of differential edges among two network
sigma2  noise variance of error
u  variance of bias in SEM model.
type  type of generated network, can be selected as DG, ER, Scale-free network
dag  network is directed-acyclic or not. Default TRUE
coef  Range of absolute value of coefficients in simulated network matrices. Default (0.2, 0.4), or (0.5, 1)
nhub  If you select to generate ER network, nhub is the number of pre-defined hub node number. Default 2

Value
list of generated data

Data  List of observed, Xs, Ys, Sk
Vars  List of model, Bs, Fs, mu, n, p, k
randomFSSEMdata4Cor

**Description**

randomFSSEMdata4Cor

**Usage**

```r
code
randomFSSEMdata4Cor(
  n,
  p,
  k,
  sparse = 0.1,
  df = 0.2,
  sigma2 = 0.01,
  u = 5,
  type = c("DG", "ER"),
  dag = TRUE,
  coef = c(0.2, 0.4),
  nhub = 2,
  r = 0.5
)
```

**Arguments**

- **n**: number of observations. Vector for unbalance observations
- **p**: number of genes
- **k**: number of eQTLs
- **sparse**: ratio of edges / gene_number
- **df**: ratio of differential edges among two network
- **sigma2**: noise variance of error
- **u**: variance of bias in SEM model.
- **type**: type of generated network, can be selected as DG, ER, Scale-free network
- **dag**: network is directed-acyclic or not. Default TRUE
- **coef**: Range of absolute value of coefficients in simulated network matrices. Default (0.2, 0.4), or (0.5, 1)
- **nhub**: If you select to generate ER network, nhub is the number of pre-defined hub node number. Default 2
- **r**: correlation between different observations
TPR

Value

list of generated data

Data List of observed, Xs, Ys, Sk

Vars List of model, Bs, Fs, mu, n, p, k

Description

Power of detection for network prediction

Usage

TPR(X, B, PREC = 0)

Arguments

X list of predicted network matrices
B list of true network matrices
PREC precision threshold for FDR test. Default 0.

transx

Description

transx

Usage

transx(data)

Arguments

data Collecting data structure generated by randomFSSEMData function

Value

transformed list of eQTL matrices
Index

cv.multiFSSEMiPALM, 2
cv.multiFSSEMiPALM2, 4
cv.multiNFSSEMiPALM2, 5
cv.multiRegression, 6
cwiseGradient4FSSEM, 6
FDR, 7
flinvB, 7
floneB, 8
fssemR, 8
implFSSEM, 9
initLambdaiPALM, 10
initLambdaiPALM2, 10
initLambdaiPALM3, 11
initRhoiPALM, 12
initRhoiPALM2, 12
initRhoiPALM3, 13
inverseB, 14
invoneB, 14
logLikFSSEM, 15
logLikNFSSEM, 15
multiFSSEMiPALM, 16
multiFSSEMiPALM2, 18
multiNFSSEMiPALM2, 20
multiRegression, 22
obj.multiRegression, 23
opt.multiFSSEMiPALM, 24
opt.multiFSSEMiPALM2, 25
opt.multiNFSSEMiPALM2, 27
package-fsseR (fssemR), 8
pnninvB, 28
pnoneB, 28
proc.centerFSSEM, 29
proc.centerFSSEM2, 29
randomFSSEMdta, 30
randomFSSEMdta2, 31
randomFSSEMdta4Cor, 32
TPR, 33
transx, 33