

Package ‘iSFun’

October 13, 2022

Type Package

Title Integrative Dimension Reduction Analysis for Multi-Source Data

Version 1.1.0

Maintainer Rui Ren <xmurr@stu.xmu.edu.cn>

Description The implement of integrative analysis methods based on a two-part penalization, which realizes dimension reduction analysis and mining the heterogeneity and association of multiple studies with compatible designs. The software package provides the integrative analysis methods including integrative sparse principal component analysis (Fang et al., 2018), integrative sparse partial least squares (Liang et al., 2021) and integrative sparse canonical correlation analysis, as well as corresponding individual analysis and meta-analysis versions. References: (1) Fang, K., Fan, X., Zhang, Q., and Ma, S. (2018). Integrative sparse principal component analysis. *Journal of Multivariate Analysis*, <doi:10.1016/j.jmva.2018.02.002>. (2) Liang, W., Ma, S., Zhang, Q., and Zhu, T. (2021). Integrative sparse partial least squares. *Statistics in Medicine*, <doi:10.1002/sim.8900>.

License GPL (>= 2)

Depends R (>= 3.5.0)

Imports caret, graphics, grDevices, irlba, stats

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

NeedsCompilation no

Author Kuangnan Fang [aut],
Rui Ren [aut, cre],
Qingzhao Zhang [aut],
Shuangge Ma [aut]

Repository CRAN

Date/Publication 2022-01-03 17:00:02 UTC

R topics documented:

iscca 2

iscca.cv	4
iscca.plot	6
ispca	7
ispca.cv	9
ispca.plot	11
ispls	12
ispls.cv	14
ispls.plot	16
meta.scca	17
meta.spca	19
meta.spls	20
preview.cca	21
preview.pca	23
preview.pls	24
scca	25
simData.cca	26
simData.pca	26
simData.pls	27
spca	27
spls	28

Index 30

iscca	<i>Integrative sparse canonical correlation analysis</i>
-------	--

Description

This function provides a penalty-based integrative sparse canonical correlation analysis method to handle the multiple datasets with high dimensions generated under similar protocols, which consists of two built-in penalty items for selecting the important variables for users to choose, and two contrasted penalty functions for eliminating the difference (magnitude or sign) between estimators within each group.

Usage

```
iscca(x, y, L, mu1, mu2, mu3, mu4, eps = 1e-04, pen1 = "homogeneity",
      pen2 = "magnitude", scale.x = TRUE, scale.y = TRUE, maxstep = 50,
      submaxstep = 10, trace = FALSE, draw = FALSE)
```

Arguments

x	list of data matrices, L datasets of explanatory variables.
y	list of data matrices, L datasets of dependent variables.
L	numeric, number of datasets.
mu1	numeric, sparsity penalty parameter for vector u.
mu2	numeric, contrasted penalty parameter for vector u.

mu3	numeric, sparsity penalty parameter for vector v.
mu4	numeric, contrasted penalty parameter for vector v.
eps	numeric, the threshold at which the algorithm terminates.
pen1	character, "homogeneity" or "heterogeneity" type of the sparsity structure. If not specified, the default is homogeneity.
pen2	character, "magnitude" or "sign" based contrasted penalty. If not specified, the default is magnitude.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
submaxstep	numeric, maximum iteration steps in the sub-iterations. The default value is 10.
trace	character, "TRUE" or "FALSE". If TRUE, prints out its screening results of variables.
draw	character, "TRUE" or "FALSE". If TRUE, plot the convergence path of loadings and the heatmap of coefficient beta.

Value

An 'iscca' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- y: list of data matrices, L datasets of dependent variables with centered columns. If scale.y is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- loading.x: the estimated canonical vector of variables x.
- loading.y: the estimated canonical vector of variables y.
- variable.x: the screening results of variables x.
- variable.y: the screening results of variables y.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.
- meany: list of numeric vectors, column mean of the original datasets y.
- normy: list of numeric vectors, column standard deviation of the original datasets y.

See Also

See Also as [preview.cca](#), [iscca.cv](#), [meta.scca](#), [scca](#).

Examples

```

# Load a list with 3 data sets
library(iSFun)
data("simData.cca")
x <- simData.cca$x
y <- simData.cca$y
L <- length(x)
mu1 <- mu3 <- 0.4
mu2 <- mu4 <- 2.5

prev_cca <- preview.cca(x = x, y = y, L = L, scale.x = TRUE, scale.y = TRUE)
res_homo_m <- iscca(x = x, y = y, L = L, mu1 = mu1, mu2 = mu2, mu3 = mu3, mu4 = mu4,
  eps = 5e-2, maxstep = 50, submaxstep = 10, trace = TRUE, draw = TRUE)

res_homo_s <- iscca(x = x, y = y, L = L, mu1 = mu1, mu2 = mu2, mu3 = mu3, mu4 = mu4,
  eps = 5e-2, pen1 = "homogeneity", pen2 = "sign", scale.x = TRUE,
  scale.y = TRUE, maxstep = 50, submaxstep = 10, trace = FALSE, draw = FALSE)

mu1 <- mu3 <- 0.3
mu2 <- mu4 <- 2
res_hete_m <- iscca(x = x, y = y, L = L, mu1 = mu1, mu2 = mu2, mu3 = mu3, mu4 = mu4,
  eps = 5e-2, pen1 = "heterogeneity", pen2 = "magnitude", scale.x = TRUE,
  scale.y = TRUE, maxstep = 50, submaxstep = 10, trace = FALSE, draw = FALSE)

res_hete_s <- iscca(x = x, y = y, L = L, mu1 = mu1, mu2 = mu2, mu3 = mu3, mu4 = mu4,
  eps = 5e-2, pen1 = "heterogeneity", pen2 = "sign", scale.x = TRUE,
  scale.y = TRUE, maxstep = 50, submaxstep = 10, trace = FALSE, draw = FALSE)

```

iscca.cv

Cross-validation for iscca

Description

Performs K-fold cross validation for the integrative sparse canonical correlation analysis over a grid of values for the regularization parameter μ_1 , μ_2 , μ_3 and μ_4 .

Usage

```

iscca.cv(x, y, L, K = 5, mu1, mu2, mu3, mu4, eps = 1e-04,
  pen1 = "homogeneity", pen2 = "magnitude", scale.x = TRUE,
  scale.y = TRUE, maxstep = 50, submaxstep = 10)

```

Arguments

x list of data matrices, L datasets of explanatory variables.
y list of data matrices, L datasets of dependent variables.

L	numeric, number of datasets.
K	numeric, number of cross-validation folds. Default is 5.
mu1	numeric, the feasible set of sparsity penalty parameter for vector u.
mu2	numeric, the feasible set of contrasted penalty parameter for vector u.
mu3	numeric, the feasible set of sparsity penalty parameter for vector v.
mu4	numeric, the feasible set of contrasted penalty parameter for vector v.
eps	numeric, the threshold at which the algorithm terminates.
pen1	character, "homogeneity" or "heterogeneity" type of the sparsity structure. If not specified, the default is homogeneity.
pen2	character, "magnitude" or "sign" based contrasted penalty. If not specified, the default is magnitude.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
submaxstep	numeric, maximum iteration steps in the sub-iterations. The default value is 10.

Value

An 'iscca.cv' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- y: list of data matrices, L datasets of dependent variables with centered columns. If scale.y is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- mu1: the sparsity penalty parameter selected from the feasible set of parameter mu1 provided by users.
- mu2: the contrasted penalty parameter selected from the feasible set of parameter mu2 provided by users.
- mu3: the sparsity penalty parameter selected from the feasible set of parameter mu3 provided by users.
- mu4: the contrasted penalty parameter selected from the feasible set of parameter mu4 provided by users.
- fold: The fold assignments for cross-validation for each observation.
- loading.x: the estimated canonical vector of variables x with selected tuning parameters.
- loading.y: the estimated canonical vector of variables y with selected tuning parameters.
- variable.x: the screening results of variables x.
- variable.y: the screening results of variables y.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.
- meany: list of numeric vectors, column mean of the original datasets y.
- normy: list of numeric vectors, column standard deviation of the original datasets y.

See Also

See Also as [iscca](#).

Examples

```
# Load a list with 3 data sets
library(iSFun)
data("simData.cca")
x <- simData.cca$x
y <- simData.cca$y
L <- length(x)
mu1 <- c(0.2, 0.4)
mu3 <- 0.4
mu2 <- mu4 <- 2.5

res_homo_m <- iscca.cv(x = x, y = y, L = L, K = 5, mu1 = mu1, mu2 = mu2, mu3 = mu3,
                      mu4 = mu4, eps = 1e-2, pen1 = "homogeneity", pen2 = "magnitude",
                      scale.x = TRUE, scale.y = TRUE, maxstep = 50, submaxstep = 10)

res_homo_s <- iscca.cv(x = x, y = y, L = L, K = 5, mu1 = mu1, mu2 = mu2, mu3 = mu3,
                      mu4 = mu4, eps = 1e-2, pen1 = "homogeneity", pen2 = "sign",
                      scale.x = TRUE, scale.y = TRUE, maxstep = 50, submaxstep = 10)

mu1 <- mu3 <- c(0.1, 0.3)
mu2 <- mu4 <- 2
res_hete_m <- iscca.cv(x = x, y = y, L = L, K = 5, mu1 = mu1, mu2 = mu2, mu3 = mu3,
                      mu4 = mu4, eps = 1e-2, pen1 = "heterogeneity", pen2 = "magnitude",
                      scale.x = TRUE, scale.y = TRUE, maxstep = 50, submaxstep = 10)

res_hete_s <- iscca.cv(x = x, y = y, L = L, K = 5, mu1 = mu1, mu2 = mu2, mu3 = mu3,
                      mu4 = mu4, eps = 1e-2, pen1 = "heterogeneity", pen2 = "sign",
                      scale.x = TRUE, scale.y = TRUE, maxstep = 50, submaxstep = 10)
```

iscca.plot

Plot the results of iscca

Description

Plot the convergence path graph in the integrative sparse canonical correlation analysis method or show the the first pair of canonical vectors.

Usage

```
iscca.plot(x, type)
```

Arguments

x	list of "iscca", which is the result of command "iscca".
type	character, "path" or "loading" type, if "path", plot the the convergence path graph of vector u and v in the integrative sparse canonical correlation analysis method, if "loading", show the the first pair of canonical vectors.

Details

See details in [iscca](#).

Value

the convergence path graph or the scatter diagrams of the first pair of canonical vectors.

Examples

```
library(iSFun)
data("simData.cca")
x <- simData.cca$x
y <- simData.cca$y
L <- length(x)
mu1 <- mu3 <- 0.4
mu2 <- mu4 <- 2.5

res_homo_m <- iscca(x = x, y = y, L = L, mu1 = mu1, mu2 = mu2, mu3 = mu3,
                   mu4 = mu4, eps = 5e-2, maxstep = 100, trace = FALSE, draw = FALSE)
iscca.plot(x = res_homo_m, type = "path")
iscca.plot(x = res_homo_m, type = "loading")
```

ispca

Integrative sparse principal component analysis

Description

This function provides a penalty-based integrative sparse principal component analysis method to obtain the direction of first principal component of the multiple datasets with high dimensions generated under similar protocols, which consists of two built-in penalty items for selecting the important variables for users to choose, and two contrasted penalty functions for eliminating the difference (magnitude or sign) between estimators within each group.

Usage

```
ispca(x, L, mu1, mu2, eps = 1e-04, pen1 = "homogeneity",
      pen2 = "magnitude", scale.x = TRUE, maxstep = 50,
      submaxstep = 10, trace = FALSE, draw = FALSE)
```

Arguments

x	list of data matrices, L datasets of explanatory variables.
L	numeric, number of data sets.
mu1	numeric, sparsity penalty parameter.
mu2	numeric, contrasted penalty parameter.
eps	numeric, the threshold at which the algorithm terminates.
pen1	character, "homogeneity" or "heterogeneity" type of the sparsity structure. If not specified, the default is homogeneity.
pen2	character, "magnitude" or "sign" based contrasted penalty. If not specified, the default is magnitude.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
submaxstep	numeric, maximum iteration steps in the sub-iterations. The default value is 10.
trace	character, "TRUE" or "FALSE". If TRUE, prints out its screening results of variables.
draw	character, "TRUE" or "FALSE". If TRUE, plot the convergence path of loadings.

Value

An 'ispca' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- eigenvalue: the estimated first eigenvalue.
- eigenvector: the estimated first eigenvector.
- component: the estimated first component.
- variable: the screening results of variables.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.

References

- Fang K, Fan X, Zhang Q, et al. Integrative sparse principal component analysis[J]. Journal of Multivariate Analysis, 2018, 166: 1-16.

See Also

See Also as [preview.pca](#), [ispca.cv](#), [meta.spca](#), [spca](#).

Examples

```
# Load a list with 3 data sets
library(iSFun)
data("simData.pca")
x <- simData.pca$x
L <- length(x)

prev_pca <- preview.pca(x = x, L = L, scale.x = TRUE)
res_homo_m <- ispca(x = x, L = L, mu1 = 0.5, mu2 = 0.002, trace = TRUE, draw = TRUE)

res_homo_s <- ispca(x = x, L = L, mu1 = 0.5, mu2 = 0.002,
  pen1 = "homogeneity", pen2 = "sign", scale.x = TRUE,
  maxstep = 50, submaxstep = 10, trace = FALSE, draw = FALSE)

res_hete_m <- ispca(x = x, L = L, mu1 = 0.1, mu2 = 0.05,
  pen1 = "heterogeneity", pen2 = "magnitude", scale.x = TRUE,
  maxstep = 50, submaxstep = 10, trace = FALSE, draw = FALSE)

res_hete_s <- ispca(x = x, L = L, mu1 = 0.1, mu2 = 0.05,
  pen1 = "heterogeneity", pen2 = "sign", scale.x = TRUE,
  maxstep = 50, submaxstep = 10, trace = FALSE, draw = FALSE)
```

ispca.cv

Cross-validation for ispca

Description

Performs K-fold cross validation for the integrative sparse principal component analysis over a grid of values for the regularization parameter μ_1 and μ_2 .

Usage

```
ispca.cv(x, L, K = 5, mu1, mu2, eps = 1e-04, pen1 = "homogeneity",
  pen2 = "magnitude", scale.x = TRUE, maxstep = 50,
  submaxstep = 10)
```

Arguments

x	list of data matrices, L datasets of explanatory variables.
L	numeric, number of datasets.
K	numeric, number of cross-validation folds. Default is 5.
mu1	numeric, the feasible set of sparsity penalty parameter.
mu2	numeric, the feasible set of contrasted penalty parameter.
eps	numeric, the threshold at which the algorithm terminates.

pen1	character, "homogeneity" or "heterogeneity" type of the sparsity structure. If not specified, the default is homogeneity.
pen2	character, "magnitude" or "sign" based contrasted penalty. If not specified, the default is magnitude.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
submaxstep	numeric, maximum iteration steps in the sub-iterations. The default value is 10.

Value

An 'ispca.cv' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- y: list of data matrices, L datasets of dependent variables with centered columns. If scale.y is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- mu1: the sparsity penalty parameter selected from the feasible set of parameter mu1 provided by users.
- mu2: the contrasted penalty parameter selected from the feasible set of parameter mu2 provided by users.
- fold: The fold assignments for cross-validation for each observation.
- eigenvalue: the estimated first eigenvalue with selected tuning parameters mu1 and mu2.
- eigenvector: the estimated first eigenvector with selected tuning parameters mu1 and mu2.
- component: the estimated first component with selected tuning parameters mu1 and mu2.
- variable: the screening results of variables.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.

References

- Fang K, Fan X, Zhang Q, et al. Integrative sparse principal component analysis[J]. Journal of Multivariate Analysis, 2018, 166: 1-16.

See Also

See Also as [ispca](#).

Examples

```
# Load a list with 3 data sets
library(iSFun)
data("simData.pca")
x <- simData.pca$x
L <- length(x)
```

```
mu1 <- c(0.3, 0.5)
mu2 <- 0.002

res_homo_m <- ispca.cv(x = x, L = L, K = 5, mu1 = mu1, mu2 = mu2, pen1 = "homogeneity",
                      pen2 = "magnitude", scale.x = TRUE, maxstep = 50, submaxstep = 10)

res_homo_s <- ispca.cv(x = x, L = L, K = 5, mu1 = mu1, mu2 = mu2, pen1 = "homogeneity",
                      pen2 = "sign", scale.x = TRUE, maxstep = 50, submaxstep = 10)

mu1 <- c(0.1, 0.15)
mu2 <- 0.05
res_hete_m <- ispca.cv(x = x, L = L, K = 5, mu1 = mu1, mu2 = mu2, pen1 = "heterogeneity",
                      pen2 = "magnitude", scale.x = TRUE, maxstep = 50, submaxstep = 10)

res_hete_s <- ispca.cv(x = x, L = L, K = 5, mu1 = mu1, mu2 = mu2, pen1 = "heterogeneity",
                      pen2 = "sign", scale.x = TRUE, maxstep = 50, submaxstep = 10)
```

ispca.plot

Plot the results of ispca

Description

Plot the convergence path graph or estimated value of the first eigenvector u in the integrative sparse principal component analysis method.

Usage

```
ispca.plot(x, type)
```

Arguments

<code>x</code>	list of "ispca", which is the result of command "ispca".
<code>type</code>	character, "path" or "loading" type, if "path", plot the the convergence path graph of the first eigenvector u in the integrative sparse principal component analysis method, if "loading", plot the first eigenvector.

Details

See details in [ispca](#).

Value

the convergence path graph or the scatter diagrams of the first eigenvector u .

Examples

```

library(iSFun)
data("simData.pca")
x <- simData.pca$x
L <- length(x)

res_homo_m <- ispca(x = x, L = L, mu1 = 0.5, mu2 = 0.002, trace = FALSE, draw = FALSE)
ispca.plot(x = res_homo_m, type = "path")
ispca.plot(x = res_homo_m, type = "loading")

```

ispls

*Integrative sparse partial least squares***Description**

This function provides a penalty-based integrative sparse partial least squares method to handle the multiple datasets with high dimensions generated under similar protocols, which consists of two built-in penalty items for selecting the important variables for users to choose, and two contrasted penalty functions for eliminating the difference (magnitude or sign) between estimators within each group.

Usage

```

ispls(x, y, L, mu1, mu2, eps = 1e-04, kappa = 0.05,
      pen1 = "homogeneity", pen2 = "magnitude", scale.x = TRUE,
      scale.y = TRUE, maxstep = 50, submaxstep = 10, trace = FALSE,
      draw = FALSE)

```

Arguments

x	list of data matrices, L datasets of explanatory variables.
y	list of data matrices, L datasets of dependent variables.
L	numeric, number of datasets.
mu1	numeric, sparsity penalty parameter.
mu2	numeric, contrasted penalty parameter.
eps	numeric, the threshold at which the algorithm terminates.
kappa	numeric, $0 < \text{kappa} < 0.5$ and the parameter reduces the effect of the concave part of objective function.
pen1	character, "homogeneity" or "heterogeneity" type of the sparsity structure. If not specified, the default is homogeneity.
pen2	character, "magnitude" or "sign" based contrasted penalty. If not specified, the default is magnitude.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.

scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
submaxstep	numeric, maximum iteration steps in the sub-iterations. The default value is 10.
trace	character, "TRUE" or "FALSE". If TRUE, prints out its screening results of variables.
draw	character, "TRUE" or "FALSE". If TRUE, plot the convergence path of loadings.

Value

An 'ispls' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- y: list of data matrices, L datasets of dependent variables with centered columns. If scale.y is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- betahat: the estimated regression coefficients.
- loading: the estimated first direction vector.
- variable: the screening results of variables x.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.
- meany: list of numeric vectors, column mean of the original datasets y.
- normy: list of numeric vectors, column standard deviation of the original datasets y.

References

- Liang W, Ma S, Zhang Q, et al. Integrative sparse partial least squares[J]. Statistics in Medicine, 2021, 40(9): 2239-2256.

See Also

See Also as [preview.pls](#), [ispls.cv](#), [meta.spls](#), [spls](#).

Examples

```
# Load a list with 3 data sets
library(iSFun)
data("simData.pls")
x <- simData.pls$x
y <- simData.pls$y
L <- length(x)

prev_pls <- preview.pls(x, y, L, scale.x = TRUE, scale.y = TRUE)
res_homo_m <- ispls(x = x, y = y, L = L, mu1 = 0.05, mu2 = 0.25,
  eps = 5e-2, trace = TRUE, draw = TRUE)
```

```

res_homo_s <- ispls(x = x, y = y, L = L, mu1 = 0.05, mu2 = 0.25,
  eps = 5e-2, kappa = 0.05, pen1 = "homogeneity",
  pen2 = "sign", scale.x = TRUE, scale.y = TRUE,
  maxstep = 50, submaxstep = 10, trace = FALSE, draw = FALSE)

res_hete_m <- ispls(x = x, y = y, L = L, mu1 = 0.05, mu2 = 0.25,
  eps = 5e-2, kappa = 0.05, pen1 = "heterogeneity",
  pen2 = "magnitude", scale.x = TRUE, scale.y = TRUE,
  maxstep = 50, submaxstep = 10, trace = FALSE, draw = FALSE)

res_hete_s <- ispls(x = x, y = y, L = L, mu1 = 0.05, mu2 = 0.25,
  eps = 5e-2, kappa = 0.05, pen1 = "heterogeneity",
  pen2 = "sign", scale.x = TRUE, scale.y = TRUE,
  maxstep = 50, submaxstep = 10, trace = FALSE, draw = FALSE)

```

ispls.cv

Cross-validation for ispls

Description

Performs K-fold cross validation for the integrative sparse partial least squares over a grid of values for the regularization parameter μ_1 and μ_2 .

Usage

```

ispls.cv(x, y, L, K, mu1, mu2, eps = 1e-04, kappa = 0.05,
  pen1 = "homogeneity", pen2 = "magnitude", scale.x = TRUE,
  scale.y = TRUE, maxstep = 50, submaxstep = 10)

```

Arguments

x	list of data matrices, L datasets of explanatory variables.
y	list of data matrices, L datasets of dependent variables.
L	numeric, number of datasets.
K	numeric, number of cross-validation folds. Default is 5.
mu1	numeric, the feasible set of sparsity penalty parameter.
mu2	numeric, the feasible set of contrasted penalty parameter.
eps	numeric, the threshold at which the algorithm terminates.
kappa	numeric, $0 < \text{kappa} < 0.5$ and the parameter reduces the effect of the concave part of objective function.
pen1	character, "homogeneity" or "heterogeneity" type of the sparsity structure. If not specified, the default is homogeneity.
pen2	character, "magnitude" or "sign" based contrasted penalty. If not specified, the default is magnitude.

scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
submaxstep	numeric, maximum iteration steps in the sub-iterations. The default value is 10.

Value

An 'ispls.cv' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- y: list of data matrices, L datasets of dependent variables with centered columns. If scale.y is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- mu1: the sparsity penalty parameter selected from the feasible set of parameter mu1 provided by users.
- mu2: the contrasted penalty parameter selected from the feasible set of parameter mu2 provided by users.
- fold: The fold assignments for cross-validation for each observation.
- betahat: the estimated regression coefficients with selected tuning parameters mu1 and mu2.
- loading: the estimated first direction vector with selected tuning parameters mu1 and mu2.
- variable: the screening results of variables x.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.
- meany: list of numeric vectors, column mean of the original datasets y.
- normy: list of numeric vectors, column standard deviation of the original datasets y.

References

- Liang W, Ma S, Zhang Q, et al. Integrative sparse partial least squares[J]. Statistics in Medicine, 2021, 40(9): 2239-2256.

See Also

See Also as [ispls](#).

Examples

```
# Load a list with 3 data sets
library(iSFun)
data("simData.pls")
x <- simData.pls$x
y <- simData.pls$y
L <- length(x)
```

```

mu1 <- c(0.04, 0.05)
mu2 <- 0.25

res_homo_m <- ispls.cv(x = x, y = y, L = L, K = 5, mu1 = mu1, mu2 = mu2, eps = 1e-2,
                      kappa = 0.05, pen1 = "homogeneity", pen2 = "magnitude",
                      scale.x = TRUE, scale.y = TRUE, maxstep = 50, submaxstep = 10)

res_homo_s <- ispls.cv(x = x, y = y, L = L, K = 5, mu1 = mu1, mu2 = mu2, eps = 1e-2,
                      kappa = 0.05, pen1 = "homogeneity", pen2 = "sign",
                      scale.x = TRUE, scale.y = TRUE, maxstep = 50, submaxstep = 10)

res_hete_m <- ispls.cv(x = x, y = y, L = L, K = 5, mu1 = mu1, mu2 = mu2, eps = 1e-2,
                      kappa = 0.05, pen1 = "heterogeneity", pen2 = "magnitude",
                      scale.x = TRUE, scale.y = TRUE, maxstep = 50, submaxstep = 10)

res_hete_s <- ispls.cv(x = x, y = y, L = L, K = 5, mu1 = mu1, mu2 = mu2, eps = 1e-2,
                      kappa = 0.05, pen1 = "heterogeneity", pen2 = "sign",
                      scale.x = TRUE, scale.y = TRUE, maxstep = 50, submaxstep = 10)

```

ispls.plot

Plot the results of ispls

Description

Plot the convergence path graph of the first direction vector w in the integrative sparse partial least squares model or show the regression coefficients.

Usage

```
ispls.plot(x, type)
```

Arguments

<code>x</code>	list of "ispls", which is the result of command "ispls".
<code>type</code>	character, "path", "loading" or "heatmap" type, if "path", plot the the convergence path graph of vector w in the integrative sparse partial least squares model, if "loading", plot the the first direction vectors, if "heatmap", show the heatmap of regression coefficients among different datasets.

Details

See details in [ispls](#).

Value

show the convergence path graph of the first direction vector w or the regression coefficients.

Examples

```

library(iSFun)
data("simData.pls")
x <- simData.pls$x
y <- simData.pls$y
L <- length(x)

res_homo_m <- ispls(x = x, y = y, L = L, mu1 = 0.05, mu2 = 0.25,
                  eps = 5e-2, trace = FALSE, draw = FALSE)
ispls.plot(x = res_homo_m, type = "path")
ispls.plot(x = res_homo_m, type = "loading")
ispls.plot(x = res_homo_m, type = "heatmap")

```

meta.scca

*Meta-analytic sparse canonical correlation analysis method in integrative study***Description**

This function provides penalty-based sparse canonical correlation meta-analytic method to handle the multiple datasets with high dimensions generated under similar protocols, which is based on the principle of maximizing the summary statistics S .

Usage

```

meta.scca(x, y, L, mu1, mu2, eps = 1e-04, scale.x = TRUE,
          scale.y = TRUE, maxstep = 50, trace = FALSE)

```

Arguments

x	list of data matrices, L datasets of explanatory variables.
y	list of data matrices, L datasets of dependent variables.
L	numeric, number of datasets.
mu1	numeric, sparsity penalty parameter for vector u.
mu2	numeric, sparsity penalty parameter for vector v.
eps	numeric, the threshold at which the algorithm terminates.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
trace	character, "TRUE" or "FALSE". If TRUE, prints out its screening results of variables.

Value

A 'meta.scca' object that contains the list of the following items.

- `x`: list of data matrices, `L` datasets of explanatory variables with centered columns. If `scale.x` is `TRUE`, the columns of `L` datasets are standardized to have mean 0 and standard deviation 1.
- `y`: list of data matrices, `L` datasets of dependent variables with centered columns. If `scale.y` is `TRUE`, the columns of `L` datasets are standardized to have mean 0 and standard deviation 1.
- `loading.x`: the estimated canonical vector of variables `x`.
- `loading.y`: the estimated canonical vector of variables `y`.
- `variable.x`: the screening results of variables `x`.
- `variable.y`: the screening results of variables `y`.
- `meanx`: list of numeric vectors, column mean of the original datasets `x`.
- `normx`: list of numeric vectors, column standard deviation of the original datasets `x`.
- `meany`: list of numeric vectors, column mean of the original datasets `y`.
- `normy`: list of numeric vectors, column standard deviation of the original datasets `y`.

References

- Cichonska A, Rousu J, Martinen P, et al. metaCCA: summary statistics-based multivariate meta-analysis of genome-wide association studies using canonical correlation analysis[J]. *Bioinformatics*, 2016, 32(13): 1981-1989.

See Also

See Also as [iscca](#), [scca](#).

Examples

```
# Load a list with 3 data sets
library(iSFun)
data("simData.cca")
x <- simData.cca$x
y <- simData.cca$y
L <- length(x)
mu1 <- 0.08
mu2 <- 0.08

res <- meta.scca(x = x, y = y, L = L, mu1 = mu1, mu2 = mu2, trace = TRUE)
```

meta.spca	<i>Meta-analytic sparse principal component analysis method in integrative study</i>
-----------	--

Description

This function provides penalty-based sparse principal component meta-analytic method to handle the multiple datasets with high dimensions generated under similar protocols, which is based on the principle of maximizing the summary statistics S .

Usage

```
meta.spca(x, L, mu1, eps = 1e-04, scale.x = TRUE, maxstep = 50,
          trace = FALSE)
```

Arguments

x	list of data matrices, L datasets of explanatory variables.
L	numeric, number of datasets.
mu1	numeric, sparsity penalty parameter.
eps	numeric, the threshold at which the algorithm terminates.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
trace	character, "TRUE" or "FALSE". If TRUE, prints out its screening results of variables.

Value

A 'meta.spca' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- eigenvalue: the estimated first eigenvalue.
- eigenvector: the estimated first eigenvector.
- component: the estimated first component.
- variable: the screening results of variables.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.

References

- Kim S H, Kang D, Huo Z, et al. Meta-analytic principal component analysis in integrative omics application[J]. *Bioinformatics*, 2018, 34(8): 1321-1328.

See Also

See Also as [ispca](#), [spca](#).

Examples

```
library(iSFun)
data("simData.pca")
x <- simData.pca$x
L <- length(x)

res <- meta.spls(x = x, L = L, mu1 = 0.5, trace = TRUE)
```

 meta.spls

Meta-analytic sparse partial least squares method in integrative study

Description

This function provides penalty-based sparse canonical correlation meta-analytic method to handle the multiple datasets with high dimensions generated under similar protocols, which is based on the principle of maximizing the summary statistics.

Usage

```
meta.spls(x, y, L, mu1, eps = 1e-04, kappa = 0.05, scale.x = TRUE,
  scale.y = TRUE, maxstep = 50, trace = FALSE)
```

Arguments

x	list of data matrices, L datasets of explanatory variables.
y	list of data matrices, L datasets of dependent variables.
L	numeric, number of datasets.
mu1	numeric, sparsity penalty parameter.
eps	numeric, the threshold at which the algorithm terminates.
kappa	numeric, $0 < \text{kappa} < 0.5$ and the parameter reduces the effect of the concave part of objective function.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
trace	character, "TRUE" or "FALSE". If TRUE, prints out its screening results of variables.

Value

A 'meta.spls' object that contains the list of the following items.

- `x`: list of data matrices, L datasets of explanatory variables with centered columns. If `scale.x` is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- `y`: list of data matrices, L datasets of dependent variables with centered columns. If `scale.y` is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- `betahat`: the estimated regression coefficients.
- `loading`: the estimated first direction vector.
- `variable`: the screening results of variables `x`.
- `meanx`: list of numeric vectors, column mean of the original datasets `x`.
- `normx`: list of numeric vectors, column standard deviation of the original datasets `x`.
- `meany`: list of numeric vectors, column mean of the original datasets `y`.
- `normy`: list of numeric vectors, column standard deviation of the original datasets `y`.

See Also

See Also as [ispls](#), [spl](#).

Examples

```
library(iSFun)
data("simData.pls")
x <- simData.pls$x
y <- simData.pls$y
L <- length(x)

res <- meta.spls(x = x, y = y, L = L, mu1 = 0.03, trace = TRUE)
```

Description

The function describes the basic statistical information of the data, including sample mean, sample variance of X and Y, and the first pair of canonical vectors.

Usage

```
preview.cca(x, y, L, scale.x = TRUE, scale.y = TRUE)
```

Arguments

x	list of data matrices, L datasets of explanatory variables.
y	list of data matrices, L datasets of dependent variables.
L	numeric, number of datasets.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.

Value

An 'preview.cca' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- y: list of data matrices, L datasets of dependent variables with centered columns. If scale.y is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- loading.x: the estimated canonical vector of variables x.
- loading.y: the estimated canonical vector of variables y.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.
- meany: list of numeric vectors, column mean of the original datasets y.
- normy: list of numeric vectors, column standard deviation of the original datasets y.

See Also

See Also as [iscca](#).

Examples

```
# Load a list with 3 data sets
library(iSFun)
data("simData.cca")
x <- simData.cca$x
y <- simData.cca$y
L <- length(x)

prev_cca <- preview.cca(x = x, y = y, L = L, scale.x = TRUE, scale.y = TRUE)
```

preview.pca

Statistical description before using function ispca

Description

The function describes the basic statistical information of the data, including sample mean, sample co-variance of X and Y, the first eigenvector, eigenvalue and principal component, etc.

Usage

```
preview.pca(x, L, scale.x = TRUE)
```

Arguments

x	list of data matrices, L datasets of explanatory variables.
L	numeric, number of data sets.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.

Value

An 'preview.pca' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- eigenvalue: the estimated first eigenvalue.
- eigenvector: the estimated first eigenvector.
- component: the estimated first component.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.

See Also

See Also as [ispca](#).

Examples

```
# Load a list with 3 data sets
library(iSFun)
data("simData.pca")
x <- simData.pca$x
L <- length(x)

prev.pca <- preview.pca(x = x, L = L, scale.x = TRUE)
```

preview.pls

Statistical description before using function ispls

Description

The function describes the basic statistical information of the data, including sample mean, sample variance of X and Y, the first direction of partial least squares method, etc.

Usage

```
preview.pls(x, y, L, scale.x = TRUE, scale.y = TRUE)
```

Arguments

x	list of data matrices, L datasets of explanatory variables.
y	list of data matrices, L datasets of dependent variables.
L	numeric, number of datasets.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.

Value

A 'preview.pls' object that contains the list of the following items.

- x: list of data matrices, L datasets of explanatory variables with centered columns. If scale.x is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- y: list of data matrices, L datasets of dependent variables with centered columns. If scale.y is TRUE, the columns of L datasets are standardized to have mean 0 and standard deviation 1.
- loading: the estimated first direction vector.
- meanx: list of numeric vectors, column mean of the original datasets x.
- normx: list of numeric vectors, column standard deviation of the original datasets x.
- meany: list of numeric vectors, column mean of the original datasets y.
- normy: list of numeric vectors, column standard deviation of the original datasets y.

See Also

See Also as [ispls](#).

Examples

```

library(iSFun)
data("simData.pls")
x <- simData.pls$x
y <- simData.pls$y
L <- length(x)

prev_pls <- preview.pls(x = x, y = y, L = L, scale.x = TRUE, scale.y = TRUE)

```

scca

*Sparse canonical correlation analysis***Description**

This function provides penalty-based sparse canonical correlation analysis to get the first pair of canonical vectors.

Usage

```

scca(x, y, mu1, mu2, eps = 1e-04, scale.x = TRUE, scale.y = TRUE,
     maxstep = 50, trace = FALSE)

```

Arguments

x	data matrix of explanatory variables
y	data matrix of dependent variables.
mu1	numeric, sparsity penalty parameter for vector u.
mu2	numeric, sparsity penalty parameter for vector v.
eps	numeric, the threshold at which the algorithm terminates.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
trace	character, "TRUE" or "FALSE". If TRUE, prints out its screening results of variables.

Value

An 'scca' object that contains the list of the following items.

- x: data matrix of explanatory variables with centered columns. If scale.x is TRUE, the columns of data matrix are standardized to have mean 0 and standard deviation 1.
- y: data matrix of dependent variables with centered columns. If scale.y is TRUE, the columns of data matrix are standardized to have mean 0 and standard deviation 1.

- loading.x: the estimated canonical vector of variables x.
- loading.y: the estimated canonical vector of variables y.
- variable.x: the screening results of variables x.
- variable.y: the screening results of variables y.
- meanx: column mean of the original dataset x.
- normx: column standard deviation of the original dataset x.
- meany: column mean of the original dataset y.
- normy: column standard deviation of the original dataset y.

See Also

See Also as [iscca](#), [meta.scca](#).

Examples

```
library(iSFun)
data("simData.cca")
x.scca <- do.call(rbind, simData.cca$x)
y.scca <- do.call(rbind, simData.cca$y)
res_scca <- scca(x = x.scca, y = y.scca, mu1 = 0.1, mu2 = 0.1, eps = 1e-3,
                scale.x = TRUE, scale.y = TRUE, maxstep = 50, trace = FALSE)
```

simData.cca

Example data for method iscca

Description

Example data for users to apply the method [iscca](#), [iscca.cv](#), [meta.scca](#) or [scca](#).

Format

list

simData.pca

Example data for method ispca

Description

Example data for users to apply the method [ispca](#), [ispca.cv](#), [meta.spca](#) or [spca](#).

Format

list

 simData.pls

Example data for method ispls

Description

Example data for users to apply the method ispls, ispls.cv, meta.spls or spls.

Format

list

spca

Sparse principal component analysis

Description

This function provides penalty-based integrative sparse principal component analysis to obtain the direction of first principal component of a given dataset with high dimensions.

Usage

```
spca(x, mu1, eps = 1e-04, scale.x = TRUE, maxstep = 50,
      trace = FALSE)
```

Arguments

x	data matrix of explanatory variables.
mu1	numeric, sparsity penalty parameter.
eps	numeric, the threshold at which the algorithm terminates.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
trace	character, "TRUE" or "FALSE". If TRUE, prints out its screening results of variables.

Value

An 'spca' object that contains the list of the following items.

- x: data matrix of explanatory variables with centered columns. If scale.x is TRUE, the columns of data matrix are standardized to have mean 0 and standard deviation 1.
- eigenvalue: the estimated first eigenvalue.
- eigenvector: the estimated first eigenvector.

- component: the estimated first principal component.
- variable: the screening results of variables.
- meanx: column mean of the original dataset x.
- normx: column standard deviation of the original dataset x.

See Also

See Also as [ispca](#), [meta.spca](#).

Examples

```
library(iSFun)
data("simData.pca")
x.spca <- do.call(rbind, simData.pca$x)
res_spca <- spca(x = x.spca, mu1 = 0.08, eps = 1e-3, scale.x = TRUE,
                maxstep = 50, trace = FALSE)
```

spls

Sparse partial least squares

Description

This function provides penalty-based sparse partial least squares analysis for single dataset with high dimensions., which aims to have the direction of the first loading.

Usage

```
spls(x, y, mu1, eps = 1e-04, kappa = 0.05, scale.x = TRUE,
     scale.y = TRUE, maxstep = 50, trace = FALSE)
```

Arguments

x	matrix of explanatory variables.
y	matrix of dependent variables.
mu1	numeric, sparsity penalty parameter.
eps	numeric, the threshold at which the algorithm terminates.
kappa	numeric, $0 < \text{kappa} < 0.5$ and the parameter reduces the effect of the concave part of objective function.
scale.x	character, "TRUE" or "FALSE", whether or not to scale the variables x. The default is TRUE.
scale.y	character, "TRUE" or "FALSE", whether or not to scale the variables y. The default is TRUE.
maxstep	numeric, maximum iteration steps. The default value is 50.
trace	character, "TRUE" or "FALSE". If TRUE, prints out its screening results of variables.

Value

An 'spls' object that contains the list of the following items.

- `x`: data matrix of explanatory variables with centered columns. If `scale.x` is `TRUE`, the columns of data matrix are standardized to have mean 0 and standard deviation 1.
- `y`: data matrix of dependent variables with centered columns. If `scale.y` is `TRUE`, the columns of data matrix are standardized to have mean 0 and standard deviation 1.
- `betahat`: the estimated regression coefficients.
- `loading`: the estimated first direction vector.
- `variable`: the screening results of variables.
- `meanx`: column mean of the original dataset `x`.
- `normx`: column standard deviation of the original dataset `x`.
- `meany`: column mean of the original dataset `y`.
- `normy`: column standard deviation of the original dataset `y`.

See Also

See Also as [ispls](#), [meta.spls](#).

Examples

```
library(iSFun)
data("simData.pls")
x.spls <- do.call(rbind, simData.pls$x)
y.spls <- do.call(rbind, simData.pls$y)
res_spls <- spls(x = x.spls, y = y.spls, mu1 = 0.05, eps = 1e-3, kappa = 0.05,
                scale.x = TRUE, scale.y = TRUE, maxstep = 50, trace = FALSE)
```

Index

* datasets

- simData.cca, 26
- simData.pca, 26
- simData.pls, 27

- iscca, 2, 6, 7, 18, 22, 26
- iscca.cv, 3, 4
- iscca.plot, 6
- ispca, 7, 10, 11, 20, 23, 28
- ispca.cv, 8, 9
- ispca.plot, 11
- ispls, 12, 15, 16, 21, 24, 29
- ispls.cv, 13, 14
- ispls.plot, 16

- meta.scca, 3, 17, 26
- meta.spca, 8, 19, 28
- meta.spls, 13, 20, 29

- preview.cca, 3, 21
- preview.pca, 8, 23
- preview.pls, 13, 24

- scca, 3, 18, 25
- simData.cca, 26
- simData.pca, 26
- simData.pls, 27
- spca, 8, 20, 27
- spls, 13, 21, 28