Package 'iFad'

February 20, 2015

Type Package

Title An integrative factor analysis model for drug-pathway association inference

Version 3.0

Date 2014-03-25

Author Haisu Ma <haisu.ma.pku.2008@gmail.com>

Maintainer Haisu Ma <haisu.ma.pku.2008@gmail.com>

Depends R (>= 2.12.1),Rlab,MASS,coda,ROCR

Description This package implements a Bayesian sparse factor model for the joint analysis of paired datasets, one is the gene expression dataset and the other is the drug sensitivity profiles, measured across the same panel of samples, e.g., cell lines. Prior knowledge about gene-pathway associations can be easily incorporated in the model to aid the inference of drug-pathway associations.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2014-03-27 23:58:34

R topics documented:

`iFad-package` *An integrative factor analysis model for drug-pathway association inference*

Description

This package implements a bayesian sparse factor model for the joint analysis of paired datasets, one is the gene expression dataset and the other is the drug sensitivity profiles, measured across the same panel of samples, e.g., cell lines. Prior knowledge about gene-pathway associations can be easily incorporated in the model to aid the inference of drug-pathway associations.

Details

Package:	iFad
Type:	Package
Version:	3.0
Date:	2014-03-25
License:	GPL (version 2 or later)
LazyLoad:	yes

install.packages("iFad")

Author(s)

Haisu Ma Maintainer: Haisu Ma <haisu.ma.pku.2008@gmail.com>

Examples

```
library(Rlab)
library(MASS)
library(coda)
library(ROCR)

#Simulate datasets

data_simulation(K=10,G1=30,G2=30,J=15,eta0=c(0.2,0.2),
eta1=c(0.2,0.2),density=c(0.1,0.1),alpha_tau=1,
beta_tau=0.01,SNR=0,file_name="demo_data.RData")


#Gibbs sampling

data(matrixY1)
data(matrixY2)
data(matrixL1)
data(matrixL2)


gibbs_sampling(matrixY1, matrixY2, matrixL1, matrixL2,
eta0=c(0.2,0.2), eta1=c(0.2,0.2), alpha_tau = 1,
beta_tau = 0.01, tau_sig = 1, max_iter = 5,
thin = 1, file_name="Demo_Gibbs_result.RData")


#Traceplot
data(tau_g_chain)
mcmc_trace_plot(tau_g_chain,plot_file_name="Demo_traceplot.pdf",
index=1:10)

#ROC plot
data(matrixZ1)
data(matrixZ2)
data(matrixZ_chain)
ROC_plot(matrixZ1, matrixZ2, matrixZ_chain, plot_name="ROC_plot.pdf",
result_file_name="ROC_result.RData", burn=1)

#RMSE plot
data(Y1_mean)
data(Y2_mean)
data(matrixY1)
data(matrixY2)
data(matrixZ_chain)
data(matrixW1)
data(matrixW2)
data(matrixW_chain)
data(matrixX)
data(matrixX_chain)
Ymean_compare(Y1_mean,Y2_mean,matrixY1, matrixY2, matrixZ_chain,
```

```
matrixW1, matrixW2, matrixW_chain, matrixX, matrixX_chain,
result_file_name="RMSE_demo.RData", plot_name="RMSE_plot.pdf")
```

`data_simulation` *Simulation of example dataset for the factor analysis model*

Description

Simulation of gene expression data, drug sensitivity data, as well as gene-pathway association matrix

Usage

```
data_simulation(K, G1, G2, J, eta0, eta1, density,
alpha_tau = 1, beta_tau = 0.01, SNR = 0, file_name)
```

Arguments

`K`	Number of latent factors (pathways)
`G1`	Number of genes in matrixY1
`G2`	Number of drugs in matrixY2
`J`	Number of samples (for example, cell lines)
`eta0`	The probability of having true value of 1 for the entries in matrixZ with value 0 in matrixL
`eta1`	The probability of having true value of 0 for the entries in matrixZ with value 1 in matrixL
`density`	Density of prior-information matrixL
`alpha_tau`	The alpha parameter of Gamma distribution used for the simulation of noise, default value=1
`beta_tau`	The beta parameter of Gamma distribution used for the simulation of noise, default value=0.01
`SNR`	The signal-to-noise ratio, which the ratio of the variance of individual genes to the variance of the noise term, default value=0
`file_name`	The name of the simulated data file

Details

When SNR is set to some non-zero value, alpha_tau and beta_tau will not be used for the simulation of noise term

Value

A ".RData" file with the following components:

`matrixL1,matrixL2`
: The binary matrix of prior information about gene/drug-pathway associations. Dim(matrixL1)=G1*K, Dim(matrixL2)=G2*K. For example, matrixL1[g,k]=1 indicates that the g-th gene is known to be associated with the k-th pathway. This information can come from some well-known database, such as KEGG pathway database.

`matrixPi1,matrixPi2`
: The matrix with the bernoulli probability for binary matrixZ1 and matrixZ2.Therefore, matrixPi[g,k]=P(matrixZ[g,k]==1). Dim(matrixPi1)=G1*K,Dim(matrixPi2)=G2*K. These two matrix are determined by matrixL1,L2 and eta0, eta1.

`matrixX`
: The factor activity matrix with dimension K*J. matrixX[k,j] is the activity value of the k-th latent factor (e.g., pathway) in the j-th sample (e.g., cell line). Real continuous value with mean 0 and SD 1.

`matrixW1,matrixW2`
: The factor loading matrix representing the degree of influence of the latent factors on individual genes/drugs. Dim(matrixW1)=G1*K; Dim(matrixW2)=G2*K. Real continuous value with mean 0 and SD 1.

`matrixY1,matrixY2`
: The paired gene expression and drug response matrix measured across the same set of samples (cell lines). Dim(matrixY1)=G1*J. Dim(matrixY2)=G2*J.

`matrixZ1,matrixZ2`
: The binary matrix indicating whether each entry of loading matrix W1 and W2 is non-zero. For example, matrixZ1[g,k]=1 indicates that matrixW1[g,k] is non-zero, and vice versa. Dim(matrixZ1)=G1*K. Dim(matrixZ2)=G2*K.

`sigma1,sigma2`
: The positive-definite symmetric matrix specifying the covariance matrix of the noise term associated with each gene or drug. Dim(sigma1)=G1*G1. Dim(sigma2)=G2*G2.

`Y1_mean,Y2_mean`
: The matrix of the mean value of matrixY before adding the noise term.Calculated by the multiplication of matrixW and matrixX. Dim(Y1_mean)=G1*J. Dim(Y2_mean)=G2*J.

Author(s)

Haisu Ma <haisu.ma@yale.edu>

Examples

```
data_simulation(K=10,G1=30,G2=30,J=15,eta0=c(0.2,0.2),
eta1=c(0.2,0.2), density=c(0.1,0.1),alpha_tau=1,
beta_tau=0.01,SNR=0,file_name="demo_data.RData")
```

gibbs_sampling *Gibbs sampling for the inference of the inference of parameters in the sparse factor analysis model*

Description

This function implements the collapsed Gibbs sampling algorithm for the inference of unknown parameters in the proposed sparse factor analysis model

Usage

```
gibbs_sampling(matrixY1, matrixY2, matrixL1, matrixL2, eta0,
eta1, alpha_tau = 1, beta_tau = 0.01, tau_sig = 0, max_iter = 1e+05,
thin = 10, alpha_sigma1 = 0.7, alpha_sigma2 = 0.7,
beta_sigma1 = 0.3, beta_sigma2 = 0.3, file_name)
```

Arguments

matrixY1	The gene expression matrix, dim(Y1)=G1*J
matrixY2	The drug sensitivity matrix, dim(Y2)=G2*J
matrixL1	The linkage matrix representing prior knowledge about the sparsity pattern of matrixZ1
matrixL2	The linkage matrix representing prior knowledge about the sparsity pattern of matrixZ2
eta0	The probability of having true value of 1 for the entries in matrixZ with value 0 in matrixL
eta1	The probability of having true value of 0 for the entries in matrixZ with value 1 in matrixL
alpha_tau	The alpha parameter of Gamma distribution used for the simulation of noise, default value=1
beta_tau	The beta parameter of Gamma distribution used for the simulation of noise, default value=0.01
tau_sig	Pre-defined precision of each entry in the factor loadings matrixW1 and matrixW2, default value=0
max_iter	The number of iterations of the collaped Gibbs sampling algorithm, default=100000
thin	The number of iteration cycle for the record of Gibbs samples. For the convenience of storage, the result of the Gibbs sampling will be kept every other "thin" iterations to alliviate the auto-correlation problem between adjacent interations of the Gibbs sampling process
alpha_sigma1,alpha_sigma2	If tau_sig=0, the precision of each entry in the factor loading matrixW1 and matrixW2 is not pre-defined, but also treated in a bayesian way. The implemented algorithm will then put a Gamma prior on the precision of matrixW. alpha_sigma1 and alpha_sigma2 are the alpha parameter for the Gamma prior for matrixW1 and matrixW2 respectively

`beta_sigma1,beta_sigma2`
: The alpha parameter for the Gamma prior for matrixW1 and matrixW2 respectively

`file_name`
: The name of the file to store the result of each thinned iteration of the Gibbs sampling

Value

A ".RData" file with the following components:

`matrixZ_chain`
: The updated matrixZ in each recorded iteration.A list of length 2, for matrixZ1 and matrixZ2 respectively. matrixZ_chain[[1]] and matrixZ2[[2]] are both matrices with dimension A*B, whereas A is the number of recorded iterations and B the length of matrixZ1/Z2. Each row of the matrix corresponds to the vectorized matrixZ1 or matrixZ2 in each iteration.

`matrixW_chain`
: The updated matrixW in each recorded iteration, with format similar to matrixZ_chain.

`matrixX_chain`
: The updated matrixX in each recorded iteration. An A*B matrix with each row corresponding to the vectorized matrixX in each recorded iteration.

`tau_g_chain`
: The updated precision of each gene in each itertaion. A list with length 2, containing 2 matrices. Each row corresponds to the updated precision of the G1 or G2 genes in each recorded iteration.

`matrixPr_chain`
: Posterior probability for each entry in matrixZ to take value of 1 in each recorded iteration. Also a list of length 2. Each element is a matrix of the same dimension with matrixZ_chain[[1]] or [[2]].

`label_chain`
: A matrix to record the factor labels updated in each iteration. The label index is relative to the starting configuration in the Gibbs sampling process, not the true factor labels.

Author(s)

Haisu Ma <haisu.ma@yale.edu>

Examples

```
library(Rlab)
library(MASS)
library(coda)
library(ROCR)

data(matrixY1)
data(matrixY2)
data(matrixL1)
data(matrixL2)

gibbs_sampling(matrixY1, matrixY2, matrixL1, matrixL2,
eta0=c(0.2,0.2), eta1=c(0.2,0.2), alpha_tau = 1,
beta_tau = 0.01, tau_sig = 1, max_iter = 5, thin = 1,
```

```
file_name="Demo_Gibbs_result.RData")
```

`label_chain` *Updated factor label configuration during the Gibbs sampling*

Description

A matrix to record the factor labels updated in each iteration. The label index is relative to the starting configuration in the Gibbs sampling process, not the true factor labels.

Usage

```
data(label_chain)
```

Format

The format is: num [1:5, 1:10] 4 4 4 4 4 7 7 7 7 7 ...

Examples

```
data(label_chain)
```

`matrixL1` *The matrix representing prior belief for matrixZ1*

Description

The binary matrix of prior information about gene-pathway associations. Dim(matrixL1)=G1*K. For example, matrixL1[g,k]=1 indicates that the g-th gene is known to be associated with the k-th pathway. This information can come from some well-known database, such as KEGG pathway database.

Usage

```
data(matrixL1)
```

Format

The format is: num [1:30, 1:10] 0 0 0 0 0 0 1 0 0 0 0 ...

Examples

```
data(matrixL1)
```

matrixL2 *The matrix representing prior belief for matrixZ2*

Description

The binary matrix of prior information about drug-pathway associations. Dim(matrixL2)=G2*K. For example, matrixL2[g,k]=1 indicates that the g-th drug is known to be associated with the k-th pathway. This information can come from some well-known database, such as KEGG pathway database.

Usage

```
data(matrixL2)
```

Format

The format is: num [1:30, 1:10] 0 0 1 0 1 0 0 0 0 0 ...

Examples

```
data(matrixL2)
```

matrixPi1 *The bernoulli probability matrix for matrixZ1*

Description

The matrix with the bernoulli probability for binary matrixZ1. Therefore, matrixPi1[g,k]=P(matrixZ1[g,k]==1). Dim(matrixPi1)=G1*K. This matrix is determined by matrixL1 and eta0, eta1.

Usage

```
data(matrixPi1)
```

Format

The format is: num [1:30, 1:10] 0.2 0.2 0.2 0.2 0.2 0.8 0.2 0.2 0.2 0.2 ...

Examples

```
data(matrixPi1)
```

matrixPi2 — *The bernoulli probability matrix for matrixZ2*

Description

The matrix with the bernoulli probability for binary matrixZ2. Therefore, matrixPi2[g,k]=P(matrixZ2[g,k]==1). Dim(matrixPi2)=G2*K. This matrix is determined by matrixL2 and eta0, eta1.

Usage

```
data(matrixPi2)
```

Format

The format is: num [1:30, 1:10] 0.2 0.2 0.8 0.2 0.8 0.2 0.2 0.2 0.2 0.2 ...

Examples

```
data(matrixPi2)
```

matrixPr_chain — *The updated posterior probability for matrixZ1&Z2 during Gibbs sampling*

Description

Posterior probability for each entry in matrixZ to take value of 1 in each recorded iteration. Also a list of length 2. Each element is a matrix of the same dimension with matrixZ_chain[[1]] or [[2]].

Usage

```
data(matrixPr_chain)
```

Format

The format is: List of 2 $: num [1:5, 1:300] 0.09996 0.00958 0.04389 0.3874 0.11508 ... $: num [1:5, 1:300] 1 1 1 1 1 ...

Examples

```
data(matrixPr_chain)
```

matrixW1 *The factor loading matrix representing the gene-pathway association*

Description

The factor loading matrix representing the degree of influence of the latent factors on individual genes. Dim(matrixW1)=G1*K; Real continuous value with mean 0 and SD 1.

Usage

```
data(matrixW1)
```

Format

The format is: num [1:30, 1:10] 0 0 1.76 0 0 ...

Examples

```
data(matrixW1)
```

matrixW2 *The factor loading matrix representing the drug-pathway association*

Description

The factor loading matrix representing the degree of influence of the latent factors on individual drugs. Dim(matrixW2)=G2*K; Real continuous value with mean 0 and SD 1.

Usage

```
data(matrixW2)
```

Format

The format is: num [1:30, 1:10] 0 0 0 2.03 -0.744 ...

Examples

```
data(matrixW2)
```

matrixW_chain *The updated matrixW during the Gibbs sampling*

Description

The updated matrixW in each recorded iteration, with format similar to matrixZ_chain.

Usage

```
data(matrixW_chain)
```

Format

The format is: List of 2 $: num [1:5, 1:300] 0 0 0 0.348 0 ... $: num [1:5, 1:300] -0.651 -0.868 -1.013 -0.868 -0.938 ...

Examples

```
data(matrixW_chain)
```

matrixX *The factor activity matrix*

Description

The factor activity matrix with dimension K*J. matrixX[k,j] is the activity value of the k-th latent factor (e.g., pathway) in the j-th sample (e.g., cell line). Real continuous value with mean 0 and SD 1.

Usage

```
data(matrixX)
```

Format

The format is: num [1:10, 1:15] -0.647 0.405 -0.824 1.463 0.998 ...

Examples

```
data(matrixX)
```

matrixX_chain *The updated matrixX in the Gibbs sampling process*

Description

The updated matrixX in each recorded iteration. An A*B matrix with each row corresponding to the vectorized matrixX in each recorded iteration.

Usage

```
data(matrixX_chain)
```

Format

The format is: num [1:5, 1:150] 0.1878 0.2226 0.1325 0.0274 -0.1524 ...

Examples

```
data(matrixX_chain)
```

matrixY1 *The gene expression dataset*

Description

The gene expression matrix, dim(Y1)=G1*J. G1 is the number of genes, J is the number of samples.

Usage

```
data(matrixY1)
```

Format

The format is: num [1:30, 1:15] 0.828 4.897 -0.609 2.856 0.497 ...

Examples

```
data(matrixY1)
```

matrixY2 *The drug sensitivity matrix*

Description

The drug sensitivity matrix, dim(Y2)=G2*J. G2 is the number of drugs, J is the number of samples.

Usage

```
data(matrixY2)
```

Format

The format is: num [1:30, 1:15] -0.157 -1.994 -0.339 -1.224 0.622 ...

Examples

```
data(matrixY2)
```

matrixZ1 *The binary indicator matrix for matrixW1*

Description

The binary matrix indicating whether each entry of loading matrix W1 is non-zero. For example, matrixZ1[g,k]=1 indicates that matrixW1[g,k] is non-zero, and vice versa. Dim(matrixZ1)=G1*K.

Usage

```
data(matrixZ1)
```

Format

The format is: num [1:30, 1:10] 0 0 1 0 0 1 0 0 0 0 ...

Examples

```
data(matrixZ1)
```

matrixZ2 *Binary indictor matrix for matrixW2*

Description

The binary matrix indicating whether each entry of loading matrixW2 is non-zero. For example, matrixZ2[g,k]=1 indicates that matrixW2[g,k] is non-zero, and vice versa. Dim(matrixZ2)=G2*K.

Usage

```
data(matrixZ2)
```

Format

The format is: num [1:30, 1:10] 0 0 0 1 1 0 0 0 0 0 ...

Examples

```
data(matrixZ2)
```

matrixZ_chain *The updated matrixZ in the Gibbs sampling process*

Description

The updated matrixZ in each recorded iteration.A list of length 2, for matrixZ1 and matrixZ2 respectively. matrixZ_chain[[1]] and matrixZ2[[2]] are both matrices with dimension A*B, whereas A is the number of recorded iterations and B the length of matrixZ1/Z2. Each row of the matrix corresponds to the vectorized matrixZ1 or matrixZ2 in each iteration.

Usage

```
data(matrixZ_chain)
```

Format

The format is: List of 2 $: num [1:5, 1:300] 0 0 0 1 0 1 1 1 1 0 ... $: num [1:5, 1:300] 1 1 1 1 1 1 1 1 1 1 ...

Examples

```
data(matrixZ_chain)
```

mcmc_trace_plot *Traceplot of the Gibbs sampling iterations*

Description

Plot the updated standard deviation of gene in every recorded Gibbs sampling iteration

Usage

```
mcmc_trace_plot(tau_g_chain, plot_file_name, index)
```

Arguments

tau_g_chain	The updated precision of each gene in each itertaion. A list with length 2, containing 2 matrices. Each row corresponds to the updated precision of the G1 or G2 genes in each recorded iteration.
plot_file_name	The name of the ".pdf" plot file
index	The index of genes/drugs whose standard deviation in each iteration of the Gibbs sampling is to be drawn in the trace plot. If there are G1 genes and G2 drugs in the dataset, "index" can be a vector with integer values ranging from 1 to (G1+G2).

Examples

```
library(Rlab)
library(MASS)
library(coda)
library(ROCR)

data(tau_g_chain)
mcmc_trace_plot(tau_g_chain,plot_file_name="Demo_traceplot.pdf", index=1:10)
```

ROC_plot *Calculate the AUC (area under curve) and generate ROC plot*

Description

This function evaluates the inference performance of the sparse factor analysis model by using different cut-off values to calculate the sensitivity and specificity of the result of the Gibbs sampling algorithm. In this way, it can give the AUC (area under curve) value and plot the receiver-operating-curve.

Usage

```
ROC_plot(matrixZ1, matrixZ2, matrixZ_chain,
plot_name, result_file_name, burn)
```

Arguments

matrixZ1,matrixZ2	The binary matrix indicating whether each entry of loading matrix W1 and W2 is non-zero. For example, matrixZ1[g,k]=1 indicates that matrixW1[g,k] is non-zero, and vice versa. Dim(matrixZ1)=G1*K. Dim(matrixZ2)=G2*K.
matrixZ_chain	The updated matrixZ in each recorded iteration. A list of length 2, for matrixZ1 and matrixZ2 respectively. matrixZ_chain[[1]] and matrixZ2[[2]] are both matrices with dimension A*B, whereas A is the number of recorded iterations and B the length of matrixZ1/Z2. Each row of the matrix corresponds to the vectorized matrixZ1 or matrixZ2 in each iteration.
plot_name	The name of the ".pdf" file of the plot
result_file_name	The name of the ".RData" file which stores the result of the performance evaluation.
burn	The number of iteration taken as the starting point for the inference of the unknown values in the model. For example,if burn=101, samples from the 101-th iteration to the final iteration of the Gibbs sampling process will be averaged for the inference.

Examples

```
library(Rlab)
library(MASS)
library(coda)
library(ROCR)


data(matrixZ1)
data(matrixZ2)
data(matrixZ_chain)
ROC_plot(matrixZ1, matrixZ2, matrixZ_chain, plot_name="ROC_plot.pdf",
result_file_name="ROC_result.RData", burn=1)
```

---

sigma1 *Covariance matrix of the noise term for the genes*

Description

The positive-definite symmetric matrix specifying the covariance matrix of the noise term associated with each gene. Dim(sigma1)=G1*G1

Usage

```
data(sigma1)
```

Format

The format is: num [1:30, 1:30] 0.108 0 0 0 0 ...

Examples

```
data(sigma1)
```

---

sigma2 *Covariance matrix of the noise term for the drugs*

Description

The positive-definite symmetric matrix specifying the covariance matrix of the noise term associated with each drug. Dim(sigma2)=G2*G2

Usage

```
data(sigma2)
```

Format

The format is: num [1:30, 1:30] 0.00881 0 0 0 0 ...

Examples

```
data(sigma2)
```

tau_g_chain *The updated tau_g in the Gibbs sampling process*

Description

The updated precision of each gene in each itertaion. A list with length 2, containing 2 matrices. Each row corresponds to the updated precision of the G1 or G2 genes in each recorded iteration.

Usage

```
data(tau_g_chain)
```

Format

The format is: List of 2 $: num [1:5, 1:30] 43.7 211.3 12.6 39.6 13.2 ... $: num [1:5, 1:30] 49.5 35.7 53.6 33.1 39.1 ...

Examples

```
data(tau_g_chain)
```

Y1_mean *The mean value used for the simulation of matrixY1*

Description

The matrix of the mean value of matrixY1 before adding the noise term.Calculated by the multiplication of matrixW and matrixX. Dim(Y1_mean)=G1*J

Usage

```
data(Y1_mean)
```

Format

The format is: num [1:30, 1:15] 0.814 4.768 -0.475 2.84 0.579 ...

Examples

```
data(Y1_mean)
```

Y2_mean — *The mean value used for the simulation of matrixY2*

Description

The matrix of the mean value of matrixY2 before adding the noise term.Calculated by the multiplication of matrixW and matrixX. Dim(Y2_mean)=G2*J

Usage

```
data(Y2_mean)
```

Format

The format is: num [1:30, 1:15] -0.138 -2.06 -0.242 -1.313 0.61 ...

Examples

```
data(Y2_mean)
```

Ymean_compare — *Compare the infered Y_mean values with the true values*

Description

Calculate the RMSE (root-mean-sqaured-error) of the inferred mean of matrixY1 and matrixY2 compared with the original true means of matrixY1 and Y2

Usage

```
Ymean_compare(Y1_mean,Y2_mean,matrixY1, matrixY2,
matrixZ_chain, matrixW1, matrixW2, matrixW_chain,
matrixX, matrixX_chain, result_file_name, plot_name)
```

Arguments

`Y1_mean,Y2_mean`
: The matrix of the mean value of matrixY before adding the noise term. Calculated by the multiplication of matrixW and matrixX. Dim(Y1_mean)=G1*J,Dim(Y2_mean)=G2*J.

`matrixY1,matrixY2`
: The paired gene expression and drug response matrix measured across the same set of samples (cell lines). Dim(matrixY1)=G1*J. Dim(matrixY2)=G2*J.

`matrixZ_chain`
: The updated matrixZ in each recorded iteration. A list of length 2, for matrixZ1 and matrixZ2 respectively. matrixZ_chain[[1]] and matrixZ2[[2]] are both matrices with dimension A*B, whereas A is the number of recorded iterations and B the length of matrixZ1/Z2. Each row of the matrix corresponds to the vectorized matrixZ1 or matrixZ2 in each iteration.

`matrixW1,matrixW2`
: The factor loading matrix representing the degree of influence of the latent factors on individual genes. Dim(matrixW1)=G1*K; Dim(matrixW2)=G2*K. Real continuous value with mean 0 and SD 1.

`matrixW_chain`
: The updated matrixW in each recorded iteration, with format similar to matrixZ_chain.

`matrixX`
: The factor activity matrix with dimension K*J. matrixX[k,j] is the activity value of the k-th latent factor (e.g., pathway) in the j-th sample (e.g., cell line). Real continuous value with mean 0 and SD 1.

`matrixX_chain`
: The updated matrixX in each recorded iteration. An A*B matrix with each row corresponding to the vectorized matrixX in each recorded iteration.

`result_file_name`
: Name of the ".RData" file for storing the RMSE related result

`plot_name`
: Name of the ".pdf" file for plotting RMSE

Value

The generated .RData file includes the following components:

`scaled_Y1_mean,scaled_Y2_mean`
: The mean of matrixY1 and Y2 after data normalization (prior to the Gibbs sampling, the original matrixY1 and matrixY2 are normalized, so their mean will change), with dimension the same to that of matrixY1 and matrixY2

`est_Y1_mean_chain,est_Y2_mean_chain`
: Estimated mean of matrixY1 and matrixY2 in each iteration of the Gibbs sampling algorithm.Each row corresponds to the vectorized estimated mean of matrixY1 or matrixY2

`RMSE_Y_MEAN`
: Root-mean-squared-error of the estimation. A matrix of two columns. The first column records the RMSE for the mean of matrixY1 in each iteration of the Gibbs sampling process and the second column records that of matrixY2

Examples

```
library(Rlab)
library(MASS)
library(coda)
library(ROCR)

data(Y1_mean)
data(Y2_mean)
data(matrixY1)
data(matrixY2)
data(matrixZ_chain)
data(matrixW1)
data(matrixW2)
data(matrixW_chain)
data(matrixX)
```

```
data(matrixX_chain)
Ymean_compare(Y1_mean,Y2_mean,matrixY1, matrixY2,
matrixZ_chain, matrixW1, matrixW2, matrixW_chain,
matrixX, matrixX_chain, result_file_name="RMSE_demo.RData",
plot_name="RMSE_plot.pdf")
```

Index