

# Package ‘tensorIA’

January 3, 2020

**Type** Package

**Title** tensorIA

**Version** 0.1.0

**Author** Xu Liu [aut,cre],  
Wei Zou [aut],  
Yiming Liu [aut],  
Mengyun Wu [aut].

**Maintainer** Xu Liu <liu.xu@sufe.edu.cn>

**Description** For a multivariates grouped gression model, with or without aparsity assumptions, treating the coefficients as a third-order tensor and borrowing Tucker decomposition to reduce the number of parameters.

**License** GPL (>= 2)

**Imports** Rcpp (>= 0.11.15), RcppEigen (>= 0.3.2.3.0)

**LinkingTo** Rcpp, RcppEigen

**RoxygenNote** 6.0.1

**NeedsCompilation** yes

**Repository** github

**URL** <https://github.com/xliusufe/tensorIA>

**Encoding** UTF-8

## R topics documented:

tensorIA-package . . . . .	2
generateData . . . . .	2
integ . . . . .	3
integ_dr . . . . .	5
TransferModalUnfoldings . . . . .	7
<b>Index</b>	<b>8</b>

---

 tensorIA-package

*Integrative analysis based on tensor modelling*


---

### Description

For a multivariate grouped regression model, with or without sparsity assumptions, treating the coefficients as a third-order tensor and borrowing Tucker decomposition to reduce the number of parameters.

### Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

### Author(s)

Xu Liu

Maintainer: Xu Liu <liu.xu@sufe.edu.cn>

### References

Integrative analysis based on tensor modelling.

---

 generateData

*Generate data from multivariate regression model.*


---

### Description

Generate data for a multivariate regression model.

### Usage

```
generateData(n, q, p, g, D3, SigmaX=NULL, SigmaE=NULL,
             mu=NULL, sigma2=NULL, seed_id=NULL)
```

### Arguments

n	Sample size.
q	The number of responses, $q \geq 1$ .
p	The number of covariates, $p \geq 1$ .
g	The number of groups.
D3	The mode of unfolding $D_{(3)}$ .
sigma2	err variance. Default is 0.2.
SigmaX	A $pg \times pg$ positive-definite matrix, which is the covariance matrix of covariates $X$ .
SigmaE	A $q \times q$ positive-definite matrix, which is the covariance matrix of error $E$ .
seed_id	A positive integer, the seed for generating the random numbers.

## Details

This function gives coefficients of multivariate regression. The singular value matrix of tensor is a  $r_1 \times r_2 \times r_3$ -tensor. We choose  $r_1$ ,  $r_2$  and  $r_3$  by BIC, AIC, EBIC, CV, or GCV.

## Value

Y	Response, a $n \times q$ -matrix.
X	Design matrix, a $n \times pg$ -matrix.

## References

Integrative analysis based on tensor modelling.

## See Also

mam\_sparse

## Examples

```
# Example 1

D3 <- matrix(runif(30, 0.7, 1), 2, 15)
mydata <- generateData(200, 3, 5, 5, D3)

Y <- mydata$Y
X <- mydata$X


# Example 2
n <- 500
p <- 10
q <- 10
g <- 10
r10=r20=r30=2
S3 <- matrix(runif(r10*r20*r30,3,7),nrow = r30)
T1 <- matrix(rnorm(s0*r10),nrow = s0)
U1 <- qr.Q(qr(T1))
T1 <- matrix(rnorm(g*r20),nrow = g)
U2 <- qr.Q(qr(T1))
T1 <- matrix(rnorm(q*r30),nrow = q)
U3 <- qr.Q(qr(T1))
D3 <- U3%*%S3%*%t(kronecker(U2,U1))
mydata <- generateData(n,q,p,g,D3)
```

## Description

Fit a multivariates grouped gression model by treating coefficients as a order 3 tensor, without aparsity assumptions, and given ranks  $r_1$ ,  $r_2$ ,  $r_3$ .

**Usage**

```
integ(Y, X, g = 1, r1 = NULL, r2 = NULL, r3 = NULL, SABC = NULL,
      intercept = TRUE, mu = NULL, eps = 1e-4, max_step = 20)
```

**Arguments**

Y	A $n \times q$ numeric matrix of responses.
X	A $n \times p$ numeric design matrix for the model.
g	The number of groups. Default is 1.
r1	The first dimension of single value matrix of the tensor. Default is 2.
r2	The second dimension of single value matrix of the tensor. Default is 2.
r3	The third dimension of single value matrix of the tensor. Default is 2.
SABC	A user-specified list of initial coefficient matrix of $S, A, B, C$ . By default, initial matrices are provided by random.
intercept	Should intercept(s) be fitted (default=TRUE) or set to zero (FALSE)?
mu	A user-specified initial of intercept(s), a $q$ -vector. Default is $\emptyset$ .
eps	Convergence threshold. The algorithm iterates until the relative change in any coefficient is less than eps. Default is $1e-4$ .
max_step	Maximum number of iterations. Default is 20.

**Details**

This function gives pq functional coefficients' estimators of MAM. The singular value matrix of tensor is a  $r_1 \times r_2 \times r_3$ -tensor. We choose  $r_1, r_2$  and  $r_3$  by BIC or CV.

**Value**

Dnew	Estimator of $D_{(3)}$ .
mu	Estimator of intercept $\mu$ .
rss	Residual sum of squares (RSS).
Y	Response $Y$ .
X	Design matrix $X$ .

**References**

Integrative analysis based on tensor modelling.

**See Also**

integ\_dr

**Examples**

```
n <- 200
p <- 5
q <- 5
g <- 5
r10 <- 2
```

```

r20 <- 2
r30 <- 2
S3 <- matrix(runif(r10*r20*r30,3,7),nrow = r30)
T1 <- matrix(rnorm(p*r10),nrow = p)
U1 <- qr.Q(qr(T1))
T1 <- matrix(rnorm(g*r20),nrow = g)
U2 <- qr.Q(qr(T1))
T1 <- matrix(rnorm(q*r30),nrow = q)
U3 <- qr.Q(qr(T1))
D3 <- U3%*%S3%*%t(kronecker(U2,U1))
X <- matrix(runif(n*p*g), nrow = n)
eps <- matrix(rnorm(n*q),n,q)
Y <- X%*%t(D3) + eps

fit <- integ(Y, X, g, r1=2, r2=2, r3=2)
D3hat <- fit$Dnew
D2hat <- TransferModalUnfoldings(D3hat,3,2,p,g,q)

```

---

integ_dr	<i>Integrative analysis for GWAS data without sparsity assumption, and with ranks selected by BIC, AIC, EBIC, CV, or GCV.</i>
----------	---

---

## Description

Fit a multivariate grouped regression model by treating coefficients as a order 3 tensor, without sparsity assumptions, and with ranks  $r_1, r_2, r_3$  selected by BIC, AIC, EBIC, CV, or GCV.

## Usage

```

integ_dr(Y, X, g = 1, method = "BIC", ncv = 10, r1_index = NULL,
         r2_index = NULL, r3_index = NULL, SABC = NULL, intercept = TRUE, mu = NULL,
         eps = 1e-4, max_step = 20)

```

## Arguments

Y	A $n \times q$ numeric matrix of responses.
X	A $n \times p$ numeric design matrix for the model.
g	The number of groups. Default is 1.
method	The method to be applied to select parameters. Either BIC (the default), AIC, EBIC, CV, or GCV.
ncv	The number of cross-validation folds. Default is 10. If method is "BIC", ncv is useless.
r1_index	A user-specified sequence of $r_1$ values, where $r_1$ is the first dimension of single value matrix of the tensor. Default is $r1\_index = 1, \dots, \min(\log(n), p)$ .
r2_index	A user-specified sequence of $r_2$ values, where $r_2$ is the second dimension of single value matrix of the tensor. Default is $r2\_index = 1, \dots, \min(\log(n), g)$ .
r3_index	A user-specified sequence of $r_3$ values, where $r_3$ is the third dimension of single value matrix of the tensor. Default is $r3\_index = 1, \dots, \min(\log(n), q)$ .
SABC	A user-specified list of initial coefficient matrix of $S, A, B, C$ , which is a list with values $S, A, B, C$ . By default, initial matrices are provided by random.

intercept	Should intercept(s) be fitted (default=TRUE) or set to zero (FALSE)?
mu	A user-specified initial of intercept(s), a $q$ -vector. Default is 0.
eps	Convergence threshold. The algorithm iterates until the relative change in any coefficient is less than eps. Default is 1e-4.
max_step	Maximum number of iterations. Default is 20.

### Details

This function gives pq functional coefficients' estimators of MAM. The singular value matrix of tensor is a  $r_1 \times r_2 \times r_3$ -tensor. We choose  $r_1$ ,  $r_2$  and  $r_3$  by BIC or CV.

### Value

Dnew	Estimator of $D_{(3)}$ .
mu	Estimator of intercept $\mu$ .
rss	Residual sum of squares (RSS).
rk_opt	The optimal parametres that slected by BIC (the default), AIC, EBIC, CV, or GCV. It is a vector with length 4, which are selected $r_1$ , $r_2$ , $r_3$ , and $K$ .
selected	Which $\lambda$ is selection.
Y	Response $Y$ .
X	Design matrix $X$ .

### References

Integrative analysis based on tensor modelling.

### See Also

integ

### Examples

```
n <- 200
p <- 5
q <- 5
g <- 5
r10 <- 2
r20 <- 2
r30 <- 2
S3 <- matrix(runif(r10*r20*r30,3,7),nrow = r30)
T1 <- matrix(rnorm(p*r10),nrow = p)
U1 <- qr.Q(qr(T1))
T1 <- matrix(rnorm(g*r20),nrow = g)
U2 <- qr.Q(qr(T1))
T1 <- matrix(rnorm(q*r30),nrow = q)
U3 <- qr.Q(qr(T1))
D3 <- U3%*%S3%*%t(kronecker(U2,U1))
X <- matrix(runif(n*p*g), nrow = n)
eps <- matrix(rnorm(n*q),n,q)
Y <- X%*%t(D3) + eps

fit <- integ_dr(Y, X, g)
```

```
D3hat <- fit$Dnew
D2hat <- TransferModalUnfoldings(D3hat,3,2,p,g,q)
opt <- fit$rk_opt
```

---

TransferModalUnfoldings

*Transfer a tensor's modal unfoldings to another.*


---

## Description

Transfer a tensor's modal unfoldings to another.

## Usage

```
TransferModalUnfoldings(S, d1, d2 , r1, r2, r3)
```

## Arguments

S	A mode-d1-unfolding of a tensor with size $r_1 \times r_2 \times r_3$ , input unfolding
d1	An integer, the mode of unfolding $S_{(d_1)}$
d2	An integer, the mode of output unfolding $S_{(d_2)}$
r1	The first dimension of tensor
r2	The second dimension of tensor
r3	The third dimension of tensor

## Details

This function transfers an input mode-d1-unfolding  $S_{(d_1)}$  to mode-d2-unfolding  $S_{(d_2)}$

## Value

D the output mode-d2-unfolding,  $S_{(d_2)}$

## References

A tensor estimation approach to multivariate additive models.

## Examples

```
D1 <- matrix(1:24,nrow = 4) # A tensor unfolding with size 4*6
D2 <- TransferModalUnfoldings(D1,1,2,4,3,2)
```

# Index

\*Topic **GWAS; Integrative analysis;  
Tensor estimation; Tucker  
decomposition.**

generateData, [2](#)

integ, [3](#)

tensorIA-package, [2](#)

\*Topic **High-dimensional, Sparse  
models; Tensor estimation;  
Tucker decomposition.**

TransferModalUnfoldings, [7](#)

generateData, [2](#)

integ, [3](#)

integ-function(integ), [3](#)

integ\_dr, [5](#)

integ\_dr-function(integ\_dr), [5](#)

tensorIA(tensorIA-package), [2](#)

tensorIA-package, [2](#)

TransferModalUnfoldings, [7](#)

TransferModalUnfoldings-function  
(TransferModalUnfoldings), [7](#)