timecourse

March 24, 2012

`abs2ratio` *Convert log-values to log-ratios*

Description

For a single gene, computes the log ratios between time courses from two paired biological conditions.

Usage

```
abs2ratio(x, mn, k, c.grp, reference)
```

Arguments

`x`	a numeric vector giving the log-values of a gene with two paired biological conditions, sorted in ascending order by biological condition, replicate, and time groups.
`mn`	a numeric matrix giving the sample sizes for the two biological conditions.
`k`	a positive integer giving the number of time points.
`c.grp`	an numeric or character vector with length equals to that of `x`, giving the biological condition group for each element of `x`.
`reference`	a numeric value or character assigning the reference biological condition.

Details

This function is for internal use only and is not to be called by the user.

Value

a numeric vector containing log-ratios between two paired biological conditions.

Author(s)

Yu Chuan Tai <yuchuan@stat.berkeley.edu>

See Also

`mb.paired`.

fruitfly *Drosophila microarray time course data in Tomancak et al. (2002)*

Description

This is a subset of the Drosophila microarray time course data in Tomancak et al. (2002).

Usage

```
data(fruitfly)
```

Format

A matrix of log_2 gene expression values for 2000 probesets.

Source

Tomancak et al. (2002) *Systematic determination of patterns of gene expression during Drosophila embryogenesis.* Genome Biology 2002, 3:research0088.1-0088.14 http://genomebiology.com/2002/3/12/research/0088.1 The complete dataset can be downloaded from the following website http://www.fruitfly.org/cgi-bin/ex/insitu.pl

MArrayTC-class *Microarray Time Course Object- class*

Description

A list-based class for storing the analysis results from the multivariate empirical Bayes models of differential expression for longitudinal replicated developmental microarray time course data. Objects are normally created by mb.long and mb.MANOVA.

Slots/Components

MArrayTC objects do not contain any slots (apart from .Data) but they should contain the following list components:

M: input matrix of log-ratios or log-values of expression for a series of microarrays.

Objects may also contain the following optional components:

prop: numeric value giving the proportion of differentially expressed genes.

nu: numeric value containing the estimated amount of moderation.

Lambda: the estimated Lambda.

Lambda1: the estimated Lambda1.

eta: the estimated prior scale parameter.

alpha: the estimated common mean of the expected time course vector under the null.

alpha.d: the estimated condition-specific means of the expected time course vectors under the alternative.

`beta`: the estimated scale parameter for the common covariance matrix of the common expected time course vector under the null.

`beta.d`: the estimated condition-specific scale parameters for the common covariance matrix of the expected time course vectors under the alternative.

`percent`: `numeric` matrix containing the percent of moderation corresponding to each sample size for the longitudinal one- and two- sample problems.

`size`: `numeric` vector or matrix containing the sample sizes for all genes corresponding to different biological conditions, when the latter are sorted in ascending order.

`con.group`: `numeric` or `character` vector giving the biological condition group of each array. The i_th element of `con.group` corresponds to the biological condition of the i_th column of `M`.

`rep.group`: `numeric` or `character` vector giving the replicate group of each array. The i_th element of `rep.group` corresponds to the replicate of the i_th column of `M`.

`time.group`: `numeric` vector giving the time group of each array. The i_th element of `time.group` corresponds to the time of the i_th column of `M`.

`HotellingT2`: `numeric` vector giving the $\tilde{T}^2$ statistics of differential expression.

`MB`: `numeric` vector giving the MB-statistics of differential expression.

`pos.HotellingT2`: `numeric` vector whose i_th element corresponds to the index of the gene with ranking i in `HotellingT2`.

`pos.MB`: `numeric` vector whose i_th element corresponds to the index of the gene with ranking i in `MB`.

`geneNames`: `character` vector giving gene names.

`descriptions`: `character` vector giving gene descriptions.

Methods

MArrayTC extends the `LargeDataObject` class in package limma, and inherits a `show` method from there.

The function `plotProfile` takes `MArrayTC` as the input argument.

Author(s)

Yu Chuan Tai <yuchuan@stat.berkeley.edu>

`matrix.cov` *Covariance*

Description

For a single gene, computes the transformed or untransformed sample covariance matrix if one biological condition, or pooled sample covariance matrix if two or more biological conditions.

Usage

```
matrix.cov(x, k, trans = TRUE, c.grp = NULL, use = "complete.obs")
```

Arguments

x	a numeric vector giving the log-ratios or log-values for a gene, sorted in ascending order by biological condition, replicate, and time groups.
k	a positive integer giving the number of time points.
trans	logical. Should the Helmert transformation be performed?
c.grp	a numeric vector corresponding to the biological condition group for each element of x.
use	character. The same as the use in stats function cov. The default uses complete observations.

Details

This function is for internal use only and is not to be called by the user.

Value

A numeric matrix.

Author(s)

Yu Chuan Tai <yuchuan@stat.berkeley.edu>

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

cov, ot.helmert.

---

mb.MANOVA *Multivariate Empirical Bayes Analysis of Variance for Longitudinal Replicated Developmental Microarray Time Course Data*

Description

Computes the MB-statistics for longitudinal replicated developmental microarray time course data with multiple biological conditions.

Usage

```
mb.MANOVA(object, times, D, size, nu = NULL, Lambda = NULL,
beta.d = NULL, beta = NULL, alpha.d = NULL, alpha = NULL,
condition.grp, time.grp = NULL, rep.grp = NULL, p = 0.02)
```

Arguments

object	Required. An object of class matrix, MAList, marrayNorm, or ExpressionSet containing log-ratios or log-values of expression for a series of microarrays.
times	Required. A positive integer giving the number of time points.
D	Required. A positive integer giving the number of biological conditions. D>1
size	Required. A numeric matrix corresponding to the sample sizes for all genes across different biological conditions, when biological conditions are sorted in ascending order. Rows represent genes while columns represent biological conditions.
nu	an optional positive value giving the degrees of moderation for the fully moderated Wilks' Lambda.
Lambda	an optional numeric matrix giving the common covariance matrix for the fully moderated Wilks' Lambda.
beta.d	an optional numeric vector of length D giving the condition-specific scale parameters for the common covariance matrix of the expected time course vectors under the alternative.
beta	an optional numeric value giving the scale parameter for the common covariance matrix of the common expected time course vector under the null.
alpha.d	an optional numeric matrix giving the condition-specific means of the expected time course vectors under the alternative.
alpha	an optional numeric vector of length times giving the common mean of the expected time course vector under the null.
condition.grp	Required. A numeric or character vector with length equals to the number of arrays, assigning the biological condition group to each array.
rep.grp	an optional numeric or character vector with length equals to the number of arrays, assigning the replicate group to each array.
time.grp	an optional numeric vector with length equals to the number of arrays, assigning the time point group to each array.
p	a numeric value between 0 and 1, assumed proportion of genes which are differentially expressed.

Details

This function implements the multivariate empirical Bayes Analysis of Variance model for identifying genes with different temporal profiles across multiple biological conditions, as described in Tai (2005).

Value

Object of MArrayTC.

Author(s)

Yu Chuan Tai <yuchuan@stat.berkeley.edu>

References

Yu Chuan Tai (2005). Multivariate empirical Bayes models for replicated microarray time course data. Ph.D. dissertation. Division of Biostatistics, University of California, Berkeley.

Yu Chuan Tai and Terence P. Speed (2005). Statistical analysis of microarray time course data. In: DNA Microarrays, U. Nuber (ed.), BIOS Scientific Publishers Limited, Taylor & Francis, 4 Park Square, Milton Park, Abingdon OX14 4RN, Chapter 20.

See Also

timecourse Vignette.

Examples

```
SS <- matrix(c(    0.01, -0.0008,   -0.003,     0.007,  0.002,
                -0.0008,    0.02,    0.002,   -0.0004, -0.001,
                 -0.003,   0.002,     0.03,   -0.0054, -0.009,
                  0.007, -0.0004, -0.00538,      0.02, 0.0008,
                  0.002,  -0.001,   -0.009,    0.0008,  0.07), ncol=5)

sim.Sigma <- function()
{
   S <- matrix(rep(0,25),ncol=5)
   x <- mvrnorm(n=10, mu=rep(0,5), Sigma=10*SS)
   for(i in 1:10)
       S <- S+crossprod(t(x[i,]))

   solve(S)

}

## Now let's simulate a dataset with three biological conditions
## 500 genes in total, 10 of them have different expected time course profiles
## across biological conditions
## the first condition has 3 replicates, while the second condition has 4 replicates,
## and the third condition has 2 replicates. 5 time points for each condition.

sim.data <- function(x, indx=1)
{
   mu <- rep(runif(1,8,x[1]),5)
   if(indx==1)
     res <- c(as.numeric(t(mvrnorm(n=3, mu=mu+rnorm(5,sd=5), Sigma=sim.Sigma()))),
              as.numeric(t(mvrnorm(n=4, mu=mu+rnorm(5,sd=3.2), Sigma=sim.Sigma()))),
              as.numeric(t(mvrnorm(n=2, mu=mu+rnorm(5,sd=2), Sigma=sim.Sigma()))))

   if(indx==0) res <- as.numeric(t(mvrnorm(n=9, mu=mu+rnorm(5,sd=3), Sigma=sim.Sigma())))
   res
}

M <- matrix(rep(14,500*45), ncol=45)
M[1:10,] <- t(apply(M[1:10,],1,sim.data))
M[11:500,] <- t(apply(M[11:500,],1,sim.data, 0))


assay <- rep(c("1.2.04","2.4.04","3.5.04","5.21.04","7.17.04","9.10.04","12.1.04","1.2.05
trt <- c(rep(c("wildtype","mutant1"),each=15),rep("mutant1",5), rep("mutant2", 10))
```

```
# Caution: since "mutant1" < "mutant2" < "wildtype", the sample sizes should be in the or
# but NOT 3,4,2.
size <- matrix(c(4,2,3), byrow=TRUE, nrow=500, ncol=3)
MB.multi <- mb.MANOVA(M, times=5, D=3, size=size, rep.grp=assay, condition.grp=trt)

plotProfile(MB.multi, stats="MB", type="b") # plots the no. 1 gene
```

`mb.long` *Multivariate Empirical Bayes Statistics for Longitudinal Replicated Developmental Microarray Time Course Data*

Description

Computes the $\tilde{T}^2$ statistics and/or the MB-statistics of differential expression for longitudinal replicated developmental microarray time course data by multivariate empirical Bayes shrinkage of gene-specific sample variance-covariance matrices towards a common matrix.

Usage

```
mb.long(object, method = c("1D", "paired", "2D"), type = c("none", "robust"),
times, reps, prior.df = NULL, prior.COV = NULL,
prior.eta = NULL, condition.grp = NULL, rep.grp = NULL, time.grp = NULL,
one.sample = FALSE, ref = NULL, p = 0.02, out.t = FALSE,
tuning = 1.345, HotellingT2.only=TRUE)
```

Arguments

`object`	Required. An object of class `matrix`, `MAList`, `marrayNorm`, or `ExpressionSet` containing log-ratios or log-values of expression for a series of microarrays.
`method`	a character string, `"1D"` for the one-sample case where genes of interest are those which change over time, `"paired"` for the one-sample case where genes of interest are those whose expected temporal profiles do not stay 0, for example, cDNA microarrays, or the paired two-sample case where genes of interest are those with different expected temporal profiles across 2 biological conditions, `"2D"` for the independent two-sample case where genes of interest are those with different expected temporal profiles across 2 biological conditions. The default is `"1D"`.
`type`	a character string, indicating whether possible outliers should be down-weighted.
`times`	Required. A positive integer giving the number of time points.
`reps`	Required. A numeric vector or matrix corresponding to the sample sizes for all genes across different biological conditions, when biological conditions are sorted in ascending order. If a matrix, rows represent genes while columns represent biological conditions.
`prior.df`	an optional positive value giving the degrees of moderation.
`prior.COV`	an optional numeric matrix giving the common covariance matrix to which the gene-specific sample covariances are smoothed toward.
`prior.eta`	an optional numeric value giving the scale parameter for the covariance matrix for the expected time course profile.

`condition.grp`
: a numeric or character vector with length equals to the number of arrays, assigning the biological condition group of each array. Required if `method=2D`.

`rep.grp`
: an optional numeric or character vector with length equals to the number of arrays, assigning the replicate group of each array.

`time.grp`
: an optional numeric vector with length equals to the number of arrays, assigning the time point group of each array.

`one.sample`
: Is it a one-sample problem? Only specify this argument when `method=paired`. The default is `FALSE` which means it is a paired two-sample problem.

`ref`
: an optional numeric value or character specifying the name of reference biological condition. The default uses the first element of `condition.grp`. Only specify this argument when `method=paired` and `one.sample` is `FALSE`.

`p`
: a numeric value between 0 and 1, assumed proportion of genes which are differentially expressed.

`out.t`
: logical. Should the moderated multivariate t-statistics be outputed? The default is `FALSE`.

`tuning`
: the tuning constant for the Huber weight function with a default 1.345.

`HotellingT2.only`
: logical. Should only the HotellingT2 statistics be outputed? This should be set as `TRUE` (default) when the sample size(s) are the same across genes, in order to reduce computational time.

Details

This function implements the multivariate empirical Bayes statistics described in Tai and Speed (2004), to rank genes in the order of interest from longitudinal replicated developmental microarray time course experiments. It calls one of the following functions, depending on which `method` is used: `mb.1D`, `mb.paired`, and `mb.2D`.

The arguments `condition.grp`, `rep.grp`, and `time.grp`, if specified, should have lengths equal to the number of arrays. The i_th elements of these three arguments should correspond to the biological condition, replicate, and time for the i_th column (array) in the expression value matrix of the input object, respectively. The default assumes the columns of `M` are in the ascending order of `condition.grp` first, and then `rep.grp`, and finally `time.grp`.

Arguments `one.sample` and `ref` are for `method=paired` only.

When `type=robust`, the numerator of the $\tilde{T}^2$ statistic is calculated using the weighted average time course vector(s), where the weight at each data point is determined using Huber's weight function with the default tuning constant 1.345.

Warning: When there are only 2 replicates within conditions, `type="robust"` produces the same rankings as `type="none"` since there is no consensus on gene expression values. Check the output weights for these outliers.

Value

Object of `MArrayTC`.

Author(s)

Yu Chuan Tai <yuchuan@stat.berkeley.edu>

References

Yu Chuan Tai and Terence P. Speed (2006). A multivariate empirical Bayes statistic for replicated microarray time course data. Annals of Statistics 34(5):2387-2412.

Yu Chuan Tai and Terence P. Speed (2005). Statistical analysis of microarray time course data. In: DNA Microarrays, U. Nuber (ed.), BIOS Scientific Publishers Limited, Taylor & Francis, 4 Park Square, Milton Park, Abingdon OX14 4RN, Chapter 20.

P. J. Huber (2004). Robust Statistics. *Wiley series in probability and mathematical statistics*.

See Also

timecourse Vignette.

Examples

```
data(fruitfly)
colnames(fruitfly) ## check if arrays are arranged in the default order
gnames <- rownames(fruitfly)
assay <- rep(c("A", "B", "C"), each = 12)
time.grp <- rep(c(1:12), 3)
size <- rep(3, nrow(fruitfly))

out1 <- mb.long(fruitfly, times=12, reps=size, rep.grp = assay, time.grp = time.grp)
summary(out1)
plotProfile(out1, type="b", gnames=gnames, legloc=c(2,15), pch=c("A","B","C"), xlab="Hour

## Simulate gene expression data
## Note: this simulation is for demonstration purpose only,
## and does not necessarily reflect the real
## features of longitudinal time course data

## one biological condition, 5 time points, 3 replicates
## 500 genes, 10 genes change over time

SS <- matrix(c(    0.01, -0.0008,   -0.003,     0.007,  0.002,
               -0.0008,    0.02,    0.002,   -0.0004, -0.001,
                -0.003,   0.002,     0.03,   -0.0054, -0.009,
                 0.007, -0.0004, -0.00538,      0.02, 0.0008,
                 0.002,  -0.001,   -0.009,    0.0008,  0.07), ncol=5)

sim.Sigma <- function()
{
   S <- matrix(rep(0,25),ncol=5)
   x <- mvrnorm(n=10, mu=rep(0,5), Sigma=10*SS)
   for(i in 1:10)
       S <- S+crossprod(t(x[i,]))

   solve(S)

}

sim.data1 <- function(x, indx=1)
{
   mu <- rep(runif(1,8,x[1]),5)
   if(indx==1) res <- as.numeric(t(mvrnorm(n=3, mu=mu+rnorm(5,sd=4), Sigma=sim.Sigma())))
   if(indx==0) res <- as.numeric(t(mvrnorm(n=3, mu=mu, Sigma=sim.Sigma())))
```

```
   res
}

M1 <- matrix(rep(14,500*15), ncol=15)
M1[1:10,] <- t(apply(M1[1:10,],1,sim.data1))
M1[11:500,] <- t(apply(M1[11:500,],1,sim.data1, 0))

## Which genes are nonconstant?
MB.1D1 <- mb.long(M1, times=5, reps=rep(3, 500))
MB.1D1$percent  # check the percent of moderation

plotProfile(MB.1D1,type="b") # plots the no. 1 gene
plotProfile(MB.1D1,type="b",ranking=10) # plots the no. 10 gene
genenames <- as.character(1:500)
plotProfile(MB.1D1, type="b", gid="8", gnames=genenames) #plots the gene with ID "8"

##
MB.1D1.r <- mb.long(M1, type="r", times=5, reps=rep(3, 500))
plotProfile(MB.1D1.r,type="b",gnames=genenames)
plotProfile(MB.1D1.r,type="b", gid="1", gnames=genenames) #plots the gene with ID "1"

## assign the following labellings to columns of M1
## which is actually the same as the default
## Not Run
trt <- rep("wildtype", 15)
assay <- rep(c("A","B","C"), rep(5,3))
time.grp <- rep(c(0, 1, 3, 4, 6), 3)

## MB.1D2 should give the same results as MB.1D1
#MB.1D2 <- mb.long(M1, times=5, reps=rep(3, 500), condition.grp = trt, rep.grp = assay,
#time.grp=time.grp)

## suppose now the replicates are in this order instead
assay <- rep(c("A","C","B"), rep(5,3))

## then
MB.1D3 <- mb.long(M1, times=5, reps=rep(3, 500), condition.grp = trt, rep.grp = assay, ti
MB.1D3$rep.group  #check the replicate and time group
MB.1D3$time.group


## Now let's simulate another dataset with two biological conditions
## 500 genes also, 10 of them have different expected time course profiles
## between these two biological conditions
## 3 replicates, 5 time points for each condition

sim.data2 <- function(x, indx=1)
{
   mu <- rep(runif(1,8,x[1]),5)
   if(indx==1)
     res <- c(as.numeric(t(mvrnorm(n=3, mu=mu+rnorm(5,sd=5), Sigma=sim.Sigma()))),
              as.numeric(t(mvrnorm(n=3, mu=mu+rnorm(5,sd=3.2), Sigma=sim.Sigma()))))

   if(indx==0) res <- as.numeric(t(mvrnorm(n=6, mu=mu+rnorm(5,sd=3), Sigma=sim.Sigma())))
   res
}
```

```
M2 <- matrix(rep(14,500*30), ncol=30)
M2[1:10,] <- t(apply(M2[1:10,],1,sim.data2))
M2[11:500,] <- t(apply(M2[11:500,],1,sim.data2, 0))

## assume it is a paired two-sample problem
trt <- rep(c("wt","mt"),each=15)
assay <- rep(rep(c("1.2.04","2.4.04","3.5.04"),each=5),2)
size <- matrix(3, nrow=500, ncol=2)
MB.paired <- mb.long(M2, method="paired", times=5, reps=size, condition.grp=trt, rep.grp=
MB.paired$con.group # check the condition, replicate and time groups
MB.paired$rep.group
MB.paired$time.group

plotProfile(MB.paired, type="b")
genenames <- as.character(1:500)
plotProfile(MB.paired, gid="12", type="b", gnames=genenames) #plots the gene with ID "12"

### assume it is a unpaired two-sample problem
assay <- rep(c("1.2.04","2.4.04","3.5.04","5.21.04","7.17.04","8.4.04"),each=5)
MB.2D <- mb.long(M2, method="2", times=5, reps=size, condition.grp=trt, rep.grp=assay)
MB.2D$con.group # check the condition, replicate and time groups
MB.2D$rep.group
MB.2D$time.group

plotProfile(MB.2D,type="b", gnames=genenames) # plot the no. 1 gene


## Now let's simulate another dataset with two biological conditions
## 500 genes also, 10 of them have different expected time course profiles
## between these two biological conditions
## the first condition has 3 replicates, while the second condition has 4 replicates,
## 5 time points for each condition

sim.data3 <- function(x, indx=1)
{
   mu <- rep(runif(1,8,x[1]),5)
   if(indx==1)
     res <- c(as.numeric(t(mvrnorm(n=3, mu=mu+rnorm(5,sd=5), Sigma=sim.Sigma()))),
              as.numeric(t(mvrnorm(n=4, mu=mu+rnorm(5,sd=3.2), Sigma=sim.Sigma()))))

   if(indx==0) res <- as.numeric(t(mvrnorm(n=7, mu=mu+rnorm(5,sd=3), Sigma=sim.Sigma())))
   res
}

M3 <- matrix(rep(14,500*35), ncol=35)
M3[1:10,] <- t(apply(M3[1:10,],1,sim.data3))
M3[11:500,] <- t(apply(M3[11:500,],1,sim.data3, 0))

assay <- rep(c("1.2.04","2.4.04","3.5.04","5.21.04","7.17.04","9.10.04","12.1.04"),each=5
trt <- c(rep(c("wildtype","mutant"),each=15),rep("mutant",5))
## Note that "mutant" < "wildtype", the sample sizes are (4, 3)
size <- matrix(c(4,3), nrow=500, ncol=2, byrow=TRUE)
MB.2D.2 <- mb.long(M3, method="2", times=5, reps=size, rep.grp=assay, condition.grp=trt)
MB.2D.2$con.group # check the condition, replicate and time groups
MB.2D.2$rep.group
MB.2D.2$time.group
```

```
plotProfile(MB.2D.2, type="b") # plot the no. 1 gene
```

`ot.helmert` *Helmert orthogonal transformation*

Description

Computes the Helmert orthogonal transformation matrix.

Usage

```
ot.helmert(k)
```

Arguments

`k`	a positive integer giving the number of time points.

Details

This function is for internal use only and is not to be called by the user.

Value

a numeric matrix.

Author(s)

Yu Chuan Tai <yuchuan@stat.berkeley.edu>

`plotProfile` *Gene Temporal Profile Plot*

Description

Plots the longitudinal temporal profile of a gene.

Usage

```
plotProfile(object, stats=c("HotellingT2", "MB"), ranking=1, gid=NULL, gnames=NU
type=c("p","l","b"), col=2:100, lty=1:100, pch=1:100, lwd=2, xlab="Time",
ylab="Expression", legloc=NULL, xlim=NULL, ylim=NULL, cex.main=1,...)
```

Arguments

object	a MArrayTC object.
stats	a character indicating which statistic the ranking is based on.
ranking	a numeric value giving the ranking of the gene to be plotted.
gid	an optional character giving the ID of the gene to be plotted.
gnames	an optional character vector with the i_th element corresponds to the gene ID of the i_th gene in object$M.
desc	an optional character vector with the i_th element corresponds to the gene description of the i_th gene in object$M.
type	a character indicating the plot type, "p" for points, "l" for lines, and "b" for both.
col	a character or numeric vector giving the colors for different biological conditions. Default is 2:100.
lty	a character or numeric vector giving the line types for different replicates. Default is 1:100.
pch	a character or numeric vector giving the point types for different replicates. Default is 1:100.
lwd	optional. The default sets to 2.
xlab	character. The label for the x-axis.
ylab	character. The label for the y-axis.
legloc	an optional vector giving the location of the legend.
xlim	an optional vector giving the upper- and lower- limits of x-axis.
ylim	an optional vector giving the upper- and lower- limits of y-axis.
cex.main	optional. The default sets to 1
...	any other arguments passed onto plot

Details

This function takes an object of MArrayTC as the input and plots the temporal profile of a single gene. The user can specify either the ranking based on stats or the gene ID of the gene to be plotted.

See points for possible values for pch, col and cex.

See mb.long for examples.

Author(s)

Yu Chuan Tai <yuchuan@stat.berkeley.edu>

univ.func *Univariate Data*

Description

Transforms multivariate vectors into univariate values using the Helmert matrix.

Usage

```
univ.func(dummy, M, k, n, indx = 1)
```

Arguments

dummy	a numeric gene index.
M	a numeric matrix containing the log-values or log-ratios of a gene.
k	a positive integer giving the number of time points.
n	a positive integer giving the number of replicates.
indx	a positive integer between 1 and k, indicating which row of the Helmert matrix to transform the vectors.

Details

This function is for internal use only and is not to be called by the user.

Value

A numeric vector with length equals to n.

Author(s)

Yu Chuan Tai <yuchuan@stat.berkeley.edu>

See Also

ot.helmert.

Index