# Package 'ktabtools'

December 10, 2012

**Type** Package

**Title** Additional R functions related to K-table analyses

**Version** 1.0

**Date** 2012-11-21

**Author** Pierre Bady <pierre.bady@unil.ch>

**Maintainer** Pierre Bady <pierre.bady@unil.ch>

**Depends** R (>= 2.15.0),ade4, mixOmics

**Suggests** ade4, boot

**Description**

This package contains additional R functions related to K-table analyses and Multiblock-(s)pls.

**License** GPL version 2 or newer

## R topics documented:

## ktabtools-package *Additional R functions related to K-table analyses*

### Description

This package contains additional R functions related to K-table analyses and Multiblock-(s)pls.

### Details

| | |
|---|---|
| Package: | ktabtools |
| Type: | Package |
| Version: | 1.0 |
| Date: | 2012-11-21 |
| License: | GPL version 2 or newer |

### Author(s)

Pierre Bady <pierre.bady@unil.ch> Maintainer: Pierre Bady <pierre.bady@unil.ch>

## kplot.mbpls *Multiple plot representation associated with object (s)mbpls*

### Description

Representation of the scores and weights of the Tables X and Y

### Usage

```
kplot.mbpls(...)
```

### Arguments

| | |
|---|---|
| ... | further arguments passed to or from other methods |

## kRVplot *Multiple plot representation of RV coefficients*

### Description

Pairwise Analyse of the similarity among several tables

### Usage

```
kRVplot(X, whichinrow = NULL, whichincol = NULL, gap = 4, nclass = 10, coeff = 1, nrepet = 99, c
```

## Arguments

| | |
|---|---|
| X | |
| whichinrow | |
| whichincol | |
| gap | |
| nclass | numeric (integer). For S(-PLUS) compatibility only, <91>nclass<92> is equivalent to <91>breaks<92> for a scalar or character argument. |
| coeff | |
| nrepet | the number of permutations |
| col | a list of colors such as that generated by <91>rainbow<92>,<91>heat.colors<92>, <91>topo.colors<92>, <91>terrain.colors<92> or similar functions. |
| digits | minimal number of _significant_ digits (by default, 2). |
| ... | further arguments passed to or from other methods |

## Examples

```
require(ade4)
require(MASS)
# k-table simulation
tab0 <- as.data.frame(expand.grid(1:10,1:5))
row.names(tab0)<- paste("s",1:50,sep="")
tab1 <- cbind(tab0, mvrnorm(50,mu=rnorm(5,0,1),Sigma=diag(abs(rnorm(5,0,1)))))
tab2 <- cbind(tab0, mvrnorm(50,mu=rnorm(5,0,1),Sigma=diag(abs(rnorm(5,0,1)))))
tab3 <- cbind(tab0, mvrnorm(50,mu=rnorm(5,0,1),Sigma=diag(abs(rnorm(5,0,1)))))
listab <- list(tab1,tab2,tab3)
names(listab) <- c("tab1","tab2","tab3")
listpca <- list()
for(i in 1:3)
listpca[[i]] <- dudi.pca(listab[[i]],scannf=FALSE,nf=2)
# construction of the K-table
ktab1 <- ktab.list.dudi(listpca,tabnames= names(listab))
# graphical representation
kRVplot(ktab1)
```

---

# mbpls *Multi-Block Partial Least Squares (MBPLS)*

---

## Description

Performs a multi-block partail least squares, using list of K data.frames (Xb) associated with a data.frame (Y) or object 'ktab' (Xb) associated with object 'dudi' (Y).

## Usage

```
mbpls(X, ...)
mbpls.default(X, Y, option = c("inertia", "lambda1", "uniform"), deflation = c("super", "block")
nf = 3, tol = 1e-06, max.iter = 100, center.x=TRUE,scale.x=TRUE,center.y=TRUE,scale.y=TRUE,...)
mbpls.ktab(X, Y, option = c("inertia", "lambda1", "uniform", "internal"), deflation = c("super",
nf = 3, tol = 1e-06, max.iter = 100, ...)
```

## Arguments

| | |
|---|---|
| X | a list of data.frame or object 'ktab' |
| Y | a data.frame or object 'dudi' |
| option | a character describing the weighting of the table (inertia, lambda1 or uniform, see details) |
| deflation | a character describing deflation method ("super" or "block", see details) |
| nf | number of components |
| tol | a positive real, the tolerance used in the iterative algorithm. |
| max.iter | integer, the maximum number of iterations. |
| center.x | either a logical value or a numeric vector of length equal to the number of columns of 'X' |
| scale.x | either a logical value or a numeric vector of length equal to the number of columns of 'X'. |
| center.y | either a logical value or a numeric vector of length equal to the number of columns of 'Y' |
| scale.y | either a logical value or a numeric vector of length equal to the number of columns of 'Y'. |
| ... | further arguments passed to or from other methods |

## Details

Two methods of deflation are proposed in this function. The first (deflation="super") is the deflation mode proposed by Westerhuis and Coenegracht (1997) using the super score 'Tt'for the deflation step. The super score are orthogonal and the block scores 'Tb' are slightly correlated (more details in Westerhuis et al. 1998). The second method (deflation="block") is suggested by Wangen and Kowalski and it used the block scores 'Tb' for the deflation of the data block Xb. This deflation strategy forces the blocks scores to be orthogonal ('Tb') and the super scores 'Tt' becomes correlated.

## Value

mpls and all the functions that use it return a list with the following components:

| | |
|---|---|
| Tt | an object 'matrix' containing the super scores |
| Qt | an object 'matrix' containing the weights of the variables in Y |
| Wt | an object 'matrix' containing the super weights |
| Pb | an object 'matrix' containing the loadings of variables in block Xb |
| U | an object 'matrix' containing score of Y |
| Wb | an object 'matrix' containing the weights of the variables in block Xb |
| Tb | an object 'matrix' containing all the block score of Xb |
| iter | the number of iteration |
| Y | an object 'data.frame' corresponding to the response data |
| X | a list of 'data.frame' corresponding to descriptor data (all Xb) |
| InerY | Inertia of Y |
| InerX | Inertia of the block Xb |
| ExpVarX | Explained variances (inertia) in block Xb by component |

| | |
|---|---|
| ExpVarY | Explained variances (inertia) in Y by component |
| blo | a vector of variable number by blocks |
| TC | a numeric vector with the factors for columns |
| TL | a numeric vector with the factors for rows |
| call | original call |

## Note

The version based on 'ktab' and 'dudi' object is in development and not stabilized!!!

## References

González I., Lé Cao, K-A. and Déjean S. (2011) mixOmics: Omics Data Integration Project. URL: http://www.math.univ-toulouse.fr/~biostat/mixOmics/ Lé Cao, K-A., González I. and Déjean S. (2009) intergrOmics: an R package to unravel relationships between two omics data sets. Bioinformatics. 25(21):2855-2856. Li W, Zhang S, Liu CC, Zhou XJ.(2012)Identifying multi-layer gene regulatory modules from multi-dimensional genomic data.Bioinformatics, 28(19):2458-66. Westerhuis et al. (1998) Analysis of multiblock and hierarchical pca and pls models, journal of chemometrics, 12, 301-321.

## Examples

```
# example1: Comparison of pls and mbpls for two tables (Y and X)
require(ade4)
require(mixOmics)
data(linnerud)
X <- linnerud$exercise
Y <- linnerud$physiological
pls1 <- pls(X, Y, mode = "classic",ncomp=3)
mb1 <- mbpls(list(X),Y,option="uniform",deflation="super")
par(mfrow=c(2,2))
plot(pls1$variates$Y[,1],mb1$U[,1])
abline(0,1,col="red")
plot(pls1$variates$Y[,2],mb1$U[,2])
abline(0,1,col="red")
plot(pls1$variates$Y[,3],mb1$U[,3])
abline(0,1,col="red")
```

---

# network *Network representation associated with object (s)mbpls*

---

## Usage

```
network.mbpls(object, comp = 1, X.names = NULL, Y.names = NULL, keep.var = TRUE, threshold = 0.1
network.smbpls(object, comp = 1, X.names = NULL, Y.names = NULL, keep.var = TRUE, threshold = 0.
```

### Arguments

object
comp
X.names
Y.names
keep.var
threshold
...

---

## plot.mbpls *Plot representation associated with object (s)mbpls*

---

### Usage

```
plot.smbpls(...)
```

### Arguments

| | |
|---|---|
| ... | further arguments passed to or from other methods |

---

## predict.mbpls *Prediction for object (s)mbpls*

---

### Usage

```
predict.mbpls(x,data=NULL,nf=NULL,...)
```

### Arguments

...

### Examples

```
# example1: Comparison of pls and mbpls for two tables (Y and X)
require(ade4)
require(mixOmics)
data(linnerud)
X <- linnerud$exercise
Y <- linnerud$physiological
pls1 <- pls(X, Y, mode = "classic",ncomp=3)
mb1 <- mbpls(list(X),Y,option="uniform",deflation="super")
par(mfrow=c(2,2))
plot(pls1$variates$Y[,1],mb1$U[,1])
abline(0,1,col="red")
plot(pls1$variates$Y[,2],mb1$U[,2])
abline(0,1,col="red")
plot(pls1$variates$Y[,3],mb1$U[,3])
abline(0,1,col="red")
pred1 <- predict(mb1)
pred2 <- predict(mb1,list(X))
```

smbpls *Sparse Multi-Block Partial Least Squares (sMBPLS)*

## Description

Performs a sparse multi-block partail least squares, using list of K data.frames (Xb) associated with a data.frame (Y) or object 'ktab' (Xb) associated with object 'dudi' (Y). Warnings!!!! presently this function is in development and the management of the weight is not correct!!!

## Usage

```
smbpls(X, ...)
smbpls.default(X, Y, keepX = NULL, keepY = NULL, nf = 3, option = c("inertia","lambda1", "unifor
max.iter = 100,center.x=TRUE,scale.x=TRUE,center.y=TRUE,scale.y=TRUE,...)
smpls.ktab(X,Y,keepX=NULL,keepY=NULL,nf=3,option = c("inertia", "lambda1", "uniform", "internal"
```

## Arguments

| | |
|---|---|
| X | a list of data.frame or object 'ktab' |
| Y | a data.frame or object 'dudi' |
| keepX | a numeric matrix containing the number of selected probes for each table by axes (dimension: number of table x nf). By default all variables are kept in the model. |
| keepY | a numeric vector of length 'nf', the number of variables to keep in Y-weights. By default all variables are kept in the model. |
| nf | the number of components to include in the model (by default nf=3). |
| option | a character describing the weighting of the table (inertia, lambda1 or uniform, see details) |
| deflation | a character describing deflation method ("super" or "block", see details) |
| tol | a positive real, the tolerance used in the iterative algorithm. |
| max.iter | integer, the maximum number of iterations. |
| center.x | either a logical value or a numeric vector of length equal to the number of columns of 'X' |
| scale.x | either a logical value or a numeric vector of length equal to the number of columns of 'X'. |
| center.y | either a logical value or a numeric vector of length equal to the number of columns of 'Y' |
| scale.y | either a logical value or a numeric vector of length equal to the number of columns of 'Y'. |
| ... | further arguments passed to or from other methods. |

## Details

Two methods of deflation are proposed in this function. The first (deflation="super") is the deflation mode proposed by Westerhuis and Coenegracht (1997) using the super score 'Tt'for the deflation step. The super score are orthogonal and the block scores 'Tb' are slightly correlated (more details in Westerhuis et al. 1998). The second method (deflation="block") is suggested by Wangen

and Kowalski and it used the block scores 'Tb' for the deflation of the data block Xb. This deflation strategy forces the blocks scores to be orthogonal ('Tb') and the super scores 'Tt' becomes correlated.

The function 'smbpls' is based on the 'sparsity' approach proposed in the R package 'mixOmics'.

## Value

smpls and all the functions that use it return a list with the following components:

| | |
|---|---|
| `Tt` | an object 'matrix' containing the super scores |
| `Qt` | an object 'matrix' containing the weights of the variables in Y |
| `Wt` | an object 'matrix' containing the super weights |
| `Pb` | an object 'matrix' containing the loadings of variables in block Xb |
| `U` | an object 'matrix' containing score of Y |
| `Wb` | an object 'matrix' containing the weights of the variables in block Xb |
| `Tb` | an object 'matrix' containing all the block score of Xb |
| `iter` | the number of iteration |
| `Y` | an object 'data.frame' corresponding to the response data |
| `X` | a list of 'data.frame' corresponding to descriptor data (all Xb) |
| `keepX` | a numeric matrix containing the number of selected probes for each table by axes (dimension: number of table x nf) |
| `keepY` | a numeric vector of length 'nf', the number of variables to keep in Y-weights |
| `InerY` | Inertia of Y |
| `InerX` | Inertia of the block Xb |
| `ExpVarX` | Explained variances (inertia) in block Xb by component |
| `ExpVarY` | Explained variances (inertia) in Y by component |
| `blo` | a vector of variable number by blocks |
| `TC` | a numeric vector with the factors for columns |
| `TL` | a numeric vector with the factors for rows |
| `call` | original call |

## Note

The version based on 'ktab' and 'dudi' object is in development and not stabilized!!!

## References

González I., Lé Cao, K-A. and Déjean S. (2011) mixOmics: Omics Data Integration Project. URL: http://www.math.univ-toulouse.fr/~biostat/mixOmics/ Lé Cao, K-A., González I. and Déjean S. (2009) intergrOmics: an R package to unravel relationships between two omics data sets. Bioinformatics. 25(21):2855-2856. Li W, Zhang S, Liu CC, Zhou XJ.(2012)Identifying multi-layer gene regulatory modules from multi-dimensional genomic data.Bioinformatics, 28(19):2458-66. Westerhuis et al. (1998) Analysis of multiblock and hierarchical pca and pls models, journal of chemometrics, 12, 301-321.

## statismod *Statis*

### Usage

```
statismod(X, scannf = TRUE, nf = 3, tol = 1e-07)
```

### Arguments

X

scannf

nf

tol

## summary.mbpls *Summary of object (s)mbpls*

### Usage

```
summary.mbpls(...)
```

### Arguments

...

## valid.mbpls *Validation of object (s)mbpls*

### Usage

```
valid.mbpls(...)
```

### Arguments

...

# Index