

# Package ‘concor’

April 17, 2009

**Version** 1.0-0.1

**Date** 2009-02-25

**Title** Concordance

**Author** R. Lafosse <lafosse@lsp.ups-tlse.fr>

**Maintainer** S. Déjean <sdejean@lsp.ups-tlse.fr>

**Depends** R (>= 0.99)

**Description** The four functions svdcp (cp for column partitioned), svdbip or svdbip2 (bip for bi-partitioned), and svdbips (s for a simultaneous optimization of one set of r solutions), correspond to a “SVD by blocks” notion, by supposing each block depending on relative subspaces, rather than on two whole spaces as usual SVD does. The other functions, based on this notion, are relative to two column partitioned data matrices x and y defining two sets of subsets xi and yj of variables and amount to estimate a link between xi and yj for the pair (xi, yj) relatively to the links associated to all the other pairs.

**Encoding** latin1

**License** GPL

**Repository** CRAN

**Date/Publication** 2009-02-25 08:29:14

## R topics documented:

concor . . . . .	2
concorcano . . . . .	3
concoreg . . . . .	4
concorgm . . . . .	6
concorgmcano . . . . .	7
concorgmreg . . . . .	8
concors . . . . .	9
concorscano . . . . .	10
concorsreg . . . . .	11

svdbip . . . . .	12
svdbip2 . . . . .	14
svdbips . . . . .	15
svdcp . . . . .	16

<b>Index</b>	<b>18</b>
--------------	-----------

---

concor *Relative links of several subsets of variables*

---

## Description

Relative links of several subsets of variables  $Y_j$  with another set  $X$ . SUCCESSIVE SOLUTIONS

## Usage

concor (x, y, py, r)

## Arguments

x, y	are $n \times p$ and $n \times q$ matrices of $p$ and $q$ centered columns
py	is a row vector which contains the numbers $q_i, i=1, \dots, k_y$ , of the $k_y$ subsets $y_i$ of $y$ : $\sum(q_i)=\sum(py)=q$ . py is the partition vector of $y$
r	is the wanted number of successive solutions

## Details

The first solution calculates  $1+k_x$  normed vectors: the vector  $u[:,1]$  of  $R_p$  associated to the  $k_y$  vectors  $v_i[:,1]$ 's of  $R_{q_i}$ , by maximizing  $\sum_i \text{cov}(x * u[,k], y_i * v_i[,k])^2$ , with  $1+k_y$  norm constraints on the axes. A component  $x * u[,k]$  is associated to  $k_y$  partial components  $y_i * v_i[,k]$  and to a global component  $y * V[,k]$ .  $\text{cov}(x * u[,k], y * V[,k])^2 = \sum \text{cov}(x * u[,k], y_i * v_i[,k])^2$ .  $y * V[,k]$  is a global component of the components  $y_i * v_i[,k]$ .

The second solution is obtained from the same criterion, but after replacing each  $y_i$  by  $y_i - y_i * v_i[:,1] * v_i[:,1]'$ . And so on for the successive solutions  $1, 2, \dots, r$ . The biggest number of solutions may be  $r = \inf(n, p, q_i)$ , when the  $x * y_i$ 's are supposed with full rank; then  $r_{\max} = \min(c(\min(py), n, p))$ . For a set of  $r$  solutions, the matrix  $u'X'YV$  is diagonal and the matrices  $u'X'Y_j v_j$  are triangular (good partition of the link by the solutions). concor.m is the svdcp.m function applied to the matrix  $x * y$ .

## Value

list with following components

u	is a $p \times r$ matrix of axes in $R_p$ relative to $x$ ; $u' * u = \text{Identity}$
v	is a $q \times r$ matrix of $k_y$ row blocks $v_i$ ( $q_i \times r$ ) of axes in $R_{q_i}$ relative to $y_i$ ; $v_i' * v_i = \text{Identity}$
V	is a $q \times r$ matrix of axes in $R_q$ relative to $y$ ; $V' * V = \text{Identity}$
cov2	is a $k_y \times r$ matrix; each column $k$ contains $k_y$ squared covariances $\text{cov}(x * u[,k], y_i * v_i[,k])^2$ , the partial measures of link

## References

Lafosse R. & Hanafi M.(1997) Concordance d'un tableau avec K tableaux: Definition de K+1 uples synthetiques. Revue de Statistique Appliquee vol.45,n.4.

## Examples

```
# To make some "GPA" : so, by posing the compromise X = Y,
# "procrustes" rotations to the "compromise X" then are :
# Yj*(vj*u').

x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
co<-concor(x,y,c(3,2,4),2)
((t(x%%co$u[,1])%%y[,1:3])%%co$v[1:3,1])/10)^2;co$cov2[1,1]
t(x%%co$u)%%y%%co$v
```

---

concorcano

*Canonical analysis of several sets with another set*

---

## Description

Relative proximities of several subsets of variables Yj with another set X. SUCCESSIVE SOLUTIONS

## Usage

```
concorcano(x,y,py,r)
```

## Arguments

x	is a n x p matrix of p centered variables
y	is a n x q matrix of q centered variables
py	is a row vector which contains the numbers qi, i=1,...,ky, of the ky subsets yi of y : $\sum_i q_i = \text{sum}(py) = q$ . py is the partition vector of y
r	is the wanted number of successive solutions

## Details

The first solution calculates a standardized canonical component cx[,1] of x associated to ky standardized components cyi[,1] of yi by maximizing  $\sum_i \rho(cx[,1], cy_i[,1])^2$ .

The second solution is obtained from the same criterion, with ky orthogonality constraints for having  $\rho(cy_i[,1], cy_i[,2]) = 0$  (that implies  $\rho(cx[,1], cx[,2]) = 0$ ). For each of the 1+ky sets, the r canonical components are 2 by 2 zero correlated.

The ky matrices (cx)'\*cyi are triangular.

This function uses concor function.

**Value**

list with following components

<code>cx</code>	is $n \times r$ matrix of the $r$ canonical components of $x$
<code>cy</code>	is $n.ky \times r$ matrix. The $ky$ blocks $cy_i$ of the rows $n*(i-1)+1 : n*i$ contain the canonical components relative to $Y_i$
<code>rho2</code>	is a $ky \times r$ matrix; each column $k$ contains $ky$ squared canonical correlations $\rho(cx[,k], cy_i[,k])^2$

**References**

Hanafi & Lafosse (2001) Generalisation de la regression lineaire simple pour analyser la dependance de  $K$  ensembles de variables avec un  $K+1$  eme. Revue de Statistique Appliquee vol.49, n.1

**Examples**

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
ca<-concorcano(x,y,c(3,2,4),2)
diag(t(ca$cx)%*%ca$cy[1:10,])/10)^2
ca$rho2[1,]
```

---

concoreg

*Redundancy of sets  $y_j$  by one set  $x$*

---

**Description**

Regression of several subsets of variables  $Y_j$  by another set  $X$ . SUCCESSIVE SOLUTIONS

**Usage**

```
concoreg(x,y,py,r)
```

**Arguments**

<code>x</code>	is a $n \times p$ matrix of $p$ centered explanatory variables
<code>y</code>	is a $n \times q$ matrix of $q$ centered variables
<code>py</code>	is a row vector which contains the numbers $q_i, i = 1, \dots, ky$ , of the $ky$ subsets $y_i$ of $y : \sum_i q_i = \text{sum}(py) = q$ . <code>py</code> is the partition vector of $y$
<code>r</code>	is the wanted number of successive solutions

## Details

The first solution calculates  $1+ky$  normed vectors: the component  $cx[,1]$  in  $R^n$  associated to the  $ky$  vectors  $vi[,1]$ 's of  $R^{q_i}$ , by maximizing  $varexp1 = \sum_i \rho(cx[,1], y_i * v_i[,1])^2 \text{var}(y_i * v_i[,1])$ , with  $1 + ky$  norm constraints. A explanatory component  $cx[,k]$  is associated to  $ky$  partial explained components  $yi*vi[,k]$  and also to a global explained component  $y*V[,k]$ .  $\rho(cx[,k], y * V[,k])^2 \text{var}(y * V[,k]) = varexp$ . The total explained variance by the first solution is maximal.

The second solution is obtained from the same criterion, but after replacing each  $y_i$  by  $y_i - y_i * v_i[,1] * v_i[,1]'$ . And so on for the successive solutions  $1,2,\dots,r$ . The biggest number of solutions may be  $r = \inf(n, p, q_i)$ , when the matrices  $x' * y_i$  are supposed with full rank. For a set of  $r$  solutions, the matrix  $(cx)' * y * V$  is diagonal : "on average", the explanatory component of one solution is only linked with the components explained by this explanatory, and is not linked with the explained components of the other solutions. The matrices  $(cx)' * y_j * v_j$  are triangular : the explanatory component of one solution is not linked with each of the partial components explained in the following solutions. The definition of the explanatory components depends on the partition vector  $py$  from the second solution.

This function is using `concor` function

## Value

list with following components

<code>cx</code>	the $n \times r$ matrix of the $r$ explanatory components
<code>v</code>	is a $q \times r$ matrix of $ky$ row blocks $v_i$ ( $q_i \times r$ ) of axes in $R_{q_i}$ relative to $y_i$ ; $v_i' * v_i = \text{Id}$
<code>V</code>	is a $q \times r$ matrix of axes in $R_q$ relative to $y$ ; $V' * V = \text{Id}$
<code>varexp</code>	is a $ky \times r$ matrix; each column $k$ contains $ky$ explained variances $\rho(cx[,k], y_i * v_i[,k])^2 \text{var}(y_i * v_i[,k])$

## References

Hanafi & Lafosse (2001) Generalisation de la regression lineaire simple pour analyser la dependance de  $K$  ensembles de variables avec un  $K+1$  eme. *Revue de Statistique Appliquee* vol.49, n.1.

Chessel D. & Hanafi M. (1996) Analyses de la Co-inertie de  $K$  nuages de points. *Revue de Statistique Appliquee* vol.44, n.2. (this ACOM analysis of one multiset is obtained by the command : `concoreg(Y,Y,py,r)`)

## Examples

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
co<-concoreg(x,y,c(3,2,4),2)
((t(co$cx[,1])%*%y[,1:3])%*%co$v[1:3,1])/10)^2;co$varexp[1,1]
t(co$cx)%*%co$cx /10
diag(t(co$cx)%*%y%*%co$V/10)^2
sum(co$varexp[,1]);sum(co$varexp[,2])
```

concorgm

*Analyzing a set of partial links between Xi and Yj***Description**

Analyzing a set of partial links between Xi and Yj, SUCCESSIVE SOLUTIONS

**Usage**

concorgm(x, px, y, py, r)

**Arguments**

x is a n x p matrix of p centered variables

y is a n x q matrix of q centered variables

px is a row vector which contains the numbers pi, i=1,...,kx, of the kx subsets xi of x : sum(pi)=sum(px)=p. px is the partition vector of x

py is the partition vector of y with ky subsets yj, j=1,...,ky

r is the wanted number of successive solutions rmax <= min(min(px),min(py),n)

**Details**

For the first solution,  $\sum_i \sum_j \text{cov2}(x_i * u_i[, 1], y_j * v_j[, 1])$  is the optimized criterion. The second solution is calculated from the same criterion, but with  $x_i - x_i * u_i[, 1] * u_i[, 1]'$  and  $y_j - y_j * v_j[, 1] * v_j[, 1]'$  instead of the kx+ky matrices xi and yj. And so on for the other solutions. When kx=1 (px=p), take concor.m

This function uses the svdbip function.

**Value**

list with following components

u is a p x r matrix of kx row blocks ui (pi x r), the orthonormed partial axes of xi; associated partial components: xi\*ui

v is a q x r matrix of ky row blocks vj (qj x r), the orthonormed partial axes of yj; associated partial components: yj\*vj

cov2 is a kx x ky x r array; for r fixed to k, the matrix contains kxky squared covariances  $\text{cov2}(x_i * u_i[, k], y_j * v_j[, k])^2$ , the partial links between xi and yj measured with the solution k.

**References**

Kissita, Cazes, Hanafi & Lafosse (2004) Deux methodes d'analyse factorielle du lien entre deux tableaux de variables partitionnées. Revue de Statistique Appliquée, Vol 52, n° 3, 73-92.

**Examples**

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
cg<-concorgm(x,c(2,3),y,c(3,2,4),2)
diag(t(x[,1:2]**%cg$u[1:2,])**%y[,1:3]**%cg$v[1:3,])/10)^2
cg$cov2[1,1,]
```

---

concorgmcano                      *Canonical analysis of subsets Yj with subsets Xi*

---

**Description**

Canonical analysis of subsets Yj with subsets Xi. Relative valuations by squared correlations of the proximities of subsets Xi with subsets Yj. SUCCESSIVE SOLUTIONS

**Usage**

```
concorgmcano(x, px, y, py, r)
```

**Arguments**

x	is a n x p matrix of p centered variables
y	is a n x q matrix of q centered variables
px	is a row vector which contains the numbers pi, i=1,...,kx, of the kx subsets xi of $x : \sum_i p_i = \text{sum}(px) = p$ . px is the partition vector of x
py	is the partition vector of y with ky subsets yj, j=1,...,ky
r	is the wanted number of successive solutions $r_{\max} \leq \min(\min(px), \min(py), n)$

**Details**

For the first solution,  $\text{sum}_i \text{sum}_j \text{rho2}(cx_i[,1], cy_j[,1])$  is the optimized criterion. The other solutions are calculated from the same criterion, but with orthogonalities for having two by two zero correlated the canonical components defined for each xi, and also for those defined for each yj. Each solution associates kx canonical components to ky canonical components. When kx = 1 (px=p), take concorecano function

This function uses the concoregm function

**Value**

list with following components

cx	is a n.kx x r matrix of kx row blocks cxi (n x r). Each row block contains r partial canonical components
cy	is a n.ky x r matrix of ky row blocks cyj (n x r). Each row block contains r partial canonical components
rho2	is a kx x ky x r array; for a fixed solution k, rho2[,k] contains kxky squared correlations $\text{rho2}(cx[n * (i - 1) + 1 : n * i, k], cy[n * (j - 1) + 1 : n * j, k])$ , simultaneously calculated between all the yj with all the xi

## References

Kissita G., Analyse canonique generalisee avec tableau de reference generalisee. Thesis, Ceremade Paris 9 Dauphine (2003).

## Examples

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
cc<-concordmcano(x,c(2,3),y,c(3,2,4),2)
diag(t(cc$cx[1:10,])**%cc$cy[1:10,])/10)^2
cc$rho2[1,1,]
```

---

concordmreg

*Regression of subsets Yj by subsets Xi*

---

## Description

Regression of subsets Yj by subsets Xi for comparing all the explanatory-explained pairs (Xi,Yj).  
SUCCESSIVE SOLUTIONS

## Usage

```
concordmreg(x, px, y, py, r)
```

## Arguments

x	is a n x p matrix of p centered variables
y	is a n x q matrix of q centered variables
px	is a row vector which contains the numbers pi, i=1,...,kx, of the kx subsets xi of x : $\sum p_i = \text{sum}(px) = p$ . px is the partition vector of the columns of x.
py	is the partition vector of y with ky subsets yj, j=1,...,ky. $\text{sum}(py) = q$
r	is the wanted number of successive solutions $r_{\max} \leq \min(\min(px), \min(py), n)$

## Details

For the first solution,  $\sum_i \sum_j \rho_2(cx_i[,1], y_j * v_j[,1]) \text{var}(y_j * v_j[,1])$  is the optimized criterion. The second solution is calculated from the same criterion, but with  $y_j - y_j * v_j[,1] * v_j[,1]'$  instead of the matrices yj and with orthogonalities for having two by two zero correlated the explanatory components defined for each matrix xi. And so on for the other solutions. One solution k associates kx explanatory components (in cx[,k]) to ky explained components. When kx =1 (px=p), take concordmreg function

This function uses the concordm function

**Value**

list with following components

<code>cx</code>	is a $n.kx \times r$ matrix of $kx$ row blocks $cxi$ ( $n \times r$ ). Each row block contains $r$ partial explanatory components
<code>v</code>	is a $q \times r$ matrix of $ky$ row blocks $vj$ ( $qj \times r$ ), the orthonormed partial axes of $yj$ ; The components $yj * vj$ are the explained components
<code>varexp</code>	is a $kx \times ky \times r$ array; for a fixed solution $k$ , the matrix <code>varexp[,k]</code> contains $kxky$ explained variances obtained by a simultaneous regression of all the $yj$ by all the $xi$ , so the values $\rho^2(cx[n * (i - 1) + 1 : n * i, k], yj * vj[, k])var(yj * vj[, k])$

**References**

Hanafi & Lafosse (2004) Regression of a multi-set by another based on an extension of the SVD. COMPSTAT'2004 Symposium

**Examples**

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
cr<-concorgmreg(x,c(2,3),y,c(3,2,4),2)
diag(t(cr$cx[1:10,])%*%y[,1:3]%*%cr$v[1:3,])/10)^2
cr$varexp[1,1,]
```

---

concors

*"simultaneous concorm"*

---

**Description**

concorgm with the set of  $r$  solutions simultaneously optimized

**Usage**

```
concors(x, px, y, py, r)
```

**Arguments**

<code>x</code>	is a $n \times p$ matrix of $p$ centered variables
<code>y</code>	is a $n \times q$ matrix of $q$ centered variables
<code>px</code>	is a row vector which contains the numbers $pi$ , $i=1, \dots, kx$ , of the $kx$ subsets $xi$ of $x$ : $\sum_i pi = \text{sum}(px) = p$ . <code>px</code> is the partition vector of $x$
<code>py</code>	is the partition vector of $y$ with $ky$ subsets $yj$ , $j=1, \dots, ky$
<code>r</code>	is the wanted number of successive solutions $r_{\max} \leq \min(\min(px), \min(py), n)$

**Details**

This function uses the `svdbips` function

**Value**

list with following components

u	is a $p \times r$ matrix of $k_x$ row blocks $u_i$ ( $p_i \times r$ ), the orthonormed partial axes of $x_i$ ; associated partial components: $x_i * u_i$
v	is a $q \times r$ matrix of $k_y$ row blocks $v_j$ ( $q_j \times r$ ), the orthonormed partial axes of $y_j$ ; associated partial components: $y_j * v_j$
cov2	is a $k_x \times k_y \times r$ array; for $r$ fixed to $k$ , the matrix contains $k_x k_y$ squared covariances $\text{cov}(x_i * u_i[, k], y_j * v_j[, k])^2$ , the partial links between $x_i$ and $y_j$ measured with the solution $k$

**References**

See svdbips

**Examples**

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
cs<-concors(x,c(2,3),y,c(3,2,4),2)
diag(t(x[,1:2])%*%cs$u[1:2,])%*%y[,1:3]%*%cs$v[1:3,]/10)^2
cs$cov2[1,1,]
```

---

concorscano                    *"simultaneous concorgmcano"*

---

**Description**

concorgmcano with the set of  $r$  solutions simultaneously optimized

**Usage**

```
concorscano(x, px, y, py, r)
```

**Arguments**

x	is a $n \times p$ matrix of $p$ centered variables
y	is a $n \times q$ matrix of $q$ centered variables
px	is a row vector which contains the numbers $p_i$ , $i=1, \dots, k_x$ , of the $k_x$ subsets $x_i$ of $x$ : $\sum_i p_i = \text{sum}(px) = p$ . $px$ is the partition vector of $x$
py	is the partition vector of $y$ with $k_y$ subsets $y_j$ , $j=1, \dots, k_y$
r	is the wanted number of successive solutions $r_{\max} \leq \min(\min(px), \min(py), n)$

**Details**

This function uses the concors function

**Value**

list with following components

`cx` is a  $n.kx \times r$  matrix of  $kx$  row blocks  $cxi$  ( $n \times r$ ). Each row block contains  $r$  partial canonical components

`cy` is a  $n.ky \times r$  matrix of  $ky$  row blocks  $cyj$  ( $n \times r$ ). Each row block contains  $r$  partial canonical components

`rho2` is a  $kx \times ky \times r$  array; for a fixed solution  $k$ , `rho2[,k]` contains  $kxky$  squared correlations  $\rho(cx[n * (i - 1) + 1 : n * i, k], cy[n * (j - 1) + 1 : n * j, k])^2$ , simultaneously calculated between all the  $y_j$  with all the  $x_i$

**References**

See `svdbips`

**Examples**

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
cca<-concorscano(x,c(2,3),y,c(3,2,4),2)
diag(t(cca$cx[1:10,])%*%cca$cy[1:10,])/10)^2
cca$rho2[1,1,]
```

---

`concorsreg`                      *"simultaneous concormreg"*

---

**Description**

`concorgmreg` with the set of  $r$  solutions simultaneously optimized

**Usage**

```
concorsreg(x,px,y,py,r)
```

**Arguments**

`x` is a  $n \times p$  matrix of  $p$  centered variables

`y` is a  $n \times q$  matrix of  $q$  centered variables

`px` is a row vector which contains the numbers  $pi$ ,  $i=1,\dots,kx$ , of the  $kx$  subsets  $x_i$  of  $x$  :  $\sum(pi)=\sum(px)=p$ . `px` is the partition vector of  $x$

`py` is the partition vector of  $y$  with  $ky$  subsets  $y_j$ ,  $j=1,\dots,ky$

`r` is the wanted number of successive solutions  $r_{max} \leq \min(\min(px),\min(py),n)$

**Details**

This function uses the `concors` function

**Value**

list with following components

<code>cx</code>	is a $n.kx \times r$ matrix of $kx$ row blocks $cxi$ ( $n \times r$ ). Each row block contains $r$ partial explanatory components
<code>v</code>	is a $q \times r$ matrix of $ky$ row blocks $vj$ ( $qj \times r$ ), the orthonormed partial axes of $yj$ ; The components $yj * vj$ are the explained components.
<code>varexp</code>	is a $kx \times ky \times r$ array; for a fixed solution $k$ , the matrix <code>varexp[,k]</code> contains $kxky$ explained variances obtained by a simultaneous regression of all the $yj$ by all the $xi$ , so the values $\rho^2(cx[n * (i - 1) + 1 : n * i, k], yj * vj[, k])var(yj * vj[, k])$

**References**

See `svdbips`

**Examples**

```
x<-matrix(runif(50),10,5);y<-matrix(runif(90),10,9)
x<-scale(x);y<-scale(y)
crs<-concorsreg(x,c(2,3),y,c(3,2,4),2)
diag(t(crs$cx[1:10,])%*%y[,1:3]%*%crs$y[1:3,])/10)^2
crs$varexp[1,1,]
```

---

svdbip

*SVD for one bipartitioned matrix x*

---

**Description**

SVD for bipartitioned matrix  $x$ .  $r$  successive Solutions

**Usage**

```
svdbip(x, K, H, r)
```

**Arguments**

<code>x</code>	is a $p \times q$ matrix
<code>K</code>	is a row vector which contains the numbers $pk$ , $k=1,\dots,kx$ , of the partition of $x$ with $kx$ row blocks : $\text{sum}(pk) = p$
<code>H</code>	is a row vector which contains the numbers $qh$ , $h=1,\dots,ky$ , of the partition of $x$ with $ky$ column blocks : $\text{sum}(qh)=q$
<code>r</code>	is the wanted number of successive solutions

## Details

The first solution calculates  $k_x+k_y$  normed vectors:  $k_x$  vectors  $uk[:,1]$  of  $R^{p_k}$  associated to  $k_y$  vectors  $vh[:,1]$ 's of  $R^{q_h}$ , by maximizing  $\sum_k \sum_h (u_k[:,1]' * x_{kh} * v_h[:,1])^2$ , with  $k_x+k_y$  norm constraints. A value  $(u_k[:,1]' * x_{kh} * v_h[:,1])^2$  measures the relative link between  $R^{p_k}$  and  $R^{q_h}$  associated to the block  $x_{kh}$ .

The second solution is obtained from the same criterion, but after replacing each  $x_{kh}$  by  $x_{kh} * v_h * v_h' - uk * uk' * x_{kh} + uk * uk' * x_{kh} * v_h * v_h'$ . And so on for the successive solutions 1,2,...,r . The biggest number of solutions may be  $r = \inf(pk, qh)$ , when the  $x_{kh}$ 's are supposed with full rank; then  $rmax = \min([\min(K), \min(H)])$ .

When  $K=p$  (or  $H=q$ , with  $t(x)$ ), `svdcp` function is better. When  $H=q$  and  $K=p$ , it is the usual `svd` (with squared singular values).

Convergence of algorithm may be not global. So the below proposed initialisation of the algorithm may be not very suitable for some data sets. Several different random initialisations with normed vectors might be considered and the best result then chosen.

## Value

list with following components

u	is a $p \times r$ matrix of $k_x$ row blocks $uk$ ( $pk \times r$ ); $uk' * uk = Identity$ .
v	is a $q \times r$ matrix of $k_y$ row blocks $vh$ ( $qh \times r$ ); $vh' * vh = Identity$
s2	is a $k_x \times k_y \times r$ array; with $r$ fixed, each matrix contains $k_x k_y$ values $(u_h' * x_{kh} * v_k)^2$ , the partial (squared) singular values relative to $x_{kh}$ .

## References

Kissita G., Cazes P., Hanafi M. & Lafosse (2004) Deux methodes d'analyse factorielle du lien entre deux tableaux de variables partitiones. *Revue de Statistique Appliquee*.

## Examples

```
x<-matrix(runif(200),10,20)
s<-svdbip(x,c(3,4,3),c(5,15),3)
zu<-cbind(x[1:3,1:5]**%s$v[1:5,1],x[1:3,6:20]**%s$v[6:20,1])
czu<-svd(zu);
czu$u[,1]**%s$u[1:3,2:3]
czu$u[,1] # is a compromise between the vectors xj*vj[,1],
# orthogonal to the partial vectors uk[,k] relative to the
# following solutions (k>1); (in a same way, the singular
# vectors ui and vj of an usual SVD of x verifies ui'*(x*vj)=0,
#when i is not equal to j)
```

svdbip2

*SVD for bipartitioned matrix x***Description**

SVD for bipartitioned matrix  $x$ .  $r$  successive Solutions. As SVDBIP, but with another algorithm and another initialisation

**Usage**

```
svdbip2(x, K, H, r)
```

**Arguments**

$x$	is a $p \times q$ matrix
$K$	is a row vector which contains the numbers $p_k$ , $k=1, \dots, k_x$ , of the partition of $x$ with $k_x$ row blocks : $\sum_k p_k = p$
$H$	is a row vector which contains the numbers $q_h$ , $h=1, \dots, k_y$ , of the partition of $x$ with $k_y$ column blocks : $\sum q_h = q$
$r$	is the wanted number of successive solutions

**Details**

The first solution calculates  $k_x+k_y$  normed vectors:  $k_x$  vectors  $u_k[:,1]$  of  $R^{p_k}$  associated to  $k_y$  vectors  $v_h[:,1]$ 's of  $R^{q_h}$ , by maximizing  $\sum_k \sum_h (u_k[:,1]' * x_{kh} * v_h[:,1])^2$ , with  $k_x+k_y$  norm constraints. A value  $(u_k[:,1]' * x_{kh} * v_h[:,1])^2$  measures the relative link between  $R^{p_k}$  and  $R^{q_h}$  associated to the block  $x_{kh}$ .

The second solution is obtained from the same criterion, but after replacing each  $x_{kh}$  by  $x_{kh} - u_k * u_k' * x_{kh} + u_k * u_k' * x_{kh} * v_h * v_h'$ . And so on for the successive solutions 1,2,...,r . The biggest number of solutions may be  $r = \inf(p_k, q_h)$ , when the  $x_{kh}$ 's are supposed with full rank; then  $r_{max} = \min([\min(K), \min(H)])$ .

When  $K=p$  (or  $H=q$ , with  $t(x)$ ), svdcp function is better. When  $H=q$  and  $K=p$ , it is the usual svd (with squared singular values).

Convergence of algorithm may be not global. So the below proposed initialisation of the algorithm may be not very suitable for some data sets. Several different random initialisations with normed vectors might be considered and the best result then chosen

**Value**

list with following components

$u$	is a $p \times r$ matrix of $k_x$ row blocks $u_k$ ( $p_k \times r$ ); $u_k' * u_k = \text{Identity}$
$v$	is a $q \times r$ matrix of $k_y$ row blocks $v_h$ ( $q_h \times r$ ); $v_h' * v_h = \text{Identity}$
$s2$	is a $k_x \times k_y \times r$ array; with $r$ fixed, each matrix contains $k_x k_y$ values $(u_h' * x_{kh} * v_k)^2$ , the partial (squared) singular values relative to $x_{kh}$

## References

Kissita G., Analyse canonique generalisee avec tableau de reference generalisee. Thesis, Ceremade Paris 9 Dauphine (2003)

## Examples

```
x<-matrix(runif(200),10,20)
s2<-svdbip2(x,c(3,4,3),c(5,5,10),3);s2$s2
s1<-svdbip(x,c(3,4,3),c(5,5,10),3);s1$s2
```

---

 svdbips

*SVD for bipartitioned matrix x*


---

## Description

SVD for bipartitioned matrix x. SIMULTANEOUS SOLUTIONS. ("simultaneous svdbip")

## Usage

```
svdbips(x, K, H, r)
```

## Arguments

x	is a p x q matrix
K	is a row vector which contains the numbers pk, k=1,...,kx, of the partition of x with kx row blocks : $\sum_k p_k = p$
H	is a row vector which contains the numbers qh, h=1,...,ky, of the partition of x with ky column blocks : $\sum_h q_h = q$
r	is the wanted number of solutions

## Details

One set of r solutions is calculated by maximizing  $\sum_i \sum_k \sum_h (u_k[,i]' * x_{kh} * v_h[,i])^2$ , with kx+ky orthonormality constraints (for each uk and each vh). For each fixed r value, the solution is totally new (does'nt consist to complete a previous calculus of one set of r-1 solutions). rmax=min([min(K),min(H)]). When r=1, it is svdbip (thus it is svdcp when r=1 and kx=1).

Convergence of algorithm may be not global. So the below proposed initialisation of the algorithm may be not very suitable for some data sets. Several different random initialisations with normed vectors might be considered and the best result then choosen....

## Value

list with following components

u	is a p x r matrix of kx row blocks uk (pk x r); uk'*uk = Identity
v	is a q x r matrix of ky row blocks vh (qh x r); vh'*vh = Identity
s2	is a kx x ky x r array; for a fixed solution k, each matrix s2[:,k] contains kxky values $(u'_h * x_{kh} * v_k)^2$ , the "partial (squared) singular values" relative to $x_{kh}$ .

## References

Lafosse R. & Ten Berge J. A simultaneous CONCOR method for the analysis of two partitioned matrices. submitted.

## Examples

```
x<-matrix(runif(200),10,20)
s1<-svdbip(x,c(3,4,3),c(5,5,10),2);sum(sum(sum(s1$s2)))
ss<-svdbips(x,c(3,4,3),c(5,5,10),2);sum(sum(sum(ss$s2)))
```

---

 svdcp

*SVD for a Column Partitioned matrix x*


---

## Description

SVD for a Column Partitioned matrix x. r global successive solutions

## Usage

```
svdcp(x, H, r)
```

## Arguments

x is a p x q matrix  
 H is a row vector which contains the numbers  $q_i$ ,  $i=1,\dots,k_x$ , of the partition of x with  $k_x$  column blocks  $x_i$  :  $\sum q_i = q$ .  
 r is the wanted number of successive solutions.

## Details

The first solution calculates  $1+k_x$  normed vectors: the vector  $u[,1]$  of  $R^p$  associated to the  $k_x$  vectors  $v_i[,1]$ 's of  $R^{q_i}$ . by maximizing  $\sum_i (u[,1]' * x_i * v_i[,1])^2$ , with  $1+k_x$  norm constraints. A value  $(u[,1]' * x_i * v_i[,1])^2$  measures the relative link between  $R^p$  and  $R^{q_i}$  associated to  $x_i$ . It corresponds to a partial squared singular value notion, since  $\sum_i (u[,1]' * x_i * v_i[,1])^2 = s^2$ , where s is the usual first singular value of x.

The second solution is obtained from the same criterion, but after replacing each  $x_i$  by  $x_i - x_i * v_i[,1] * v_i[,1]'$ . And so on for the successive solutions 1,2,...,r. The biggest number of solutions may be  $r = \inf(p, q_i)$ , when the  $x_i$ 's are supposed with full rank; then  $r_{max} = \min([\min(H), p])$ .

## Value

list with following components

u is a p x r matrix;  $u' * u = \text{Identity}$   
 v is a q x r matrix of  $k_x$  row blocks  $v_i$  ( $q_i$  x r);  $v_i' * v_i = \text{Identity}$   
 s2 is a  $k_x$  x r matrix; each column k contains  $k_x$  values  $(u[,k]' * x_i * v_i[,k])^2$ , the partial (squared) singular values relative to  $x_i$

**References**

Lafosse R. & Hanafi M.(1997) Concordance d'un tableau avec K tableaux: Definition de K+1 uples synthetiques. Revue de Statistique Appliquee vol.45,n.4.

**Examples**

```
x<-matrix(runif(200),10,20)
s<-svdcp(x,c(5,5,10),1)
ss<-svd(x);ss$d[1]^2
sum(s$s2)
```

# Index

concor, [2](#)  
concorcano, [3](#)  
concoreg, [4](#)  
concorgm, [6](#)  
concorgmcano, [7](#)  
concorgmreg, [8](#)  
concors, [9](#)  
concorscano, [10](#)  
concorsreg, [11](#)

svdbip, [12](#)  
svdbip2, [14](#)  
svdbips, [15](#)  
svdcp, [16](#)