Package ‘vcvComp’

June 28, 2019

Title Comparison of Variance - Covariance Patterns

Version 1.0.1

Date 2019-06-24

Description Comparison of variance - covariance patterns using relative principal component analysis (relative eigenanalysis). Also provides functions to compute group covariance matrices, distance matrices, and perform proportionality tests. A working sample on the body shape of cichlid fishes is included.

Depends R (>= 3.6.0)

License GPL-3

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, geomorph, rgl, scatterplot3d, RRPP

VignetteBuilder knitr

RoxygenNote 6.1.1

NeedsCompilation no

Author Anne Le Maitre [aut, cre],
Philipp Mitteroecker [aut]

Maintainer Anne Le Maitre <anne.le.maitre@univie.ac.at>

Repository CRAN

Date/Publication 2019-06-28 10:10:03 UTC

R topics documented:

cov.B 2
cov.group 3
cov.W 4
eigen.test 5
euclidean.dist 6
mat.sq.dist 7
minv 8
pr.coord 9

prop.vcv.test 11
relative.eigen 12
relGV.multi 13
scaling.BW 14
Tropheus 15

Index **17**

`cov.B` *Between-group covariance matrix*

Description

Computes the between-group covariance matrix. The effect of sexual dimorphism can be removed by using, for each group, the average of the mean of males and the mean of females.

Usage

```
cov.B(X, groups, sex = NULL, center = FALSE, weighted = FALSE)
```

Arguments

`X`	a data matrix with variables in columns and group names as row names
`groups`	a character / factor vector containing grouping variable
`sex`	NULL (default). A character / factor vector containing sex variable, to remove sexual dimorphism by averaging males and females in each group
`center`	either a logical value or a numeric vector of length equal to the number of columns of X
`weighted`	logical. Should the between-group covariance matrix be weighted?

Value

The between-group covariance matrix

See Also

`cov`, `cov.wt`

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
```

```
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Between-group covariance matrix for all populations
B <- cov.B(proc.coord, groups = Tropheus$POP.ID)

# Between-group covariance matrix for all populations, pooled by sex
B.mf <- cov.B(proc.coord, groups = Tropheus$POP.ID, sex = Tropheus$Sex)
```

`cov.group` *Group covariance matrices*

Description

Computes the covariance matrix of each group. The effect of sexual dimorphism can be removed by using, for each group, the average of the covariance matrix of males and the covariance matrix of females.

Usage

```
cov.group(X, groups, sex = NULL, use = "everything")
```

Arguments

`X` a data matrix with variables in columns and group names as row names

`groups` a character / factor vector containing grouping variable

`sex` NULL (default). A character / factor vector containing sex variable, to remove sexual dimorphism by averaging males and females in each group

`use` an optional character string giving a method for computing covariances in the presence of missing values. This must be (an abbreviation of) one of the strings "everything", "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs".

Value

A (p x p x m) array of covariance matrices, where p is the number of variables and m the number of groups.

See Also

`cov` and `scale`

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Covariance matrix of each population
S.phen.pop <- cov.group(proc.coord, groups = Tropheus$POP.ID)

# Covariance matrix of each population, pooled by sex
S.phen.pooled <- cov.group(proc.coord, groups = Tropheus$POP.ID, sex = Tropheus$Sex)
```

---

cov.W *Within-group covariance matrix*

Description

Computes the pooled within-group covariance matrix. The effect of sexual dimorphism can be removed by using, for each group, the average of the covariance matrix of males and the covariance matrix of females.

Usage

```
cov.W(X, groups, sex = NULL)
```

Arguments

X	a data matrix with variables in columns and group names as row names
groups	a character / factor vector containing grouping variable
sex	NULL (default). A character / factor vector containing sex variable, to remove sexual dimorphism by averaging males and females in each group

Value

The pooled within-group covariance matrix

See Also

cov

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Within-group covariance matrix for all populations
W <- cov.W(proc.coord, groups = Tropheus$POP.ID)

# Within-group covariance matrix for all populations, pooled by sex
W.mf <- cov.W(proc.coord, groups = Tropheus$POP.ID, sex = Tropheus$Sex)
```

eigen.test *Difference test for successive relative eigenvalues*

Description

Tests the difference between two successive relative eigenvalues

Usage

```
eigen.test(n, relValues)
```

Arguments

n	the sample size(s), given as a number or a vector of length 2
relValues	a vector of relative eigenvalues

Value

The P-values for the test of difference between successive eigenvalues

References

Mardia KV, Kent JT, Bibby JM (1979) *Multivariate analysis.* Academic Press, London.

See Also

relative.eigen for the computation of relative eigenvalues,

pchisq for Chi-squared distribution

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Data reduction
phen.pca <- prcomp(proc.coord, rank. = 5, tol = sqrt(.Machine$double.eps))
pc.scores <- phen.pca$x

# Covariance matrix of each population
S.phen.pop <- cov.group(pc.scores, groups = Tropheus$POP.ID)

# Relative PCA = relative eigenanalysis between 2 covariance matrices
# (population IKA1 relative to IKS5)
relEigen.a1s5 <- relative.eigen(S.phen.pop[, , "IKA1"], S.phen.pop[, , "IKS5"])

# Test of the difference between 2 successives eigenvalues
# of the covariance matrix of IKA1 relative to IKS5
eigen.test(n = c(71, 75), relValues = relEigen.a1s5$relValues)  # 71 and 75 are the sample sizes
```

euclidean.dist *Euclidean distance between two covariance matrices*

Description

Computes the Euclidean distance (Frobenius norm) between two variance-covariance matrices of same dimensions

Usage

```
euclidean.dist(S1, S2)
```

Arguments

S1	a variance-covariance matrix
S2	a variance-covariance matrix

Value

Euclidean distance between S1 and S2 following Dryden et al. (2009).

References

Dryden IL, Koloydenko A, Zhou D (2009) Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *The Annals of Applied Statistics 3*:1102-1123. https://projecteuclid.org/euclid.aoas/1254773280

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Data reduction
phen.pca <- prcomp(proc.coord, rank. = 5, tol = sqrt(.Machine$double.eps))
pc.scores <- phen.pca$x

# Covariance matrix of each population
S.phen.pop <- cov.group(pc.scores, groups = Tropheus$POP.ID)

# Euclidean distance between the covariance matrices of 2 populations
# (IKA1 relative to IKS5)
dist.a1s5 <- euclidean.dist(S.phen.pop[, , "IKA1"], S.phen.pop[, , "IKS5"])
```

`mat.sq.dist` *Squared distance matrix*

Description

Computes the squared distance matrix of a set of covariance matrices

Usage

```
mat.sq.dist(Sm, dist. = "Riemannian", method = 0, pa = 0)
```

Arguments

`Sm`	a (p x p x m) array of covariance matrices, where p is the number of variables and m the number of groups.
`dist.`	"Riemannian" or "Euclidean"
`method`	an integer for the method of matrix inversion
`pa`	an integer for the parameter of matrix inversion

Value

The matrix of squared Riemannian or Euclidean distances

See Also

See `minv` for the method and the parameter used for the matrix inversion

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Data reduction
phen.pca <- prcomp(proc.coord, rank. = 5, tol = sqrt(.Machine$double.eps))
pc.scores <- phen.pca$x

# Covariance matrix of each population
S.phen.pop <- cov.group(pc.scores, groups = Tropheus$POP.ID)

# Squared Riemannian distance matrix of the covariance matrices of all populations
eigen.phen.r <- mat.sq.dist(S.phen.pop, dist. = "Riemannian")

# Squared Euclidean distance matrix of the covariance matrices of all populations
eigen.phen.e <- mat.sq.dist(S.phen.pop, dist. = "Euclidean")
```

`minv` *Matrix pseudoinverse*

Description

Computes the inverse or the pseudoinverse of a matrix

Usage

```
minv(M, method = 0, pa = 0)
```

Arguments

`M` a numeric matrix (square matrix)

`method` an integer for the method of inversion. If method = 0, only the nonzero eigenvalues are kept; if method = 1, only the eigenvalues above a threshold are kept; if method = 2, only the several first eigenvalues are kept; if method = 3, a Tikhonov regularization (= ridge regression) is performed.

`pa` an integer for the parameter of inversion. If method = 1, pa is the threshold below which the eigenvalues are not kept; if method = 2, pa is an positive integer number corresponding to number of eigenvalues that are kept; if method = 3, pa is the scaling factor for the identity matrix

Value

A numeric matrix corresponding to the pseudoinverse of M

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Covariance matrix of each population
S.phen.pop <- cov.group(proc.coord, groups = Tropheus$POP.ID)

# Pseudo-inversion of a square matrix (covariance matrix of the population IKS5)
S2 <- S.phen.pop[, , "IKS5"]
invS2 <- minv(S2, method = 0, pa = 0)  # Pseudoinverse keeping non-zero eigenvalues
invS2 <- minv(S2, method = 1, pa = 10^-8)  # Pseudoinverse keeping eigenvalues above 10^-8
invS2 <- minv(S2, method = 2, pa = 5)  # Pseudoinverse keeping the first five eigenvalues
invS2 <- minv(S2, method = 3, pa = 0.5)  # Ridge regression with Tikhonov factor of 0.5
```

pr.coord *Principal coordinates ordination*

Description

Performs a principal coordinates analysis of a distance matrix

Usage

```
pr.coord(V)
```

Arguments

`V` a square distance matrix

Value

A list containing the following named components:

`k` the number of groups (value)

`vectors` the eigenvectors of the centered inner product matrix (matrix)

`values` the eigenvalues of the centered inner product matrix (vector)

`PCoords` the principal coordinates = scaled eigenvectors (matrix)

`Variance` a dataframe containing the following named variables:

- **eigenvalues** eigenvalues of the centered inner product matrix
- **variance** variance of each principal coordinate
- **exVar** proportion of the total variation accounted by each principal coordinate
- **cumVar** cumulative proportion of the total variation accounted by principal coordinate

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Data reduction
phen.pca <- prcomp(proc.coord, rank. = 5, tol = sqrt(.Machine$double.eps))
pc.scores <- phen.pca$x

# Covariance matrix of each population
S.phen.pop <- cov.group(pc.scores, groups = Tropheus$POP.ID)

# Squared distance matrix of the covariance matrices of all populations
eigen.phen.pop <- mat.sq.dist(S.phen.pop, dist. = "Riemannian")  # Riemannian distances

# Ordination of the squared distance matrix
prcoa.pop <- pr.coord(eigen.phen.pop)

# Visualization
plot(prcoa.pop$PCoords[, 1], prcoa.pop$PCoords[, 2])
abline(h = 0) ; abline(v = 0)
text(prcoa.pop$PCoords[, 1], prcoa.pop$PCoords[, 1], labels = rownames(prcoa.pop$PCoords))
```

`prop.vcv.test` *Proportionality test of two variance-covariance matrices*

Description

Tests the proportionality of two variance-covariance matrices

Usage

```
prop.vcv.test(n, S1, S2, method = 0, pa = 0)
```

Arguments

`n`	the sample size(s), given as a number or a vector of length 2
`S1`	a variance-covariance matrix
`S2`	a variance-covariance matrix
`method`	an integer for the method of matrix inversion (see function 'minv')
`pa`	an integer for the parameter of matrix inversion (see function 'minv')

Value

The P-value for the test of proportionality between two variance-covariance matrices

References

Mardia KV, Kent JT, Bibby JM (1979) *Multivariate analysis*. Academic Press, London.

See Also

`relative.eigen` for the computation of relative eigenvalues,

`minv` for the method and the parameter used for the matrix inversion,

`pchisq` for Chi-squared distribution

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Data reduction
```

```
phen.pca <- prcomp(proc.coord, rank. = 5, tol = sqrt(.Machine$double.eps))
pc.scores <- phen.pca$x

# Covariance matrix of each population
S.phen.pop <- cov.group(pc.scores, groups = Tropheus$POP.ID)

# Maximum likelihood test of proportionality between 2 covariance matrices
# (IKA1 relative to IKS5) - 71 and 75 are the sample sizes
prop.vcv.test(n = c(71, 75), S.phen.pop[,,"IKA1"], S.phen.pop[,,"IKS5"])
```

relative.eigen *Relative eigenanalysis*

Description

Computes the Riemanian distance between two variance-covariance matrices of same dimensions and the relative eigenvectors and eigenvalues of S1 with respect to S2

Usage

```
relative.eigen(S1, S2, method = 0, pa = 0)
```

Arguments

`S1`	a variance-covariance matrix
`S2`	a variance-covariance matrix
`method`	an integer for the method of matrix inversion (see function 'minv')
`pa`	an integer for the parameter of matrix inversion (see function 'minv')

Value

A list containing the following named components:

`relValues`	the vector of relative eigenvalues
`relVectors`	the matrix of relative eigenvectors
`distCov`	the distance between the two covariance matrices
`relGV`	the product of the nonzero relative eigenvalues = the ratio of the generalized variances. The generalized variance corresponds to the determinant of the covariance matrix.
`logGV`	the log ratio of the generalized variances
`q`	the number of nonzero eigenvalues

References

Bookstein F, Mitteroecker P (2014) Comparing covariance matrices by relative eigenanalysis, with applications to organismal biology. *Evolutionary Biology 41*: 336-350. https://doi.org/10.1007/s11692-013-9260-5

See Also

See minv for the method and the parameter used for the matrix inversion

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Data reduction
phen.pca <- prcomp(proc.coord, rank. = 5, tol = sqrt(.Machine$double.eps))
pc.scores <- phen.pca$x

# Covariance matrix of each population
S.phen.pop <- cov.group(pc.scores, groups = Tropheus$POP.ID)

# Relative PCA = relative eigenanalysis between 2 covariance matrices
# (population IKA1 relative to IKS5)
relEigen.a1s5 <- relative.eigen(S.phen.pop[, , "IKA1"], S.phen.pop[, , "IKS5"])
```

---

relGV.multi *Ratio of generalized variances*

Description

Computes the (log-transformed) ratios of the generalized variances of a set of covariance matrices

Usage

```
relGV.multi(Sm, logGV = TRUE)
```

Arguments

Sm	a (p x p x m) array of covariance matrices, where p is the number of variables and m the number of groups.
logGV	a logical argument to indicate if the ratios should be log-transformed

Value

The matrix of the (log-transformed) ratios of the generalized variances. For each row, the ratio corrresponds to the group of the row relative to the group of a column.

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Data reduction
phen.pca <- prcomp(proc.coord, rank. = 5, tol = sqrt(.Machine$double.eps))
pc.scores <- phen.pca$x

# Covariance matrix of each population
S.phen.pop <- cov.group(pc.scores, groups = Tropheus$POP.ID)

# Ratio of the generalized variances of 2 populations (IKA1 and IKS5)
relGV.multi(S.phen.pop[, , c("IKA1", "IKS5")], logGV = FALSE)
```

`scaling.BW` *Scaling factor between two matrices*

Description

Computes the maximum-likelihood estimate of the scaling factor between two proportional covariance matrices. Note that the scaling factor between the two matrices is equal to the arithmetic mean of their relative eigenvalues.

Usage

```
scaling.BW(S1, S2, method = 0, pa = 0)
```

Arguments

`S1`	a variance-covariance matrix
`S2`	a variance-covariance matrix
`method`	an integer for the method of matrix inversion (see function ’minv’)
`pa`	an integer for the parameter of matrix inversion (see function ’minv’)

Value

The scaling factor between the two matrices.

See Also

See minv for the method and the parameter used for the matrix inversion

Examples

```
# Data matrix of 2D landmark coordinates
data("Tropheus")
PHEN <- as.matrix(Tropheus[which(names(Tropheus) == "X1"):which(names(Tropheus) == "Y19")])

# Procrustes superimposition
library("geomorph")
PHEN_array <- arrayspecs(PHEN, p = 19, k = 2)
phen.gpa <- gpagen(PHEN_array, print.progress = FALSE)
proc.coord <- two.d.array(phen.gpa$coords)

# Between-group (B) and within-group (W) covariance matrices for all populations
B <- cov.B(proc.coord, groups = Tropheus$POP.ID, sex = Tropheus$Sex)
W <- cov.W(proc.coord, groups = Tropheus$POP.ID, sex = Tropheus$Sex)

# ML estimate of the scaling factor between B and W
sc <- scaling.BW(B, W)

# Scaling of B to W
Bsc <- B / sc
```

`Tropheus` *vcvComp dataset*

Description

A data frame of 723 observations of 57 variables extracted from a freely available dataset, downloaded from the Dryad digital repository (https://doi.org/10.5061/dryad.fc02f). The observations correspond to cichlid fishes of the species *Tropheus moorii* (color morphs 'Kaiser' and 'Kirschfleck') and *T. polli* collected from eight locations of Lake Tanganyika (Kerschbaumer et al., 2014). The main numerical variables provided are the 2D Cartesian coordinates of 19 landmarks quantifying the external body morphology of adult fishes and the genotypes for 6 microsatellite markers.

Usage

```
data(Tropheus)
```

Format

A data frame with 723 rows and 57 variables

Details

- **List_TropheusData_ID** Specimen ID
- **Extractionnr.** Extraction number for genomic DNA
- **G** Group number
- **POP.ID** Population Id
- **Sex** Sex
- **Allo.Symp** Allopatric or sympatric population
- **X1 ... Y19** Cartesian coordinates of 19 landmarks
- **Pzep3_1 ... UME003_2** Genotype for 6 microsatellite markers

References

Kerschbaumer M, Mitteroecker P, Sturmbauer C (2014) Evolution of body shape in sympatric versus non-sympatric Tropheus populations of Lake Tanganyika. *Heredity 112(2)*: 89–98. https://doi.org/10.1038/hdy.2013.78

Kerschbaumer M, Mitteroecker P, Sturmbauer C (2013) Data from: Evolution of body shape in sympatric versus non-sympatric Tropheus populations of Lake Tanganyika. *Dryad Digital Repository*. https://doi.org/10.5061/dryad.fc02f

Index

cov, *2–4*
cov.B, 2
cov.group, 3
cov.W, 4
cov.wt, *2*

eigen.test, 5
euclidean.dist, 6

mat.sq.dist, 7
minv, *8*, 8, *11*, *13*, *15*

pchisq, *5*, *11*
pr.coord, 9
prop.vcv.test, 11

relative.eigen, *5*, *11*, 12
relGV.multi, 13

scale, *3*
scaling.BW, 14

Tropheus, 15