Package 'HDTD'

October 16, 2018

Type Package

Title Statistical Inference about the Mean Matrix and the Covariance Matrices in High-Dimensional Transposable Data (HDTD)

Version 1.14.0

Depends R (>= 3.4)

Suggests knitr, markdown

Imports stats, Rcpp (>= 0.12.13)

LinkingTo Rcpp, RcppArmadillo

Description

Characterization of intra-individual variability using physiologically relevant measurements provides important insights into fundamental biological questions ranging from cell type identity to tumor development. For each individual, the data measurements can be written as a matrix with the different subsamples of the individual recorded in the columns and the different phenotypic units recorded in the rows. Datasets of this type are called high-dimensional transposable data. The HDTD package provides functions for conducting statistical inference for the mean relationship between the row and column variables and for the covariance structure within and between the row and column variables.

License GPL-3

biocViews DifferentialExpression, Genetics, GeneExpression, Microarray, Sequencing, StatisticalMethod, Software

VignetteBuilder knitr

URL http://github.com/AnestisTouloumis/HDTD

BugReports http://github.com/AnestisTouloumis/HDTD/issues

LazyData true

NeedsCompilation no

RoxygenNote 6.0.1

git_url https://git.bioconductor.org/packages/HDTD

git_branch RELEASE_3_7

git_last_commit 8e7323c

git_last_commit_date 2018-04-30

Date/Publication 2018-10-15

Author Anestis Touloumis [cre, aut],
John C. Marioni [aut],
Simon Tavar\{}'{e} [aut]

Maintainer Anestis Touloumis <A.Touloumis@brighton.ac.uk>

R topics documented:

`HDTD-package` *Estimation and Hypothesis Testing in High-Dimensional Transposable Data*

Description

The package HDTD offers functions to estimate and test the matrix parameters of transposable data in high-dimensional settings.

Details

The term transposable data refers to datasets that are structured in a matrix form such that both the rows and columns correspond to variables of interest. For example, consider microarray studies in genetics where multiple RNA samples across different tissues are available per subject. In this case, a data matrix can be created with row variables the genes, column variables the tissues and measurements the corresponding expression levels.

The function `meanmat.hat` estimates the mean matrix of the transposable data.

The mean relationship of the row and column variables can be tested using the function `meanmat.ts`. The implemented test is nonparametric and not seriously restricted by the dependence structure among and/or between the row and column variables. See *Touloumis et al. (2015)* for more details.

The function `covmat.hat` provides Stein-type shrinkage estimators for the row covariance matrix and/or for the column covariance matrix under a matrix-variate normal model. See *Touloumis et al. (2016)* for more details.

The sphericity and identity hypothesis for the row or column covariance matrix can be tested using the function `covmat.ts`. Both tests are nonparametric, i.e., they do not rely on a normality assumption. See *Touloumis et al. (2017)* for more details.

There are three utility functions that allow the user to change to interchange the role of row and column variables (`transposedata`), to center the transposable data (`centerdata`) or to rearrange the order of the row and/or column variables (`orderdata`).

Author(s)

Anestis Touloumis, John Marioni, Simon Tavare.

Maintainer: Anestis.Touloumis <A.Touloumis@brighton.ac.uk>

References

Touloumis, A., Tavare, S. and Marioni, J. C. (2015) Testing the Mean Matrix in High-Dimensional Transposable Data. *Biometrics* **71**, 157–166

Touloumis, A., Marioni, J. C. and Tavare, S. (2016) HDTD: Analyzing multi-tissue gene expression data. *Bioinformatics* **32**, 2193–2195.

Touloumis, A., Marioni, J. C. and Tavare, S. (2017) Hypothesis Testing for the Covariance Matrix in High-Dimensional Transposable Data with Kronecker Product Dependence Structure.

Examples

```
data(VEGFmouse)
## The sample mean matrix.
sample_mean <- meanmat.hat(datamat = VEGFmouse, N = 40)
sample_mean
## Testing conservation of the overall gene expression across tissues.
tissues_mean_test <- meanmat.ts(datamat = VEGFmouse, N = 40, group.sizes = 9)
tissues_mean_test
# Estimating the gene and column covariance matrices.
est_cov_mat <- covmat.hat(datamat = VEGFmouse, N = 40)
est_cov_mat
## Hypothesis tests for the covariance matrix of the genes (rows).
genes_cov_test <- covmat.ts(datamat = VEGFmouse, N = 40)
genes_cov_test
## Hypothesis tests for the covariance matrix of the tissues (columns).
tissues_cov_test <- covmat.ts(datamat = VEGFmouse, N = 40, voi = 'columns')
tissues_cov_test
```

---

centerdata *Centering Transposable Data*

Description

This function centers the transposable data around their sample mean matrix.

Usage

```
centerdata(datamat, N)
```

Arguments

`datamat`	numeric matrix containing the transposable data.
`N`	positive integer number indicating the sample size, e.g., the number of subjects.

Details

It is assumed that there are `nrow(datamat)` row variables and `ncol(datamat)/N` column variables in `datamat`. Further, `datamat` should be written in such a way that every `ncol(datamat)/N` consecutive columns belong to the same subject and the order of the column variables in each block is preserved across subjects.

Value

Returns a matrix of the same size as datamat.

Author(s)

Anestis Touloumis

See Also

covmat.hat and covmat.ts.

Examples

```
data(VEGFmouse)
## Centering the VEGF dataset around the sample mean matrix.
VEGFcen <- centerdata(datamat = VEGFmouse, N = 40)
```

---

covmat.hat *Estimation of the Row and of the Column Covariance Matrices.*

Description

This function provides the row and/or column covariance matrix estimators.

Usage

```
covmat.hat(datamat, N, shrink = "both", centered = FALSE, voi = "both")
```

Arguments

datamat	numeric matrix containing the transposable data.
N	positive integer number indicating the sample size, i.e., the number of subjects.
shrink	character indicating if shrinkage estimation should be performed. Options include 'rows', 'columns', 'both' and 'none'.
centered	logical indicating if the transposable data are centered. Options include TRUE or FALSE.
voi	character indicating if the row, column or both covariance matrices should be printed. Options include 'rows', 'columns' and 'both'.

Details

It is assumed that there are nrow(datamat) row variables and ncol(datamat)/N column variables in datamat. Further, datamat should be written in such a way that every ncol(datamat)/N consecutive columns belong to the same subject and the order of the column variables in each block is preserved across subjects.

For identifiability reasons, the trace of the row covariance matrix is set equal to its dimension. If you want to place the equivalent restriction on the column covariance matrix, interchange the role of row and column variables by utilizing the function transposedata.

Value

Returns a list with components:

rows.covmat	the estimated row covariance matrix.
rows.intensity	the estimated row intensity.
cols.covmat	the estimated column covariance matrix.
cols.intensity	the estimated column intensity.
N	the sample size.
n.rows	the number of row variables.
n.cols	the number of column variables.
shrink	character indicating if shrinkage estimation was performed.
centered	logical indicating if the transposable data were centered.

Author(s)

Anestis Touloumis

References

Touloumis, A., Marioni, J. C. and Tavare, S. (2016) HDTD: Analyzing multi-tissue gene expression data, *Bioinformatics* **32**, 2193–2195.

Examples

```
data(VEGFmouse)
# Estimating the gene and tissue covariance matrices.
est_cov_mat <- covmat.hat(datamat = VEGFmouse, N = 40)
est_cov_mat
```

---

covmat.ts *Nonparametric Tests for the Row or Column Covariance Matrix*

Description

Testing the sphericity, identity and diagonality hypotheses for the row or column covariance matrix.

Usage

```
covmat.ts(datamat = datamat, N = N, voi = "rows", centered = FALSE)
```

Arguments

datamat	numeric matrix containing the transposable data.
N	positive integer number indicating the sample size, i.e., the number of subjects.
voi	character indicating if the test should be applied on the row or column covariance matrix. Options include 'rows' or 'columns'.
centered	logical indicating if the transposable data are centered. Options include TRUE or FALSE.

Details

It is assumed that there are `nrow(datamat)` row variables and `ncol(datamat)/N` column variables in `datamat`. Further, `datamat` should be written in such a way that every `ncol(datamat)/N` consecutive columns belong to the same subject and the order of the column variables in each block is preserved across subjects.

The tests are nonparametric and thus robust to some departures from the matrix-variate normal model.

Value

It returns a list with components:

`diagonality.ts`	a list containing the test statistic and p-value of the diagonality hypothesis test.
`sphericity.ts`	a list containing the test statistic and p-value of the sphericity hypothesis test.
`identity.ts`	a list containing the test statistic and p-value of the identity hypothesis test.
`N`	the sample size.
`n.rows`	the number of row variables.
`n.cols`	the number of column variables.
`variables`	character indicating if the tests were applied to the row or column covariance matrix.
`centered`	logical indicating if the transposable data were centered.

Author(s)

Anestis Touloumis

References

Touloumis, A., Marioni, J.C. and Tavare, S. (2017). Hypothesis Testing for the Covariance Matrix in High-Dimensional Transposable Data with Kronecker Product Dependence Structure.

Examples

```
data(VEGFmouse)
## Hypothesis tests for the covariance matrix of the genes (rows).
genes_cov_test <- covmat.ts(datamat = VEGFmouse, N = 40)
genes_cov_test
## Hypothesis tests for the covariance matrix of the tissues (columns).
tissues_cov_test <- covmat.ts(datamat = VEGFmouse, N = 40, voi = 'columns')
tissues_cov_test
```

meanmat.hat *Estimation the Mean Matrix*

Description

This function estimates the mean matrix.

Usage

```
meanmat.hat(datamat, N, group.sizes = NULL, group.vars = NULL)
```

Arguments

datamat	numeric matrix containing the transposable data.
N	positive integer number indicating the sample size, i.e., the number of subjects.
group.sizes	numeric vector indicating the size of the row or column groups that share the same mean vector. It should be used only when group.vars='rows' or 'columns'.
group.vars	character indicating that the mean matrix can be simplified over the row or column variables. Options include 'rows' or 'columns'.

Details

It is assumed that there are nrow(datamat) row variables and ncol(datamat)/N column variables in datamat. Further, datamat should be written in such a way that every ncol(datamat)/N consecutive columns belong to the same subject and the order of the column variables in each block is preserved across subjects.

Value

Returns a list with components:

estmeanmat	the estimated mean matrix.
N	the sample size.
n.rows	the number of row variables.
n.cols	the number of column variables.

Author(s)

Anestis Touloumis

References

Touloumis, A., Marioni, J. C. and Tavare, S. (2016) HDTD: Analyzing multi-tissue gene expression data. *Bioinformatics* **32**, 2193–2195.

Examples

```
data(VEGFmouse)
## The sample mean matrix of the VEGF mouse data.
sample_mean <- meanmat.hat(datamat = VEGFmouse, N = 40)
sample_mean
sample_mean$estmeanmat
```

meanmat.ts *Nonparametric Tests for the Mean Matrix*

Description

This function performs hypothesis testing for the mean matrix.

Usage

```
meanmat.ts(datamat, N, group.sizes, voi = "columns")
```

Arguments

datamat	numeric matrix containing the transposable data.
N	positive integer number indicating the sample size, i.e., the number of subjects.
group.sizes	numeric vector indicating the group sizes under the null hypothesis.
voi	character indicating if the test will be applied to the row or column variables. Options include 'rows' or 'columns'.

Details

It is assumed that there are nrow(datamat) row variables and ncol(datamat)/N column variables in datamat. Further, datamat should be written in such a way that every ncol(datamat)/N consecutive columns belong to the same subject and the order of the column variables in each block is preserved across subjects.

Value

Returns a list with components:

statistic	the value of the test statistic.
p.value	the corresponding p-value.
voi	the set of variables that the test was applied to.
n.groups	the number of groups under the null hypothesis.
group.sizes	the size of each group under the null hypothesis.
N	the sample size.
n.rows	the number of row variables.
n.cols	the number of column variables.

Author(s)

Anestis Touloumis

References

Touloumis, A., Tavare, S. and Marioni, J. C. (2015) Testing the Mean Matrix in High-Dimensional Transposable Data. *Biometrics* **71**, 157–166.

Examples

```
data(VEGFmouse)
## Testing conservation of the overall gene expression across tissues.
tissues_mean_test <- meanmat.ts(datamat = VEGFmouse, N = 40, group.sizes = 9)
tissues_mean_test
## Testing if the adrenal and the cerebrum tissues have the same mean vector.
test2 <- meanmat.ts(VEGFmouse, N = 40, group.sizes = c(2, rep(1,7)))
test2
```

`orderdata` *Reordering Row and Column Variables*

Description

This utility function rearranges the row and/or the column variables in a desired order.

Usage

```
orderdata(datamat, N, order.rows = NULL, order.cols = NULL)
```

Arguments

`datamat`	numeric matrix containing the transposable data.
`N`	positive integer number indicating the sample size, i.e., the number of subjects.
`order.rows`	numeric vector displaying the desired order of the row variables.
`order.cols`	numeric vector displaying the desired order of the column variables.

Details

It is assumed that there are `nrow(datamat)` row variables and `ncol(datamat)/N` column variables in `datamat`. Further, `datamat` should be written in such a way that every `ncol(datamat)/N` consecutive columns belong to the same subject and the order of the column variables in each block is preserved across subjects.

Value

Returns a matrix of the same size as `datamat`.

Author(s)

Anestis Touloumis

See Also

`meanmat.ts` and `meanmat.hat`.

Examples

```
data(VEGFmouse)
set.seed(1)
tissuesold <-  colnames(VEGFmouse[ ,1:9])
## Suppose that you want to order the tissues in the folowing order.
tissuesnew <- colnames(VEGFmouse[ ,1:9])[sample(9)]
tissuesnew
## To do this, create a numeric vector with the desired order.
ordtis <- pmatch(tissuesnew, tissuesold)
VEGFmousenew <- orderdata(datamat = VEGFmouse, N = 40, order.cols = ordtis)
colnames(VEGFmousenew)[1:9]
```

---

transposedata *Interchanging the Row and Column Variables in Transposable Data*

Description

This function interchanges the row and column variables in transposable data so that the original row variables will be treated as column variables and the original column variables as row variables.

Usage

```
transposedata(datamat, N)
```

Arguments

datamat	numeric matrix containing the transposable data.
N	positive integer number indicating the sample size, i.e., the number of subjects.

Details

It is assumed that there are `nrow(datamat)` row variables and `ncol(datamat)/N` column variables in `datamat`. Further, `datamat` should be written in such a way that every `ncol(datamat)/N` consecutive columns belong to the same subject and the order of the column variables in each block is preserved across subjects.

Value

Returns a matrix with `ncol(datamat)` rows and `nrow(datamat)/N` columns.

Author(s)

Anestis Touloumis

See Also

`centerdata` and `orderdata`.

Examples

```
data(VEGFmouse)
## Transposing the VEGF dataset.
VEGFtr <- transposedata(datamat = VEGFmouse, N = 40)
```

VEGFmouse *Vascular Endothelial Growth Factor Mouse Dataset*

Description

Log2 normalized mouse gene expression data in the vascular endothelial growth factor signalling pathway across multiple tissues.

Format

A data frame with 46 rows and 360 columns. The rows corresponds to 46 genes in the VEGF signalling pathway. The column names indicate the mouse and the tissue on which gene expression levels were measured. Since there are 40 mice and 9 tissues, we have a total of 360 columns. Every 9 consecutive columns belong to the same mouse and the tissues are ordered in the same way in each mouse.

Source

Zahn et al. (2007). AGEMAP: A gene expression database for aging in mice. *PLoS Genetics* **3**, e201.

Examples

```
data(VEGFmouse)
## Check the order of the tissues from the first mouse.
colnames(VEGFmouse[,1:9])
```

Index