Package 'longitudinal'

November 13, 2021

Version 1.1.13

Date 2021-11-13

Title Analysis of Multiple Time Course Data

Author Rainer Opgen-Rhein and Korbinian Strimmer.

Maintainer Korbinian Strimmer <strimmerlab@gmail.com>

Depends R (>= 3.0.2), corpcor (>= 1.6.10)

Suggests

Imports graphics, grDevices, stats

Description Contains general data structures and
functions for longitudinal data with multiple variables,
repeated measurements, and irregularly spaced time points.
Also implements a shrinkage estimator of dynamical correlation
and dynamical covariance.

License GPL (>= 3)

URL https://strimmerlab.github.io/software/longitudinal/

NeedsCompilation no

Repository CRAN

Date/Publication 2021-11-13 20:30:08 UTC

R topics documented:

longitudinal-package *The longitudinal package*

Description

This package contains general data structures and functions for longitudinal data with multiple variables, repeated measurements, and irregularly spaced time points. It also implements a shrinkage estimator of dynamical correlation and dynamical covariance.

Author(s)

Rainer Opgen-Rhein and Korbinian Strimmer (https://strimmerlab.github.io/)

References

See website: https://strimmerlab.github.io/software/longitudinal/

See Also

longitudinal, dyn.cor.

dyn.cor *Dynamical Correlation and Covariance*

Description

The functions estimate dynamical correlation and covariance, and related quantities.

Usage

```
dyn.cor(x, lambda, verbose=TRUE)
dyn.var(x, lambda.var, verbose=TRUE)
dyn.cov(x, lambda, lambda.var, verbose=TRUE)
dyn.invcor(x, lambda, verbose=TRUE)
dyn.invcov(x, lambda, lambda.var, verbose=TRUE)
dyn.pvar(x, lambda, lambda.var, verbose=TRUE)
dyn.pcor(x, lambda, verbose=TRUE)
```

Arguments

x	a data matrix
lambda	the correlation shrinkage intensity (range 0-1). If lambda is not specified (the default) it is estimated using an analytic formula from Sch\"afer and Strimmer (2005) and Opgen-Rhein and Strimmer (2006a,b). For lambda=0 the empirical correlations are recovered. See also cor.shrink.

lambda.var	the variance shrinkage intensity (range 0-1). If lambda.var is not specified (the default) it is estimated using an analytic formula from Sch\"afer and Strimmer (2005) and Opgen-Rhein and Strimmer (2006a,b). For lambda.var=0 the empirical variances are recovered. See also var.shrink.
verbose	report progress while computing (default: TRUE)

Details

The dynamical correlation and related quantities implemented here follow the definition of Opgen-Rhein and Strimmer (2006a,b). This approach is derived from a FDA perspective. Essentially, it takes account of the distances between the various time points by assigning weights to samples. If these weights are all equal the usual iid estimators are obtained.

For details about the analytic shrinkage procedure consult Opgen-Rhein and Strimmer (2006b) and Sch\"afer and Strimmer (2005) as well as the help page of cov.shrink.

Value

dyn.cor returns the dynamical correlation matrix. dyn.var returns the vector of dynamical variances. dyn.cov returns the dynamical covariance matrix.

dyn.invcor returns the inverse dynamical correlation matrix. dyn.invcov returns the inverse dynamical covariance matrix.

dyn.pvar returns the vector of partial dynamical variances. dyn.pcor returns the partial dynamical correlation matrix.

Author(s)

Rainer Opgen-Rhein and Korbinian Strimmer (https://strimmerlab.github.io).

References

Opgen-Rhein, R., and K. Strimmer. 2006a. Inferring gene dependency networks from genomic longitudinal data: a functional data approach. REVSTAT **4**:53-65.

Opgen-Rhein, R., and K. Strimmer. 2006b. Using regularized dynamic correlation to infer gene dependency networks from time-series microarray data. The 4th International Workshop on Computational Systems Biology, WCSB 2006 (June 12-13, 2006, Tampere, Finland).

Schaefer, J., and Strimmer, K. (2005). A shrinkage approach to large-scale covariance estimation and implications for functional genomics. Statist. Appl. Genet. Mol. Biol. **4**:32. <DOI:10.2202/1544-6115.1175>

See Also

dyn.weights, cov.shrink, pcor.shrink

Examples

```
# load "longitudinal" library
library("longitudinal")
```

```
# load tcell data
data(tcell)
get.time.repeats(tcell.34)


# dynamical partial correlation
# (this takes into account of the unequal spacings between time points)
dynpc <- dyn.pcor(tcell.34, lambda=0)

# static partial correlation
statpc <- pcor.shrink(tcell.34, lambda=0)

# this is NOT the same
sum((dynpc - statpc)^2)
```

`dyn.scale` *Dynamical Scale, Moments, and Weights*

Description

`time2weights` computes weights corresponding to time points

`dyn.weights` computes these weights for a given `longitudinal` matrix.

`dyn.moments` computes means and variances for the variables in a `longitudinal` object.

`dyn.scale` centers and standardizes a `longitudinal` matrix.

Usage

```
time2weights(t)
dyn.weights(x)
dyn.moments(x)
dyn.scale(x, center=TRUE, scale=TRUE)
```

Arguments

`t`	a vector with time points
`x`	a `longitudinal` object, or a matrix
`center`	logical value
`scale`	logical value

Details

The dynamical weights are computed assuming a linear spline - see Opgen-Rhein and Strimmer (2006a,b). The dynamical mean and variance etc. are then simply weighted versions of the usual empirical estimators.

Value

A vector with weights (`time2weights` and `dyn.weights`), a list containing the column means and variances (`dyn.moments`), or a rescaled longitudinal matrix (`dyn.scale`).

Author(s)

Rainer Opgen-Rhein and Korbinian Strimmer (https://strimmerlab.github.io).

References

Opgen-Rhein, R., and K. Strimmer. 2006a. Inferring gene dependency networks from genomic longitudinal data: a functional data approach. REVSTAT **4**:53-65.

Opgen-Rhein, R., and K. Strimmer. 2006b. Using regularized dynamic correlation to infer gene dependency networks from time-series microarray data. The 4th International Workshop on Computational Systems Biology, WCSB 2006 (June 12-13, 2006, Tampere, Finland).

See Also

`wt.scale`.

Examples

```
# load "longitudinal" library
library("longitudinal")

# weights of for the data points in tcell data
data(tcell)
dyn.weights(tcell.34)

# dynamical moments
dyn.moments(tcell.34)
```

`longitudinal` *Data Structure for Longitudinal Data*

Description

The data type `longitudinal` stores multiple time series data. It allows repeated measurements, irregular sampling, and unequal temporal spacing of the time points.

`as.longitudinal` converts a matrix into a `longitudinal` object. The columns of the input matrix are considered as individual variables (time series). The rows contain the measurements in temporal order (for instance, rows 1-10 could contain 10 repeated measurements taken at time point 1, rows 11-20 further 10 measurements taken at time point 2, and so on). The dates for the time points can be specified with the argument `times` and need not be equally spaced. With the argument `repeats` it is possible to specify the number of measurements per time point (this may be different from time point to time point). In the resulting `longitudinal` matrix object the row names will indicate both

the time points and the repetition number (e.g., "10-1", "10-2", "10-3", ..., "20-1", "20-2", "20-3", etc.).

`is.longitudinal` checks whether a matrix has the `longitudinal` attributes.

The functions `summary`, `print`, `plot` are the standard generic functions adapted to `longitudinal` objects.

Usage

```
as.longitudinal(x, repeats=1, time)
is.longitudinal(x)
## S3 method for class 'longitudinal'
summary(object, ...)
## S3 method for class 'longitudinal'
print(x, ...)
## S3 method for class 'longitudinal'
plot(x, series=1, type=c("median", "mean"), autolayout=TRUE, ...)
```

Arguments

`x, object`	Time series data, contained in a `longitudinal` object or in matrix form (`as.longitudinal`).
`repeats`	Integer, or a vector of integers, that specifies the number of available measurements per time point. If only one number is given then it is assumed the time series is regularly sampled. If instead a vector is specified, then each time point may have a different number of samples.
`time`	A vector with the dates for the time points. If not specified, equally spaced time points 1, 2, 3, ... are assumed.
`series`	Number, or a vector of numbers, that indicates which time series (=variables and columns of x) are plotted.
`type`	Determines whether the plotted line corresponds to the mean or median value of the samples per time point (default: "median").
`autolayout`	determine the number of columns and rows in the plot automatically in the display of multiple time series (default: TRUE).
`...`	Additional optional parameters

Value

`as.longitudinal` returns a `longitudinal` object.

`is.longitudinal` returns TRUE or false.

Author(s)

Korbinian Strimmer (https://strimmerlab.github.io).

See Also

`longitudinal.util`, `tcell`, `ts`.

Examples

```
# load "longitudinal" library
library("longitudinal")

# load data set
data(tcell)
is.longitudinal(tcell.34)
attributes(tcell.34)
tcell.34[,1:3]

# how many samples and how many genes?
dim(tcell.34)
summary(tcell.34)

# plot first nine time series
plot(tcell.34, 1:9)

#####

# an artificial example with repeated measurements, irregular sampling, and unequal spacing
m <- matrix(rnorm(200), 50, 4)
z <- as.longitudinal(m, repeats=c(10,5,5,10,20), time=c(2,8,9,15,16))
plot(z, 1:4)
```

`longitudinal.util` *Utility Functions for the "Longitudinal" Data Structure*

Description

The above functions are all utility functions for `longitudinal` objects.

`get.time.repeats` returns the measurement design, i.e. the time points and the number of repeats per time point.

`is.equally.spaced` checks whether the distances between subsequent time points are all equal.

`is.regularly.sampled` checks whether the number of measurements are identical across time points.

`has.repeated.measurements` checks whether any time point as been measured more than once.

`combine.longitudinal` combines the measurements of two `longitudinal` objects. These objects must have the same (number of) variables.

`condense.longitudinal` condenses the multiple measurements per time point using an arbitrary function (e.g., mean, median, var).

Usage

```
get.time.repeats(x)
is.equally.spaced(x)
is.regularly.sampled(x)
```

```
has.repeated.measurements(x)
combine.longitudinal(x1, x2)
condense.longitudinal(x, s, func=median)
```

Arguments

x, x1, x2	longitudinal time series objects
s	An integer, or a vector of integers, that designate the set of time series (variables) to condense.
func	Univariate function used to summarize the multiple measurements per time point.

Value

get.time.repeats returns a list containing two vectors (time and repeats).

is.equally.spaced, is.regularly.sampled, and has.repeated.measurements return either TRUE or FALSE.

combine.longitudinal returns a longitudinal object.

condense.longitudinal returns a matrix.

Author(s)

Korbinian Strimmer (https://strimmerlab.github.io).

See Also

longitudinal, tcell.

Examples

```
# load "longitudinal" library
library("longitudinal")


# load tcell data set
data(tcell)
dim(tcell.34)
is.longitudinal(tcell.34)
summary(tcell.34)

# information
get.time.repeats(tcell.34)
is.equally.spaced(tcell.34)
is.regularly.sampled(tcell.34)
has.repeated.measurements(tcell.34)

# compute the mean value at each time point for the first two gene
condense.longitudinal(tcell.34, 1:2, mean)


# combine two time series
```

```
m1 <- matrix(rnorm(100), 50, 2)
m2 <- matrix(rnorm(100), 50, 2)
z1 <- as.longitudinal(m1, repeats=c(10,5,5,10,20), time=c(2,8,9,15,16))
z2 <- as.longitudinal(m2, repeats=c(10,5,5,10,20), time=c(1,8,9,15,20))

z3 <- combine.longitudinal(z1,z2)
summary(z3)
get.time.repeats(z3)  # compare with z1 and z2
```

`tcell` *Microarray Time Series Data for T-Cell Activation*

Description

The data result from two experiments investigating the expression response of human T cells to PMA and ionomicin treatment.

The first data set (`tcell.34`) contains the temporal expression levels of 58 genes for 10 unequally spaced time points. At each time point there are 34 separate measurements. The second data set (`tcell.10`) stems from a related experiment considering the same genes and identical time points, and contains 10 further measurements per time point. See Rangel et al. (2004) for more details.

Usage

```
data(tcell)
```

Format

`tcell.10` and `tcell.34` are `longitudinal` objects, i.e. matrices with 58 colums each and a number of extra attributes (see `longitudinal` and `longitudinal.util`).

The vector `tcell.gene.descriptions` contains the description of the functions of the 58 investigated genes.

Source

This data is described in Rangel et al. (2004).

References

Rangel, C., Angus, J., Ghahramani, Z., Lioumi, M., Sotheran, E., Gaiba, A., Wild, D. L., and Falciani, F. (2004) Modeling T-cell activation using gene expression profiling and state-space models. *Bioinformatics*, **20**, 1361–1372.

Examples

```
# load "longitudinal" library
library("longitudinal")

# load data sets
data(tcell)

# data set with 10 repeats
dim(tcell.10)
summary(tcell.10)
is.longitudinal(tcell.10)
is.regularly.sampled(tcell.10)
is.equally.spaced(tcell.10)
get.time.repeats(tcell.10)

# data set with 34 repeats
dim(tcell.34)
summary(tcell.34)
is.longitudinal(tcell.34)
is.regularly.sampled(tcell.34)
is.equally.spaced(tcell.34)
get.time.repeats(tcell.34)

# descriptions of the first nine genes
tcell.gene.description[1:9]

# plot the first nine time series
plot(tcell.10, 1:9)
plot(tcell.34, 1:9)

# Rangel et al. use the combined data set
tcell.44 <- combine.longitudinal(tcell.34, tcell.10)
plot(tcell.44, 1:9)
```

Index