

Package: LPWC (via r-universe)

July 5, 2026

Version 1.0.1

Title Lag Penalized Weighted Correlation for Time Series Clustering

Description Computes a time series distance measure for clustering based on weighted correlation and introduction of lags. The lags capture delayed responses in a time series dataset. The timepoints must be specified. T. Chandereeng, A. Gitter (2020) <[doi:10.1186/s12859-019-3324-1](https://doi.org/10.1186/s12859-019-3324-1)>.

Depends R (>= 3.0.2)

Suggests testthat, rmarkdown, pkgdown, ggplot2, knitr, devtools

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

URL <https://github.com/gitter-lab/LPWC>

BugReports <https://github.com/gitter-lab/LPWC/issues>

Imports nleqslv

NeedsCompilation yes

Author Thevaa Chandereeng [aut, cre, cph] (ORCID: <<https://orcid.org/0000-0003-4078-9176>>), Anthony Gitter [aut, cph] (ORCID: <<https://orcid.org/0000-0002-5324-9833>>)

Maintainer Thevaa Chandereeng <thevaasiinen@gmail.com>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-05 12:44:37 UTC

RemoteUrl <https://github.com/cran/LPWC>

RemoteRef HEAD

RemoteSha 5fb76794248740c6756507ba22da03ded868d80e

Contents

| | |
|--------------|-----------|
| best.lag | 2 |
| comp.corr | 3 |
| corr.bestlag | 4 |
| findC | 5 |
| prep.data | 6 |
| score | 7 |
| simdata | 7 |
| weight | 8 |
| weight.lag | 9 |
| wt.corr | 9 |
| Index | 11 |

| | |
|----------|-----------------|
| best.lag | <i>Best Lag</i> |
|----------|-----------------|

Description

best.lag computes the best lags for a dataset using weighted correlation. The lags obtained are in reference to the original timepoints.

Usage

```
best.lag(data, timepoints, max.lag = NULL, C)
```

Arguments

| | |
|------------|--|
| data | a matrix or data frame with rows representing genes and columns representing different timepoints. If data is a data frame, the gene names can be specified using the row.names(). |
| timepoints | a vector of time points used in the dataset |
| max.lag | a numeric value of the maximum lag allowed, if null, defaults to the floor of the number of timepoints divided by 4 |
| C | a numeric value of C used in computing weighted correlation |

Value

a vector of best lags used in the dataset, one per gene

Author(s)

Thevaa Chandereeng, Anthony Gitter

Examples

```
best.lag(data = array(rnorm(20), c(4, 5)), timepoints = c(0, 5, 10, 20, 40), C = 300)
best.lag(data = array(runif(100), c(5, 20)), timepoints = seq(2, 40, 2), C = 10)
best.lag(data = array(runif(100), c(5, 20)), timepoints = seq(2, 40, 2), max.lag = 2, C = 10)
```

comp.corr

*Computing corr***Description**

This function computes the weighted correlation with a penalty for lags. It should only be used after the fixed lags have already been applied to the dataset and timepoints using the functions `prep.data()` and `best.lag()`.

Usage

```
comp.corr(data, time, C)
```

Arguments

| | |
|-------------------|--|
| <code>data</code> | a lagged matrix or data frame with rows representing genes and columns representing different timepoints (NAs added when lags are needed) |
| <code>time</code> | a lagged matrix with rows representing each gene's timepoint and columns representing the number of timepoints, NA is introduced when it is lagged |
| <code>C</code> | a numeric value of C used in computing weighted correlation |

Value

a similarity matrix with values between -1 and 1 (1 highly correlated, 0 no correlation)

Author(s)

Thevaa Chandereeng, Anthony Gitter

Examples

```
## This function computes the correlation after the lags (or shifts) have
## been computed. In this example, the lags argument is randomly sampled
## for the sake of illustrating how prep.data() applies the lags and
## prepares a transformed dataset for comp.corr().
lagged <- prep.data(array(rnorm(30), c(3, 10)), timepoints = seq(0, 45, 5),
  lags = sample(c(0, 1, -1, 2, -2), size = 3))
comp.corr(data = lagged$data, time = lagged$time, C = 10)

## This example shows how comp.corr is used in practice with real data.
## The best.lag() function is called first to pre-compute the lags, which
```

```
## are passed to prep.data().
randdata <- array(rnorm(120), c(10, 12))
bl <- best.lag(data = randdata, timepoints = 1:12, C = 5)
lag.data <- prep.data(randdata, timepoints = 1:12, lags = bl)
comp.corr(lag.data$data, time = lag.data$time, C = 5)
```

| | |
|--------------|--------------------------------------|
| corr.bestlag | <i>Computes best lag correlation</i> |
|--------------|--------------------------------------|

Description

This function computes correlation based on best picked lags. The lags indicate delayed changes.

Usage

```
corr.bestlag(data, timepoints, max.lag = NULL, C = NULL,
  penalty = "high", iter = 10)
```

Arguments

| | |
|------------|--|
| data | a matrix or data frame with rows representing genes and columns representing different timepoints. If data is a data frame, the gene names can be specified using the <code>row.names()</code> . |
| timepoints | a vector of time points used in the dataset |
| max.lag | a integer value of the maximum lags allowed in the dataset, if null, defaults to the floor of the number of timepoints divided by 4 |
| C | a numeric value of C used in computing weighted correlation, if null, a default is computed based on the penalty argument |
| penalty | a factor with two levels high and low penalty on the weighted correlation |
| iter | an integer indicating the number of C values to test for low penalty |

Value

a list containing weighted correlation and best lags used in each row

Author(s)

Thevaa Chandereeng, Anthony Gitter

Examples

```
corr.bestlag(array(rnorm(30), c(5, 6)), max.lag = 1,
               timepoints = c(0, 5, 10, 15, 20, 25), C = 10, penalty = "high")
corr.bestlag(array(runif(40, 0, 20), c(4, 10)),
               timepoints = c(0, 0.5, 1.5, 3, 6, 12, 18, 26, 39, 50), penalty = "high")
corr.bestlag(matrix(data = rexp(n = 40, 2), nrow = 8),
                 timepoints = c(0, 5, 15, 20, 40), penalty = "low", iter = 5)
```

findC

Finding best C

Description

This function computes the values of C to test using the timepoints and max lag in the dataset

Usage

```
findC(timepoints, max.lag = NULL, pi = 0.95, iter = 10)
```

Arguments

| | |
|------------|--|
| timepoints | a vector of timepoints used in the dataset |
| max.lag | a numeric value with maximum lags allowed, if null, defaults to the floor of the number of timepoints divided by 4 |
| pi | a numeric value between 0.5 and 1 for the upper bound on the penalty |
| iter | a numeric value with the number of penalties to test |

Value

a vector of length iter of the different values of C to test

Author(s)

Thevaa Chandereeng, Anthony Gitter

Examples

```
findC(c(0, 5, 10, 15, 20, 25), max.lag = 1, iter = 15)
findC(c(2, 4, 8, 16, 32, 64, 128, 256), iter = 5)
findC(c(2, 6, 10, 15, 22, 30, 40, 55, 80), pi = 0.8, iter = 20)
findC(c(1, 2, 3.2, 4, 5.3, 7), pi = 0.99)
```

prep.data

Preparing Data

Description

This function prepares the data to compute correlation by introducing NA's when lags are needed

Usage

```
prep.data(data, lags, timepoints)
```

Arguments

| | |
|------------|--|
| data | a matrix or data frame with rows representing genes and columns representing different timepoints. If data is a data frame, the gene names can be specified using the <code>row.names()</code> . |
| lags | a vector of same length as the number of rows in the data column indicating the best lags |
| timepoints | a vector of time points used in the dataset |

Value

a list of two matrices, one matrix with NA's for the lags for the dataset and another matrix with the timepoints used for each row in the dataset

Author(s)

Thevaa Chandereeng, Anthony Gitter

Examples

```
prep.data(array(rnorm(20), c(5, 4)), c(0, 0, 0, -1, 1),
           timepoints = c(0, 5, 15, 30))
prep.data(array(runif(100, 0, 10), c(10, 10)), sample((-2:2), size = 10, replace = TRUE),
           timepoints = c(0, 5, 15, 30, 45, 60, 75, 80, 100, 120))
```

| | |
|-------|----------------------|
| score | <i>Score of Lags</i> |
|-------|----------------------|

Description

This function computes the score of best lags by summing the correlation of corresponding lags

Usage

```
score(corr, lags)
```

Arguments

| | |
|------|--|
| corr | a vector of computed correlation |
| lags | a vector of same length with corr that holds the lags corresponding to the corr vector |

Value

a numerical value of best lag picked based on corr and lags

Author(s)

Thevaa Chandereeng, Anthony Gitter

Examples

```
score(runif(10, 0, 1), c(2, 0, 0, 0, 3, 2, -1, 2, 0, 1))  
score(runif(20, 0.5, 0.8), sample(-3:3, size = 20, replace = TRUE))
```

| | |
|---------|---------------------------------|
| simdata | <i>Example datasets for LPC</i> |
|---------|---------------------------------|

Description

Data is simulated from ImpulseDE function with 8 timepoints c(0, 2, 4, 6, 8, 18, 24, 32, 48, 72) See <https://doi.org/10.1093/bioinformatics/btw665>

Usage

```
simdata
```

Format

data matrix

Examples

```
data(simdata)
```

| | |
|--------|------------------------------|
| weight | <i>Weight in correlation</i> |
|--------|------------------------------|

Description

This function computes the weight used for correlation using timepoints used and lags used

Usage

```
weight(t, lag, C)
```

Arguments

| | |
|-----|---|
| t | a vector of timepoints |
| lag | a integer value of the lag used |
| C | a numeric of the constant used in the penalty and weight inside the Gaussian kernel |

Value

a list containing w0 and vector w used for computing weighted correlation

Author(s)

Thevaa Chandereeng, Anthony Gitter

Examples

```
weight(t = c(0, 5, 10, 15, 20), lag = 1, C = 20)  
weight(t = c(0, 2, 5, 10, 14, 19, 22), lag = 1, C = 100)
```

`weight.lag`*Weight Lag*

Description

This function matches vectors of two different lengths

Usage

```
weight.lag(x1, x2)
```

Arguments

x1 a vector

x2 a vector

Value

a matrix with two rows with the shortest length of the vector as the number of columns

Author(s)

Thevaa Chandereeng, Anthony Gitter

Examples

```
weight.lag(1:5, 2:9)
weight.lag (seq(0, 10, 2), seq(4, 10, 2))
```

`wt.corr`*Weighted correlation*

Description

This function computes weighted Pearson correlation between two vectors with weights given. The output is between -1 and 1 with 1 being highly positively correlated, -1 being highly negatively correlated, and 0 being no correlation

Usage

```
wt.corr(x, y, w)
```

Arguments

| | |
|---|--------------------------------------|
| x | a vector |
| y | a vector with same length of x |
| w | a vector with same length of x and y |

Value

a numerical value of weighted Pearson correlation

Source

https://en.wikipedia.org/wiki/Pearson_correlation_coefficient#Weighted_correlation_coefficient

Examples

```
wt.corr(c(1, 2, -9, 4, 5), c(2:6), c(0.5, 1, 2, 0.5, 2))  
wt.corr(rnorm(5), rnorm(5), runif(5, 0, 1))
```

Index

* **datasets**

 simdata, 7

best.lag, 2

comp.corr, 3

corr.bestlag, 4

findC, 5

prep.data, 6

score, 7

simdata, 7

weight, 8

weight.lag, 9

wt.corr, 9