Package: ecoXCorr (via r-universe)

May 29, 2026

Type Package

Title Lagged Cross-Correlation Analysis of Environmental Time Series

Version 0.2.2

Description Tools for analysing lagged relationships between environmental variables and ecological or epidemiological time series. The package implements a workflow to aggregate meteorological data over multiple lagged intervals, fit regression models, including mixed-effect models using 'glmmTMB', for each lag window, and visualise varied models outcomes (effect strength and direction, model prediction error...) using cross-correlation maps ('CCM').

License GPL (>= 3)

URL https://github.com/Nmoiroux/ecoXCorr

BugReports https://github.com/Nmoiroux/ecoXCorr/issues

Depends R (>= 3.5)

Imports ggplot2, glmmTMB, performance, Rdpack, scales, shiny, stats

Suggests testthat (>= 3.0.0)

RdMacros Rdpack

Config/testthat/edition 3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev zlib1g-dev

Repository https://r-multiverse-staging.r-universe.dev

Date/Publication 2026-05-01 09:46:39 UTC

RemoteUrl https://github.com/Nmoiroux/ecoXCorr

RemoteRef 0936697d8620330696c4e99310e4018be970f244

RemoteSha 0936697d8620330696c4e99310e4018be970f244

Contents

`aggregate_lagged_intervals` *Aggregate meteorological time series over multiple lagged time intervals*

Description

This function computes aggregated values of one or several meteorological time series over all possible lagged time intervals defined relative to one or more reference dates.

Usage

```
aggregate_lagged_intervals(
  data,
  date_col,
  value_cols,
  id_col = NULL,
  ref_date,
  interval = 1,
  max_lag,
  shift = 1,
  funs = list(mean = mean, min = min, max = max, sum = sum),
  na.rm = TRUE
)
```

Arguments

`data`	A data.frame containing the meteorological time series.
`date_col`	Character string giving the name of the date column in `data`. The column must be of class `Date` or `POSIXct`.
`value_cols`	Character vector giving the names of numeric variables to be aggregated (e.g. rainfall, temperature).
`id_col`	Optional character string giving the name of a column identifying independent time series (e.g. site, station, individual). When provided, aggregation is performed separately within each level of `id_col`, ensuring that lagged intervals do not combine values from different time series.

If NULL (default), the input data are assumed to represent a single continuous time series.

ref_date — Vector of reference dates d. Can be of class Date or coercible to Date. Aggregations are computed independently for each date.

interval — Integer giving the length of the base time interval i, expressed in days. Each elementary lag block contains exactly i calendar days. (e.g. 1 for daily data, 7 for weekly intervals, 14 for fortnightly intervals).

max_lag — Integer giving the maximum lag (number of intervals) m to consider. All combinations of lag windows with 1 <= lag_end <= lag_start <= m are evaluated.

shift — Integer specifying how many days x the reference date d should be shifted back in time when constructing lag windows.

When x = 0, lag intervals can end exactly at and include the reference date d. When x = 1 (default), the reference date itself is excluded and lag intervals end at d - 1 (included). More generally, increasing x shifts the reference date d back in time and excludes the x most recent days from the aggregation. This parameter is useful when the predictor measured at or immediately before the sampling date should not contribute to the lagged summary.

funs — Named list of aggregation functions to apply to each variable. Each function must accept a numeric vector as first argument. The names of the list are used to construct output column names (e.g. rain_mean, temp_max). Defaults to mean, min, max and sum.

na.rm — Logical indicating whether missing values should be removed before aggregation. Passed to the aggregation functions (default: TRUE).

Details

For each reference date d, lag windows are constructed backward in time as $[(d - x) - k \times i + 1,\ (d - x) - (l - 1) \times i]$, where x shifts the reference date d back in time (excluding d when x >= 1), i is the base interval length (in days), m is the maximum number of intervals considered, and k, l range from 1 to m with k >= l.

The function supports multiple reference dates, multiple variables, and multiple aggregation functions, and returns all combinations as additional columns in the output data frame.

Reference dates for which at least one interval contains no observations are reported in the console as having missing data. Reference dates for which at least one interval partially lies outside the temporal bounds of the input time series are reported as having truncated intervals.

When id_col is specified, lagged aggregation is applied independently to each time series defined by this column. This is particularly useful when the dataset contains multiple independent temporal replicates (e.g. multiple sampling locations or experimental units). In that case, aggregation windows are constructed within each group, and no information is shared across groups.

Value

A data.frame with one row per reference date and lag window, containing:

- date: reference date
- start, end: start (inclusive) and end (inclusive) of the aggregation interval

- `lag_start`, `lag_end`: lag indices defining the interval
- One column per combination of variable and aggregation function (e.g. `rain_mean`, `temperature_sum`)

If `id_col` is provided, the output includes this column to identify the corresponding time series for each aggregated interval.

Examples

```
sampling_dates <- unique(albopictusMPL2023$date)

met_agg <- aggregate_lagged_intervals(
data       = meteoMPL2023,
date_col   = "date",
value_cols = c("rain_sum", "temp_mean"),
ref_date   = sampling_dates,
interval   = 7,
max_lag    = 8
)

head(met_agg)
```

`albopictusMPL2023` *Entomological records of female *Aedes albopictus* in Montpellier (2023)*

Description

This dataset contains entomological sampling records of female *Aedes albopictus* collected in Montpellier (France) during 2023. The data originate from GBIF and correspond to a zero-truncated subset (i.e. only positive captures) of adult female mosquitoes collected using fixed traps across different areas of the city.

Usage

```
albopictusMPL2023
```

Format

A data.frame with the following variables:

species Scientific name of the species (*Aedes albopictus*).

individualCount Number of female individuals captured during the sampling event.

eventDate Original event date as provided by GBIF.

trap Identifier of the trap used for sampling.

area Identifier of the area where the trap is located.

date Sampling date (class `Date`).

ID Unique sampling event identifier.

Details

The dataset was derived from a GBIF annotated archive (DOI: 10.15468/4qafbu) and processed to extract sampling dates, trap identifiers, and spatial grouping variables.

Only records corresponding to adult females of *Aedes albopictus* were retained. Observations with zero counts were removed, resulting in a zero-truncated dataset suitable for abundance modelling.

The dataset can be used to illustrate lagged associations between environmental variables and mosquito abundance, for example in conjunction with the functions `aggregate_lagged_intervals()`, `fit_models_by_lag()`, and `plotCCM()`.

Source

GBIF occurrence data: doi:10.15468/4qafbu

`ecoXCorr` *Run a complete ecoXCorr analysis: aggregation + lagged modelling*

Description

This wrapper function combines `aggregate_lagged_intervals()` and `fit_models_by_lag()` into a single workflow. It aggregates environmental predictors over multiple lag windows relative to sampling dates, merges them with a response dataset, and fits regression models separately for each lag window.

Usage

```
ecoXCorr(
  meteo_data,
  response_data,
  date_col_meteo = "date",
  date_col_resp = "date",
  value_cols,
  agg_fun = "mean",
  interval = 1,
  max_lag,
  shift = 1,
  response,
  covariates = character(0),
  random = character(0),
  family = "gaussian",
  na.rm = TRUE,
  track = FALSE,
  ...
)
```

Arguments

meteo_data	Data frame containing a **unique** time series (i.e. from one station of site, no date duplication) of meteorological variables.
response_data	Data frame containing the response variable and sampling dates.
date_col_meteo	Name of the date column in meteo_data.
date_col_resp	Name of the date column in response_data.
value_cols	Name of one meteorological variables to aggregate.
agg_fun	Name (character string) of the aggregation function. Function must accept a numeric vector as first argument. Default to "mean".
interval	Length of the base lag interval (in days).
max_lag	Maximum number of lag intervals.
shift	Integer specifying how many days x the reference date d should be shifted back in time when constructing lag windows. When x = 0, lag intervals can end exactly at and include the reference date d. When x = 1 (default), the reference date itself is excluded and lag intervals end at d - 1 (included). More generally, increasing x shifts the reference date d back in time and excludes the x most recent days from the aggregation. This parameter is useful when the predictor measured at or immediately before the sampling date should not contribute to the lagged summary.
response	Name of the response variable.
covariates	Optional fixed-effect covariates (adjustment covariates).
random	Optional random-effects structure (passed to fit_models_by_lag).
family	Model family (GLM or glmmTMB).
na.rm	Logical indicating whether NA values are removed before aggregation.
track	Logical; if TRUE, prints lag windows during model fitting.
...	Additional arguments passed to the model fitting function.

Details

Both fixed-effect and mixed-effect models are supported.

The modelling function used depends on the random and family arguments:

- random is not specified (default): stats::glm()
- random is note an empty string OR family is a valid glmmTMB family: glmmTMB::nbinom2()

For mixed-effects models, marginal R^2 (Nakagawa) is returned. For fixed-effects models, appropriate R^2 is used (see performance::r2()). Depending on model specification (depending on random and/or family), R^2 for glmmTMB::glmmTMB() models may not be computed: the returned error or warning is printed in the console.

For each unique combination of lag_start and lag_end, the function:

1. Subsets the data to the corresponding lag window,
2. Removes rows with missing values in the response or predictor,

3. Fits the specified model,
4. Extracts beta parameter of the linear predictor,
5. Extracts the p-value of the predictor effect,
6. Computes AIC for the specified and null models (i.e. excluding 'predictors' in the fixed part),
7. Computes the appropriate modelR^2 (marginal Nakagawa R^2 for mixed models),
8. Records the sign of the estimated effect and the sample size,
9. Computes delta-AIC and Akaike weight of each model (van de Pol et al. 2016),
10. Computes adjusted p-value using the False Discovery Rate method (Benjamini and Yekutieli 2001)

The returned table is suitable for lag-window screening, heatmap visualisation, or sensitivity analyses in epidemiological or ecological studies.

Value

A data frame with one row per lag window, containing:

lag_start Start lag index of the aggregation window.

lag_end End lag index of the aggregation window.

response Name of the response variable.

predictor Name of the predictor variable.

R2... Model's coefficient of determination (marginal R^2 for mixed models).

betas Estimated beta parameter of the linear predictor.

sign Sign of the estimated predictor effect (-1 or +1).

d_aic AIC reduction compared to the null model.

n Number of observations used to fit the model.

p_value P-value associated with the predictor effect.

weight Akaike weight.

p_adj P-value of the estimated predictor effect, adjusted for multiple testing (false discovery rate method.

References

Benjamini Y, Yekutieli D (2001). "The Control of the False Discovery Rate in Multiple Testing under Dependency." *The Annals of Statistics*, **29**(4), 1165–1188. ISSN 0090-5364, 2168-8966. doi:10.1214/aos/1013699998.

van de Pol M, Bailey LD, McLean N, Rijsdijk L, Lawson CR, Brouwer L (2016). "Identifying the Best Climatic Predictors in Ecology and Evolution." *Methods in Ecology and Evolution*, **7**(10), 1246–1257. doi:10.1111/2041210X.12590.

See Also

[aggregate_lagged_intervals()] [fit_models_by_lag()]

Examples

```
res_glmm <- ecoXCorr(
meteo_data    = meteoMPL2023,
response_data = albopictusMPL2023,
date_col_meteo   = "date",
date_col_resp    = "date",
value_cols    = "rain_sum",
agg_fun       = "sum",
response      = "individualCount",
interval      = 7,
max_lag       = 8,
family        = "poisson"
)

head(res_glmm)
```

`ecoXCorrApp` *Launch the ecoXCorr Shiny application*

Description

This function launches an interactive Shiny application allowing users to run a complete ecoXCorr workflow (aggregation, lagged modelling and visualisation) using either example datasets included in the package or user-provided data.

Usage

```
ecoXCorrApp()
```

Value

No return value, the function is a wrapper for `shiny::runApp()`

`fit_models_by_lag` *Fit regression models by lag window on aggregated meteorological predictors*

Description

This function fits a regression model separately for each lag window defined by the `lag_start` and `lag_end` columns of the input data frame. For each lag window, the model is fitted using observations corresponding to different reference dates (`date`), and summary statistics (betas, sign of effect, R^2, delta AIC, Akaike weight, sample size, p-value, p-value adjusted for multiple testing) are returned for the specified predictor.

Usage

```
fit_models_by_lag(
  data,
  response,
  predictors,
  covariates = character(0),
  random = character(0),
  family = "gaussian",
  min_n = 10,
  track = FALSE,
  ...
)
```

Arguments

data	A data frame containing, at minimum, the columns lag_start, lag_end, date, the response variable, response unique identifier ID, the predictor variable(s) and optional covariates and random-effect variables.
response	Character string giving the name of the response variable.
predictors	Character vector of predictor names. Currently, only a single predictor is supported; providing more than one predictor will result in an error.
covariates	Character vector of predictor names to be included in the fixed part of the model to control for their effect.
random	Optional character string specifying random-effects terms or covariance structure to be added to the model formula (without a leading +), e.g. "(1 | site/year)", "(1 | site) + (1 | year)" or "ar1(times + 0 | group)" (glmmTMB::glmmTMB()). If empty (default), a fixed-effect model is fitted.
family	Character string. The name of a family function to be used in GLM or GLMM models. Default to "gaussian" (Linear model). see stats::family() and glmmTMB::glmmTMB() for valid family functions.
min_n	Minimum number of observations required to fit a model. (Currently not enforced; retained for future extensions.)
track	If TRUE, lag window is printed in the console before model fitting.
...	Additional arguments passed to the underlying modelling function (stats::glm(), or glmmTMB::glmmTMB()).

Details

Both fixed-effect and mixed-effect models are supported.

The modelling function used depends on the random and family arguments:

- random is not specified (default): stats::glm()
- random is note an empty string OR family is a valid glmmTMB family: glmmTMB::nbinom2()

For mixed-effects models, marginal R^2 (Nakagawa) is returned. For fixed-effects models, appropriate R^2 is used (see performance::r2()). Depending on model specification (depending on

`random` and/or `family`), R^2 for `glmmTMB::glmmTMB()` models may not be computed: the returned error or warning is printed in the console.

For each unique combination of `lag_start` and `lag_end`, the function:

1. Subsets the data to the corresponding lag window,
2. Removes rows with missing values in the response or predictor,
3. Fits the specified model,
4. Extracts beta parameter of the linear predictor,
5. Extracts the p-value of the predictor effect,
6. Computes AIC for the specified and null models (i.e. excluding 'predictors' in the fixed part),
7. Computes the appropriate modelR^2 (marginal Nakagawa R^2 for mixed models),
8. Records the sign of the estimated effect and the sample size,
9. Computes delta-AIC and Akaike weight of each model (van de Pol et al. 2016),
10. Computes adjusted p-value using the False Discovery Rate method (Benjamini and Yekutieli 2001)

The returned table is suitable for lag-window screening, heatmap visualisation, or sensitivity analyses in epidemiological or ecological studies.

Value

A data frame with one row per lag window, containing:

lag_start Start lag index of the aggregation window.

lag_end End lag index of the aggregation window.

response Name of the response variable.

predictor Name of the predictor variable.

R2... Model's coefficient of determination (marginal R^2 for mixed models).

betas Estimated beta parameter of the linear predictor.

sign Sign of the estimated predictor effect (-1 or +1).

d_aic AIC reduction compared to the null model.

n Number of observations used to fit the model.

p_value P-value associated with the predictor effect.

weight Akaike weight.

p_adj P-value of the estimated predictor effect, adjusted for multiple testing (false discovery rate method.

References

Benjamini Y, Yekutieli D (2001). "The Control of the False Discovery Rate in Multiple Testing under Dependency." *The Annals of Statistics*, **29**(4), 1165–1188. ISSN 0090-5364, 2168-8966. doi:10.1214/aos/1013699998.

van de Pol M, Bailey LD, McLean N, Rijsdijk L, Lawson CR, Brouwer L (2016). "Identifying the Best Climatic Predictors in Ecology and Evolution." *Methods in Ecology and Evolution*, **7**(10), 1246–1257. doi:10.1111/2041210X.12590.

See Also

glmmTMB::glmmTMB(), performance::r2(), performance::r2_nakagawa()

Examples

```
sampling_dates <- unique(albopictusMPL2023$date)

met_agg <- aggregate_lagged_intervals(
data       = meteoMPL2023,
date_col   = "date",
value_cols = c("rain_sum", "temp_mean"),
ref_date   = sampling_dates,
interval   = 7,
max_lag    = 8
)

albo_lag <- merge(met_agg, albopictusMPL2023, by = "date", all = TRUE)

res_glm <- fit_models_by_lag(
data       = albo_lag,
response   = "individualCount",
predictors = "rain_sum_sum",
random     = "",
family     = "poisson"
)

head(res_glm)
```

---

meteoMPL2023 *Daily meteorological conditions in Montpellier (2023)*

Description

This dataset contains daily meteorological variables for Montpellier (Fréjorgues Airport, WMO station 07643) during the year 2023. The data were derived from 3-hourly SYNOP observations provided by Météo-France and aggregated to daily summaries.

Usage

```
meteoMPL2023
```

Format

A data.frame with one row per day and the following variables:

date Date of observation (class `Date`).

wind_mean Daily mean wind speed (m/s).

wind_min Daily minimum wind speed (m/s).

wind_max Daily maximum wind speed (m/s).

temp_mean Daily mean air temperature (°C).

temp_min Daily minimum air temperature (°C).

temp_max Daily maximum air temperature (°C).

dew.p_mean Daily mean dew point temperature (°C).

dew.p_min Daily minimum dew point temperature (°C).

dew.p_max Daily maximum dew point temperature (°C).

rh_mean Daily mean relative humidity (%).

rh_min Daily minimum relative humidity (%).

rh_max Daily maximum relative humidity (%).

pres_mean Daily mean atmospheric pressure (Pa).

pres_min Daily minimum atmospheric pressure (Pa).

pres_max Daily maximum atmospheric pressure (Pa).

rain_mean Daily mean precipitation (mm).

rain_min Daily minimum precipitation (mm).

rain_max Daily maximum precipitation (mm).

rain_sum Daily total precipitation (mm).

Details

The dataset includes temperature, humidity, wind speed, atmospheric pressure and precipitation, and is intended for use in studies of lagged associations between environmental conditions and ecological or epidemiological time series.

Original 3-hourly SYNOP observations were filtered to retain data from Montpellier–Fréjorgues Airport (WIGOS station 0-20000-0-07643). Air temperature and dew point temperature were converted from Kelvin to degrees Celsius. Negative precipitation values were set to zero prior to aggregation.

Daily statistics (mean, minimum and maximum) were computed for all variables. For precipitation, daily totals are also provided.

This dataset is designed to be used in combination with `aggregate_lagged_intervals()` to generate lagged environmental predictors for ecological or epidemiological modelling.

Source

Météo-France SYNOP data, distributed via data.gouv.fr: https://meteo.data.gouv.fr/datasets/686f8595b351c06a3a790867

plotCCM *Plot a cross-correlation map (CCM) from lagged regression results*

Description

This function visualises the strength and direction of associations between a response variable and a lagged predictor across multiple lag windows (Curriero et al. 2005), using the output of `fit_models_by_lag()`. The resulting plot is a two-dimensional "cross-correlation map", where each tile represents a lag window defined by `lag_start` and `lag_end`.

Usage

```
plotCCM(
  data,
  model_outcome = c("d_aic", "R2sign", "R2", "betas", "weight"),
  threshold_p = 1
)
```

Arguments

`data`	A data.frame produced by `fit_models_by_lag()`, containing at least the columns `lag_start`, `lag_end`, `r2`, `p_value`, and `sign`.
`model_outcome`	Character string specifying the model's outcomes to plot. Either: • `"d_aic"` for the delta AIC (compared to the null model) (default), • `"R2sign"` for the signed coefficient of determination, computed as the marginal or classical R^2 multiplied by the sign of the estimated effect, • `"R2"` for the coefficient of determination (R^2), • `"betas"` for the estimated beta parameter (slope) of the linear predictor or, • `"weight"` for the Akaike weight.
`threshold_p`	Numeric value giving the p-value threshold above which associations are masked (set to `NA`) in the plot. Filtering is performed on the adjusted (for multiple testing) p-values `p_adj`. (Default is `1`, meaning that no filtering is applied.

Details

The color of each tile corresponds to `model_outcome`. Positive associations are shown in red, negative associations in blue, and non-significant or filtered values (using `threshold_p`) are shown in grey.

The lag window yielding the maximum absolute value of `model_outcome` is highlighted with a colored border.

The x-axis corresponds to `lag_start` (displayed in reverse order), and the y-axis corresponds to `lag_end`. Tiles are colored using a diverging color scale centered on zero. Lag windows with `p_value >= threshold_p` are not displayed and appear in grey.

This function does not perform any modelling itself; it is intended solely for visualising results obtained from `fit_models_by_lag()`.

Value

A ggplot2 object representing the cross-correlation map.

References

Curriero FC, Shone SM, Glass GE (2005). "Cross Correlation Maps: A Tool for Visualizing and Modeling Time Lagged Associations." *Vector Borne and Zoonotic Diseases (Larchmont, N.Y.)*, **5**(3), 267–275. ISSN 1530-3667. doi:10.1089/vbz.2005.5.267.

See Also

fit_models_by_lag(), ggplot2::ggplot()

Examples

```
res_glm <- ecoXCorr(
meteo_data    = meteoMPL2023,
response_data = albopictusMPL2023,
date_col_meteo   = "date",
date_col_resp    = "date",
value_cols    = "rain_sum",
agg_fun       = "sum",
response      = "individualCount",
interval      = 7,
max_lag       = 8,
family        = "poisson"
)

plotCCM(res_glm, model_outcome = "R2sign", threshold_p = 0.05)
plotCCM(res_glm, model_outcome = "R2")
plotCCM(res_glm, model_outcome = "d_aic")
plotCCM(res_glm, model_outcome = "betas", threshold_p = 0.05)
```

Index