Package ‘patterncausality’

February 15, 2025

Type Package

Title Pattern Causality Algorithm

Version 0.2.1

Maintainer Hui Wang <huiw1128@gmail.com>

Description A comprehensive package for detecting and analyzing causal relationships in complex systems using pattern-based approaches. Key features include state space reconstruction, pattern identification, and causality strength evaluation.

License GPL-3 | file LICENSE

Depends R (>= 4.1.0)

Imports stats, plot3D, ggplot2, reshape2, grid, ggthemes, tidyr,
statebins, ggrepel, RColorBrewer, scales, gridExtra, parallel

Suggests knitr, rmarkdown, stringr, testthat (>= 3.0.0), lintr, zoo,
graphics,

Encoding UTF-8

LazyData true

Config/Needs/website tidyverse/tidytemplate

URL https://github.com/skstavroglou/pattern_causality/

BugReports https://github.com/skstavroglou/pattern_causality/issues

Config/testthat/edition 3

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Author Stavros Stavroglou [aut] (<https://orcid.org/0000-0003-3931-0391>),
Athanasios Pantelous [aut] (<https://orcid.org/0000-0001-5738-1471>),
Hui Wang [aut, cre] (<https://orcid.org/0009-0006-0095-0243>)

Repository CRAN

Date/Publication 2025-02-15 13:40:09 UTC

Contents

AUCO *Illapel Ecological Dataset*

Description

Raw rodent and rainfall data collected from the Las Chinchillas National Reserve near Illapel, Coquimbo Region of Chile. This dataset provides ecological time series for studying species interactions and environmental effects.

Usage

```
AUCO
```

Format

A data frame with rodent and rainfall data.

Details

Illapel Ecological Dataset

Source

Las Chinchillas National Reserve Research Station

Examples

```
data(AUCO)
head(AUCO)
summary(AUCO)
```

climate_indices *Climate Indices Dataset*

Description

A comprehensive time series dataset containing various climate indices used for pattern causality analysis. This dataset includes multiple climate indicators measured over time.

Usage

```
climate_indices
```

Format

A data frame with 100 rows and 5 columns:

Date Date; Date of the measurement

AO Numeric; Arctic Oscillation index

AAO Numeric; Antarctic Oscillation index

NAO Numeric; North Atlantic Oscillation index

PNA Numeric; Pacific/North American index

Details

Climate Indices Dataset

Source

https://www.cpc.ncep.noaa.gov/

Examples

```
data(climate_indices)
head(climate_indices)
summary(climate_indices)
```

---

distanceMetric *Distance Metric Interface*

Description

A generic interface for computing distances between observations using either built-in or custom distance metrics.

Usage

```
distanceMetric(x, method = "euclidean", ...)

## Default S3 method:
distanceMetric(x, method = "euclidean", ...)

## S3 method for class 'custom'
distanceMetric(x, method, ...)
```

Arguments

x	Input data matrix or vector
method	Custom function to compute distances
...	Additional arguments passed to methods

Details

Generic Interface for Distance Metrics

Value

A distance object or matrix containing pairwise distances

Methods (by class)

- `distanceMetric(default)`: Default method using stats::dist
- `distanceMetric(custom)`: Custom distance metric implementation

Examples

```
## Not run:
# Using default method
x <- matrix(rnorm(100), ncol=2)
d1 <- distanceMetric(x, "euclidean")

# Using custom method
custom_dist <- function(x) as.dist(crossprod(x))
d2 <- distanceMetric(x, method=custom_dist)

## End(Not run)
```

`DJS` *Dow Jones Stock Price Dataset*

Description

A comprehensive dataset containing daily stock prices for 29 companies listed in the Dow Jones Industrial Average (DJIA). The dataset includes opening, closing, high, and low prices for each stock.

Usage

```
DJS
```

Format

A data frame with daily stock prices for 29 companies.

Details

Dow Jones Stock Price Dataset

Source

Yahoo Finance

Examples

```
data(DJS)
head(DJS)
summary(DJS)
```

optimalParametersSearch

Search for Optimal Parameters in Pattern Causality Analysis

Description

Searches for the optimal embedding dimension (E) and time delay (tau) to maximize the accuracy of causality predictions in a dataset. This function implements a grid search approach to evaluate different parameter combinations.

Usage

```
optimalParametersSearch(
  Emax,
  tauMax,
  metric = "euclidean",
  distance_fn = NULL,
  state_space_fn = NULL,
  dataset,
  h = 0,
  weighted = FALSE,
  relative = TRUE,
  verbose = FALSE
)
```

Arguments

`Emax`	Positive integer > 2; maximum embedding dimension to test
`tauMax`	Positive integer; maximum time delay to test
`metric`	Character string; distance metric for causality analysis ('euclidean', 'manhattan', 'maximum'). Defaults to "euclidean". Ignored if `distance_fn` is provided.
`distance_fn`	Optional custom distance function; takes two numeric vectors as input and returns a numeric distance. (default: NULL)
`state_space_fn`	Optional custom function for state space reconstruction; takes a numeric vector and parameters E and tau as input and returns a reconstructed state space. (default: NULL)
`dataset`	Numeric matrix; each column represents a time series.
`h`	Positive integer; prediction horizon.
`weighted`	Logical; if TRUE, weighted causality analysis is performed.

`relative` Logical; if TRUE calculates relative changes ((new-old)/old), if FALSE calculates absolute changes (new-old) in signature space. Default is TRUE.

`verbose` Logical; if TRUE, prints progress information. (default: FALSE)

Details

Search for Optimal Parameters in Pattern Causality Analysis

This function evaluates each combination of embedding dimension and time delay for their effectiveness in detecting different types of causality:

- Total causality: Overall causal relationship strength
- Positive causality: Direct positive influences
- Negative causality: Direct negative influences
- Dark causality: Complex or indirect causal relationships

Value

A `pc_params` object containing:

- accuracy_summary: A data frame summarizing the accuracy for each parameter combination.
- computation_time: The time taken for the analysis.
- parameters: A list of the input parameters used.

Examples

```
data(climate_indices)
dataset <- climate_indices[, -1]
optimalParams <- optimalParametersSearch(
  Emax = 3,
  tauMax = 3,
  metric = "euclidean",
  dataset = dataset,
  h = 1,
  weighted = FALSE
)
print(optimalParams)
```

`pcAccuracy` *Calculate Pattern Causality Accuracy*

Description

Evaluates the causality prediction accuracy across multiple time series within a dataset using the PC Mk. II Light method. This function analyzes pairwise causality relationships and computes different types of causality measures.

Usage

```
pcAccuracy(
  dataset,
  E,
  tau,
  metric = "euclidean",
  h,
  weighted,
  distance_fn = NULL,
  state_space_fn = NULL,
  relative = TRUE,
  verbose = FALSE
)
```

Arguments

`dataset` A matrix or data frame where each column represents a time series

`E` Integer; embedding dimension for state space reconstruction (E > 1)

`tau` Integer; time delay for state space reconstruction (tau > 0)

`metric` Character; distance metric to use, one of "euclidean", "manhattan", or "maximum"

`h` Integer; prediction horizon, indicating forecast distance (h >= 0)

`weighted` Logical; whether to use weighted approach in calculating causality strengths

`distance_fn` Optional custom distance function for computing distances (default: NULL)

`state_space_fn` Optional custom function for state space reconstruction (default: NULL)

`relative` Logical; if TRUE calculates relative changes ((new-old)/old), if FALSE calculates absolute changes (new-old) in signature space. Default is TRUE.

`verbose` Logical; whether to display progress information (default: FALSE)

Details

Calculate Pattern Causality Accuracy

Value

An object of class "pc_accuracy" containing:

- parameters: List of input parameters (E, tau, metric, h, weighted)
- total: Mean total causality across all pairs
- positive: Mean positive causality across all pairs
- negative: Mean negative causality across all pairs
- dark: Mean dark causality across all pairs
- matrices: Raw causality matrices for each type

See Also

pcMatrix for analyzing individual causality matrices pcLightweight for pairwise causality analysis

Examples

```
data(climate_indices)
data <- climate_indices[, -1]
results <- pcAccuracy(dataset = data, E = 3, tau = 1,
                     metric = "euclidean", h = 1,
                     weighted = TRUE, verbose = TRUE)
print(results)
```

pcCrossMatrix *Cross Pattern Causality Matrix Analysis*

Description

Analyzes pattern causality relationships between multiple time series in X and multiple time series in Y by computing pairwise causality measures and organizing them into a matrix.

Usage

```
pcCrossMatrix(
  X,
  Y,
  E,
  tau,
  metric = "euclidean",
  h,
  weighted = TRUE,
  distance_fn = NULL,
  state_space_fn = NULL,
  relative = TRUE,
  verbose = FALSE,
  n_cores = 1
)
```

Arguments

X	Matrix or data frame of time series for the cause
Y	Matrix or data frame of time series for the effect
E	Integer; embedding dimension
tau	Integer; time delay

`metric` Character; distance metric ("euclidean", "manhattan", "maximum")

`h` Integer; prediction horizon

`weighted` Logical; whether to use weighted causality

`distance_fn` Optional custom distance function

`state_space_fn` Optional custom state space reconstruction function

`relative` Logical; if TRUE calculates relative changes ((new-old)/old), if FALSE calculates absolute changes (new-old) in signature space. Default is TRUE.

`verbose` Logical; whether to print progress

`n_cores` Integer; number of cores for parallel computation

Details

Compute Cross Pattern Causality Matrix Analysis

The function performs these key steps:

- Validates input data and parameters
- Computes pairwise causality measures between X and Y
- Organizes results into a causality matrix
- Provides summary statistics for each causality type

Value

A pc_matrix object containing causality matrices

Related Packages

- **vars**: Vector autoregression analysis
- **tseries**: Time series analysis tools
- **forecast**: Time series forecasting methods

`pcCrossValidation` *Pattern Causality Cross-Validation Analysis*

Description

Evaluates the robustness of pattern causality measures through repeated sampling analysis. This function performs cross-validation by analyzing multiple subsets of the data to assess the stability of causality relationships.

Usage

```
pcCrossValidation(
  X,
  Y,
  E,
  tau,
  metric = "euclidean",
  h,
  weighted,
  distance_fn = NULL,
  state_space_fn = NULL,
  numberset,
  random = TRUE,
  bootstrap = 1,
  verbose = FALSE,
  n_cores = 1,
  relative = TRUE
)
```

Arguments

`X` Numeric vector representing the first time series.

`Y` Numeric vector representing the second time series.

`E` Integer specifying the embedding dimension.

`tau` Integer specifying the time delay.

`metric` Character string specifying the distance metric to use.

`h` Integer specifying the prediction horizon.

`weighted` Logical indicating whether to use weighted calculations.

`distance_fn` Optional custom distance function.

`state_space_fn` Optional custom state space function.

`numberset` Numeric vector of sample sizes to analyze.

`random` Logical indicating whether to use random sampling (default: TRUE).

`bootstrap` Integer specifying the number of bootstrap iterations (default: 1).

`verbose` Logical indicating whether to display progress messages.

`n_cores` Integer specifying the number of cores to use for parallel computation (default: 1).

`relative` Logical; if TRUE calculates relative changes ((new-old)/old), if FALSE calculates absolute changes (new-old) in signature space. Default is TRUE.

Details

Perform Pattern Causality Cross-Validation Analysis

The function implements these key steps:

- Validates input parameters and data
- Performs stratified sampling of time series data
- When random=TRUE and bootstrap>1, performs bootstrap sampling
- Computes pattern causality measures for each sample
- Aggregates results across all samples

When bootstrap sampling is enabled (random=TRUE and bootstrap>1), the function returns statistics including mean, 5% quantile, 95% quantile, and median for each sample size.

Value

A pc_cv object containing:

- samples: Vector of sample sizes used
- results: Array of causality results
- parameters: List of analysis parameters

The results array structure depends on the bootstrap parameter:

- If bootstrap>1: A three-dimensional array where first dimension represents sample sizes, second dimension contains statistics (mean, quantiles, median), and third dimension represents causality types (positive, negative, dark)
- If bootstrap=1: A three-dimensional array where first dimension represents sample sizes, second dimension contains single values, and third dimension represents causality types (positive, negative, dark)

See Also

`plot.pc_cv` for visualizing cross-validation results `print.pc_cv` for printing cross-validation results `summary.pc_cv` for summarizing cross-validation results

Examples

```
data(climate_indices)
X <- climate_indices$AO
Y <- climate_indices$AAO

# Basic cross-validation
cv_result <- pcCrossValidation(
  X, Y,
  E = 3, tau = 1,
  metric = "euclidean",
  h = 1,
  weighted = FALSE,
  numberset = c(100, 200, 300)
)

# Cross-validation with bootstrap
cv_result_boot <- pcCrossValidation(
  X, Y,
  E = 3, tau = 1,
  metric = "euclidean",
  h = 1,
  weighted = FALSE,
  numberset = c(100, 200, 300),
  random = TRUE,
  bootstrap = 100
)
```

pcEffect *Pattern Causality Effect Analysis*

Description

Analyzes pattern causality matrices to compute and summarize the directional effects of different causality types (positive, negative, dark) between system components.

Usage

```
pcEffect(pcmatrix, verbose = FALSE)
```

Arguments

`pcmatrix`	An object of class "pc_matrix" containing causality matrices
`verbose`	Logical; whether to display computation progress (default: FALSE)

Details

Calculate Pattern Causality Effect Analysis

The function performs these key steps:

- Processes raw causality matrices
- Computes received and exerted influence for each component
- Calculates net causality effect (difference between received and exerted)
- Normalizes results to percentage scale

Value

An object of class "pc_effect" containing:

- positive: Data frame of positive causality effects
- negative: Data frame of negative causality effects
- dark: Data frame of dark causality effects
- items: Vector of component names
- summary: Summary statistics for each causality type

Related Packages

- **vars**: Vector autoregression for multivariate time series
- **lmtest**: Testing linear regression models
- **causality**: Causality testing and modeling

See Also

`pcMatrix` for generating causality matrices `plot.pc_effect` for visualizing causality effects

Examples

```
data(climate_indices)
dataset <- climate_indices[, -1]
pcmatrix <- pcMatrix(dataset, E = 3, tau = 1,
                    metric = "euclidean", h = 1,
                    weighted = TRUE)
effects <- pcEffect(pcmatrix)
print(effects)
plot(effects)
```

pcFullDetails *Calculate Full Details Pattern Causality Analysis*

Description

Implements an advanced pattern causality algorithm to explore the causal relationships between two time series datasets. This function provides comprehensive analysis of causality patterns, including state space reconstruction, pattern identification, and causality strength evaluation.

Usage

```
pcFullDetails(
  X,
  Y,
  E,
  tau,
  h,
  weighted,
  metric = "euclidean",
  distance_fn = NULL,
  state_space_fn = NULL,
  relative = TRUE,
  verbose = FALSE
)
```

Arguments

`X` Numeric vector; the first time series data

`Y` Numeric vector; the second time series data

`E` Integer; embedding dimension for state space reconstruction

`tau` Integer; time delay between data points

`h` Integer; prediction horizon for causality analysis

`weighted` Logical; whether to weight causality strength

`metric` Character; distance metric ('euclidean', 'manhattan', or 'maximum')

`distance_fn` Optional custom distance function for computing distances (default: NULL)

`state_space_fn` Optional custom function for state space reconstruction (default: NULL)

`relative` Logical; if TRUE calculates relative changes ((new-old)/old), if FALSE calculates absolute changes (new-old) in signature space. Default is TRUE.

`verbose` Logical; if TRUE, prints computation progress (default: FALSE)

Details

Calculate Full Details Pattern Causality Analysis

The function implements these key steps:

- State Space Reconstruction: Creates shadow attractors using embedding
- Pattern Analysis: Converts time series into signature and pattern spaces
- Nearest Neighbor Analysis: Identifies and analyzes local dynamics
- Causality Evaluation: Computes predicted and actual causality matrices
- Results Validation: Provides detailed diagnostics and quality metrics

Value

A pc_full_details object containing:

- backtest_time: Time points used for backtesting
- valid_time: Valid time points for analysis
- causality_real: Real causality spectrum
- causality_pred: Predicted causality spectrum
- state_spaces: State space reconstructions
- neighbors: Nearest neighbor information
- patterns: Pattern and signature information
- matrices: Causality matrices
- predictions: Predicted and actual values
- weighted: A logical indicating if weighted calculations were used
- E: Embedding dimension used for the analysis

`pcLightweight` *Calculate Pattern Causality Using Lightweight Algorithm*

Description

Implements a computationally efficient version of the Pattern Causality Model Mk. II for analyzing causal interactions between two time series. This function uses pattern and signature spaces to assess causality through reconstructed state spaces and hashed pattern analysis.

Usage

```
pcLightweight(
  X,
  Y,
  E,
  tau,
  h,
  weighted,
  metric = "euclidean",
  distance_fn = NULL,
  state_space_fn = NULL,
  relative = TRUE,
  verbose = FALSE
)
```

Arguments

`X` A numeric vector representing the first time series

`Y` A numeric vector representing the second time series

`E` Integer; embedding dimension for state space reconstruction (E > 1)

`tau` Integer; time delay for state space reconstruction (tau > 0)

`h` Integer; prediction horizon for future projections (h >= 0)

`weighted` Logical; whether to use weighted causality strength calculations

`metric` Character string specifying the distance metric; one of "euclidean", "manhattan", or "maximum"

`distance_fn` Custom distance function for state space reconstruction

`state_space_fn` Custom function for state space transformation

`relative` Logical; if TRUE calculates relative changes ((new-old)/old), if FALSE calculates absolute changes (new-old) in signature space. Default is TRUE.

`verbose` Logical; whether to display progress information (default: FALSE)

Details

Calculate Pattern Causality Using Lightweight Algorithm

The function implements these key steps:

- State space reconstruction using embedding parameters
- Pattern and signature space transformation
- Nearest neighbor analysis in reconstructed spaces
- Causality strength calculation using prediction accuracy
- Classification of causality types (positive/negative/dark)

Value

An object of class "pc_fit" containing:

- total: Total causality strength (0-1)
- positive: Proportion of positive causality (0-1)
- negative: Proportion of negative causality (0-1)
- dark: Proportion of dark causality (0-1)

See Also

pcFullDetails for detailed analysis pcMatrix for analyzing multiple time series

Examples

```
data(climate_indices)
X <- climate_indices$AO
Y <- climate_indices$AAO
result <- pcLightweight(X, Y, E = 3, tau = 1,
                       metric = "euclidean", h = 2,
                       weighted = TRUE, verbose = FALSE)
print(result)
summary(result)
plot(result)
```

pcMatrix *Pattern Causality Matrix Analysis*

Description

Analyzes pattern causality relationships between multiple time series by computing pairwise causality measures and organizing them into matrices.

Usage

```
pcMatrix(
  dataset,
  E,
  tau,
  metric = "euclidean",
  h,
  weighted = TRUE,
  distance_fn = NULL,
  state_space_fn = NULL,
  relative = TRUE,
  verbose = FALSE,
  n_cores = 1
)
```

Arguments

`dataset`	Matrix or data frame of time series
`E`	Integer; embedding dimension
`tau`	Integer; time delay
`metric`	Character; distance metric ("euclidean", "manhattan", "maximum")
`h`	Integer; prediction horizon
`weighted`	Logical; whether to use weighted causality
`distance_fn`	Optional custom distance function
`state_space_fn`	Optional custom state space reconstruction function
`relative`	Logical; if TRUE calculates relative changes ((new-old)/old), if FALSE calculates absolute changes (new-old) in signature space. Default is TRUE.
`verbose`	Logical; whether to print progress
`n_cores`	Integer; number of cores for parallel computation

Details

Compute Pattern Causality Matrix Analysis

The function performs these key steps:

- Validates input data and parameters
- Computes pairwise causality measures
- Organizes results into causality matrices
- Provides summary statistics for each causality type

Value

A pc_matrix object containing causality matrices

Related Packages

- **vars**: Vector autoregression analysis
- **tseries**: Time series analysis tools
- **forecast**: Time series forecasting methods

pc_cv *Pattern Causality Cross-Validation Object*

Description

Creates a pattern causality cross-validation object containing results from repeated sampling analysis. This function constructs an object of class pc_cv to store the results of cross-validation analysis.

Usage

```
pc_cv(samples = NULL, results = NULL, parameters = NULL)
```

Arguments

samples	Numeric vector of sample sizes used.
results	Matrix containing causality results for each sample.
parameters	List of analysis parameters.

Value

An object of class "pc_cv".

pc_effect *Pattern Causality Effect Object*

Description

Creates a pattern causality effect object that contains information about the received and exerted influences for different causality types. This function constructs an object of class pc_effect to store the results of effect analysis.

Usage

```
pc_effect(positive = NULL, negative = NULL, dark = NULL, items = NULL)
```

Arguments

`positive`	Data frame containing positive causality effects.
`negative`	Data frame containing negative causality effects.
`dark`	Data frame containing dark causality effects.
`items`	Names of items in the analysis.

Value

An object of class "pc_effect".

`pc_matrix` *Pattern Causality Matrix Object*

Description

Creates a pattern causality matrix object. This function constructs an object of class `pc_matrix` containing the positive, negative, and dark causality matrices, along with item names.

Usage

```
pc_matrix(
  positive = NULL,
  negative = NULL,
  dark = NULL,
  items = NULL,
  verbose = TRUE
)
```

Arguments

`positive`	Positive causality matrix.
`negative`	Negative causality matrix.
`dark`	Dark causality matrix.
`items`	Names of items in the matrices.
`verbose`	Logical, whether to print progress information.

Value

An object of class "pc_matrix".

Examples

```
data(climate_indices)
dataset <- climate_indices[, -1]
pc_matrix_obj <- pcMatrix(dataset, E = 3, tau = 1,
  metric = "euclidean", h = 1, weighted = TRUE,
  verbose = FALSE)
print(pc_matrix_obj)
```

`pc_params` *Pattern Causality Parameter Optimization Results*

Description

Creates an object containing parameter optimization results for pattern causality analysis

Usage

```
pc_params(accuracy_summary, computation_time, parameters)
```

Arguments

`accuracy_summary`	Data frame containing accuracy results for different parameter combinations
`computation_time`	Time taken for optimization
`parameters`	List of optimization parameters

Details

Pattern Causality Parameter Optimization Results

Value

An object of class "pc_params"

`plot.pc_cv` *Plot Pattern Causality Cross Validation Results*

Description

Visualizes the pattern causality cross-validation results. This function generates a line plot showing the causality strengths for different sample sizes.

Usage

```
## S3 method for class 'pc_cv'
plot(x, fr = FALSE, separate = FALSE, ...)
```

Arguments

`x`	A `pc_cv` object.
`fr`	Boolean for frame display.
`separate`	Boolean for separate plots.
`...`	Additional arguments passed to the `plot` function.

Value

Invisibly returns the input object.

Examples

```
data(climate_indices)
X <- climate_indices$AO
Y <- climate_indices$AAO
numbersets <- c(100, 150, 200)
cv_results <- pcCrossValidation(X, Y, 3, 2, "euclidean", 1, FALSE, numberset = numbersets)
plot(cv_results)
```

plot.pc_effect *Plot Pattern Causality Effect*

Description

Generates a plot to visualize the effects of positive, negative, or dark causality. Displays the influence exerted versus influence received for each item. This function generates a scatter plot showing the influence exerted versus influence received for each item, colored by the difference between exerted and received influence.

Usage

```
## S3 method for class 'pc_effect'
plot(
  x,
  status = "positive",
  add_label = TRUE,
  point_size = 3,
  label_size = 3,
  ...
)
```

Arguments

`x`	A `pc_effect` object.
`status`	Status of the effect to plot ("positive", "negative", or "dark").
`add_label`	Logical, whether to add labels to the plot.
`point_size`	Numeric value for point size (default: 3).
`label_size`	Numeric value for label text size (default: 3).
`...`	Additional arguments passed to plotting functions.

Value

Invisibly returns the ggplot object.

Examples

```
data(climate_indices)
dataset <- climate_indices[, -1]
pc_matrix_obj <- pcMatrix(dataset, E = 3, tau = 1,
  metric = "euclidean", h = 1, weighted = TRUE,
  verbose = FALSE)
effects <- pcEffect(pc_matrix_obj)
plot(effects, status = "positive")
```

---

plot.pc_fit *Plot Pattern Causality Results*

Description

Generates a combined plot of total causality and causality components for a `pc_fit` object. This function combines the visualizations from `plot_total` and `plot_components` into a single plot.

Usage

```
## S3 method for class 'pc_fit'
plot(x, ...)
```

Arguments

`x`	A `pc_fit` object.
`...`	Additional arguments passed to the underlying plotting functions.

Value

NULL invisibly.

---

plot.pc_matrix *Plot Pattern Causality Matrix*

Description

Creates a heatmap visualization of the pattern causality matrix for positive, negative, or dark causality relationships. This function generates a heatmap using `ggplot2` to visualize the specified causality matrix.

Usage

```
## S3 method for class 'pc_matrix'
plot(
  x,
  status,
  width = 0.85,
  height = 0.75,
  radius = grid::unit(3, "pt"),
  alpha = 0.53,
  show_text = FALSE,
  show_legend_title = FALSE,
  ...
)
```

Arguments

`x`	A `pc_matrix` object containing causality matrices.
`status`	The type of causality to plot ("positive", "negative", or "dark").
`width`	Numeric value specifying the width of the bars (default: 0.85).
`height`	Numeric value specifying the height of the bars (default: 0.75).
`radius`	Grid unit specifying the corner radius of the bars.
`alpha`	Numeric value specifying the transparency (default: 0.53).
`show_text`	Logical, whether to show numerical values on the plot.
`show_legend_title`	Logical, whether to display the legend title.
`...`	Additional arguments passed to plotting functions.

Value

A ggplot object invisibly.

References

Stavroglou et al. (2020) doi:10.1073/pnas.1918269117

Examples

```
data(climate_indices)
dataset <- climate_indices[, -1]
pc_matrix_obj <- pcMatrix(dataset, E = 3, tau = 1,
  metric = "euclidean", h = 1, weighted = TRUE,
  verbose = FALSE)
plot(pc_matrix_obj, status = "positive")
```

`plot.pc_state` *Plot State Space Reconstruction*

Description

Visualizes the state space reconstruction in 3D. This function generates a 3D scatter plot of the reconstructed state space.

Usage

```
## S3 method for class 'pc_state'
plot(x, style = 2, verbose = FALSE, ...)
```

Arguments

`x`	A `pc_state` object.
`style`	Integer; plot style (1 or 2).
`verbose`	Logical; whether to print verbose output.
`...`	Additional arguments passed to the plotting functions.

Value

Invisibly returns the input object.

`plot_causality` *Plot Pattern Causality Time Series*

Description

Visualizes the positive, negative and dark causality components over time

Usage

```
plot_causality(x, type, ...)
```

Arguments

`x`	An object containing pattern causality results
`type`	The type of causality to plot ("total", "positive", "negative", or "dark")
`...`	Additional arguments passed to plotting functions

Value

Invisibly returns the ggplot object

plot_causality.pc_full_details *Plot Pattern Causality Time Series*

Description

Visualizes the positive, negative and dark causality components over time

Usage

```
## S3 method for class 'pc_full_details'
plot_causality(x, type, ...)
```

Arguments

x	A pc_full_details object
type	The type of causality to plot ("total", "positive", "negative", or "dark")
...	Additional arguments passed to plotting functions

Value

Invisibly returns the ggplot object

plot_components *Plot Pattern Causality Components*

Description

Visualizes the positive, negative, and dark causality components as a barplot. This function takes a `pc_fit` object and generates a barplot showing the strength of each causality component.

Usage

```
plot_components(x, ...)
```

Arguments

x	An object containing pattern causality results, typically a `pc_fit` object.
...	Additional arguments passed to the underlying plotting functions.

Value

NULL invisibly.

Examples

```
data(climate_indices)
X <- climate_indices$AO
Y <- climate_indices$AAO
pc_result <- pcLightweight(X, Y, E = 3, tau = 2, metric = "euclidean", h = 1, weighted = TRUE)
plot_components(pc_result)
```

plot_components.pc_fit *Plot Causality Components*

Description

Visualizes the positive, negative, and dark causality components as a barplot for a pc_fit object. This function generates a barplot showing the strength of each causality component.

Usage

```
## S3 method for class 'pc_fit'
plot_components(x, ...)
```

Arguments

x	A pc_fit object.
...	Additional arguments passed to the underlying plotting functions.

Value

NULL.

plot_total *Plot Total Pattern Causality*

Description

Visualizes the total pattern causality strength as a barplot. This function takes a pc_fit object and generates a barplot showing the overall causality strength.

Usage

```
plot_total(x, ...)
```

Arguments

x	An object containing pattern causality results, typically a pc_fit object.
...	Additional arguments passed to the underlying plotting functions.

Value

NULL invisibly.

References

Stavroglou et al. (2020) doi:10.1073/pnas.1918269117

See Also

plot_components for visualizing individual causality components.

Examples

```
data(climate_indices)
X <- climate_indices$AO
Y <- climate_indices$AAO
pc_result <- pcLightweight(X, Y, E = 3, tau = 2, metric = "euclidean", h = 1, weighted = TRUE)
plot_total(pc_result)
```

plot_total.pc_fit *Plot Total Causality*

Description

Visualizes the total causality strength as a barplot for a pc_fit object. This function generates a barplot showing the total causality strength and its complement.

Usage

```
## S3 method for class 'pc_fit'
plot_total(x, ...)
```

Arguments

x	A pc_fit object.
...	Additional arguments passed to the underlying plotting functions.

Value

NULL.

print.pc_accuracy *Print Method for Pattern Causality Accuracy Results*

Description

Print Method for Pattern Causality Accuracy Results

Usage

```
## S3 method for class 'pc_accuracy'
print(x, verbose = FALSE, ...)
```

Arguments

x	A pc_accuracy object
verbose	Logical; whether to display detailed information (default: FALSE)
...	Additional arguments passed to print

Value

Invisibly returns the input object

print.pc_cv *Print Pattern Causality Cross Validation Results*

Description

Prints the pattern causality cross-validation results. This function displays the parameters used for cross-validation, the sample sizes, and the summary statistics.

Usage

```
## S3 method for class 'pc_cv'
print(x, ...)
```

Arguments

x	A pc_cv object.
...	Additional arguments passed to the print function.

Value

Invisibly returns the input object.

Examples

```
data(climate_indices)
X <- climate_indices$AO
Y <- climate_indices$AAO
numberset <- c(100, 150, 200)
cv_results <- pcCrossValidation(X, Y, 3, 2, "euclidean", 1, FALSE, numberset = numberset)
print(cv_results)
```

print.pc_effect *Print Pattern Causality Effect*

Description

Prints the pattern causality effect analysis results. This function displays the received and exerted influences for each item for positive, negative, and dark causality types.

Usage

```
## S3 method for class 'pc_effect'
print(x, ...)
```

Arguments

x	A pc_effect object.
...	Additional arguments passed to the print function.

Value

Invisibly returns the input object.

Examples

```
data(climate_indices)
dataset <- climate_indices[, -1]
pc_matrix_obj <- pcMatrix(dataset, E = 3, tau = 1,
  metric = "euclidean", h = 1, weighted = TRUE,
  verbose = FALSE)
effects <- pcEffect(pc_matrix_obj)
print(effects)
```

print.pc_fit *Print Pattern Causality Results*

Description

Prints the pattern causality analysis results from a pc_fit object. This function displays the total, positive, negative, and dark causality strengths.

Usage

```
## S3 method for class 'pc_fit'
print(x, ...)
```

Arguments

x	A pc_fit object.
...	Additional arguments passed to the print function.

Value

Invisibly returns the input object.

print.pc_matrix *Print Pattern Causality Matrix*

Description

Prints the pattern causality matrix object. This function displays the specified causality matrix (or all matrices) with a preview of the first 5 rows and columns.

Usage

```
## S3 method for class 'pc_matrix'
print(x, type = "all", ...)
```

Arguments

x	A pc_matrix object.
type	The type of matrix to print ("all" or "positive", "negative", "dark").
...	Additional arguments passed to the print function.

Value

Invisibly returns the input object.

Examples

```
data(climate_indices)
dataset <- climate_indices[, -1]
pc_matrix_obj <- pcMatrix(dataset, E = 3, tau = 1,
  metric = "euclidean", h = 1, weighted = TRUE,
  verbose = FALSE)
print(pc_matrix_obj, type = "positive")
```

print.pc_params *Print Method for Pattern Causality Parameter Results*

Description

Print Method for Pattern Causality Parameter Results

Usage

```
## S3 method for class 'pc_params'
print(x, verbose = FALSE, ...)
```

Arguments

x	A pc_params object
verbose	Logical; whether to display detailed information
...	Additional arguments passed to print

Value

Invisibly returns the input object

print.pc_state *Print State Space Reconstruction*

Description

Prints the state space reconstruction results. This function displays the parameters used for state space reconstruction and a preview of the reconstructed points.

Usage

```
## S3 method for class 'pc_state'
print(x, ...)
```

Arguments

`x`	A `pc_state` object.
`...`	Additional arguments passed to the `print` function.

Value

Invisibly returns the input object.

`print.summary.pc_accuracy` *Print Method for Pattern Causality Accuracy Summary*

Description

Print Method for Pattern Causality Accuracy Summary

Usage

```
## S3 method for class 'summary.pc_accuracy'
print(x, ...)
```

Arguments

`x`	A summary.pc_accuracy object
`...`	Additional arguments passed to print

Value

Invisibly returns the input object

`stateSpace` *State Space Reconstruction*

Description

Reconstructs the state space of a time series using delay embedding, creating a matrix where each row represents a point in the reconstructed space.

Usage

```
stateSpace(ts, E, tau, verbose = FALSE)
```

Arguments

`ts`	Numeric vector; time series data
`E`	Integer; embedding dimension (E > 1)
`tau`	Integer; time delay (tau > 0)
`verbose`	Logical; whether to display progress information

Details

State Space Reconstruction Analysis

The function implements Takens' embedding theorem to reconstruct state space:

- Creates delay vectors using specified embedding dimension (E)
- Applies time delay (tau) between consecutive elements
- Handles boundary conditions and missing values

Value

An object of class "pc_state" containing:

- matrix: The reconstructed state space matrix
- parameters: List of reconstruction parameters
- original: Original time series data

Related Packages

- **nonlinearTseries**: Nonlinear time series analysis
- **tseriesChaos**: Chaos theory analysis tools
- **fractal**: Fractal analysis methods

Examples

```
ts <- c(1:100)
result <- stateSpace(ts, E = 3, tau = 2)
plot(result)
```

stateSpaceMethod *State Space Reconstruction Interface*

Description

A generic interface for reconstructing state spaces from time series data using either built-in or custom methods.

Usage

```
stateSpaceMethod(x, E, tau, ...)

## Default S3 method:
stateSpaceMethod(x, E, tau, ...)

## S3 method for class 'custom'
stateSpaceMethod(x, E, tau, method, ...)
```

Arguments

`x`	Input time series
`E`	Embedding dimension (positive integer)
`tau`	Time delay (positive integer)
`...`	Additional arguments passed to methods
`method`	Custom function for state space reconstruction

Details

Generic Interface for State Space Reconstruction

Value

A list containing the reconstructed state space components

Methods (by class)

- `stateSpaceMethod(default)`: Default state space reconstruction
- `stateSpaceMethod(custom)`: Custom state space reconstruction

Examples

```
## Not run:
# Using default method
x <- rnorm(100)
s1 <- stateSpaceMethod(x, E=3, tau=2)

# Using custom method
```

```
custom_space <- function(x, E, tau) {
  list(matrix=embed(x, E))
}
s2 <- stateSpaceMethod(x, E=3, tau=2, method=custom_space)

## End(Not run)
```

summary.pc_accuracy *Summary Method for Pattern Causality Accuracy Results*

Description

Summary Method for Pattern Causality Accuracy Results

Usage

```
## S3 method for class 'pc_accuracy'
summary(object, ...)
```

Arguments

object	A pc_accuracy object
...	Additional arguments passed to summary

Value

A summary object for pc_accuracy

summary.pc_cv *Summary of Pattern Causality Cross Validation Results*

Description

Provides a summary of the pattern causality cross-validation results. This function calculates and displays summary statistics for the cross-validation results, including sample statistics, causality statistics, and convergence.

Usage

```
## S3 method for class 'pc_cv'
summary(object, ...)
```

Arguments

object	A pc_cv object.
...	Additional arguments passed to the summary function.

Value

Invisibly returns the input object.

Examples

```
data(climate_indices)
X <- climate_indices$AO
Y <- climate_indices$AAO
numberset <- c(100, 150, 200)
cv_results <- pcCrossValidation(X, Y, 3, 2, "euclidean", 1, FALSE, numberset = numberset)
summary(cv_results)
```

---

summary.pc_effect *Summarize Pattern Causality Effect*

Description

Provides a summary of the pattern causality effect analysis results. This function displays the summary statistics for the effects, including the number of components and the strongest effects.

Usage

```
## S3 method for class 'pc_effect'
summary(object, ...)
```

Arguments

object	A pc_effect object.
...	Additional arguments passed to the summary function.

Value

Invisibly returns the input object.

Examples

```
data(climate_indices)
dataset <- climate_indices[, -1]
pc_matrix_obj <- pcMatrix(dataset, E = 3, tau = 1,
  metric = "euclidean", h = 1, weighted = TRUE,
  verbose = FALSE)
effects <- pcEffect(pc_matrix_obj)
summary(effects)
```

summary.pc_fit *Summarize Pattern Causality Results*

Description

Provides a summary of the pattern causality analysis results from a pc_fit object. This function displays a table of causality strengths for total, positive, negative, and dark components.

Usage

```
## S3 method for class 'pc_fit'
summary(object, ...)
```

Arguments

object	A pc_fit object.
...	Additional arguments passed to the summary function.

Value

Invisibly returns the input object.

summary.pc_matrix *Summarize Pattern Causality Matrix*

Description

Provides a summary of the pattern causality matrix object. This function calculates and displays descriptive statistics (mean, SD, min, max) for each causality matrix (positive, negative, dark).

Usage

```
## S3 method for class 'pc_matrix'
summary(object, ...)
```

Arguments

object	A pc_matrix object.
...	Additional arguments passed to the summary function.

Value

Invisibly returns the input object.

Examples

```
data(climate_indices)
dataset <- climate_indices[, -1]
pc_matrix_obj <- pcMatrix(dataset, E = 3, tau = 1,
  metric = "euclidean", h = 1, weighted = TRUE,
  verbose = FALSE)
summary(pc_matrix_obj)
```

summary.pc_params *Summary Method for Pattern Causality Parameter Results*

Description

Summary Method for Pattern Causality Parameter Results

Usage

```
## S3 method for class 'pc_params'
summary(object, ...)
```

Arguments

object	A pc_params object
...	Additional arguments passed to summary

Value

A summary object for pc_params

summary.pc_state *Summarize State Space Reconstruction*

Description

Provides a summary of the state space reconstruction results. This function displays the dimensions, number of points, parameters, summary statistics for each dimension, and the number of missing values.

Usage

```
## S3 method for class 'pc_state'
summary(object, ...)
```

Arguments

object	A pc_state object.
...	Additional arguments passed to the summary function.

Value

Invisibly returns the input object.

---

validate_custom_fn_output

Validate Custom Function Output for Pattern Causality Analysis

---

Description

Validates the Output Format from Custom Distance and State Space Functions to ensure compatibility with the package's internal processing.

Usage

```
validate_custom_fn_output(output, fn_name)
```

Arguments

output	The output from a custom function to validate
fn_name	The name of the function type being validated ("distance_fn" or "state_space_fn")

Details

Validate Custom Function Output

Value

Nothing. Throws an error if validation fails.

Examples

```
# Example 1: Validating custom distance function output
custom_dist <- function(x) {
  # Create distance matrix
  dist_mat <- as.matrix(dist(x))
  # Validate output
  validate_custom_fn_output(dist_mat, "distance_fn")
  return(dist_mat)
}

# Example 2: Validating custom state space function output
custom_state_space <- function(x, E, tau) {
  # Create state space matrix
  n <- length(x) - (E-1)*tau
  state_mat <- matrix(nrow = n, ncol = E)
  for(i in 1:E) {
    state_mat[,i] <- x[1:n + (i-1)*tau]
  }
  # Create output list
  result <- list(matrix = state_mat,
                 parameters = list(E = E, tau = tau))
  # Validate output
  validate_custom_fn_output(result, "state_space_fn")
  return(result)
}

# Using the custom functions
x <- sin(seq(0, 4*pi, length.out = 100))
dist_result <- custom_dist(x)
space_result <- custom_state_space(x, E = 3, tau = 2)
```

Index