Package 'spEDM'

April 5, 2026

Title Spatial Empirical Dynamic Modeling

Version 1.12

Description Inferring causation from spatial cross-sectional data through empirical dynamic modeling (EDM), with methodological extensions including geographical convergent cross mapping from Gao et al. (2023) <doi:10.1038/s41467-023-41619-6>, as well as the spatial causality test following the approach of Herrera et al. (2016) <doi:10.1111/pirs.12144>, together with geographical pattern causality proposed in Zhang & Wang (2025) <doi:10.1080/13658816.2025.2581207>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

URL https://stscl.github.io/spEDM/, https://github.com/stscl/spEDM

BugReports https://github.com/stscl/spEDM/issues

Depends R (>= 4.1.0)

LinkingTo Rcpp, RcppThread, RcppArmadillo

Imports dplyr, ggplot2, methods, sdsfun (>= 0.7.0), sf, terra

Suggests knitr, Rcpp, RcppThread, RcppArmadillo, rmarkdown, readr, plot3D

VignetteBuilder knitr

NeedsCompilation yes

Author Wenbo Lyu [aut, cre, cph] (ORCID: <https://orcid.org/0009-0002-6003-3800>)

Maintainer Wenbo Lyu <lyu.geosocial@gmail.com>

Repository CRAN

Date/Publication 2026-04-05 14:20:02 UTC

Contents

`detectThreads` *detect the number of available threads*

Description

detect the number of available threads

Usage

```
detectThreads()
```

Value

An integer

Examples

```
spEDM::detectThreads()
```

`embedded` *embedding spatial cross sectional data*

Description

embedding spatial cross sectional data

Usage

```
## S4 method for signature 'sf'
embedded(
  data,
  target,
  E = 3,
  tau = 1,
  style = 1,
  stack = FALSE,
  detrend = FALSE,
  nb = NULL
)

## S4 method for signature 'SpatRaster'
embedded(
  data,
  target,
  E = 3,
  tau = 1,
  style = 1,
  stack = FALSE,
  detrend = FALSE,
  grid.coord = TRUE,
  embed.direction = 0
)
```

Arguments

`data`	observation data.
`target`	name of target variable.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of spatial lags.
`style`	(optional) embedding style (`0` includes current state, `1` excludes it).
`stack`	(optional) whether to stack embeddings.
`detrend`	(optional) whether to remove the linear trend.
`nb`	(optional) neighbours list.

`grid.coord` (optional) whether to detrend using cell center coordinates (`TRUE`) or row/column numbers (`FALSE`).

`embed.direction` (optional) direction selector for embeddings (`0` returns all directions, `1-8` correspond to NW, N, NE, W, E, SW, S, SE).

Value

A matrix (when `stack` is `FALSE`) or list

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))
v = spEDM::embedded(columbus,"crime")
v[1:5,]

npp = terra::rast(system.file("case/npp.tif",package="spEDM"))
r = spEDM::embedded(npp,"npp")
r[which(!is.na(r),arr.ind = TRUE)[1:5],]
```

---

fnn *false nearest neighbours*

Description

false nearest neighbours

Usage

```
## S4 method for signature 'sf'
fnn(
  data,
  target,
  E = 1:10,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L1",
  rt = 10,
  eps = 2,
  threads = detectThreads(),
  detrend = TRUE,
  nb = NULL
)
```

```
## S4 method for signature 'SpatRaster'
fnn(
  data,
  target,
  E = 1:10,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L1",
  rt = 10,
  eps = 2,
  threads = detectThreads(),
  detrend = TRUE,
  grid.coord = TRUE,
  embed.direction = 0
)
```

Arguments

`data`	observation data.
`target`	name of target variable.
`E`	(optional) embedding dimensions.
`tau`	(optional) step of spatial lags.
`style`	(optional) embedding style (`0` includes current state, `1` excludes it).
`stack`	(optional) whether to stack embeddings.
`lib`	(optional) libraries indices (input needed: `vector` - spatial vector, `matrix` - spatial raster).
`pred`	(optional) predictions indices (input requirement same as `lib`).
`dist.metric`	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).
`rt`	(optional) escape factor.
`eps`	(optional) neighborhood diameter.
`threads`	(optional) number of threads to use.
`detrend`	(optional) whether to remove the linear trend.
`nb`	(optional) neighbours list.
`grid.coord`	(optional) whether to detrend using cell center coordinates (`TRUE`) or row/column numbers (`FALSE`).
`embed.direction`	(optional) direction selector for embeddings (`0` returns all directions, `1-8` correspond to NW, N, NE, W, E, SW, S, SE).

Value

A vector

References

Kennel, M.B., Brown, R., Abarbanel, H.D.I., 1992. Determining embedding dimension for phase-space reconstruction using a geometrical construction. Physical Review A 45, 3403–3411.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

spEDM::fnn(columbus,"crime")
```

---

gccm *geographical convergent cross mapping*

Description

geographical convergent cross mapping

Usage

```
## S4 method for signature 'sf'
gccm(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
  theta = 1,
  algorithm = "simplex",
  threads = detectThreads(),
  detrend = TRUE,
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE,
  nb = NULL
)

## S4 method for signature 'SpatRaster'
gccm(
```

```
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
  theta = 1,
  algorithm = "simplex",
  threads = detectThreads(),
  detrend = TRUE,
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE,
  grid.coord = TRUE,
  embed.direction = 0,
  win.ratio = 0
)
```

Arguments

`data`	observation data.
`cause`	name of causal variable.
`effect`	name of effect variable.
`libsizes`	(optional) number of spatial units used (input needed: `vector` - spatial vector, `matrix` - spatial raster).
`E`	(optional) embedding dimensions.
`k`	(optional) number of nearest neighbors.
`tau`	(optional) step of spatial lags.
`style`	(optional) embedding style (`0` includes current state, `1` excludes it).
`stack`	(optional) whether to stack embeddings.
`lib`	(optional) libraries indices (input requirement same as `libsizes`).
`pred`	(optional) predictions indices (input requirement same as `libsizes`).
`dist.metric`	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).
`dist.average`	(optional) whether to average distance.
`theta`	(optional) weighting parameter for distances, useful when `algorithm` is `smap`.
`algorithm`	(optional) prediction algorithm.
`threads`	(optional) number of threads to use.

detrend	(optional) whether to remove the linear trend.
parallel.level	(optional) level of parallelism, low or high.
bidirectional	(optional) whether to examine bidirectional causality.
progressbar	(optional) whether to show the progress bar.
nb	(optional) neighbours list.
grid.coord	(optional) whether to detrend using cell center coordinates (TRUE) or row/column numbers (FALSE).
embed.direction	(optional) direction selector for embeddings (0 returns all directions, 1-8 correspond to NW, N, NE, W, E, SW, S, SE).
win.ratio	(optional) ratio of sliding window scale to speed up state-space predictions.

Value

A list

xmap cross mapping results

varname names of causal and effect variables

bidirectional whether to examine bidirectional causality

References

Gao, B., Yang, J., Chen, Z., Sugihara, G., Li, M., Stein, A., Kwan, M.-P., Wang, J., 2023. Causal inference from cross-sectional earth system data with geographical convergent cross mapping. Nature Communications 14.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

g = spEDM::gccm(columbus,"hoval","crime",libsizes = seq(5,45,5),E = 6)
g
plot(g,ylimits = c(0,0.85))
```

---

gcmc *geographical cross mapping cardinality*

Description

geographical cross mapping cardinality

Usage

```
## S4 method for signature 'sf'
gcmc(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  k = min(E^2),
  tau = 1,
  style = 1,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  threads = detectThreads(),
  detrend = FALSE,
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE,
  nb = NULL
)

## S4 method for signature 'SpatRaster'
gcmc(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  k = min(E^2),
  tau = 1,
  style = 1,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  threads = detectThreads(),
  detrend = FALSE,
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE,
  grid.coord = TRUE
)
```

Arguments

data	observation data.
cause	name of causal variable.
effect	name of effect variable.

libsizes	(optional) number of spatial units used (input needed: vector - spatial vector, matrix - spatial raster).
E	(optional) embedding dimensions.
k	(optional) number of nearest neighbors.
tau	(optional) step of spatial lags.
style	(optional) embedding style (0 includes current state, 1 excludes it).
lib	(optional) libraries indices (input requirement same as libsizes).
pred	(optional) predictions indices (input requirement same as libsizes).
dist.metric	(optional) distance metric (L1: Manhattan, L2: Euclidean).
threads	(optional) number of threads to use.
detrend	(optional) whether to remove the linear trend.
parallel.level	(optional) level of parallelism, low or high.
bidirectional	(optional) whether to examine bidirectional causality.
progressbar	(optional) whether to show the progress bar.
nb	(optional) neighbours list.
grid.coord	(optional) whether to detrend using cell center coordinates (TRUE) or row/column numbers (FALSE).

Value

A list

xmap cross mapping results

cs causal strength

varname names of causal and effect variables

bidirectional whether to examine bidirectional causality

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

g = spEDM::gcmc(columbus,"hoval","crime",E = 7,k = 19)
g
```

gpc *geographical pattern causality*

Description

geographical pattern causality

Usage

```
## S4 method for signature 'sf'
gpc(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  k = E + 2,
  tau = 1,
  style = 1,
  lib = NULL,
  pred = NULL,
  boot = 99,
  replace = FALSE,
  seed = 42L,
  dist.metric = "L2",
  zero.tolerance = max(k),
  relative = TRUE,
  weighted = TRUE,
  threads = detectThreads(),
  detrend = FALSE,
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE,
  nb = NULL
)

## S4 method for signature 'SpatRaster'
gpc(
  data,
  cause,
  effect,
  libsizes = NULL,
  E = 3,
  k = E + 2,
  tau = 1,
  style = 1,
  lib = NULL,
```

```
  pred = NULL,
  boot = 99,
  replace = FALSE,
  seed = 42L,
  dist.metric = "L2",
  zero.tolerance = max(k),
  relative = TRUE,
  weighted = TRUE,
  threads = detectThreads(),
  detrend = FALSE,
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE,
  grid.coord = TRUE
)
```

Arguments

data	observation data.
cause	name of causal variable.
effect	name of effect variable.
libsizes	(optional) number of spatial units used (input needed: `vector` - spatial vector, `matrix` - spatial raster).
E	(optional) embedding dimensions.
k	(optional) number of nearest neighbors.
tau	(optional) step of spatial lags.
style	(optional) embedding style (`0` includes current state, `1` excludes it).
lib	(optional) libraries indices (input requirement same as `libsizes`).
pred	(optional) predictions indices (input requirement same as `libsizes`).
boot	(optional) number of bootstraps to perform.
replace	(optional) should sampling be with replacement?
seed	(optional) random seed.
dist.metric	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).
zero.tolerance	(optional) maximum number of zeros tolerated in signature space.
relative	(optional) whether to calculate relative changes in embedding.
weighted	(optional) whether to weight causal strength.
threads	(optional) number of threads to use.
detrend	(optional) whether to remove the linear trend.
parallel.level	(optional) level of parallelism, `low` or `high`.
bidirectional	(optional) whether to examine bidirectional causality.
progressbar	(optional) whether to show the progress bar.
nb	(optional) neighbours list.
grid.coord	(optional) whether to detrend using cell center coordinates (`TRUE`) or row/column numbers (`FALSE`).

Value

A list

`xmap` cross mapping results (only present if `libsizes` is not `NULL`)

`causality` per-sample causality statistics (present if `libsizes` is `NULL`)

`summary` overall causal strength (present if `libsizes` is `NULL`)

`pattern` pairwise pattern relationships (present if `libsizes` is `NULL`)

`varname` names of causal and effect variables

`bidirectional` whether to examine bidirectional causality

References

Zhang, Z., Wang, J., 2025. A model to identify causality for geographic patterns. International Journal of Geographical Information Science 1–21.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

spEDM::gpc(columbus,"crime","hoval",E = 7,k = 9)

# convergence diagnostics
g = spEDM::gpc(columbus,"crime","hoval",libsizes = seq(7,49,7),E = 7,k = 9)
plot(g)
```

---

ic *optimal parameter search for intersectional cardinality*

Description

optimal parameter search for intersectional cardinality

Usage

```
## S4 method for signature 'sf'
ic(
  data,
  column,
  target,
  E = 2:10,
  k = E + 2,
  tau = 1,
  style = 1,
  lib = NULL,
  pred = NULL,
```

```
  dist.metric = "L2",
  threads = detectThreads(),
  detrend = FALSE,
  nb = NULL
)

## S4 method for signature 'SpatRaster'
ic(
  data,
  column,
  target,
  E = 2:10,
  k = E + 2,
  tau = 1,
  style = 1,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  threads = detectThreads(),
  detrend = FALSE,
  grid.coord = TRUE
)
```

Arguments

`data`	observation data.
`column`	name of library variable.
`target`	name of target variable.
`E`	(optional) embedding dimensions.
`k`	(optional) number of nearest neighbors used.
`tau`	(optional) step of spatial lags.
`style`	(optional) embedding style (`0` includes current state, `1` excludes it).
`lib`	(optional) libraries indices (input needed: `vector` - spatial vector, `matrix` - spatial raster).
`pred`	(optional) predictions indices (input requirement same as `lib`).
`dist.metric`	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).
`threads`	(optional) number of threads to use.
`detrend`	(optional) whether to remove the linear trend.
`nb`	(optional) neighbours list.
`grid.coord`	(optional) whether to detrend using cell center coordinates (`TRUE`) or row/column numbers (`FALSE`).

Value

A list

`xmap` cross mapping performance

`varname` name of target variable

`method` method of cross mapping

References

Tao, P., Wang, Q., Shi, J., Hao, X., Liu, X., Min, B., Zhang, Y., Li, C., Cui, H., Chen, L., 2023. Detecting dynamical causality by intersection cardinal concavity. Fundamental Research.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

spEDM::ic(columbus,"hoval","crime",E = 7,k = 15:25)
```

`multiview` *multiview embedding forecast*

Description

multiview embedding forecast

Usage

```
## S4 method for signature 'sf'
multiview(
  data,
  column,
  target,
  nvar,
  E = 3,
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
  top = NULL,
  threads = detectThreads(),
  detrend = TRUE,
  nb = NULL
```

```
)

## S4 method for signature 'SpatRaster'
multiview(
  data,
  column,
  target,
  nvar,
  E = 3,
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
  top = NULL,
  threads = detectThreads(),
  detrend = TRUE,
  grid.coord = TRUE
)
```

Arguments

`data`	observation data.
`column`	name of library variable.
`target`	name of target variable.
`nvar`	number of variable combinations.
`E`	(optional) embedding dimensions.
`k`	(optional) number of nearest neighbors used.
`tau`	(optional) step of spatial lags.
`style`	(optional) embedding style (`0` includes current state, `1` excludes it).
`stack`	(optional) whether to stack embeddings.
`lib`	(optional) libraries indices (input needed: `vector` - spatial vector, `matrix` - spatial raster).
`pred`	(optional) predictions indices (input requirement same as `lib`).
`dist.metric`	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).
`dist.average`	(optional) whether to average distance.
`top`	(optional) number of reconstructions used in MVE forecast.
`threads`	(optional) number of threads to use.
`detrend`	(optional) whether to remove the linear trend.
`nb`	(optional) neighbours list.
`grid.coord`	(optional) whether to detrend using cell center coordinates (`TRUE`) or row/column numbers (`FALSE`).

Value

A vector (when input is sf object) or matrix

References

Ye H., and G. Sugihara, 2016. Information leverage in interconnected ecosystems: Overcoming the curse of dimensionality. Science 353:922-925.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

spEDM::multiview(columbus,
                 column = c("inc","crime","open","plumb","discbd"),
                 target = "hoval", nvar = 3)
```

---

pc *optimal parameter search for pattern causality*

Description

optimal parameter search for pattern causality

Usage

```
## S4 method for signature 'sf'
pc(
  data,
  column,
  target,
  E = 2:10,
  k = E + 2,
  tau = 1,
  style = 1,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  zero.tolerance = max(k),
  relative = TRUE,
  weighted = TRUE,
  maximize = "dark",
  threads = detectThreads(),
  detrend = FALSE,
  nb = NULL
)
```

```
## S4 method for signature 'SpatRaster'
pc(
  data,
  column,
  target,
  E = 2:10,
  k = E + 2,
  tau = 1,
  style = 1,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  zero.tolerance = max(k),
  relative = TRUE,
  weighted = TRUE,
  maximize = "dark",
  threads = detectThreads(),
  detrend = FALSE,
  grid.coord = TRUE
)
```

Arguments

data	observation data.
column	name of library variable.
target	name of target variable.
E	(optional) embedding dimensions.
k	(optional) number of nearest neighbors used.
tau	(optional) step of spatial lags.
style	(optional) embedding style (0 includes current state, 1 excludes it).
lib	(optional) libraries indices (input needed: vector - spatial vector, matrix - spatial raster).
pred	(optional) predictions indices (input requirement same as lib).
dist.metric	(optional) distance metric (L1: Manhattan, L2: Euclidean).
zero.tolerance	(optional) maximum number of zeros tolerated in signature space.
relative	(optional) whether to calculate relative changes in embeddings.
weighted	(optional) whether to weight causal strength.
maximize	(optional) causality metric to maximize: one of "positive", "negative", or "dark".
threads	(optional) number of threads to use.
detrend	(optional) whether to remove the linear trend.
nb	(optional) neighbours list.
grid.coord	(optional) whether to detrend using cell center coordinates (TRUE) or row/column numbers (FALSE).

Value

A list

`xmap` cross mapping performance

`varname` name of target variable

`method` method of cross mapping

`maximize` maximized causality metric

References

Stavroglou, S.K., Pantelous, A.A., Stanley, H.E., Zuev, K.M., 2020. Unveiling causal interactions in complex systems. Proceedings of the National Academy of Sciences 117, 7599–7605.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

spEDM::pc(columbus,"crime","hoval",E = 5:10,maximize = "negative")
```

`sc.test` *spatial causality test*

Description

spatial causality test

Usage

```
## S4 method for signature 'sf'
sc.test(
  data,
  cause,
  effect,
  k,
  block = 3,
  boot = 399,
  seed = 42L,
  base = 2,
  lib = NULL,
  pred = NULL,
  threads = detectThreads(),
  detrend = TRUE,
  normalize = FALSE,
  progressbar = FALSE,
  nb = NULL
)
```

```
## S4 method for signature 'SpatRaster'
sc.test(
  data,
  cause,
  effect,
  k,
  block = 3,
  boot = 399,
  seed = 42L,
  base = 2,
  lib = NULL,
  pred = NULL,
  threads = detectThreads(),
  detrend = TRUE,
  normalize = FALSE,
  progressbar = FALSE,
  grid.coord = TRUE
)
```

Arguments

data	observation data.
cause	name of causal variable.
effect	name of effect variable.
k	(optional) number of nearest neighbors used in symbolization.
block	(optional) number of blocks used in spatial block bootstrap.
boot	(optional) number of bootstraps to perform.
seed	(optional) random seed.
base	(optional) logarithm base.
lib	(optional) libraries indices (input needed: `vector` - spatial vector, `matrix` - spatial raster).
pred	(optional) predictions indices (input requirement same as `lib`).
threads	(optional) number of threads to use.
detrend	(optional) whether to remove the linear trend.
normalize	(optional) whether to normalize the result.
progressbar	(optional) whether to show the progress bar.
nb	(optional) neighbours list.
grid.coord	(optional) whether to detrend using cell center coordinates (`TRUE`) or row/column numbers (`FALSE`).

Value

A list

`sc` statistic for spatial causality

`varname` names of causal and effect variables

References

Herrera, M., Mur, J., Ruiz, M., 2016. Detecting causal relationships between spatial processes. Papers in Regional Science 95, 577–595.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

spEDM::sc.test(columbus,"hoval","crime",k = 15)
```

---

scpcm *spatially convergent partial cross mapping*

Description

spatially convergent partial cross mapping

Usage

```
## S4 method for signature 'sf'
scpcm(
  data,
  cause,
  effect,
  conds,
  libsizes = NULL,
  E = 3,
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
  theta = 1,
  algorithm = "simplex",
  threads = detectThreads(),
  detrend = TRUE,
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE,
  cumulate = FALSE,
  nb = NULL
)
```

```
## S4 method for signature 'SpatRaster'
scpcm(
  data,
  cause,
  effect,
  conds,
  libsizes = NULL,
  E = 3,
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
  theta = 1,
  algorithm = "simplex",
  threads = detectThreads(),
  detrend = TRUE,
  parallel.level = "low",
  bidirectional = TRUE,
  progressbar = TRUE,
  cumulate = FALSE,
  grid.coord = TRUE,
  embed.direction = 0,
  win.ratio = 0
)
```

Arguments

`data`	observation data.
`cause`	name of causal variable.
`effect`	name of effect variable.
`conds`	name of conditioning variables.
`libsizes`	(optional) number of spatial units used (input needed: `vector` - spatial vector, `matrix` - spatial raster).
`E`	(optional) embedding dimensions.
`k`	(optional) number of nearest neighbors.
`tau`	(optional) step of spatial lags.
`style`	(optional) embedding style (`0` includes current state, `1` excludes it).
`stack`	(optional) whether to stack embeddings.
`lib`	(optional) libraries indices (input requirement same as `libsizes`).
`pred`	(optional) predictions indices (input requirement same as `libsizes`).
`dist.metric`	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).

dist.average	(optional) whether to average distance.
theta	(optional) weighting parameter for distances, useful when algorithm is smap.
algorithm	(optional) prediction algorithm.
threads	(optional) number of threads to use.
detrend	(optional) whether to remove the linear trend.
parallel.level	(optional) level of parallelism, low or high.
bidirectional	(optional) whether to examine bidirectional causality.
progressbar	(optional) whether to show the progress bar.
cumulate	(optional) serial or cumulative computation of partial cross mapping.
nb	(optional) neighbours list.
grid.coord	(optional) whether to detrend using cell center coordinates (TRUE) or row/column numbers (FALSE).
embed.direction	(optional) direction selector for embeddings (0 returns all directions, 1-8 correspond to NW, N, NE, W, E, SW, S, SE).
win.ratio	(optional) ratio of sliding window scale to speed up state-space predictions.

Value

A list

pxmap partial cross mapping results

xmap cross mapping results

varname names of causal, effect and conditioning variables

bidirectional whether to examine bidirectional causality

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

g = spEDM::scpcm(columbus,"hoval","crime","inc",libsizes = seq(5,45,5),E = 6)
g
plot(g,ylimits = c(-0.1,0.4),ybreaks = seq(-0.1,0.4,0.1))
```

---

simplex *optimal parameter search for simplex projection*

Description

optimal parameter search for simplex projection

Usage

```
## S4 method for signature 'sf'
simplex(
  data,
  column,
  target,
  E = 2:10,
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
  threads = detectThreads(),
  detrend = TRUE,
  nb = NULL
)

## S4 method for signature 'SpatRaster'
simplex(
  data,
  column,
  target,
  E = 2:10,
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
  threads = detectThreads(),
  detrend = TRUE,
  grid.coord = TRUE,
  embed.direction = 0
)
```

Arguments

data	observation data.
column	name of library variable.
target	name of target variable.
E	(optional) embedding dimensions.
k	(optional) number of nearest neighbors used.

`tau`	(optional) step of spatial lags.
`style`	(optional) embedding style (`0` includes current state, `1` excludes it).
`stack`	(optional) whether to stack embeddings.
`lib`	(optional) libraries indices (input needed: `vector` - spatial vector, `matrix` - spatial raster).
`pred`	(optional) predictions indices (input requirement same as `lib`).
`dist.metric`	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).
`dist.average`	(optional) whether to average distance.
`threads`	(optional) number of threads to use.
`detrend`	(optional) whether to remove the linear trend.
`nb`	(optional) neighbours list.
`grid.coord`	(optional) whether to detrend using cell center coordinates (`TRUE`) or row/column numbers (`FALSE`).
`embed.direction`	(optional) direction selector for embeddings (`0` returns all directions, `1-8` correspond to NW, N, NE, W, E, SW, S, SE).

Value

A list

`xmap` forecast performance

`varname` name of target variable

`method` method of cross mapping

References

Sugihara G. and May R. 1990. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. Nature, 344:734-741.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

spEDM::simplex(columbus,"inc","crime")
```

slm *spatial logistic map*

Description

spatial logistic map

Usage

```
## S4 method for signature 'sf'
slm(
  data,
  x = NULL,
  y = NULL,
  z = NULL,
  k = 4,
  step = 15,
  alpha_x = 0.28,
  alpha_y = 0.25,
  alpha_z = 0.22,
  beta_xy = 0.05,
  beta_xz = 0.05,
  beta_yx = 0.2,
  beta_yz = 0.2,
  beta_zx = 0.35,
  beta_zy = 0.35,
  threshold = Inf,
  transient = 1,
  interact = "local",
  aggregate_fn = NULL,
  noise = 0,
  seed = 42L,
  nb = NULL
)

## S4 method for signature 'SpatRaster'
slm(
  data,
  x = NULL,
  y = NULL,
  z = NULL,
  k = 4,
  step = 15,
  alpha_x = 0.28,
  alpha_y = 0.25,
  alpha_z = 0.22,
  beta_xy = 0.05,
```

```
  beta_xz = 0.05,
  beta_yx = 0.2,
  beta_yz = 0.2,
  beta_zx = 0.35,
  beta_zy = 0.35,
  threshold = Inf,
  transient = 1,
  interact = "local",
  aggregate_fn = NULL,
  noise = 0,
  seed = 42L
)
```

Arguments

`data`	observation data.
`x`	(optional) name of first spatial variable.
`y`	(optional) name of second spatial variable.
`z`	(optional) name of third spatial variable.
`k`	(optional) number of neighbors to used.
`step`	(optional) number of simulation time steps.
`alpha_x`	(optional) growth parameter for x.
`alpha_y`	(optional) growth parameter for y.
`alpha_z`	(optional) growth parameter for z.
`beta_xy`	(optional) cross-inhibition from x to y.
`beta_xz`	(optional) cross-inhibition from x to z.
`beta_yx`	(optional) cross-inhibition from y to x.
`beta_yz`	(optional) cross-inhibition from y to z.
`beta_zx`	(optional) cross-inhibition from z to x.
`beta_zy`	(optional) cross-inhibition from z to y.
`threshold`	(optional) set to `NaN` if the absolute value exceeds this threshold.
`transient`	(optional) transients to be excluded from the results.
`interact`	(optional) type of cross-variable interaction (`local` or `neighbors`).
`aggregate_fn`	(optional) custom aggregation function (must accept a numeric vector and return a single numeric value).
`noise`	(optional) standard deviation of white noise.
`seed`	(optional) random seed.
`nb`	(optional) neighbours list.

Value

A list

References

Willeboordse, F.H., 2003. The spatial logistic map as a simple prototype for spatiotemporal chaos. Chaos: An Interdisciplinary Journal of Nonlinear Science 13, 533–540.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))
columbus$inc = sdsfun::normalize_vector(columbus$inc)
spEDM::slm(columbus,"inc")
```

---

smap *optimal parameter search for s-mapping*

Description

optimal parameter search for s-mapping

Usage

```
## S4 method for signature 'sf'
smap(
  data,
  column,
  target,
  E = 3,
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
 theta = c(0, 1e-04, 3e-04, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 0.5, 0.75, 1, 1.5, 2, 3,
    4, 6, 8),
  threads = detectThreads(),
  detrend = TRUE,
  nb = NULL
)

## S4 method for signature 'SpatRaster'
smap(
  data,
  column,
  target,
  E = 3,
```

```
  k = E + 2,
  tau = 1,
  style = 1,
  stack = FALSE,
  lib = NULL,
  pred = NULL,
  dist.metric = "L2",
  dist.average = TRUE,
 theta = c(0, 1e-04, 3e-04, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 0.5, 0.75, 1, 1.5, 2, 3,
    4, 6, 8),
  threads = detectThreads(),
  detrend = TRUE,
  grid.coord = TRUE,
  embed.direction = 0
)
```

Arguments

`data`	observation data.
`column`	name of library variable.
`target`	name of target variable.
`E`	(optional) embedding dimensions.
`k`	(optional) number of nearest neighbors used.
`tau`	(optional) step of spatial lags.
`style`	(optional) embedding style (`0` includes current state, `1` excludes it).
`stack`	(optional) whether to stack embeddings.
`lib`	(optional) libraries indices (input needed: `vector` - spatial vector, `matrix` - spatial raster).
`pred`	(optional) predictions indices (input requirement same as `lib`).
`dist.metric`	(optional) distance metric (`L1`: Manhattan, `L2`: Euclidean).
`dist.average`	(optional) whether to average distance.
`theta`	(optional) weighting parameter for distances.
`threads`	(optional) number of threads to use.
`detrend`	(optional) whether to remove the linear trend.
`nb`	(optional) neighbours list.
`grid.coord`	(optional) whether to detrend using cell center coordinates (`TRUE`) or row/column numbers (`FALSE`).
`embed.direction`	(optional) direction selector for embeddings (`0` returns all directions, `1-8` correspond to NW, N, NE, W, E, SW, S, SE).

Value

A list

`xmap` forecast performance

`varname` name of target variable

`method` method of cross mapping

References

Sugihara G. 1994. Nonlinear forecasting for the classification of natural time series. Philosophical Transactions: Physical Sciences and Engineering, 348 (1688):477-495.

Examples

```
columbus = sf::read_sf(system.file("case/columbus.gpkg",package="spEDM"))

spEDM::smap(columbus,"inc","crime",E = 5,k = 6)
```

Index