Package ‘spatialising’

November 27, 2023

Title Ising Model for Spatial Data

Version 0.6.0

Description Performs simulations of binary spatial raster data using the Ising model (Ising (1925) <doi:10.1007/BF02980577>; Onsager (1944) <doi:10.1103/PhysRev.65.117>). It allows to set a few parameters that represent internal and external pressures, and the number of simulations (Stepinski and Nowosad (2023) <doi:10.1098/rsos.231005>).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports comat, Rcpp, terra

URL `https://github.com/Nowosad/spatialising`

BugReports `https://github.com/Nowosad/spatialising/issues`

Suggests covr, knitr, rmarkdown, optimization, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

LazyData false

LinkingTo Rcpp

VignetteBuilder knitr

NeedsCompilation yes

Author Jakub Nowosad [aut, cre] (<`https://orcid.org/0000-0002-1057-3721`>)

Maintainer Jakub Nowosad <`nowosad.jakub@gmail.com`>

Repository CRAN

Date/Publication 2023-11-27 17:10:02 UTC

R topics documented:

`composition_index` *Composition imbalance index*

Description

Calculates composition imbalance index (also known as the m index) – a sum of cell's values over the entire site divided by the number of cell in the site. m has a range from -1 (site completely dominated by the -1 values) to 1 (site completely dominated by the 1 values).

Usage

```
composition_index(x)
```

Arguments

`x`	SpatRaster or matrix containing two values: -1 and 1

Value

A numeric vector

See Also

`kinetic_ising()`

Examples

```
data(r_start, package = "spatialising")
composition_index(r_start)
ts1 = kinetic_ising(r_start, B = -0.3, J = 0.7)
composition_index(ts1)
ts2 = kinetic_ising(r_start, B = -0.3, J = 0.7, updates = 2)
composition_index(ts2)


  library(terra)
  r1 = rast(system.file("raster/r_start.tif", package = "spatialising"))
```

```
composition_index(r1)
r2 = kinetic_ising(r1, B = -0.3, J = 0.7)
composition_index(r2)
```

kinetic_ising *Ising model for spatial data*

Description

Performs simulations based on the given parameters of the Ising model

Usage

```
kinetic_ising(
  x,
  B,
  J,
  updates = 1,
  iter,
  rule = "glauber",
  inertia = 0,
  version = 1,
  progress = FALSE
)
```

Arguments

`x` SpatRaster or matrix containing two values: -1 and 1

`B` External pressure (positive or negative): it tries to align cells' values with its sign

`J` Strength of the local autocorrelation tendency (always positive): it tries to align signs of neighboring cells

`updates` Specifies how many sets of iterations are performed on the input object. The output of this function has as many layers as the `updates` value.

`iter` Specifies how many iterations are performed on the input object. By default it equals to the number of values in the input object.

`rule` IM temporal evolution rule: either `"glauber"` (default) or `"metropolis"`

`inertia` Represents the modification of the algorithm aimed at suppressing the salt-and-pepper noise of the focus category present when simulating evolution of a coarse-textured pattern. With Q > 0, small patches of the focus category are not generated, thus eliminating the salt-and-pepper noise of the focus category

`version` By default, `1`, the `x` object is converted into a matrix (fast, but can be memory consuming); `version = 2` has a lower RAM impact, but is much slower

`progress` TRUE/FALSE

Value

Object of the same class as x with the number of layers specified by updates

References

Ising, E., 1924. Beitrag zur theorie des ferro-und paramagnetismus. Ph.D. thesis, Grefe & Tiedemann.

Onsager, L., 1944. Crystal statistics. I. A two-dimensional model with an order-disorder transition. Physical Review 65 (3-4), 117.

Brush, S. G., 1967. History of the Lenz-Ising model. Reviews of modern physics 39 (4), 883.

Cipra, B. A., 1987. An introduction to the Ising model. The American Mathematical Monthly 94 (10), 937–959.

Examples

```
data(r_start, package = "spatialising")
ts1 = kinetic_ising(r_start, B = -0.3, J = 0.7)
ts10 = kinetic_ising(r_start, B = -0.3, J = 0.7, updates = 10)


  r1 = terra::rast(system.file("raster/r_start.tif", package = "spatialising"))
  terra::plot(r1)
  r2 = kinetic_ising(r1, B = -0.3, J = 0.7)
  terra::plot(r2)

  library(terra)
  ri1 = kinetic_ising(r1, B = -0.3, J = 0.7, updates = 9)
  plot(ri1)

  ri2 = kinetic_ising(r1, B = 0.3, J = 0.7, updates = 9)
  plot(ri2)

  ri3 = kinetic_ising(r1, B = -0.3, J = 0.4, updates = 9)
  plot(ri3)
```

kinetic_ising_ensemble

Ensemble of Ising models for spatial data

Description

Creates an ensemble of simulations based on the given parameters of the Ising model

Usage

```
kinetic_ising_ensemble(runs, ...)
```

Arguments

`runs`	A number of simulations to perform
`...`	Arguments for `kinetic_ising()`

Value

A list of objects of the same class as `x`

See Also

`kinetic_ising()`, `kinetic_ising_exemplar()`

Examples

```
data(r_start, package = "spatialising")
l = kinetic_ising_ensemble(100, r_start, B = -0.3, J = 0.7)


  library(terra)
  r1 = rast(system.file("raster/r_start.tif", package = "spatialising"))
  plot(r1)
  r2 = kinetic_ising_ensemble(100, r1, B = -0.3, J = 0.7)
```

`kinetic_ising_exemplar` *Exemplar of an Ising model for spatial data*

Description

Creates an ensemble of simulations based on the given parameters of the Ising model and selects an exemplar (a model that is closest to the average of the ensemble)

Usage

```
kinetic_ising_exemplar(runs, ...)
```

Arguments

`runs`	A number of simulations to perform
`...`	Arguments for `kinetic_ising()`

Value

Object of the same class as `x`

See Also

kinetic_ising(), kinetic_ising_ensemble()

Examples

```
data(r_start, package = "spatialising")
l = kinetic_ising_exemplar(100, r_start, B = -0.3, J = 0.7)


  library(terra)
  r1 = rast(system.file("raster/r_start.tif", package = "spatialising"))
  plot(r1)
  r2 = kinetic_ising_exemplar(100, r1, B = -0.3, J = 0.7)
  plot(r2)
```

r_end *An example matrix object*

Description

A matrix has 50 columns and 50 rows. The matrix contains two values: -1 and 1. This object was created with `r_end = kinetic_ising(r_start, B = -0.3, J = 0.7)`

Usage

```
data(r_end)
```

Format

A matrix

r_start *An example matrix object*

Description

A matrix has 50 columns and 50 rows. The matrix contains two values: -1 and 1.

Usage

```
data(r_start)
```

Format

A matrix

`r_start.tif` *An example binary raster*

Description

A raster file covering an area of 50x50 cells. The raster file contains two values: -1 and 1. `system.file("raster/r_start.tif", package = "spatialising")`

Format

A raster file

`texture_index` *Texture index*

Description

Calculates texture index – an average (over an array) of a product of the values of neighboring cells. The value of texture index is between 0 (fine texture), and 1 (coarse texture).

Usage

```
texture_index(x, ...)
```

Arguments

`x`	SpatRaster or matrix containing two values: -1 and 1
`...`	Arguments for `comat::get_coma()`

Value

A numeric vector

See Also

`kinetic_ising()`

Examples

```
data(r_start, package = "spatialising")
texture_index(r_start)
ts1 = kinetic_ising(r_start, B = -0.3, J = 0.7)
texture_index(ts1)
ts2 = kinetic_ising(r_start, B = -0.3, J = 0.7, updates = 2)
texture_index(ts2)


  library(terra)
  r1 = rast(system.file("raster/r_start.tif", package = "spatialising"))
  texture_index(r1)
  r2 = kinetic_ising(r1, B = -0.3, J = 0.7)
  texture_index(r2)
```

Index