Package ‘spatialsegregation’

April 18, 2017

Type Package
Title Segregation Measures for Multitype Spatial Point Patterns
Version 2.44
Date 2017-04-18
Author Tuomas Rajala
Maintainer Tuomas Rajala <tuomas.rajala@iki.fi>
Description Summaries for measuring segregation/mingling in multitype spatial point patterns with graph based neighbourhood description. Included indices: Mingling, Shannon, Simpson (also the non-spatial) Included functionals: Mingling, Shannon, Simpson, ISAR, MCI. Included neighbourhoods: Geometric, k-nearest neighbours, Gabriel, Delaunay. Dixon's test.
License GPL
Depends spatstat
RoxygenNote 6.0.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-04-18 16:06:17 UTC

R topics documented:

spatialsegregation-package ........................................... 2
biomassF ................................................................. 3
dixon ................................................................. 4
exposurepps ............................................................. 4
helper functions ....................................................... 5
isarF ................................................................. 6
mciF ................................................................. 7
minglingF ............................................................. 8
print.segtest ........................................................... 9
segregationFun ......................................................... 9
sg.modify.pp .......................................................... 11
shannonF ............................................................ 11
simpsonF ........................................................... 12
Spatial Segregation Measures

Description

Collection of measures or summaries of spatial multitype exposure: segregation vs. mingling of different types of points in a plane.

Details

This is a collection of summaries for multitype spatial point patterns (see package spatstat for more).

The package is developed for an article Rajala & Illian 2010, and provides summaries for detecting simple inter-type effects in the pattern.

See the help of the functions for further information.

Package provides an example dataset object called exposurepps, documented separately.

Also, the Dixon bivariate test based on contingency tables is available.

Functions

- segregationFun: General calculation function, please use one of the following wrappers:
  - minglingF: Mingling index
  - shannonF: Spatial Shannon index
  - simpsonF: Spatial Simpson index
  - isarf: ISAR function

- mingling.index: Shortcut for a single value
- shannon.index: =''=
- simpson.index: =''=
- isar.index: =''=

- dixon: Dixon's 2-type contingency table tests

Author(s)

Tuomas Rajala tuomas.rajala@iki.fi

References


Examples

data(exposurepps)
help(exposurepps)

biomassF

Local biomass summary

Description

Compute the biomass around points as a function of spatial scale.

Usage

biomassF(X, r = NULL, target = NULL, v2 = FALSE, ...)

Arguments

X Multitype point pattern of class ppp (see package ‘spatstat’). The biomass (e.g. size) is to be in an element $mass.

r Vector of sizes for neighbourhoods, e.g. ranges in the geometric graph neighbourhoods.

target Default NULL. Calculate only for target type. If NULL compute mean over all types.

v2 Logical. Return the average biomass instead of just sum (development nomenclature)

... Further parameters for the function segregationFun.

Value

Returns an fv-object, see spatstat for more information.
dixon

Dixon’s 2-type contingency table tests

Description

Dixon’s 2-type contingency table tests

Usage

dixon(xL prepr \] 0I

Arguments

x bivariate ppp
prepr for large patterns, makes it a bit easier to find nearest neighbours.

Details

Computes the tests of segregation using nearest neighbour contingency tables introduced by Philip Dixon in his paper “Testing spatial segregation using a nearest-neighbor contingency table”, Ecology, 75, p.1940-1948 (1994). The tests are an improvement on the Pielou’s test of segregation. The test is defined only for two-type spatial pattern.

References


exposureepps

Example datasets for package spatialsegregation

Description

Example datasets with 9 different scattering+exposure combinations.

Usage

data(exposureepps)

Format

A list with nine elements of class ppp.
Details

A list of 9 point patterns with different degrees of intra-species clustering and inter-species mingling.

The patterns are synthetically produced using a combination of the functions of this package as an energy function of a Gibbs model.

Author(s)

Tuomas Rajala tuomas.rajala@iki.fi

Examples

data(exposurepps)
par(mfrow=c(3,3), mar=c(2,2,2,2))
for(i in 1:9) plot(exposurepps[[i]])

# upper row has strong inter-type mingling effect
# lower row has strong inter-type repulsion or segregation
# left column has strong intra-type clustering
# right column has strong intra-type repulsion.

par(mfrow=c(3,3), mar=c(3,3,4,3))
for(i in 1:9) plot( isarf(exposurepps[[i]]), cbind(ISARmean,tho=par)

Description

Small functions included in package spatial.segregation, used for manipulation of forest datasets which have dbh-values (pp with an element $dbh$).

Usage

clean.up.data(pp, dbh = 10, atleast = 10)
freqs(pp)
minusID(pp, minusR, dbh, atleast=0)
shake(pp, a = 0.001)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pp</td>
<td>Multitype point pattern (see package 'spatstat')</td>
</tr>
<tr>
<td>atleast</td>
<td>Include specii with abundance at least atleast.</td>
</tr>
<tr>
<td>dbh</td>
<td>Include only those points with dbh at least dbh.</td>
</tr>
<tr>
<td>minusR</td>
<td>Range from the border withing which to exclude points (used for correction of estimates).</td>
</tr>
<tr>
<td>a</td>
<td>Size of displacement: x+Unif(-a,a), y+Unif(-a,a).</td>
</tr>
</tbody>
</table>
Details

Small functions to manipulate multitype point patterns.
cleanNupNdata: Returns a subsample fullfilling the given constrains.
freqs: Returns the abundance vector.
minusID: Returns a 0-1-vector indicating inclusion in a simple minus-correction.
shake: Shakes the pattern, i.e. adds a random displacement shift to each point.

Author(s)

Tuomas Rajala tuomas.rajala@iki.fi

isarF Individual Species Area Relationship

Description

Compute the Individual Species Area Relationship (ISAR) or Local Species Richness, for a given multitype point pattern.

Usage

NNNI
isarNindex(xL r] TL ntype] "knn"L NNNI

Arguments

X Multitype point pattern of class ppp (see package ‘spatstat’)
r Vector of sizes for neighbourhoods, e.g. geometric graph with different ranges.
target Default NULL. Calculate only for target type. If NULL computes for each type + mean over all types.
v2 Logical. Estimate species-to-neighbours-ratio instead of just total number of species.
v3 Logical. Instead of summing number 1 for each species present, sum the average X$mass of each species present.
v4 Logical. Estimate ISAR using empty space probabilities instead of direct counts (equals the normal version in all my tests)
... Further parameters for the function segregationFun.
notype Sets the n’hood type to knn by default in isar.index.
Details

Extension of ISAR-function introduced in WGGH07. In effect calculates the expected amount of different types present in the neighbourhood of a point in the pattern.

The function isarF is the calculation function for different neighbourhoods. Uses function segregationFun.

The function isar.index is a shortcut to get a single value for the pattern. Uses 4-nn graph by default.

Functions

- isar.index: Shortcut for 4-nearest neighbour value.

References


---

mcif  

**Mean Composite Information**

Description

Compute the Mean Composite Information for a given multitype point pattern. See Podani&Czaran 1997.

Usage

mcif(X, r = NULL, target = NULL, ...)

Arguments

- **X**
  Multitype point pattern of class ppp (see package 'spatstat')
- **r**
  Vector of sizes for neighbourhoods, e.g. geometric graph with different ranges.
- **target**
  If given, look at the surroundings of this type only.
- **...**
  Further parameters for the function segregationFun.

Value

Returns an fv-object, see spatstat for more information.

References

**minglingF**  

*Spatial Minling index*

**Description**

Compute the Mingling index for a given multitype point pattern.

**Usage**

```
minglingF(x, r = NULL, target = NULL, ratio = FALSE, ...)

mingling.index(x, r = 4, ntype = "knn", ...)
```

**Arguments**

- `x`: Multitype point pattern of class `ppp` (see package 'spatstat')
- `r`: Vector of sizes for neighbourhoods, e.g. geometric graph with different ranges.
- `target`: Default NULL. Calculate only for target type. If NULL computes for each type + mean over all types.
- `ratio`: Default FALSE. If TRUE, scale the typewise values $D_{m_{tau}}$ using formula $(1-M_{tau})/lambda_{tau}$ which equals 1 for Poisson CSR.
- `...`: Further parameters for the function `segregationFun`.
- `ntype`: The original mingling index uses knn neighbourhood type.

**Details**

Extension of Mingling index introduced by Lewandowski&Pommerening 1997. Measures the proportion of alien points in the neighbourhood of a specific type typical point of the pattern.

If no specific type is given, the function takes mean over all types. A typewise value is more useful, so they are also included.

The function `minglingF` is the main calculation function. Uses function `segregationFun`.

The function `mingling.index` is a shortcut to get a single value for the pattern. Uses 4-nn graph by default, which is the original Mingling index used by Lewandowski&Pommerening 1997 and Graz 2004.

**Functions**

- `mingling.index`: Shortcut to 4-nearest neighbours index.

**References**


print.segtest

### Description

Print Segtest Objects

### Usage

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

### Arguments

- `x` segtest object
- `...` ignored

---

segregationFun

### Main computer function for spatial segregation

#### Description

Compute the spatial exposure (segregation vs. mingling) features from a given multitype point pattern. Usage of shortcuts minglingF, isarF, shannonF, simpsonF etc. highly recommended.

#### Usage

```
segregationFun(X, fun = "isar", r = NULL, ntype = "geometric",
        funpars = NULL, toroidal = FALSE, minusRange = TRUE, included = NULL,
        dbg = FALSE, doDists = FALSE, prepRange = 0, prepGraph = NULL,
        prepGraphIsTarget = FALSE, weightMatrix = NULL, translate = FALSE, ...)
```

#### Arguments

- `X` Multitype point pattern of class ppp (see package 'spatstat')
- `fun` Default "isar". Takes "isar", "mingling", "shannon", "simpson", "mci" and "biomass", see below.
- `r` Vector for the neighbourhood defining graph, e.g. "geometric" graph with different ranges. See below.
- `ntype` Default "geometric". Type of the neighbourhood graph. Accepts: "knn", "geometric", "delauney", "gabriel".
- `funpars` Default NULL. Parameter(s) for the measure. Mingling: c(i,j), where i= only for type i (0 for all), j=1 -> ratio version. ISAR: i, i=type (integer). Shannon: 0 or 1, see v2 in shannonF. Simpson: none.
This is the general function for computing the spatial exposure (segregation/mingling) features. Used by minglingF, shannonF, simpsonF, isarF, mciF and biomassF, which should be preferred for better (and nicer) outcome.

Possible neighbourhood relations include geometric, k-nearest neighbours, Delaunay, and Gabriel. Delaunay and Gabriel are parameter free, so given \( r \) has no meaning. In geometric graph, \( r \) is a vector of distances (sizes of the surrounding 'disc') and for k-nn \( r \) is the vector of neighbourhood abundances (so \( r \) is \( k \) in k-nn) for each point to consider in the calculation of the spatial exposure measures. In general, the basic type of spatial summary in literature uses 'geometric' graph with several ranges.

For geometric and knn, the calculations are done by shrinking the graph given by the largest value of \( r \). If dealing with large datasets, it is advisable to give preprocessing range, prepRange. The algorithm first calculates a geometric graph with parameter prepRange, and uses this as basis for finding the needed neighbourhoods. Speeds up calculations, but make sure prepRange is large enough e.g. in geometric, prepRange>max(r). prepGraph, if given, works as the preprocessed geometric graph (overrides prepRange), and can be computed using the package spatgraphs; useful for huge datasets, where the dominating graph needs to be computed only one. The dobotts option speeds up calculations by precomputing the pairwise distances but requires approx. \( n*(n-1)/2 * 32 \) bytes of memory.

For border correction, use minusRange for reduced border correction (for rectangular windows only). If using geometric or knn neighbourhoods, the option toroidal for toroidal correction is...
also available. The vector included can be given for more specific minus-correction, only those points with TRUE (or 1) value are used in calculation. However, the neighbourhoods will include all points.

**Value**

Returns an object of class fv, see spatstat for more details. Basically a list with the computed values and parameter values.

---

**sg.modify.pp**

Prepare the input data pattern object

---

**Description**

Prepares given point pattern object for computations.

**Usage**

sg.modify.pp(pp)

**Arguments**

- **pp**
  - Point pattern object

---

**shannonF**

Spatial Shannon index

---

**Description**

Compute the spatial and aspatial Shannon index for a given multitype point pattern.

**Usage**

shannonF(X, r = NULL, v2 = FALSE, ...)

shannon.index(X, spatial = FALSE, ...)

**Arguments**

- **X**
  - Multitype point pattern of class ppp (see package 'spatstat')
- **r**
  - Vector of sizes for neighbourhoods, e.g. geometric graph with different ranges.
- **v2**
  - If TRUE, use the real number of types in neighbourhoods as the log-base instead of total population type count.
- **...**
  - Further parameters for the function segregationFun.
- **spatial**
  - If FALSE, return the classical aspatial index value.
Details

The form of Shannon index is \( H = 1 - \frac{E(o)}{E(N)} \), where \( E(N) \) is the global entropy and \( E(o) \) is the local entropy calculated as \( E(o) = - \sum \pi_{\tau} \log(\pi_{\tau}) \), where the sum is over the different types present in the pattern, and \( \pi_{\tau} \) is the expected frequency of type \( \tau \) points in a neighbourhood of a typical point of the pattern.

The function \texttt{shannonF} is the calculation function. Uses function \texttt{segregationFun}.

The function \texttt{shannon.index} is a shortcut to get the non-spatial Shannon index.

Value

Returns an \texttt{fv}-object, see \texttt{spatstat} for more information. The index returns a scalar.

Functions

- \texttt{shannon.index}: Traditional index.

References


\begin{itemize}
\item \texttt{simpsonF} \hspace{1cm} \textit{Spatial Simpson index}
\end{itemize}

Description

Compute the spatial and non-spatial Simpson index for a given multitype point pattern.

Usage

\begin{verbatim}
simpsonF(X, r = NULL, ...)
simpson.index(X, spatial = FALSE, ...)
\end{verbatim}

Arguments

\begin{itemize}
\item \texttt{X} \hspace{1cm} Multitype point pattern of class \texttt{ppp} (see package ’spatstat’)
\item \texttt{r} \hspace{1cm} Vector of sizes for neighbourhoods, e.g. geometric graph with different ranges.
\item \ldots \hspace{1cm} Further parameters for the function \texttt{segregationFun}.
\item \texttt{spatial} \hspace{1cm} If FALSE, return the classical aspatial index value.
\end{itemize}
Details
The form of Simpson index is $S = 1 - \sum \pi_{\tau}$, where the sum is over the types of the pattern, and $\pi_{\tau}$ is like in Shimatani & Kubota 2004. The function simpsonF is the main calculation function. Uses function segregationFun.
The function simpson.index is a shortcut to get a single value for the pattern using 4-nearest neighbours graph by default.

Value
If spatial, returns an fv-object, see spatstat for more information. Otherwise a numeric value.

Functions
• simpson.index: The Spatial Simpson Index

References
Index

*Topic **package**
  spatialseggregation-package, 2

*Topic **spatial**
  exposurereps, 4
  helper functions, 5

biomassF, 3, 10

clean.up.data (helper functions), 5
dixon, 4

exposurereps, 4

defs (helper functions), 5

helper functions, 5

isar.index (isarF), 6
isarF, 6, 10

mcif, 7, 10

mingling.index (minglingF), 8
minglingF, 8, 10

minusID (helper functions), 5

ppp, 3

print.segtest, 9

segregationFun, 3, 7, 8, 9, 11–13
sg.modify.pp, 11

shake (helper functions), 5
shannon.index (shannonF), 11
shannonF, 10, 11

simpson.index (simpsonF), 12
simpsonF, 10, 12

spatialseggregation
  (spatialseggregation-package), 2

spatialseggregation-package, 2

spatstat, 2