

Package ‘spatialwarnings’

July 23, 2026

Type Package

Title Spatial Early Warning Signals of Ecosystem Degradation

Version 3.1.2

License MIT + file LICENSE

Maintainer Alexandre Genin <alexandre.genin@inrae.fr>

Description Tools to compute and assess significance of early-warnings signals (EWS) of ecosystem degradation. EWS are spatial metrics derived from raster data -- e.g. spatial autocorrelation - that increase before an ecosystem undergoes a non-linear transition (Genin et al. (2018) <doi:10.1111/2041-210X.13058>).

URL <https://github.com/spatial-ews/spatialwarnings>

BugReports <https://github.com/spatial-ews/spatialwarnings/issues>

Depends R (>= 3.5.0), future

Imports Rcpp, ggplot2, plyr, stats, utils, future.apply, gsl,
segmented

Suggests moments, powerLaw, reshape2, testthat, covr, acss, acss.data,
mgcv, gstat, sp, raster

LinkingTo Rcpp, RcppArmadillo

LazyData true

RoxygenNote 7.3.2.9000

Encoding UTF-8

NeedsCompilation yes

Author Alain Danet [aut],
Alexandre Genin [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-3333-1338>>),
Vishwesh Guttal [aut],
Sonia Kefi [aut],
Sabiha Majumder [aut],
Sumithra Sankaran [aut],
Florian Schneider [aut]

Repository CRAN

Date/Publication 2026-07-23 15:50:02 UTC

Contents

arizona	3
coarse_grain	3
convert_to_matrix	4
create_indicator	5
dda	7
display_matrix	8
display_size_info	9
dLSW	9
extract_spectrum	11
extract_variogram	11
flowlength_sews	12
forestgap	14
generic_sews	15
indicator_plrange	17
indicator_psdtype	18
indictest	20
kbdm_sews	22
label	23
lsw_sews	24
patchdistr_sews	26
patchdistr_sews_plot	28
patchdistr_sews_predict	30
patchsizes	31
plot.simple_sews_test	32
plot_spectrum	33
pl_fit	33
raw_cg_moran	36
raw_cg_skewness	37
raw_cg_variance	38
raw_clustering	40
raw_flowlength_uniform	41
raw_kbdm	42
raw_moran	43
raw_plrange	44
raw_sdr	45
raw_structvar	46
raw_variogram_metrics	47
rspectrum	48
serengeti	49
simple_sews	50
spectral_sews	51
variogram_sews	53
variogram_sews_plot	55
variogram_sews_predict	57
xmin_estim	58

arizona	<i>Aerial views of vegetation from Arizona, USA</i>
---------	---

Description

Aerial views of vegetation from Arizona, USA

Usage

arizona

Format

A list of logical matrices which were obtained through the classification of aerial images of vegetation taken in Arizona (USA).

Source

Derived from the images provided in the Supplementary Material of Rodriguez et al. (2017).

References

Rodriguez, F., A. G. Mayor, M. Rietkerk, and S. Bautista. 2017. A null model for assessing the cover-independent role of bare soil connectivity as indicator of dryland functioning and dynamics. Ecological Indicators.

coarse_grain	<i>Matrix coarse-graining</i>
--------------	-------------------------------

Description

This function averages the spatial data locally. It divides the input matrix into submatrices of dimension subsize and averages the spatial data in these submatrices. By doing this, the dimension of resultant matrix is reduced by a factor of subsize.

Usage

coarse_grain(mat, subsize)

Arguments

mat	A matrix
subsize	Dimension of the submatrix. This has to be a positive integer smaller than the dimension of input matrix.

Details

If the data is classified into discrete units, the calculation of variance and skewness can give spurious results irrelevant to the proximity to transition. Therefore, discrete data should be 'coarse-grained' before calculating the spatial early warning signals. However, this can also be applied to continuous state data.

Value

A matrix of reduced dimension.

References

Sankaran, S., Majumder, S., Kefi, S. and Guttal, V. (2017). Implications of being discrete and spatial for detecting early warning signals of regime shifts. *Ecological Indicators*.

See Also

[generic_sews](#)

Examples

```
rmat <- matrix(runif(20*10) > .5,
               ncol = 20, nrow = 10)
rmat.cg <- coarse_grain(rmat, subsize = 2)

par(mfrow = c(1, 2))
image(rmat)
title('Raw matrix')
image(rmat.cg)
title('Coarse-grained matrix')
```

convert_to_matrix	<i>Convert an object to a matrix</i>
-------------------	--------------------------------------

Description

This function is mainly for internal use by the [spatialwarnings](#) package to convert objects before they are processed by *_sews functions.

Usage

```
convert_to_matrix(object, ...)
```

Arguments

object	An object (typically, a matrix or a list of matrices)
...	Additional arguments (currently ignored)

Details

This generic function is here so that other packages can extend it. For example, `spatialwarningsGis` will provide methods so that GIS objects can be handled (e.g. `RasterLayer` from package `raster`).

Examples

```
# this does nothing
convert_to_matrix(serengeti[2:3])
```

create_indicator	<i>Custom Spatial Early-Warning signals</i>
------------------	---

Description

Computation, significance assessment and display of trends of a custom, user-defined indicator.

Usage

```
create_indicator(fun, taskname = as.character(substitute(fun)))

compute_indicator(mat, fun, taskname = as.character(substitute(fun)), ...)
```

Arguments

fun	A function that takes a matrix as input and returns a vector of numerical values. If the function returns a named vector, then the names will be used in plots and summaries. The function may also accept extra arguments.
taskname	The task name. A character string used for plots and textual summaries that describes the indicator (or set of indicators) being computed. If a task name cannot be derived from fun, then a default name is used.
mat	A matrix or a list of matrices.
...	Additional arguments to pass to the function fun

Details

`spatialwarnings` provides "workflow functions", named `*_sews`, that assist the user in computing, displaying and assessing the significance of indicator values. The functions `create_indicator` and `compute_indicator` provides such workflow for any arbitrary function.

`create_indicator` takes a function 'fun' and returns another function that can be used as an indicator similar to the `*_sews` functions. The results of this function can be assessed for significance using `indictest` and trends can be displayed using `plot`, `summary`, etc. (see Examples). `compute_indicator` does the same but without needing an intermediate indicator function.

Value

create_indicator returns a function that can be used in the same way than the other *_sews functions (e.g. generic_sews). This function as well as compute_indicator will return [simple_sews_*](#) objects.

See Also

[simple_sews](#)

Examples

```
# Use the maximum patch size as indicator of degradation
maxpatchsize <- function(mat) {
  max(patchsizes(mat))
}

# Create the indicator function
maxpatch_sews <- create_indicator(maxpatchsize)

# Then work with this function as if it were a function from the *_sews
# family.
mp_indic <- maxpatch_sews(forestgap)
summary(mp_indic)

# Assess significance and display trends
mp_test <- indictest(mp_indic, nulln = 49)
plot(mp_test)

# Try spatial coefficient of variation as a spatial EWS. This function can
# have arguments.
spatial_cv <- function(mat, subsize) {
  matc <- coarse_grain(mat, subsize)
  return( sd(matc) / mean(matc) )
}

# Create indicator function
cv_sews <- create_indicator(spatial_cv)

# Compute and display trends
cv_indic <- cv_sews(serengeti, subsize = 3)
plot(cv_indic, along = serengeti.rain)

# We can do the same work in one go using compute_indicator
cv_indic2 <- compute_indicator(serengeti, spatial_cv, subsize = 3)
plot(cv_indic2, along = serengeti.rain)

indictest(cv_indic, nulln = 99)
```

`dda`*Density-dependent aggregation model*

Description

`dda` is a list of matrices representing results from the density-dependent aggregation model (Siteur et al. 2023) in `dda`. `dda.pars` is a data frame with the model parameters. Each row of `dda.pars` corresponds to a matrix in `dda`, in the same order. All parameters were maintained constant, except for `tau` (see model definition in Siteur et al. 2023).

Usage

`dda``dda.pars`

Format

An object of class `list` of length 4.

An object of class `data.frame` with 4 rows and 5 columns.

Source

Kindly provided by Koen Siteur

References

Siteur, Koen, Quan-Xing Liu, Vivi Rottschäfer, Tjisse van der Heide, Max Rietkerk, Arjen Doelman, Christoffer Boström, and Johan van de Koppel. 2023. "Phase-Separation Physics Underlies New Theory for the Resilience of Patchy Ecosystems." *Proceedings of the National Academy of Sciences* 120 (2): e2202683120. <https://doi.org/10.1073/pnas.2202683120>.

Examples

```
ddasews <- lsw_sews(dda)
plot(ddasews, along = dda.pars[, "tau"]) # tau is the changing parameter

display_matrix(dda[[4]])
```

display_matrix	<i>Plot a matrix</i>
----------------	----------------------

Description

Display a matrix or a list of matrices in a plot

Usage

```
display_matrix(object, palette = "RdYlBu", along = NULL, ...)
```

Arguments

object	A matrix, a list of matrices, an object produced by *_sews functions or indicest()
palette	A color palette to use in the plot. It can be any color palette understood by scale_fill_brewer .
along	A vector of values used in facet headers. If object is a matrix, this is ignored.
...	Other arguments are ignored.

Details

This function will plot a matrix using ggplot2, using the provided palette. Its use-case is very much like image(), but it produces nicer plots by default (image() is much faster, however).

Value

A ggplot2 object, which is printed when this function is used interactively.

Examples

```
# display_matrix works with single matrices or lists of matrices
display_matrix(serengeti[2:3])

# display_matrix is compatible with "*_sews" objects
indics <- compute_indicator(serengeti[2:3], raw_moran)
display_matrix(indics)
```

display_size_info	<i>Display size information of a sews_result object</i>
-------------------	---

Description

Display size information of a sews_result object

Usage

```
display_size_info(x, ...)
```

```
## S3 method for class 'sews_result_single'
```

```
display_size_info(x, ...)
```

```
## S3 method for class 'sews_result_list'
```

```
display_size_info(x, ...)
```

```
## S3 method for class 'list'
```

```
display_size_info(x, ...)
```

Arguments

x	A sews_result object (as provided by <code>**_sews</code> functions, such as <code>generic_sews()</code>)
...	Other arguments are ignored.

Details

This function is used to display information about the size of matrices used by various functions in spatialwarnings. It is mostly destined to internal use.

dLSW	<i>The Lifshitz-Slyozov-Wagner distribution</i>
------	---

Description

Density and distribution function for the Lifshitz-Slyozov-Wagner (LSW) distribution with mean μ .

Usage

```
dLSW(x, mu, log = FALSE)
```

```
pLSW(x, mu, lower.tail = TRUE, log.p = FALSE)
```

```
LSW_fit(x)
```

Arguments

<code>x</code>	vector of quantiles
<code>mu</code>	the mean of the distribution
<code>log, log.p</code>	logical; if TRUE, probabilities <code>p</code> are given as $\log(p)$
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$

Details

The LSW distribution is a continuous distribution with density

$$f(x) = \frac{4x^2}{9\mu^3} \left(\frac{3\mu}{3\mu + x} \right)^{7/3} \left(\frac{3\mu}{3\mu - 2x} \right)^{11/3} e^{\frac{2x}{2x-3\mu}}$$

where μ is the mean of the distribution.

The functions `dLSW` gives the probability density, `pLSW` gives the distribution function. `qLSW` and `rLSW` are not implemented. You can use `LSW_fit` to fit an LSW distribution to a set of observations.

The length of the results is determined by the length of `x`, and `mu` can only be a single value.

Please note that this distribution has support only on the interval $[0, \mu \cdot 3/2)$. Probabilities outside this interval are returned as 0.

Value

`dLSW` gives the density, `pLSW` gives the distribution function, both as numerical vectors determined by the length of `x`.

References

Siteur, Koen, Quan-Xing Liu, Vivi Rottschäfer, Tjisse van der Heide, Max Rietkerk, Arjen Doelman, Christoffer Boström, and Johan van de Koppel. 2023. "Phase-Separation Physics Underlies New Theory for the Resilience of Patchy Ecosystems." *Proceedings of the National Academy of Sciences* 120 (2): e2202683120. <https://doi.org/10.1073/pnas.2202683120>.

See Also

[lsw_sews](#)

Examples

```
# Plot the density
x <- seq(0, 10, l = 128)
plot(x, dLSW(x, mu = 3), type = "l", col = "black")
lines(x, dLSW(x, mu = 5), type = "l", col = "red")
lines(x, dLSW(x, mu = 7), type = "l", col = "blue")
legend(x = 0, y = max(dLSW(x, mu = 3)), lty = 1, col = c("black", "red", "blue"),
       legend = paste("mu =", c(3, 5, 7)))
```

extract_spectrum	<i>Extract the r-spectrum from objects</i>
------------------	--

Description

Extract the r-spectrum from objects produced by spectral_sews.

Usage

```
extract_spectrum(x, ...)
```

Arguments

x	An object produced by spectral_sews or the result of the indictest function called on such object
...	Other arguments are ignored

Value

The empirical r-spectrum as a data.frame

See Also

[spectral_sews](#), [rspectrum](#)

Examples

```
# Extract the r-spectrum after computing indicators
indics <- spectral_sews(serengeti[2:3])
extract_spectrum(indics)
```

extract_variogram	<i>extract_variogram() method for variogram_sews objects</i>
-------------------	--

Description

Extract the empirical variogram from a variogram_sews object

Usage

```
extract_variogram(x, ...)
```

Arguments

x	An object produced by variogram_sews, or indictest
...	Additional arguments (ignored)

Value

A data.frame containing the variogram with the distances (column dist), the empirical semivariance values (gamma), and if object contains more than one matrix, a column named matrixn.

See Also

[variogram_sews](#)

Examples

```
vario_indics <- variogram_sews(serengeti)
predict(vario_indics)
vario_test <- indictest(vario_indics, nulln = 19)
predict(vario_test) # same result
```

flowlength_sews	<i>Flowlength connectivity indicator (uniform topography)</i>
-----------------	---

Description

Measures the connectivity of runoff-source areas as determined by vegetation patterns and (uniform) topography

Usage

```
flowlength_sews(mat, slope = 20, cell_size = 1)
```

Arguments

- mat The input matrix (must be a logical matrix)
- slope The slope of the area documented by the matrix (in degree).
- cell_size The horizontal size of a cell in the matrix (as viewed from above).

Details

This function computes Flowlength, a simple metric that measures the potential hydrological connectivity of runoff-source areas (e.g., bare soil) considering vegetation cover, vegetation patterns and topography. Flowlength is defined as the average length of all the potential runoff pathways in the target area. Thus, a higher value of the index indicates a higher hydrologic connectivity of runoff source areas. This function is designed for an idealized uniform hillslope (e.g., with constant slope angle, the direction of maximum slope being from the top to the bottom of the input matrices). The deviations of Flowlength from its expected values under random or aggregated-pattern null models can be used as an indicator of imminent transition to a degraded state (Rodriguez et al. 2017) in the context of arid drylands. An increased deviation of flowlength compared to its null

values is expected as a possible transition gets closer. This deviation can be computed using any null model provided by `spatialwarnings` (see [indictest](#) for more details), but a specific null model is provided for Flowlength based on a much-faster analytical approximation, using the argument `null_method = "approx_rand"` when calling `indictest` (see examples below).

In general, Flowlength can be used as indicator of dryland functional status by assessing potential water and soil losses in patchy landscapes (Mayor et al. 2008, Moreno-de las Heras et al. 2012, Mayor et al. 2013, Okin et al. 2015). Finally, the combination of observed and expected Flowlength under null models for random or aggregated vegetation cover can be used for assessing the cover-independent role of bare- soil connectivity (Rodriguez et al. 2018).

Value

A ‘`simple_sews`’ object containing the flow length value, among other things, see [simple_sews_object](#) for more information.

References

- Rodriguez, F., A. G. Mayor, M. Rietkerk, and S. Bautista. 2017. A null model for assessing the cover-independent role of bare soil connectivity as indicator of dryland functioning and dynamics. *Ecological Indicators*.
- Mayor, A.G., Bautista, S., Small, E.E., Dixon, M., Bellot, J., 2008. Measurement of the connectivity of runoff source areas as determined by vegetation pattern and topography: a tool for assessing potential water and soil losses in drylands. *Water Resour. Res.* 44, W10423.
- Mayor, A.G., Kefi, S., Bautista, S., Rodriguez, F., Carteni, F., Rietkerk, M., 2013. Feedbacks between vegetation pattern and resource loss dramatically decrease ecosystem resilience and restoration potential in a simple dryland model. *Landsc. Ecol.* 28, 931-942.
- Moreno-de las Heras, M., Saco, P.M., Willgoose, G.R., Tongway, D.J., 2012. Variations in hydrological connectivity of Australian semiarid landscapes indicate abrupt changes in rainfall-use efficiency of vegetation. *J. Geophys. Res.* 117, G03009.
- Okin, G.S., Moreno-de las Heras, M., Saco, P.M., Throop, H.L., Vivoni, E.R., Parsons, A.J., Wainwright, J., Peters, D.P.C., 2015. Connectivity in dryland landscapes: shifting concepts of spatial interactions. *Front. Ecol. Environ.* 13 (1), 20-27.

See Also

[raw_flowlength_uniform](#), [indictest](#) to test the significance of indicator values.

Examples

```
fl_result <- flowlength_sews(arizona, slope = 20, cell_size = 1)

# Compute the Z-score (standardized deviation to null distribution) and plot
# its variations along the gradient. This Z-score is suggested by
# Rodriguez et al. (2017) as an indicator of degradation.
fl_test <- indictest(fl_result, nulln = 19)
plot(fl_test, what = "z_score")

# Use the analytical approximation suggested in Rodriguez et al. (2017),
```

```
# instead of permuting the original values in the matrix (much faster)
fl_test <- indictest(fl_result, null_method = "approx_rand")
plot(fl_test, what = "z_score")
```

forestgap

A list of binary matrices and their associated parameters

Description

A list of binary matrices and their associated parameters

Usage

```
forestgap
```

```
forestgap.pars
```

Format

A list of logical matrices which are the end results of simulations from Kubo's Forest Gap model along a gradient of increasing values of stress (see references).

The parameters used for the simulations, as a data frame.

Details

Kubo's forest gap model has three parameters, α that controls the reproductive rate of trees, d controls the non-spatialized mortality and δ the increased mortality due to the presence of a neighboring gap.

Source

Generated using the implementation of Kubo's model in caspr 0.2.0 <https://github.com/fdschneider/caspr>.

References

Kubo, T., Iwasa, Y., & Furumoto, N. (1996). Forest spatial dynamics with gap expansion: Total gap area and gap size distribution. *Journal of Theoretical Biology*, 180(3), 229-246. doi:10.1006/jtbi.1996.0099

generic_sews

*Generic Spatial Early-Warning signals***Description**

Computation, significance assessment and display of spatial generic early warning signals (Moran's I, variance and skewness)

Usage

```
generic_sews(
  mat,
  subsize = 4,
  abs_skewness = FALSE,
  moranI_coarse_grain = FALSE
)
```

Arguments

mat	A matrix (quantitative data), a binary matrix (which contains TRUE or FALSE values) or a list of those
subsize	The subsize used for the coarse-graining phase (see Details)
abs_skewness	Should the absolute skewness be used instead of its raw values ?
moranI_coarse_grain	Should the input matrix be coarse-grained before computing the Moran's I indicator value ?

Details

The Generic Early warning signal are based on the property of a dynamical system to "slow down" when approaching a critical point, that is, take more time to return to equilibrium after a perturbation. This is expected to be reflected in several spatial characteristics: the variance, the spatial autocorrelation (at lag-1) and the skewness. This function provides a convenient workflow to compute these indicators, assess their significance and display the results.

Before computing the actual indicators, the matrix can be "coarse-grained". This process reduces the matrix by averaging the nearby cells using a square window defined by the subsize parameter. This makes spatial variance and skewness reflect actual spatial patterns when working with binary (TRUE/FALSE data), but is optional when using continuous data. Keep in mind that it effectively reduces the size of the matrix by approximately subsize on each dimension.

The significance of generic early-warning signals can be estimated by reshuffling the original matrix (function `indictest`). Indicators are then recomputed on the shuffled matrices and the values obtained are used as a null distribution. P-values are obtained based on the rank of the observed value in the null distribution. A small P-value means that the indicator is significantly above the null values, as expected before a critical point.

The plot method can displays the results graphically. A text summary can be obtained using the summary method.

Value

`generic_sews` returns an object of class `simple_sews_single` (a list) if `mat` is a single matrix or an object of class `simple_sews_list` if `mat` is a list. You probably want to use some of the methods written for these complicated objects instead of extracting values directly (they are displayed using `print(<object>)`).

`indictest` returns an object of class `generic_test` (a `data.frame`).

plot methods return `ggplot` objects, usually immediately displayed when plot is being called at the R prompt.

References

Kefi, S., Guttal, V., Brock, W.A., Carpenter, S.R., Ellison, A.M., Livina, V.N., et al. (2014). Early Warning Signals of Ecological Transitions: Methods for Spatial Patterns. *PLoS ONE*, 9, e92097.

Dakos, V., van Nes, E. H., Donangelo, R., Fort, H., & Scheffer, M. (2010). Spatial correlation as leading indicator of catastrophic shifts. *Theoretical Ecology*, 3(3), 163-174.

Guttal, V., & Jayaprakash, C. (2008). Spatial variance and spatial skewness: leading indicators of regime shifts in spatial ecological systems. *Theoretical Ecology*, 2(1), 3-12.

See Also

`indictest`, to test the significance of indicator values. Individual indicators: `raw_cg_moran` `raw_cg_variance`, `raw_cg_skewness`, `simple_sews`

Examples

```
data(serengeti)
gen_indic <- generic_sews(serengeti, subsize = 5,
                          moranI_coarse_grain = TRUE)

# Display results
summary(gen_indic)

# Display trends along the varying model parameter
plot(gen_indic, along = serengeti.rain)

# Compute significance (long)

gen_test <- indictest(gen_indic, nulln = 199)

print(gen_test)

# Display the trend, now with a grey ribbon indicating the 5%-95% quantile
# range of the null distribution
plot(gen_test, along = serengeti.rain)

# Display the effect size compared to null distribution
plot(gen_test, along = serengeti.rain, what = "z_score")

# Note that plot() method returns a ggplot object that can be modified
```



```
# for convenience
if ( require(ggplot2) ) {
  plot(gen_test, along = serengeti.rain) +
    geom_vline(xintercept = 733, color = "red", linetype = "dashed") +
    xlab('Annual rainfall') +
    theme_minimal()
}
```

indicator_plrange	<i>Power-law range indicator</i>
-------------------	----------------------------------

Description

Compute the power-law range of a matrix

Usage

```
indicator_plrange(mat, merge = FALSE, xmin_bounds = NULL)
```

Arguments

mat	A logical matrix, or a list of logical matrices
merge	Controls whether the patch-size distributions of the input matrices are merged together before computing the power-law range. Setting this value to TRUE makes the function return a single value even if multiple matrices are given as input.
xmin_bounds	A vector of two integer values, defining a range in which to search for the best xmin (see Details).

Details

Some ecosystems show typical changes in their patch-size distribution as they become more and more degraded. In particular, an increase in the truncation of the patch-size distribution (PSD) is expected to occur. The power-law range (PLR) measures the truncation of the PSD in a single value (see also [patchdistr_sews](#) for more details).

To compute the PLR, power-laws are fitted with a variable minimum patch size (x_{min}) and the one with the lowest Kolmogorov-Smirnov distance to the empirical distribution is retained. PLR is then computed using this best-fitting x_{min} as:

$$\frac{\log(x_{max}) - \log(x_{min})}{\log(x_{max}) - \log(x_{smallest})}$$

where x_{max} is the maximum observed patch size, and $x_{smallest}$ is the minimum observed patch size.

Value

A data.frame with columns minsize, maxsize which are the observed minimum and maximum patch sizes, the estimated x_{min} as column xmin and the value of the power-law range as plrange. If multiple matrices were provided, then a list of data.frames is returned

References

Clauset, A., Shalizi, C. R., & Newman, M. E. (2009). Power-law distributions in empirical data. SIAM review, 51(4), 661-703.

Berdugo, M., Kefi, S., Soliveres, S. & Maestre, F.T. (2017). Plant spatial patterns identify alternative ecosystem multifunctionality states in global drylands. Nature in Ecology and Evolution.

See Also

[patchdistr_sews](#)

Examples

```
forestgap.plr <- indicator_plrange(forestgap)
do.call(rbind, forestgap.plr) # convert results to data.frame

# Restrict to small xmins
forestgap.plr2 <- indicator_plrange(forestgap, xmin_bounds = c(1, 10))
do.call(rbind, forestgap.plr2)
```

indicator_psdtype	<i>Change in patch-size distributions types</i>
-------------------	---

Description

This functions fits different patch size distributions types (power-law, log-normal, exponential and truncated power-law) to the patches contained in a matrix. The distributions are returned with their corresponding AIC, BIC and AICc to select the best fit.

Usage

```
indicator_psdtype(
  x,
  xmin = 1,
  merge = FALSE,
  fit_lnorm = FALSE,
  xmin_bounds = NULL,
  best_by = "AIC",
  wrap = FALSE,
  nbmask = "von_neumann"
)
```

Arguments

x	A logical (TRUE/FALSE values) matrix or a list of these.
xmin	The xmin to be used to fit the patch size distributions. Use the special values "estimate" to use an estimated xmin for each fit
merge	The default behavior is to produce indicators values for each matrix. If this parameter is set to TRUE then the patch size distributions are pooled together for fitting.
fit_lnorm	Fit also a log-normal distribution
xmin_bounds	Restrict the possible xmins in this range (defaults to the whole range of observed patch sizes)
best_by	The criterion used to select the best distribution type (one of "AIC", "BIC" or "AICc").
wrap	Determines whether patches are considered to wrap around the matrix when reaching the side
nbmask	Either "moore" for 8-way neighborhood, "von_neumann" for four-way neighborhood (default), or a 3x3 matrix describing which neighbors to consider around a cell. See patchsizes for details on how to specify such neighborhoods.

Details

Patterned ecosystems can exhibit a change in their spatial structure as they become more and more stressed. It has been suggested that this should be reflected in changes in the observed patch size distributions (PSD). The following sequence is expected to occur (Kefi et al. 2011) as patterned ecosystems become more and more degraded:

- Percolation of vegetation patches occurs (a patch has a width or height equal to the size of the system)
- The patch-size distribution follows a power-law
- The patch-size distribution deviates from a power-law as larger patches break down
- The patch-size distribution is closer to an exponential distribution

This indicator fits the observed patch size distribution based on maximum-likelihood (following Clauset et al. 2009 recommendations), then select the best model using AIC, BIC (default) or AICc.

Value

A data.frame (or a list of these if x is a list) with the following columns:

- method the method used for fitting (currently: only log-likelihood is implemented, "ll")
- type the type of distribution
- npars the number of parameters of the distribution type
- AIC, 'AICc' and 'BIC' the values for Akaike Information Criterion (or the corrected for small samples equivalent AICc), and Bayesion Information Criterion (BIC)
- best A logical vector indicating which distribution is the best fit
- plexpo, cutoff, meanlog, sdlog the estimates for distribution parameters (see [pl_fit](#))
- percolation A logical value indicating whether there is [percolation](#) in the system.

References

Kefi, S., Rietkerk, M., Roy, M., Franc, A., de Ruiter, P.C. & Pascual, M. (2011). Robust scaling in ecosystems and the meltdown of patch size distributions before extinction: Patch size distributions towards extinction. *Ecology Letters*, 14, 29-35.

Kefi, S., Rietkerk, M., Alados, C.L., Pueyo, Y., Papanastasis, V.P., ElAich, A., et al. (2007). Spatial vegetation patterns and imminent desertification in Mediterranean arid ecosystems. *Nature*, 449, 213-217.

Clauset, A., Shalizi, C. R., & Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review*, 51(4), 661-703.

See Also

[patchdistr_sews](#)

[patchdistr_sews](#)

Examples

```
data(forestgap)

# One logical matrix only
indicator_psdtype(forestgap[[1]])

# A list of these matrices

indicator_psdtype(forestgap)
```

indicest

Significance-assessment of spatial early-warning signals

Description

Assess the significance of spatial early-warning indicators

Usage

```
indicest(x, nulln = 999, null_method = "perm", null_control = NULL, ...)
```

Arguments

x	An object such as one produced by the *_sews functions, or compute_indicator
nulln	The number of values to produce the null distribution
null_method	The method used to produce the null values (see Details)
null_control	List of arguments used to control the generation of null matrices. If NULL, then sensible defaults are chosen (see Details)
...	Additional arguments are ignored

Details

`indictest` is used to test the significance of early-warning signals against 'null matrices', which represent the expected spatial structure in the absence of the biological process of interest.

For a given indicator, a null distribution is obtained by producing a set of 'null' matrices, from which indicator values are recomputed. This produces a null distribution of null indicator values against which the observed value is tested.

Several methods are available to produce the set of null matrices. If `null_method` is set to "perm", the original matrix is reshuffled to obtain a null matrix. If `null_method` is set to "intercept", then a generalized linear model of the form $y \sim 1$ (where y represents the values of the matrix) is fitted, then values are drawn from this model. If `null_method` is set to "smooth", then a smooth surface is fitted based on a generalized additive model (using [gam](#)) to the matrix, then values are drawn from this model. When using the "intercept" or "smooth" null models, it is important to make sure the model 'family' corresponds to the type of values present in the matrix. By default, if a matrix contains TRUE/FALSE values, a 'binomial()' family is used, otherwise a 'gaussian()' family is used. More information about null models is available in the [spatialwarnings FAQ](#).

Please note that specific null methods may exist for some indicators, such as [flowlength](#). These are often based on analytical approximation and allow faster computations.

If a matrix has attributes, then these are preserved and passed to the function used to compute the indicator value, except when using the null method 'perm', in which case matrix attributes are discarded.

The list `null_control` can be used to adjust the computation of null matrices. It can have the following components:

`family` The family used in the model used to produce the null matrices. Typically, it is one of [binomial\(\)](#), [gaussian\(\)](#), etc.

`qinf` The lower quantile to compute from the null distribution and display in summaries/plots. A numeric value between 0 and 1.

`qsup` The upper quantile to compute from the null distribution and display in summaries/plots. A numeric value between 0 and 1.

Value

An object with a class ending in `*_sews_test`, whose exact class depends on the input object. `plot`, summary methods are available to display the results of computations, and additional methods may be available depending on the input object (e.g. see [patchdistr_sews_plot](#)).

References

Kefi, S., Guttal, V., Brock, W.A., Carpenter, S.R., Ellison, A.M., Livina, V.N., et al. (2014). Early Warning Signals of Ecological Transitions: Methods for Spatial Patterns. PLoS ONE, 9, e92097

See Also

[generic_sews](#), [spectral_sews](#), [kbdm_sews](#), [compute_indicator](#), [flowlength_sews](#)

kbdm_sews

*Indicator based on Kolmogorov Complexity***Description**

Computes the Kolmogorov Complexity on a set of matrices, using the Block Decomposition Method.

Usage

```
kbdm_sews(mat, subsize = 3)
```

Arguments

mat	A logical matrix (TRUE/FALSE values) or a list of logical matrices
subsize	A submatrix size to carry out the Block Decomposition Method (must be between 1 and 3)

Details

be a useful indicator to anticipate transitions in model ecological systems (Dakos and Soler-Toscano, 2017). When close to the transition critical point, the complexity is expected to decrease.

The Kolmogorov complexity cannot be computed directly for large strings (i.e. matrices). However, the complexity of smaller submatrices can be estimated, then combined to obtain an approximation of the complexity of the whole matrix. This method, the Block Decomposition Method is implemented in this indicator following Dakos and Soler-Toscano (2017).

Value

kbdm_sews returns an object of class `simple_sews_single` (a list) if `mat` is a single matrix, and an object of class `simple_sews_list` if `mat` is a list of matrices. These objects can be used with generic methods `indictest` (to test significance) or `plot` (to display trends), see also the examples below.

References

Dakos, V., and F. Soler-Toscano. 2017. Measuring complexity to infer changes in the dynamics of ecological systems under stress. *Ecological Complexity* 32:144-155.

See Also

[raw_kbdm](#), [acss](#), [indictest](#), to test the significance of indicator values.

Examples

```
kbdm_result <- kbdm_sews(serengeti, subsize = 3)
plot(kbdm_result, along = serengeti.rain)

kbdm_test <- indictest(kbdm_result, nulln = 49)
plot(kbdm_test, along = serengeti.rain)

# Plot deviation to null expectation
plot(kbdm_test, along = serengeti.rain, what = "z_score")
```

label	<i>Labelling of unique patches and detection of percolation.</i>
-------	--

Description

Label each patch with a number in a binary matrix

`percolation()` detects whether percolation occurs in the matrix (i.e. a patch has a width or a height equal to the size of the matrix)

Usage

```
label(mat, nbmask = "von_neumann", wrap = FALSE)

percolation(mat, nbmask = "von_neumann")
```

Arguments

<code>mat</code>	A binary matrix
<code>nbmask</code>	Either "moore" for 8-way neighborhood, "von_neumann" for four-way neighborhood (default), or a 3x3 matrix describing which neighbors to consider around a cell. See patchsizes for details on how to specify such neighborhoods.
<code>wrap</code>	Whether to wrap around lattice boundaries ('TRUE'/'FALSE'), effectively using periodic boundaries.

Details

The `label` function "labels" the patches of a binary (TRUE/FALSE) matrix. It returns a matrix of similar height and width, with integer values representing the ID of each unique patch (contiguous cells). Empty cells are labelled as NA.

Value

A matrix containing ID numbers for each connected patch. Default parameters assume 4-cell neighborhood and periodic boundaries. The distribution of patch sizes is returned as the attribute "psd" and the percolation status as "percolation" (whether a TRUE patch has a width or height equal to the size of the matrix).

See Also

[patchsizes](#), [patchdistr_sews](#)

Examples

```
data(forestgap)

rmat <- matrix(rnorm(100) > .1, ncol = 10)
display_matrix(label(rmat))

# With 8-way neighborhood mask and no wrapping around borders
display_matrix(label(rmat, "moore", wrap = FALSE))

# On real data:
display_matrix(label(forestgap[[5]], "moore", wrap = FALSE))
```

lsw_sews

Indicators based on the LSW distribution

Description

LSW indicators for systems with density-dependent aggregation

Usage

```
lsw_sews(mat, wrap = FALSE)

raw_patch_radii_skewness(mat, wrap = FALSE)

raw_lsw_aicw(mat, wrap = FALSE)
```

Arguments

mat	A logical matrix (TRUE/FALSE values) or a list of such matrices
wrap	Determines whether patches are considered to wrap around the matrix when reaching on of its edges

Details

In systems where a mobile resource or consumer can be fixed in space by a sessile species, specific patterns are expected to appear. Such systems can include situations where nutrients available in the environment are fixed by a sessile species (e.g. seagrasses or corals), or where the behavior of herbivores is altered to restrict them to certain areas (see full theoretical background in Siteur et al. 2023).

In those systems, as environmental conditions change and the global density of the sessile species decreases, its spatial structure is expected to change. The area of patches of the sessile species (as measured by their radii, which assumes circular patches), is expected to go from a log-normal to a Lifshitz–Slyozov–Wagner (LSW) distribution. Thus, measuring how close the observed distribution of radii are to those two candidate distributions can constitute an indicator of ecosystem degradation.

This function measures this through the relative support based on AIC for the two distributions (equal to 1 when the empirical distribution is best-approximated by an LSW, and 0 when it is a log-normal distribution ([dlnorm](#)), and the skewness of the observed patch radii, which should approach a value around -0.92 as conditions worsen.

Value

`lsw_sews` returns an object of class `simple_sews_single` (a list) if `mat` is a single matrix or an object of class `simple_sews_list` if `mat` is a list. You probably want to use some of the methods written for these complicated objects instead of extracting values directly (they are displayed using `print(<object>)`).

Author(s)

This code has received contributions from Koen Siteur

References

Siteur, Koen, Quan-Xing Liu, Vivi Rottschäfer, Tjisse van der Heide, Max Rietkerk, Arjen Doelman, Christoffer Boström, and Johan van de Koppel. 2023. "Phase-Separation Physics Underlies New Theory for the Resilience of Patchy Ecosystems." *Proceedings of the National Academy of Sciences* 120 (2): e2202683120. <https://doi.org/10.1073/pnas.2202683120>.

See Also

[dLSW](#), [dda](#), [raw_patch_radii_skewness](#), [raw_lsw_aicw](#)

Examples

```
data(dda)
data(dda.pars)

# Compute all indicators at once (skewness and relative AIC support)
indics <- lsw_sews(dda)
plot(indics, along = dda.pars[, "tau"])

# Compute individual indicators
```

```
# Skewness of the distribution of patch radii
radii_skewness <- compute_indicator(dda, raw_patch_radii_skewness)
plot(radii_skewness, along = dda.pars[,"tau"])

# Aic weight of LSW distribution relative to a lognormal distribution. tau here
# represents the density at equilibrium in Siteur et al's model (2023)
lsw_aicw <- compute_indicator(dda, raw_lsw_aicw)
plot(lsw_aicw, along = 1 - dda.pars[,"tau"])
```

patchdistr_sews

Early-warning signals based on patch size distributions

Description

Compute early-warning signals based on patch size distributions

Usage

```
patchdistr_sews(
  mat,
  merge = FALSE,
  fit_lnorm = FALSE,
  best_by = "BIC",
  xmin = 1,
  xmin_bounds = NULL,
  wrap = FALSE,
  nbmask = "von_neumann"
)
```

Arguments

mat	A logical matrix (TRUE/FALSE values) or a list of these
merge	The default behavior is to produce indicators values for each matrix. If this parameter is set to TRUE then the patch size distributions are pooled together before fitting, yielding only one final indicator value for the set of input matrices (argument mat).
fit_lnorm	When patch size distributions are compared, should we consider lognormal type ? (see details)
best_by	The criterion to use to select the best fit (one of "AIC", "BIC" or "AICc")
xmin	The x_{min} to be used to fit the patch size distributions. Use the special value "estimate" to compute first the x_{min} that produces the best power-law fit, then use this estimated value to fit all distributions.
xmin_bounds	Bounds when estimating x_{min} for power-law distributions
wrap	Determines whether patches are considered to wrap around the matrix when reaching the side

nbmask Either "moore" for 8-way neighborhood, "von_neumann" for four-way neighborhood (default), or a 3x3 matrix describing which neighbors to consider around a cell. See [patchsizes](#) for details on how to specify such neighborhoods.

Details

Patterned ecosystems can exhibit a change in their spatial structure as they become more and more stressed. It has been suggested that this should be reflected in changes in the observed patch size distributions (PSD). The following sequence is expected to occur (Kefi et al. 2011) as patterned ecosystems become more and more degraded:

- Percolation of vegetation patches occurs (a patch has a width or height equal to the size of the system)
- The patch-size distribution follows a power-law
- The patch-size distribution deviates from a power-law as larger patches break down
- The patch-size distribution is closer to an exponential distribution

Additionally, it has been suggested that these changes in patch size distribution shape should be reflected in the power-law range (PLR). This function carries out all the required computations and helps display the results in a convenient form.

The fitting of PSDs is based on maximum-likelihood following Clauset et al.'s procedure. The best discrete distribution is estimated among these candidates: a power-law x^λ , an exponential $\exp(\alpha x)$, a truncated power-law and $x^\lambda \exp(\alpha x)$, and optionally, a log-normal. Each distribution parameter is estimated using maximum-likelihood, with a minimum patch size (xmin) fixed to one. The best distribution is selected based on BIC by default. In raw results, plexpo refers to the power-law exponent (λ in the previous equations) and cutoff refers to the exponential decay rate α .

To compute the Power-law range (PLR), power-laws are fitted with a variable minimum patch size (xmin) and the one with the lowest Kolmogorov-Smirnov distance to the empirical distribution is retained. PLR is then computed using this best-fitting xmin:

$$\frac{\log(x_{max}) - \log(x_{min})}{\log(x_{max}) - \log(x_{smallest})}$$

Results can be displayed using the text-based summary and print, but graphical options are also available to plot the trends (plot) and the fitted distributions (plot_distr). Plotting functions are documented in a [separate page](#). Observed and fitted distributions can be produced using the predict function, as documented on [this page](#).

Value

A list object of class 'psdfit' containing among other things - the observed patch size distribution data - the model outputs for the candidate distribution fits - the power-law range values - the percolation values (if several matrices were provided and 'merge' was TRUE, then the average percolation value is returned)

References

Kefi, S., Rietkerk, M., Alados, C. L., Pueyo, Y., Papanastasis, V. P., ElAich, A., & De Ruiter, P. C. (2007). Spatial vegetation patterns and imminent desertification in Mediterranean arid ecosystems. *Nature*, 449(7159), 213-217.

Kefi, S., Rietkerk, M., Roy, M., Franc, A., de Ruiter, P.C. & Pascual, M. (2011). Robust scaling in ecosystems and the meltdown of patch size distributions before extinction: Patch size distributions towards extinction. *Ecology Letters*, 14, 29-35.

Berdugo, M, Sonia Kefi, Santiago Soliveres, and Fernando T. Maestre. (2017) Plant Spatial Patterns Identify Alternative Ecosystem Multifunctionality States in Global Drylands. *Nature in Ecology and Evolution*, no. 1.

Clauset, A., Shalizi, C. R., & Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review*, 51(4), 661-703.

See Also

[patchsizes](#), [plot_distr](#), [predict](#), [plot](#),
[indictest](#), to test the significance of indicator values.

Examples

```
data(forestgap)

psd_indic <- patchdistr_sews(forestgap)

summary(psd_indic)
plot(psd_indic)

# Plots can be modified using ggplot2 directives
if ( require(ggplot2) ) {
  plot(psd_indic) +
    theme_minimal()
}

# Export results to a data.frame
psd_indic_export <- as.data.frame(psd_indic)
head(psd_indic_export)
```

patchdistr_sews_plot *Early-warning signals based on patch size distributions*

Description

Plot early-warning signals based on patch size distributions

Usage

```
## S3 method for class 'patchdistr_sews'
plot(x, along = NULL, ...)

plot_distr(x, along = NULL, best_only = TRUE, plrange = TRUE)
```

Arguments

<code>x</code>	An object as produced by <code>patchdistr_sews</code> , or the result of <code>indictest</code> called on such object
<code>along</code>	A vector providing values along which the indicator trends will be plotted. If NULL then the values are plotted sequentially in their original order.
<code>...</code>	Ignored
<code>best_only</code>	Plot only the best fit the empirical (inverse cumulative) patch-size distribution with an overlay of the estimated fits.
<code>plrange</code>	Plot the power-law range

Details

The plot function will produce a figure summarizing the changes in patch size distributions along a set of values. The figure has two panels:

- the upper panel shows the percolation status of empty (FALSE) and occupied cells (TRUE), and shows the mean value (proportion of TRUE values). The background shows the proportion of each type of distribution for each unique values of the along vector.
- the bottom panel displays the power-law range

The `plot_distr` function displays each distribution in an individual facet, with an overlay of the best distribution fit and a blue bar showing the power-law range. If appropriate, a grey ribbon is used to display the expected distribution given the null expectation (i.e. when `plot_distr` is called on the results of `indictest()`). This function can produce quite crowded graphs, but it displays in full the shape of the distributions, and can be useful e.g. to assess the quality of the fits.

See Also

[patchdistr_sews](#)

Examples

```
data(forestgap)
psd_indic <- patchdistr_sews(forestgap)

plot(psd_indic, along = forestgap.pars[ ,"d"])

# When along is non-numeric, bars are used for display
plot(psd_indic, along = as.factor(forestgap.pars[ ,"d"]))

# Display individual distributions
plot_distr(psd_indic, along = forestgap.pars[ ,"d"])

# We can display the distributions along with the null expectation after
# indictest() is run
psd_test <- indictest(psd_indic, nulln = 19)
plot_distr(psd_test, along = forestgap.pars[ ,"d"])
```

```
patchdistr_sews_predict
```

predict method for patchdistr_sews objects

Description

Export the observed and fitted patch size distributions

Usage

```
## S3 method for class 'patchdistr_sews_single'
predict(object, ..., newdata = NULL, best_only = FALSE, xmin_rescale = FALSE)
```

Arguments

object	An patchdistr_sews object
...	Additional arguments (ignored)
newdata	A vector of patch sizes at which the fit is returned (default to 200 regularly-spaced values).
best_only	Return values for only the best fit of each element (matrix) in object, or return the values for all fitted distribution.
xmin_rescale	If the xmin value used for fits is above one, then setting this to TRUE will rescale the predicted probabilities so that they align on the cumulative distribution of the observed patch sizes

Details

The function [patchdistr_sews](#) fits competing distribution models to the observed patch size distributions. This functions is able to export the observed values and the fitted values altogether.

Value

A list with component obs, a `data.frame` containing the observed distribution values and pred, a `data.frame` containing the fitted values. Both `data.frames` have columns `matrixn`, the number of the matrix for which values are given, `type`, the fitted type of distribution, as well as `patchsize` and `y`, the patch size and value of the inverse cumulative distribution function (i.e. $1 - P(x < k)$).

See Also

[patchdistr_sews](#)

Examples

```
patch_indics <- patchdistr_sews(forestgap)

predict(patch_indics)
```

patchsizes	<i>Get patch sizes.</i>
------------	-------------------------

Description

Get the distribution of patch sizes from a logical matrix

Usage

```
patchsizes(mat, merge = FALSE, nbmask = "von_neumann", wrap = FALSE)
```

Arguments

mat	A logical matrix or a list of such matrices.
merge	Controls whether the obtained patch size distributions are to be pooled together if mat is a list of matrices.
nbmask	Either "moore" for 8-way neighborhood, "von_neumann" for four-way neighborhood (default), or a 3x3 matrix describing which neighbors to consider around a cell. See patchsizes for details on how to specify such neighborhoods.
wrap	Whether to wrap around lattice boundaries ('TRUE'/'FALSE'), effectively using periodic boundaries.

Value

If mat is a logical matrix, then the function returns a vector of patch sizes. If mat is a list of logical matrices, then it returns a list of vectors of patch sizes: this list is flattened if merge is TRUE.

See Also

[label](#)

Examples

```
data(forestgap)
patchsizes(forestgap[[5]]) # Use a single matrix

# Compute the average patch size of each matrix
list_patches <- patchsizes(forestgap) # get the patch size for each matrix
print( sapply(list_patches, mean)) # print the average patch size

# Example with 8-way neighborhood
patchsizes(forestgap[[5]], nbmask = "moore")

# Same neighborhood as above, but specified in matrix form
moore_nb <- matrix(c(1, 1, 1,
                    1, 0, 1,
                    1, 1, 1),
                  nrow = 3, ncol = 3, byrow = TRUE)
patchsizes(forestgap[[5]], nbmask = moore_nb)
```

plot.simple_sews_test *Spatial early-warning signals: display of trends*

Description

Spatial early-warning signals: display of trends

Usage

```
## S3 method for class 'simple_sews_test'
plot(x, along = NULL, what = "value", display_null = TRUE, ...)

## S3 method for class 'simple_sews_list'
plot(x, along = NULL, ...)
```

Arguments

x	A simple_sews_test object (as provided by <code>**_sews</code> functions, such as <code>generic_sews()</code>)
along	A vector providing values over which the indicator trend will be plotted. If NULL then the values are plotted sequentially in their original order.
what	The trendline to be displayed. Defaults to the indicator's values ("value") but other metrics can be displayed. Accepted values are "value", "pval", "difference" (obs - null mean), or "z_score" ((obs - null mean) / (null sd)).
display_null	Chooses whether a grey ribbon should be added to reflect the null distribution. Note that it can not be displayed when the trend line reflects something else than the indicator values (when what is not set to "value").
...	Other arguments are ignored.

Details

Note that the produced plot is adjusted depending on whether `along` is numeric or not.

plot_spectrum	<i>Display the r-spectrum of a spectral_sews object</i>
---------------	---

Description

Display the r-spectrum (or multiple spectra) that are contained in an object returned by [spectral_sews](#) object (or the result of [indictest](#) applied on such object).

Usage

```
plot_spectrum(x, along = NULL, log = TRUE, display_null = TRUE, ...)
```

Arguments

<code>x</code>	An object produced by spectral_sews or the result returned by indictest applied on such object
<code>along</code>	A vector providing numerical or categorical values along which the indicator trends will be plotted. If <code>NULL</code> , then the indicator values are plotted sequentially in their original order.
<code>log</code>	Whether to use a log scale or a linear scale on the y axis
<code>display_null</code>	Whether to display null information. This argument is ignored if <code>x</code> has not been produced through indictest (and thus does not contain data regarding the null model)
<code>...</code>	Other arguments are ignored

See Also

[rspectrum](#), [spectral_sews](#), [extract_spectrum](#)

pl_fit	<i>Distribution-fitting functions</i>
--------	---------------------------------------

Description

These functions fit parametric distributions to a set of discrete values.

Usage

```
pl_fit(dat, xmin = 1)

exp_fit(dat, xmin = 1)

lnorm_fit(dat, xmin = 1)

tpl_fit(dat, xmin = 1)
```

Arguments

<code>dat</code>	The set of values to which the distribution are fit
<code>xmin</code>	The minimum possible value to consider when fitting the distribution

Details

These functions will fit distributions to a set of values using maximum-likelihood estimation. In the context of the 'spatialwarnings' package, they are most-often used to fit parametric distributions on patch size distributions. As a result, these functions assume that the data contains only integer, strictly positive values. The type of distribution depends on the prefix of the function: 'pl' for power-law, 'tpl' for truncated power-law, 'lnorm' for lognormal and 'exp' for an exponential distribution.

In the context of distribution-fitting, 'xmin' represents the minimum value that a distribution can take. It is often used to represent the minimum scale at which a power-law model is appropriate (Clauset et al. 2009), and can be estimated on an empirical distribution using [xmin_estim](#). Again, please note that the fitting procedure assumes here that xmin is equal or greater than one.

Please note that a best effort is made to have the fit converge, but it may sometimes fail when the parameters are far from their usual range, and numerical issues may occur. It is good practice to make sure the fits are sensible when convergence warnings are reported.

For reference, the shape of the distributions is as follow:

power-law x^{-a} where a is the power-law exponent

exponential $\exp(-bx)$ where b is the truncation rate of the exponential

truncated power-law $x^{-a}\exp(-bx)$ where a and b are the exponent of the power law and the rate of truncation

The lognormal form follows the [standard definition](#).

The following global options can be used to change the behavior of fitting functions and/or produce more verbose output:

spatialwarnings.constants.reltol the relative tolerance to use to compute the power-law normalizing constant

$$\sum_{k=1}^{\infty} x^{ak} e^{-bk}$$

. Increase to increase the precision of this constant, which can be useful in some cases, typically with large sample sizes. Default is 1e-8.

spatialwarnings.constants.maxit the maximum number of iterations to compute the normalizing constant of a truncated power-law. Increase if you get a warning that the relative tolerance level (defined above) was not reached. Default is 1e8

spatialwarnings.debug.fit_warn_on_bound logical value. Warn if the fit is at the boundary of the valid range for distribution parameter

spatialwarnings.debug.fit_warn_on_NA logical value. Warn if the returned fit has NA/NaN parameters

Value

A list containing at list the following components:

type The type of distribution fitted (as a character string)

method The method used for the fit - here, maximum likelihood, 'll'

ll The log likelihood at the estimated parameter values

xmin The value of xmin used for the fit

npars The number of parameters of the distribution

Additionally, this list may have one or more of the following elements depending on the type of distribution that has been fitted:

plexpo The exponent of the power-law

cutoff The rate of truncation, for truncated power law and exponential fits

meanlog The mean of the lognormal distribution

sdlog The s.d. of the lognormal distribution

References

Clauset, Aaron, Cosma Rohilla Shalizi, and M. E. J. Newman. 2009. "Power-Law Distributions in Empirical Data." *SIAM Review* 51 (4): 661–703. <https://doi.org/10.1137/070710111>.

See Also

[patchdistr_sews](#), [xmin_estim](#)

Examples

```
# Fit an exponential model to patch size distribution
exp_fit(patchsizes(forestgap[[8]]))

# Use the estimated parameters as an indicator function

get_truncation <- function(mat) {
  c(exp_cutoff = exp_fit(patchsizes(mat))$cutoff)
}
trunc_indic <- compute_indicator(forestgap, get_truncation)
plot(trunc_indic)
plot(indictest(trunc_indic, nulln = 19))
```

raw_cg_moran	<i>Moran's Index at lag of 1</i>
--------------	----------------------------------

Description

This functions computes the Moran's spatial correlation index (with lag one) on a matrix.

Usage

```
raw_cg_moran(mat, subsize = 1)
```

Arguments

mat	A matrix
subsize	logical. Dimension of the submatrix used to coarse-grain the original matrix (set to 1 for no coarse-graining).

Value

The Moran's I index measuring autocorrelation at lag 1 as a named vector

References

Dakos, V., van Nes, E. H., Donangelo, R., Fort, H., & Scheffer, M. (2010). Spatial correlation as leading indicator of catastrophic shifts. *Theoretical Ecology*, 3(3), 163-174.

Legendre, P., & Legendre, L. F. J. (2012). *Numerical Ecology*. Elsevier Science.

See Also

[generic_sews](#)

Examples

```
data(serengeti)
raw_cg_moran(serengeti[[1]], subsize = 1)
```

raw_cg_skewness	<i>Skewness indicator</i>
-----------------	---------------------------

Description

Compute the spatial skewness of spatial data (a matrix).

Usage

```
raw_cg_skewness(mat, subsize = 5, absolute = TRUE)
```

Arguments

mat	A matrix. The matrix values can be logical, with FALSE (empty) or TRUE (occupied) values. The entries can also be continuous (like NDVI or EVI data).
subsize	Dimension of the submatrix used to coarse-grain the original matrix. This must be an integer less than size of the full matrix. Coarse-graining reduces the size of the matrix by a factor subsize in each dimension of the matrix. Skewness is calculated on the coarse-grained matrix.
absolute	Should the function return the absolute value or raw value of skewness ?

Details

Spatial skewness is a measure of fluctuations in space; specifically, it measures if fluctuations are getting biased (skewed) in one direction. Based on the theory of critical slowing down, when systems approach critical points they are expected to show increased fluctuations in space. Thus, increasing spatial skewness is proposed as an early warning signal of impending critical transitions.

Computing spatial skewness is straightforward. However, detecting trends of skewness that correspond to critical slowing down can be tricky, especially if data come from discrete classification of state variable.

Many high resolution spatial data are classified as FALSE (empty) or TRUE (occupied by plant). In such cases, spatial skewness captures just the skewness in data, but not that of spatial structure. To resolve the issue, this function employs a method called coarse-graining, proposed in Kefi et al (2014), and described in detail in Sankaran et al. (2017). One must specify a subsize above one for binary valued data sets to obtain meaningful values.

subsize has to be an integer. It has to be less than or equal to half of matrix size (N). subsize must also be preferably a divisor of N. If it is not a divisor of N, the remainder rows and columns are discarded when computing coarse-graining matrices.

Null model evaluations are also done on coarse-grained matrices.

Value

The spatial skewness of the matrix as a named vector

References

Guttal, V., and Jayaprakash, C. (2009). Spatial variance and spatial skewness: leading indicators of regime shifts in spatial ecological systems. *Theoretical Ecology*, 2(1), 3-12.

Kefi, S., Guttal, V., Brock, W.A., Carpenter, S.R., Ellison, A.M., Livina, V.N., et al. (2014). Early Warning Signals of Ecological Transitions: Methods for Spatial Patterns. *PLoS ONE*, 9, e92097.

Sankaran, S., Majumder, S., Kefi, S., and Guttal, V. (2017). Implication of being discrete and spatial in detecting early warning signals of regime shifts. *Ecological indicators*.

See Also

[generic_sews](#)

Examples

```
data(serengeti)

raw_cg_skewness(serengeti[[1]])
compute_indicator(serengeti, fun = raw_cg_skewness, subsize = 5)
```

raw_cg_variance	<i>Spatial variance indicator</i>
-----------------	-----------------------------------

Description

This functions computes the spatial variance of a matrix.

Usage

```
raw_cg_variance(mat, subsize = 5)
```

Arguments

mat	A matrix. Its values can be logical, with FALSE (empty) or TRUE (occupied) values. The entries can also be continuous (like NDVI or EVI data).
subsize	Dimension of the submatrix used to coarse-grain the original matrix. This must be an integer less than size of the full matrix. Coarse-graining reduces the size of the matrix by a factor subsize in each dimension of the matrix. Variance is calculated on the coarse-grained matrix.

Details

Spatial variance is a measure of fluctuations in space. Based on the theory of critical slowing down, when systems approach critical points they are expected to show increased fluctuations in space. Thus, increasing spatial variance is proposed as an early warning signal of impending critical transitions.

Many high resolution spatial data are classified as FALSE (empty) or TRUE (occupied). In such cases, spatial variance captures just the variance in data, but not that of spatial structure. To resolve the issue, this function employs a method called coarse-graining, proposed in Kefi et al (2014), and described in detail in Sankaran et al. (2017). One must specify a subsize above one for binary valued data sets to obtain meaningful values.

subsize has to be an integer. It has to be less than or equal to half of matrix size (N). subsize must also be preferably a divisor of N. If it is not a divisor of N, the remainder rows and columns are discarded when computing coarse-graining matrices.

Null model evaluations are also done on coarse-grained matrices.

Value

The variance of the coarse-grained matrix as a named vector

References

Guttal, V., and Jayaprakash, C. (2009). Spatial variance and spatial skewness: leading indicators of regime shifts in spatial ecological systems. *Theoretical Ecology*, 2(1), 3-12.

Kefi, S., Guttal, V., Brock, W.A., Carpenter, S.R., Ellison, A.M., Livina, V.N., et al. (2014). Early Warning Signals of Ecological Transitions: Methods for Spatial Patterns. *PLoS ONE*, 9, e92097.

Sankaran, S., Majumder, S., Kefi, S., and Guttal, V. (2017). Implication of being discrete and spatial in detecting early warning signals of regime shifts. *Ecological Indicators*.

See Also

[generic_sews](#)

Examples

```
data(serengeti)
raw_cg_variance(serengeti[[1]])
compute_indicator(serengeti, fun = raw_cg_variance, subsize = 5)
```

raw_clustering	<i>Clustering of pairs</i>
----------------	----------------------------

Description

Compute the number of pairs of neighbor cells in a landscape and derive from it clustering indices

Usage

```
raw_clustering(mat, wrap = TRUE, use_8_nb = FALSE)
```

```
pair_counts(mat, wrap = TRUE, use_8_nb = FALSE, prop = TRUE)
```

Arguments

mat	A matrix, usually with discrete values (logical, integers, etc.)
wrap	Whether space should be considered to wrap at the edges
use_8_nb	Set to TRUE to use an 8-way neighborhood. The default is set to FALSE, which uses a 4-way neighborhood
prop	Return the counts for all pairs in the matrix (FALSE), or the proportions for each pair (out of all possible pairs)

Details

The clustering of pairs is defined as the density of pairs, i.e. the proportion of all neighboring pairs of cells that share the same state, divided by the null expectation given a random, homogeneous spatial structure.

For example, let's consider a matrix with two states, 'a' and 'b'. `raw_clustering` will count all pairs of cells 'a-a' or 'b-b', and divide this by the total number of pairs. This proportion is then again divided by the probability of obtaining these proportion of pairs under the assumption of no spatial structure (random mixing of cells in the matrix).

Clustering is equal to one when there is no spatial structure. It is above one when two states are found next to each other (i.e. cluster) more than expected by chance. Values below one means that those two states tend to be neighbors less frequently than expected by chance.

If you are only interested in the proportion of pairs for each combination of states, you can use the function `pair_counts`, which returns a matrix with as many rows and columns as there are states in the matrix, and contains the counts for all possible pairs of cells found in the matrix (or their relative proportions).

Value

A vector with the requested clustering values for `raw_clustering`, whose names are equal to each state (unique value) found in the original matrix, preceded by 'clust_' (to make sure names are compatible with other functions).

`pair_counts` returns the counts of pairs of states in the matrix, or the proportion of each pair, depending on the value of `prop`

Examples

```
# The clustering of a random matrix is close to one
ls <- 100 # lattice size
mm <- matrix(sample(c("sp1", "sp2", "sp3", "sp4"), size = ls^2, replace = TRUE),
             nrow = ls, ncol = ls)
clust <- raw_clustering(mm, wrap = TRUE, use_8_nb = TRUE)
print(clust)

# Compute clustering along the gradient for the serengeti dataset

data(forestgap)
clust_indic <- compute_indicator(serengeti, raw_clustering,
                               wrap = TRUE, use_8_nb = FALSE)
# The interesting one is the clustering of state 0 (FALSE in the original matrix),
# which corresponds to grassland pixels, which get more and more clustered with
# increasing rainfall (see also ?generic_sews for how that compares with generic
# indicators)
plot(clust_indic, along = serengeti.rain)

# Add null trend
clust_test <- indictest(clust_indic, nulln = 19)
plot(clust_test, along = serengeti.rain)

# Show the proportion of each pairs of states in the matrix...
pair_counts(serengeti[[5]])

# ... or the total count
pair_counts(serengeti[[5]], prop = FALSE)
```

raw_flowlength_uniform

Flow length (uniform slope)

Description

Compute a simple approximation of the flow length assuming a constant slope

Usage

```
raw_flowlength_uniform(mat, slope, cell_size)
```

Arguments

mat	The input matrix (must be a logical matrix)
slope	The slope of the area documented by the matrix (in degrees).
cell_size	The horizontal size of a cell in the matrix (as viewed from above).

Details

This function computes the Flowlength of a given matrix, using a uniform approximation (the slope is constant across the whole matrix, with maximum slope being from the top of the matrix to its bottom), as per Rodriguez et al. (2017). See [flowlength_sews](#) for more details.

Value

A named vector of length 1 containing the flow length numerical value

References

Rodriguez, F., A. G. Mayor, M. Rietkerk, and S. Bautista. 2017. A null model for assessing the cover-independent role of bare soil connectivity as indicator of dryland functioning and dynamics. Ecological Indicators.

See Also

[flowlength_sews](#)
[indictest](#), to test the significance of indicator values.

Examples

```
raw_flowlength_uniform(arizona[[1]], slope = 20, cell_size = 1)
```

raw_kbdm	<i>Kolmogorov complexity of a matrix</i>
----------	--

Description

Compute the Kolmogorov complexity of a matrix using the Block Decomposition Method (requires the acss package).

Usage

```
raw_kbdm(mat, subsize)
```

Arguments

mat	A logical matrix (with TRUE/FALSE values)
subsize	A submatrix size to carry out the Block Decomposition Method (must be between 1 and 3)

Details

The Kolmogorov complexity cannot be computed directly for large strings (i.e. matrices). However, the complexity of smaller submatrices can be estimated, then combined to obtain an approximation of the complexity of the whole matrix. This method, the Block Decomposition Method is implemented in this function. See also [kbdm_sews](#) for more details.

Value

The KBDM numeric value as a named vector

See Also

[kbdm_sews](#), [acss](#)

Examples

```
raw_kbdm(forestgap[[1]], subsize = 3)
```

raw_moran

Spatial correlation at lag 1

Description

This function computes the Moran's I index of spatial correlation at lag 1.

Usage

```
raw_moran(mat)
```

Arguments

mat A matrix containing logical, numeric or integer values

Details

This function returns the spatial correlation as measured by the Moran's I index. If the variance of the matrix is zero, then NaN is returned. This function assumes a 4-way neighborhood (a.k.a. von-Neumann neighborhood), and does not wrap around at the sides of the matrix.

Value

The Moran's I numeric value as a numeric number.

See Also

[generic_sews](#)

Examples

```
# Spatial correlation of white noise is close to zero
rmat <- matrix(runif(1000) > .5, ncol = 100)
raw_moran(rmat)

# Spatial correlation of a half-ones / half-zeros matrix is close to one.
# This would produce close but inaccurate results in version <3.0.2
m <- cbind(matrix(1, nrow = 100, ncol = 50),
            matrix(0, nrow = 100, ncol = 50))

raw_moran(m)
```

raw_plrange	<i>Power-law range indicator</i>
-------------	----------------------------------

Description

Compute the power-law range of a matrix

Usage

```
raw_plrange(mat, xmin_bounds = NULL)
```

Arguments

mat	A logical matrix, or a list of logical matrices
xmin_bounds	A vector of two integer values, defining a range in which to search for the best xmin (see Details).

Details

Some ecosystems show typical changes in their patch-size distribution as they become more and more degraded. In particular, an increase in the truncation of the patch-size distribution (PSD) is expected to occur. The power-law range (PLR) measures the truncation of the PSD in a single value (see also [patchdistr_sews](#) for more details).

To compute the PLR, power-laws are fitted with a variable minimum patch size (xmin) and the one with the lowest Kolmogorov-Smirnov distance to the empirical distribution is retained. PLR is then computed using this best-fitting xmin:

$$\frac{\log(x_{max}) - \log(x_{min})}{\log(x_{max}) - \log(x_{smallest})}$$

where x_{max} is the maximum observed patch size, and $x_{smallest}$ is the minimum observed patch size.

Value

A named vector containing the power-law range value

References

Clauset, A., Shalizi, C. R., & Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review*, 51(4), 661-703.

Berdugo, M., Kefi, S., Soliveres, S. & Maestre, F.T. (2017). Plant spatial patterns identify alternative ecosystem multifunctionality states in global drylands. *Nature in Ecology and Evolution*.

See Also

[patchdistr_sews](#)

Examples

```
forestgap.plr <- raw_plrange(forestgap[[2]])

# Restrict to small xmins
forestgap.plr2 <- indicator_plrange(forestgap[[2]], xmin_bounds = c(1, 10))
```

raw_sdr

Spectral Density Ratio (SDR) indicator

Description

Compute the ratio of low frequencies over high frequencies of the r-spectrum.

Usage

```
raw_sdr(mat, sdr_low_range = NULL, sdr_high_range = NULL)
```

Arguments

mat	A matrix with continuous values, or a logical matrix (TRUE/FALSE values).
sdr_low_range	The range of values (in proportion) to use for the computation of the spectral density ratio. For example, for the lowest 20% (default value), set sdr_low_range to <code>c(0, .2)</code> .
sdr_high_range	The range of values (in proportion) to use for the computation of the spectral density ratio. For example, for the highest 20% (default value), set sdr_high_range to <code>c(.8, 1)</code> .

Details

SDR measures the increase in long-range correlations before a critical point. It is the ratio of the average low frequency value over high frequency values. In this implementation, an increase in SDR implies a "reddening" of the [r-spectrum](#). See also [spectral_sews](#) for a more complete description.

Low and high frequencies are averaged in order to compute the SDR. The parameters `sdr_low_range` and `sdr_high_range` control which frequencies are selected for averaging. For example `sdr_low_range = c(0, .2)` (default) uses the lower 20 the average of low frequencies. `sdr_high_range = c(.8, 1)` uses the higher 20

Value

The SDR values computed on the matrix as a named vector

References

Carpenter, S.R. & Brock, W.A. (2010). Early warnings of regime shifts in spatial dynamics using the discrete Fourier transform. *Ecosphere*

See Also

[indictest](#), [rspectrum](#), [plot_spectrum](#), [spectral_sews](#), [extract_spectrum](#)

Examples

```
data(serengeti)
serengeti.sdr <- raw_sdr(serengeti[[1]],
                        sdr_low_range = c(0, 0.2),
                        sdr_high_range = c(0.8, 1))
compute_indicator(serengeti, raw_sdr)
```

raw_structvar

Structural variance

Description

Compute the structural variance on a matrix.

Usage

```
raw_structvar(mat, model = "sph", nmax = 100000L, nbins = 32, cutoff = NULL)
```

Arguments

mat	A matrix
model	The variogram model to use, either "sph" (for a spherical model) or "exp" (for an exponential model)
nmax	The maximum number of pairs of cells to use when computing the variogram
nbins	Number of distance bins to use to compute the variogram
cutoff	Maximum distance to consider in the variogram. If NULL, then a distance equal to one third of the diagonal of the matrix is used

See Also

raw_variogram_metrics, variogram_sews

Examples

```
raw_structvar(serengeti[[5]])
```

raw_variogram_metrics *Variogram parameters*

Description

Compute the nugget, partial sill, correlation range and structural variance on a matrix.

Usage

```
raw_variogram_metrics(  
  mat,  
  model = "sph",  
  nmax = 100000L,  
  nbins = 32,  
  cutoff = NULL  
)
```

Arguments

mat	A matrix
model	The variogram model to use, either "sph" (for a spherical model) or "exp" (for an exponential model)
nmax	The maximum number of pairs of cells to use when computing the variogram
nbins	Number of distance bins to use to compute the variogram
cutoff	Maximum distance to consider in the variogram. If NULL, then a distance equal to one third of the diagonal of the matrix is used

See Also

variogram_sews, raw_structvar

Examples

```
raw_variogram_metrics(serengeti[[5]])
```

rspectrum	<i>r-spectrum</i>
-----------	-------------------

Description

Compute the r-spectrum of a matrix

Usage

```
rspectrum(mat)
```

Arguments

mat A matrix with logical or numeric values

Details

This functions returns a data.frame with NAs in the rspec column if the input matrix has zero variance. Note that if the matrix is not square, then only the largest square matrix fitting in the upper left corner is used.

Value

A data.frame with two columns: dist, the wave number and rspec, the normalized value of the r-spectrum

See Also

[spectral_sews](#)

Examples

```
# Spectrum of white noise
rmat <- matrix(runif(100*100) > .5, ncol = 100)
spec <- rspectrum(rmat)
plot(spec, type = "l")

# Add some spatial correlation and compare the two spectra
rmat.cor <- rmat
for (i in seq(1, nrow(rmat)-1)) {
  for (j in seq(1, nrow(rmat)-1)) {
```



```

    rmat.cor[i,j] <- mean(rmat[(i-1):(i+1), (j-1):(j+1)])
  }
}
spec.cor <- rspectrum(rmat.cor)
plot(spec.cor, type = "n")
lines(spec, col = "black")
lines(spec.cor, col = "blue")

```

serengeti

Serengeti dataset

Description

Vegetation data along a rainfall gradient in Serengeti national park.

Usage

```
serengeti
```

```
serengeti.rain
```

Format

A list of logical matrices

The annual rainfall corresponding to the matrices in the serengeti dataset, in the corresponding order.

Details

The data-set consists of a rectangular area of size 7.5 km x 90 km. These data are represented as a list of matrices. Each matrix is a moving window of 7.5 km x 7.5 km which moves my 2.5 km along the length of the rectangular data-set.

Each entry in the matrix is vegetation data at a resolution of 30m as classified into binary units with FALSE (grass) and TRUE (forest). The rainfall data provided here is the average rainfall (mm/yr) of a moving window of dimension 7.5km which moves my 2.5 km along the length of the rectangular data-set.

Source

Extracted from Eby's et al (2017) supplementary material <https://github.com/tee-lab/spacetime-csd/>

References

- Eby, S., Agrawal, A., Majumder, S., Dobson, A.P. & Guttal, V. (2017). Alternative stable states and spatial indicators of critical slowing down along a spatial gradient in a savanna ecosystem: *Global Ecology and Biogeography*, 26, 638-649
- Reed, D. N., Anderson, T. M., Dempewolf, J., Metzger, K., & Serneels, S. (2009). The spatial distribution of vegetation types in the Serengeti ecosystem: the influence of rainfall and topographic relief on vegetation patch characteristics. *Journal of Biogeography*, 36(4), 770-782.

simple_sews	simple_sews objects
-------------	---------------------

Description

This help page describes the structure of `simple_sews_*` objects, such as those defined by the classes `simple_sews_single`, `simple_sews_list`

Details

The `spatialwarnings` uses S3 objects (lists) internally to store indicator values, along with the necessary data to plot and display results. It is not recommended to extract data directly from these objects, as they are subject to change with different releases of the package. The preferred method is to use dedicated generic functions such as `plot()` or `as.data.frame()` to display or export the results. Nonetheless, we document the structure of these objects here for reference.

`simple_sews` objects are returned by all indicator functions that return numeric values. This includes for example `generic_sews`, `flowlength_sews`, `compute_indicator` but *not* `patchdistr_sews` or `spectral_sews`, which provide indicators that depend on non-numeric values (e.g. patch-size distribution type), or need to store more information than just a single numerical value (e.g. the spectrum of the input matrix).

`simple_sews` objects come in multiple variants: `simple_sews_single` is the result of an indicator function applied to a single matrix, and `simple_sews_test_single` is the result of `indictest` applied to a `simple_sews_single` object. Both these objects have list equivalents, `simple_sews_list` and `simple_sews_test_list` which are simply a collection of their 'single' equivalent. These 'list' objects are used to store the results of computations when working with multiple matrices.

A `simple_sews_single` object is a list with the following components

`itemvalue` the indicator values. A vector of length one if there is only one numeric value returned by the indicator function (e.g. `flowlength_sews`, or with a length above one otherwise)

`orig_data` the original matrix on which the indicator was computed

`fun.args` the argument used in the call to the indicator function (the function that given a matrix, returns the spatial metrics of interest)

`taskname` a character string describing the current indicator(s) being computed

`indicf` the indicator function, which given the matrix, returns the spatial metric(s) of interest

`simple_sews_test_single` have all of the above components, plus the following:

`nulldistr` the null distribution of values, with `nulln` rows and as many columns as the number of values returned by the indicator function
`null_mean` the mean indicator values in the null distribution
`null_sd` the standard deviation of the null distribution
`null_qsup` the upper quantile of the null distribution, by default the 95 adjust this
`null_qinf` the upper quantile of the null distribution, by default the 05 adjust this
`z_score` the `z_score` of the observed value relative to the null distribution, i.e. $(\text{value} - \text{null_mean}) / \text{null_sd}$
`pval` the p-value of the indicator, i.e. the proportion of values of the null distribution that fall below the observed indicator value
`null_method` the method used to produce the null matrices. See [indictest](#) for details
`nulln` the number of null matrices used
`get_nullmat` a function that can be called to obtain a randomized matrix
`matrixn` the number of the matrix, can be above one if the computations have been run on a list of matrices, or non-existent if only one matrix was used

See Also

[custom_indicator](#)

spectral_sews

Spectrum-based spatial early-warning signals.

Description

Computation of spatial early warning signals based on spectral properties.

Usage

```
spectral_sews(mat, sdr_low_range = NULL, sdr_high_range = NULL, quiet = FALSE)
```

Arguments

<code>mat</code>	The input matrix or a list of matrices.
<code>sdr_low_range</code>	The range of values (in proportion) to use for the computation of the spectral density ratio. For example, for the lowest 20% (default value), set <code>sdr_low_range</code> to <code>c(0, .2)</code> . See also the Details section.
<code>sdr_high_range</code>	The range of values (in proportion) to use for the computation of the spectral density ratio. For example, for the higher 20% (default value), set <code>sdr_high_range</code> to <code>c(.8, 1)</code> . See also the Details section.
<code>quiet</code>	Do not display some warnings

Details

Spectral early warning signals are based on the fact that some dynamical systems can exhibit an change in specific characteristics of their spatial structure when approaching a transition. In particular, long-range correlations are expected to have an increased importance. This is expected to be reflected in the spectrum of the spatial structure as an increase of the relative importance of lower frequencies over higher frequencies ("reddening" of the spatial spectrum).

This task allows computing the radial-spectrum which gives the relative importance of each space scale as a function of distance, from 1 to $N/2$ (N being the minimum between the number of rows and columns). If the matrix is not square, then it is cropped to biggest square that fits within the left side of the matrix.

Additionally, it summarizes this spectrum into a Spectral Density Ratio (SDR), which is the ratio of low frequencies over high frequencies of the r-spectrum. The SDR value is expected to increase before a transition point.

The significance of spectral early-warning signals can be estimated by reshuffling the original matrix (function `indictest`). Indicators are then recomputed on the shuffled matrices and the values obtained are used as a null distribution. P-values are obtained based on the rank of the observed value in the null distribution.

The trend of SDR values can be plotted using the `plot()` method. Alternatively, the spectrum itself can be plotted (with facets if multiple input matrices were used) using the `plot_spectrum` method.

Value

Function `spectral_sews` object of class `spectral_sews_list` or `spectral_sews_single` depending on whether the input was a list of matrices or a single matrix.

Function `indictest`

The plot methods returns a ggplot object (usually displayed immediately when called interactively).

References

Kefi, S., Guttal, V., Brock, W.A., Carpenter, S.R., Ellison, A.M., Livina, V.N., et al. (2014). Early Warning Signals of Ecological Transitions: Methods for Spatial Patterns. PLoS ONE, 9, e92097.

See Also

[rspectrum](#), [plot_spectrum](#), [raw_sdr](#), [extract_spectrum](#)
[indictest](#), to test the significance of indicator values.

Examples

```
data(serengeti)
data(serengeti.rain)

spec_indic <- spectral_sews(serengeti,
                           sdr_low_range = c(0, .2),
```

```

sdr_high_range = c(.8, 1))

summary(spec_indic)

# Display trends along the varying model parameter
plot(spec_indic, along = serengeti.rain)

# Computing spectra many times is expensive, consider setting parallel
# computing using: options(mc.cores = n)

# Assess significance
spec_test <- indictest(spec_indic, nulln = 199)
summary(spec_test)

# Display the SDR trend, now with a grey ribbon representing 5%-95%
# quantiles of the null distribution
plot(spec_test, along = serengeti.rain)

# Add a line highlighting the shift
if (require(ggplot2)) {
  plot(spec_test, along = serengeti.rain) +
    geom_vline(xintercept = 590, color = "red", linetype = "dashed")
}

# Display radial-spectra
plot_spectrum(spec_indic, along = serengeti.rain)

# Graphics can be modified using ggplot2 functions
if (require(ggplot2)) {
  plot_spectrum(spec_indic, along = serengeti.rain) +
    scale_y_log10()
}

```

variogram_sews

Early-Warning signals based on variograms (EXPERIMENTAL)

Description

Compute Early-warning signals based on metrics derived from semi-variograms.

Usage

```
variogram_sews(mat, model = "sph", nmax = 1e+05, nbins = 32, cutoff = NULL)
```

Arguments

<code>mat</code>	A matrix (TRUE/FALSE values) or a list of matrices
<code>model</code>	The variogram model to use, either "sph" (for a spherical model) or "exp" (for an exponential model)
<code>nmax</code>	The maximum number of pairs of cells to use when computing the variogram
<code>nbins</code>	Number of distance bins to use to compute the variogram
<code>cutoff</code>	Maximum distance to consider in the variogram. If NULL, then a distance equal to one third of the diagonal of the matrix is used

Details

During ecosystem degradation and especially before a regime shift occurs in some ecosystems, spatial autocorrelation is expected to increase in a landscape. This increase can be measured based on variograms, which represent how the difference (variance) between two points in a landscape varies as a function of distance.

The approach used to derive variogram-based EWS is to compute the empirical variogram of a landscape (represented passed as a matrix of values), then fit a variogram model to it. Three parameters are then extracted from the variogram model (see Nijp et al. 2019 for a visual description of these parameters):

1. The *nugget* (intercept)
2. The *partial sill*, i.e. the reduction in semivariance at distance zero
3. The *correlation range*, i.e. the distance at which the relationship between semivariance and distance flattens

Additionally, the *structural variance* is computed as (partial sill)/(nugget + partial sill), which quantifies whether the data are spatially structured (structural variance of one), or completely unstructured (value of zero). Theoretical work suggests that partial sill, correlation range and structural variance should increase before a regime shift occurs in an ecosystem (Nijp et al. 2019).

This function offers to fit a spherical model or an exponential model. The best-fitting model depends on your data, you should try different options and review the fits using [plot_variogram](#).

Please note that this part of the package is still experimental and deserves more testing.

Value

A list object of class "variogram_sews", that can be displayed using `summary()`, `plot()`, etc. Significance of values can be tested using [indictest](#).

References

Nijp, Jelmer J., Arnaud J.A.M. Temme, George A.K. Voorn, Lammert Kooistra, Geerten M. Hengeveld, Merel B. Soons, Adriaan J. Teuling, and Jakob Wallinga. (2019) Spatial Early Warning Signals for Impending Regime Shifts: A Practical Framework for Application in Real-world Landscapes. *Global Change Biology* 25 (6): 1905-21. doi:10.1111/gcb.14591

See Also

[raw_structvar](#), [plot_variogram](#), [extract_variogram](#), [predict_indicest](#), to test the significance of indicator values.

Examples

```
serengeti_ews <- variogram_sews(serengeti,
                               model = "exp")
plot(serengeti_ews, along = serengeti.rain)
summary(serengeti_ews)

plot_variogram(serengeti_ews)

# nulln should be set to a higher values for meaningful results
serengeti_test <- indicest(serengeti_ews, nulln = 9)
plot(serengeti_test) # gray ribbons indicate the null indicator values
summary(serengeti_test)
```

variogram_sews_plot	<i>Early-warning signals based on variograms</i>
---------------------	--

Description

Plot trends of indicators based on variograms

Usage

```
## S3 method for class 'variogram_sews'
plot(x, along = NULL, ...)

plot_variogram(x, along = NULL, ...)

## S3 method for class 'variogram_sews'
plot_variogram(x, along = NULL, ...)

## S3 method for class 'variogram_sews_test'
plot(x, along = NULL, what = "value", display_null = TRUE, ...)

## S3 method for class 'variogram_sews_test'
plot_variogram(x, along = NULL, what = "value", display_null = TRUE, ...)
```

Arguments

<code>x</code>	An object produced by <code>variogram_sews</code> , or the result of applying <code>indictest</code> on such object.
<code>along</code>	A vector providing values along which the indicator trends will be plotted. If <code>NULL</code> then the indicator values are plotted sequentially in their original order.
<code>...</code>	Other arguments are ignored.
<code>what</code>	The trendline to be displayed. Defaults to the indicator's values ("value") but other metrics can be displayed. Accepted values are "value", "pval", "difference" (obs - null mean), or "z_score" ((obs - null mean) / (null sd)).
<code>display_null</code>	Chooses whether a grey ribbon should be added to reflect the null distribution. Note that it can not be displayed when the trend line reflects something else than the indicator values (when <code>what</code> is not set to "value").

Details

The `plot()` function will display how the estimated variogram parameters change along a set of values (passed with argument `along`). If the object passed has been processed through `indictest`, then the null values are also displayed. `plot_variogram()` can be used to display the individual variograms that have been fit to the data.

See Also

`variogram_sews`, `indictest`, `plot_variogram`

Examples

```
serengeti_ews <- variogram_sews(serengeti, model = "exp")

# Display the change in variogram parameters
plot(serengeti_ews, along = serengeti.rain) +
  ggplot2::labs(x = "Rainfall (mm)")

# Visualize the fitted variograms
plot_variogram(serengeti_ews, along = serengeti.rain)

# Test the trends (nulln should be set to a higher value to obtain
# meaningful results
serengeti_test <- indictest(serengeti_ews, nulln = 19)
plot(serengeti_test, along = serengeti.rain)
plot_variogram(serengeti_test, along = serengeti.rain)
```

`variogram_sews_predict`*predict() method for variogram_sews objects*

Description

Export the fitted variogram(s)

Usage

```
## S3 method for class 'variogram_sews_list'  
predict(object, newdist = NULL, ...)
```

Arguments

<code>object</code>	An object produced by <code>variogram_sews</code>
<code>newdist</code>	A vector of distances at which to return the variogram fit values (defaults to 128 regularly-spaced values).
<code>...</code>	Additional arguments (ignored)

Value

A data.frame with the distances (column `dist`), the fitted values (`gamma`), and if object contains more than one matrix, a column `matrixn`.

See Also

[variogram_sews](#)

Examples

```
vario_indics <- variogram_sews(serengeti)  
predict(vario_indics)  
vario_test <- indictest(vario_indics, nulln = 19)  
predict(vario_test) # same result
```

`xmin_estim`*Estimate the minimum patch size of a power-law distribution*

Description

When fitting a power-law to a discrete distribution, it might be worth discarding points below a certain threshold (`xmin`) to improve the fit. This function estimates the optimal `xmin` based on the Kolmogorov-Smirnoff distance between the fit and the empirical distribution, as suggested by Clauset et al. (2009).

Usage

```
xmin_estim(dat, bounds = range(dat))
```

Arguments

<code>dat</code>	A vector of integer values
<code>bounds</code>	A vector of two values representing the bounds in which the best <code>xmin</code> is searched

Details

The function returns NA if `dat` has only three unique values or if the power-law fit failed.

Value

The estimated `xmin` as an integer value

References

Clauset, A., Shalizi, C. R., & Newman, M. E. (2009). Power-law distributions in empirical data. SIAM review, 51(4), 661-703.

See Also

[patchdistr_sews](#)

Examples

```
psd <- patchsizes(forestgap[[5]])  
xmin_estim(psd)
```

Index

- * **datasets**
 - arizona, [3](#)
 - dda, [7](#)
 - forestgap, [14](#)
 - serengeti, [49](#)
- acss, [22](#), [43](#)
- arizona, [3](#)
- binomial, [21](#)
- coarse_grain, [3](#)
- compute_indicator, [20](#), [21](#), [50](#)
- compute_indicator (create_indicator), [5](#)
- convert_to_matrix, [4](#)
- create_indicator, [5](#)
- custom_indicator, [51](#)
- custom_indicator (create_indicator), [5](#)
- dda, [7](#), [25](#)
- display_matrix, [8](#)
- display_size_info, [9](#)
- dlnorm, [25](#)
- dLSW, [9](#), [25](#)
- exp_fit (pl_fit), [33](#)
- extract_spectrum, [11](#), [33](#), [46](#), [52](#)
- extract_variogram, [11](#), [55](#)
- flowlength, [21](#)
- flowlength_sews, [12](#), [21](#), [42](#), [50](#)
- forestgap, [14](#)
- gam, [21](#)
- gaussian, [21](#)
- generic_sews, [4](#), [15](#), [21](#), [36](#), [38](#), [39](#), [43](#), [50](#)
- indicator_plrange, [17](#)
- indicator_psdtype, [18](#)
- indictest, [13](#), [16](#), [20](#), [22](#), [28](#), [29](#), [33](#), [42](#), [46](#), [50–52](#), [54–56](#)
- kbdm_sews, [21](#), [22](#), [43](#)
- label, [23](#), [31](#)
- lnorm_fit (pl_fit), [33](#)
- LSW_fit (dLSW), [9](#)
- lsw_sews, [10](#), [24](#)
- pair_counts (raw_clustering), [40](#)
- patchdistr_sews, [17](#), [18](#), [20](#), [24](#), [26](#), [29](#), [30](#), [35](#), [44](#), [45](#), [50](#), [58](#)
- patchdistr_sews_plot, [21](#), [28](#)
- patchdistr_sews_predict, [30](#)
- patchsizes, [19](#), [23](#), [24](#), [27](#), [28](#), [31](#), [31](#)
- percolation, [19](#)
- percolation (label), [23](#)
- pl_fit, [19](#), [33](#)
- plot, [28](#)
- plot.patchdistr_sews
 - (patchdistr_sews_plot), [28](#)
- plot.simple_sews_list
 - (plot.simple_sews_test), [32](#)
- plot.simple_sews_test, [32](#)
- plot.variogram_sews
 - (variogram_sews_plot), [55](#)
- plot.variogram_sews_test
 - (variogram_sews_plot), [55](#)
- plot_distr, [27](#), [28](#)
- plot_distr (patchdistr_sews_plot), [28](#)
- plot_spectrum, [33](#), [46](#), [52](#)
- plot_variogram, [54–56](#)
- plot_variogram (variogram_sews_plot), [55](#)
- pLSW (dLSW), [9](#)
- predict, [28](#), [55](#)
- predict.patchdistr_sews_single
 - (patchdistr_sews_predict), [30](#)
- predict.variogram_sews_list
 - (variogram_sews_predict), [57](#)
- r-spectrum, [46](#)
- raw_cg_moran, [16](#), [36](#)

raw_cg_skewness, [16](#), [37](#)
raw_cg_variance, [16](#), [38](#)
raw_clustering, [40](#)
raw_flowlength_uniform, [13](#), [41](#)
raw_kbdm, [22](#), [42](#)
raw_lsw_aicw, [25](#)
raw_lsw_aicw(lsw_sews), [24](#)
raw_moran, [43](#)
raw_patch_radii_skewness, [25](#)
raw_patch_radii_skewness(lsw_sews), [24](#)
raw_plrange, [44](#)
raw_sdr, [45](#), [52](#)
raw_structvar, [46](#), [55](#)
raw_variogram_metrics, [47](#)
rspectrum, [11](#), [33](#), [46](#), [48](#), [52](#)

scale_fill_brewer, [8](#)
separate page, [27](#)
serengeti, [49](#)
simple_sews, [6](#), [16](#), [50](#)
simple_sews_*, [6](#)
simple_sews_object, [13](#)
simple_sews_object(simple_sews), [50](#)
spatialwarnings, [4](#), [50](#)
spectral_sews, [11](#), [21](#), [33](#), [46](#), [48](#), [50](#), [51](#)
standard definition, [34](#)

this page, [27](#)
tpl_fit(pl_fit), [33](#)

variogram_sews, [12](#), [53](#), [56](#), [57](#)
variogram_sews_plot, [55](#)
variogram_sews_predict, [57](#)

xmin_estim, [34](#), [35](#), [58](#)