Package 'geohabnet'

June 27, 2024

Title Geographical Risk Analysis Based on Habitat Connectivity

Version 2.1.3

Date 2024-06-26

Description

The geohabnet package is designed to perform a geographically or spatially explicit risk analysis of habitat connectivity. Xing et al (2021) <doi:10.1093/biosci/biaa067> proposed the concept of cropland connectivity as a risk factor for plant pathogen or pest invasions. As the functions in geohabnet were initially developed thinking on cropland connectivity, users are recommended to first be familiar with the concept by looking at the Xing et al paper. In a nutshell, a habitat connectivity analysis combines information from maps of host density, estimates the relative likelihood of pathogen movement between habitat locations in the area of interest, and applies network analysis to calculate the connectivity of habitat locations.

The functions of geohabnet are built to conduct a habitat connectivity analysis relying on geographic parameters (spatial resolution and spatial extent), dispersal parameters (in two commonly used dispersal kernels: inverse power law and negative exponential models), and network parameters (link weight thresholds and network metrics).

The functionality and main extensions provided by the functions in geohabnet to habitat connectivity analysis are

a) Capability to easily calculate the connectivity of locations in a landscape using a single function, such as sensitivity_analysis() or msean().

b) As backbone datasets, the geohabnet package supports the use of two publicly available global datasets to calculate cropland density. The backbone datasets in the geohabnet package include crop distribution maps from Monfreda, C., N. Ramankutty, and J. A. Foley (2008) <doi:10.1029/2007gb002947> ``Farming the planet: 2. Geographic distribution of crop areas, yields, physiological types, and net primary production in the year 2000, Global Biogeochem. Cycles, 22, GB1022'' and International Food Policy Research Institute (2019) <doi:10.7910/DVN/PRFF8V> ``Global Spatially-Disaggregated Crop Production Statistics Data for 2010 Version 2.0, Harvard Dataverse, V4''. Users can also provide any other geographic dataset that represents host density.

c) Because the geohabnet package allows R users to provide maps of host density (as originally in Xing et al (2021)), host landscape density (representing the geographic distribution of either crops or wild species), or habitat distribution (such as host landscape density adjusted by climate suitability) as inputs, we propose the term habitat connectivity.

d) The geohabnet package allows R users to customize parameter values in the habitat connectivity analysis, facilitating context-specific (pathogen- or pest-specific) analyses.

e) The geohabnet package allows users to automatically visualize maps of the habitat connectivity of locations resulting from a sensitivity analysis across all customized parameter combinations. The primary function is sean() and sensitivity analysis().

Most functions in geohabnet provide as three main outcomes: i) A map of mean habitat connectivity across parameters selected by the user, ii) a map of variance of habitat connectivity across the selected parameters, and iii) a map of the difference between the ranks of habitat connectivity and habitat density.

Each function can be used to generate these maps as 'final' outcomes.

Each function can also provide intermediate outcomes, such as the adjacency matrices built to perform the analysis, which can be used in other network analysis.

Refer to article at <https://garrettlab.github.io/HabitatConnectivity/articles/analysis.html> to see examples of each function and how to access each of these outcome types.

To change parameter values, the file called parameters.yaml stores the parameters and their values, can be accessed using get_parameters() and set new parameter values with set_parameters(). Users can modify up to ten parameters.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Imports config (>= 0.3.1), geodata (>= 0.6.2), geosphere (>= 1.5.18), igraph (>= 2.0.3), terra (>= 1.7.29), easycsv (>= 1.0.8), yaml (>= 2.3.7), stats, stringr (>= 1.5.0), memoise (>= 2.0.1), graphics, rlang (>= 1.1.1), viridisLite (>= 0.4.2), beepr (>= 1.3), rnaturalearth (>= 0.3.3), tools, methods, future.apply (>= 1.11.0), future (>= 1.33.0), magrittr (>= 2.0.3), ggplot2 (>= 3.5.1), patchwork (>= 1.2.0)

Suggests devtools, knitr, lintr (>= 3.0.2), mockthat (>= 0.2.8), pkgdown, rmarkdown, testthat (>= 3.1.7)

URL https://garrettlab.github.io/HabitatConnectivity/,

https://CRAN.R-project.org/package=geohabnet/,

https://github.com/GarrettLab/HabitatConnectivity/tree/main/geohabnet/,

https://www.garrettlab.com/

BugReports https://github.com/GarrettLab/HabitatConnectivity/issues

VignetteBuilder knitr

NeedsCompilation no

Author Krishna Keshav [aut, cre],
Aaron Plex [aut] (<https://orcid.org/0000-0001-7317-3090>),
Garrett Lab [ctb] (https://garrettlab.com),
Karen Garrett [aut] (<https://orcid.org/0000-0002-6578-1616>),
University of Florida [cph, fnd] (https://www.ufl.edu)

Maintainer Krishna Keshav <kkeshav@ufl.edu>

Repository CRAN

Date/Publication 2024-06-27 11:50:02 UTC

Contents

.indices   *Internal function to extract risk indices from a list of crop rasters.*

Description

Internal function to extract risk indices from a list of crop rasters.

Usage

```
.indices(crop_rasters)
```

Arguments

crop_rasters   List of raster objects.

Value

A list of risk indices.

---

connectivity   *Calculate and plot maps*

Description

Calculate mean, variance and difference. The result is produced in form of maps plotted with predefined settings. Currently, the settings for plot cannot be customized. Default value is TRUE for all logical arguments

Usage

```
connectivity(
  host,
  indices,
  global = FALSE,
  east = NULL,
  west = NULL,
  geoscale = NULL,
  res = reso(),
  pmean = TRUE,
  pvar = TRUE,
  pdiff = TRUE,
  outdir = tempdir()
)
```

Arguments

host	SpatRaster. Host density map aka `SpatRaster` object
indices	SpatRaster. Collection of risk indices.
global	Logical. `TRUE` if global analysis is desired, `FALSE` otherwise. `east` and `west` are required when `TRUE`.
east	SpatRaster. Collection of risk indices on eastern extent.
west	SpatRaster. Collection of risk indices on western extent. When `TRUE`, `geoscale` is ignored. Default is `TRUE`.
geoscale	Vector. geographical scale. Default is `NULL`.
res	Numeric. Map resolution. This value is used in aggregation and dis-aggregation operation. Default is [`reso()`].
pmean	Logical. `TRUE` if map of mean should be plotted, `FALSE` otherwise.
pvar	Logical. `TRUE` if variance map should be plotted, `FALSE` otherwise.
pdiff	Logical. `TRUE` if difference map should be plotted, `FALSE` otherwise.
outdir	Character. Output directory for saving raster in TIFF format. Default is [`tempdir()`].

Details

`indexes` are actually risk indices i.e. lis of `spatRaster` objects resulting from operations on crop's raster and parameters provided in either `parameters.yaml` or [`sean()`].

It will save all the opted plots using - `pmean`, `pvar` and `pdiff`. File will be saved in provided value of `outdir` or [`tempdir()`].If [`interactive()`] is `TRUE`, then plots can be seen in active plot window. E.g. Rstudio. The maps are plotted using SpatRaster object. These objects are available as a return value of this function.

Value

Gmap. See details.

References

Yanru Xing, John F Hernandez Nopsa, Kelsey F Andersen, Jorge L Andrade-Piedra, Fenton D Beed, Guy Blomme, Mónica Carvajal-Yepes, Danny L Coyne, Wilmer J Cuellar, Gregory A Forbes, Jan F Kreuze, Jürgen Kroschel, P Lava Kumar, James P Legg, Monica Parker, Elmar Schulte-Geldermann, Kalpana Sharma, Karen A Garrett, *Global Cropland connectivity: A Risk Factor for Invasion and Saturation by Emerging Pathogens and Pests*, BioScience, Volume 70, Issue 9, September 2020, Pages 744–758, doi:10.1093/biosci/biaa067

Hijmans R (2023). *terra: Spatial Data Analysis*. R package version 1.7-46, https://CRAN.R-project.org/package=terra

See Also

[`hci_mean()`], [`hci_variance()`], [`hci_diff()`]

---

cropharvest_rast   *Get raster object for crop*

Description

Get cropland information in a form of raster object from data source for crop

Usage

```
cropharvest_rast(crop_name, data_source)
```

Arguments

crop_name   Name of the crop

data_source   Data source for cropland information

Value

Raster.

Examples

```
cropharvest_rast("avocado", "monfreda")
```

---

crops_rast   *Get sum of rasters for individual crops*

Description

Takes crop names and returns raster object which is sum of raster of individual crops. Currently, only supports crops listed in geodata::monfredaCrops(), geodata::spamCrops() If crop is present in multiple sources, then their mean is calculated.

Usage

```
crops_rast(crop_names)
```

Arguments

crop_names   A named list of source along with crop names

Value

SpatRaster. Raster object which is sum of all the individual crop raster

See Also

cropharvest_rast()

Examples

```
crops_rast(list(monfreda = c("wheat", "barley"), mapspam2010 = c("wheat", "potato")))
```

---

dist_methods   *Distance methods supported*

Description

Contains supported strategies to calculate distance between two points. Use of one the methods in sean() or sensitivity_analysis().

Usage

```
dist_methods()
```

Value

vector

Examples

```
dist_methods()
```

GeoModel-class   *GeoModel class*

Description

A ref class to represent results of dispersal models.

Slots

amatrix An adjacency matrix to represent network.

index A raster object representing the index.

hdthreshold A numeric value representing the threshold.

aggregation A character value representing the aggregation method.

linkthreshold A numeric value representing the link threshold.

beta A numeric value representing the beta parameter.

gamma A numeric value representing the gamma parameter.

GeoNetwork-class *GeoNetwork*

Description

An S4 class representing a network of geographical data. This will wrap all the results from the risk analysis using `sean()` or `sensitivity_analysis()`. This class contains the field from `Gmap` class which has results in the form of `SpatRaster` and TIFF file.

Usage

```
## S4 replacement method for signature 'GeoNetwork'
host_density(x) <- value
```

Arguments

x   GeoNetwork.

value   SpatRaster.

Value

GeoNetwork.

Slots

rasters A list of `GeoRasters` objects.

host_density A `SpatRaster` representing the host density.

me_rast A raster representing mean risk index.

me_out Character. A file path to the mean risk index raster.

diff_rast A raster representing difference.

diff_out Character. A file path to the difference raster.

var_rast A raster representing variance.

var_out Character. A file path to the variance raster.

GeoRasters-class *GeoRaster class*

Description

A class to represent raster vis-a-vis risk indices. This class encapsulates the results of apply dispersal models and metrics.

Fields

host_density SpatRaster. A spatial raster representing host density.

rasters List. List of raster representing risk indices. These are of type GeoModels.

global Boolean. True if contains GlobalRast object, False otherwise.

---

geoscale_param *Get geographical scales from the parameters*

Description

This function returns a list of geographical scales set in global and custom extent in parameters.yaml. If global is TRUE, the CustomExt is ignored.

Usage

```
geoscale_param(gparams = load_parameters())
```

Arguments

gparams   Optional. [load_parameters()](#) or null

Value

Vector. A set of geographical scales

---

get_parameters   *Get Parameters*

Description

This function retrieves the parameters and copies the parameter file to the specified output path.

Usage

```
get_parameters(out_path = tempdir(), iwindow = FALSE)
```

Arguments

out_path	character. The output path where the parameter file will be copied. The default is a temporary directory `tempdir()`
iwindow	logical. If window = TRUE, this will prompt the user to select the output directory using a file chooser window. The default value is FALSE.

Details

Using configuration file is an alternative to `sean()`

Value

character. The path to the copied parameter file.

See Also

`set_parameters()`

Examples

```
get_parameters()
get_parameters(out = tempdir())
```

---

get_rasters   *Get rasters object from parameters*

Description

Takes named list of hosts as an input. See host object in `get_parameters()` or `load_parameters()`. This is also a wrapper of `crops_rast()`. Function creates 2 raster object - one is a sum of all the crops specified under sources and other using the provided raster file. See `tiff_torast()`

Usage

```
get_rasters(hosts)
```

Arguments

hosts   List of hosts and values. It is synonym to Hosts object in parameters

Value

List of SpatRaster.

See Also

[load_parameters()](), [get_parameters()](), [tiff_torast()](), [cropharvest_rast()]()

Examples

```
# Get default rasters
## Not run:
get_rasters(list(mapspam2010 = c("wheat"), monfreda = c("avocado"), file = "some_raster.tif"))

## End(Not run)
```

GlobalRast-class   *GlobalRast class*

Description

A class to represent raster for global scales. Global scales are accessible using [global_scales()](). However, this class encapsulates the results of apply dispersal models and metrics.

Fields

east A list of raster for eastern hemisphere.

west A list of raster for western hemisphere.

---

global_scales   *Global geographical extent*

Description

This function provides the geographical extent used in a global analysis. This function returns the geographic extents of the eastern and western hemispheres used in the analysis. Each geographic extent is in the form of c(Xmin, Xmax, Ymin, Ymax). The geohabnet functions are designed to work with the coordinate reference system: lon/lat WGS 84 (EPSG:4326).

Usage

```
global_scales()
```

Details

When a habitat connectivity analysis is global, the functions in geohabnet will conduct two separate analyses, one on the geographical scale of the eastern hemisphere and another for the western hemisphere. The final outcomes (such as maps or adjacency matrices) are then combined for the global analysis.

Value

List. Named list with scales for eastern and western hemispheres

See Also

set_global_scales()

Gmap-class   *Gmap class*

Description

An S4 class to represent various maps in the form of SpatRaster.

Usage

```
setmaps(x, me, vari, dif)
```

Arguments

x	Gmap object.
me	SpatRaster. A raster representing mean risk index.
vari	SpatRaster. A raster representing variance.
dif	SpatRaster. A raster representing difference.

Value

Gmap object.

Slots

`me_rast` A raster representing mean risk index.

`me_out` Character. A file path to the mean risk index raster.

`diff_rast` A raster representing difference.

`diff_out` Character. A file path to the difference raster.

`var_rast` A raster representing variance.

`var_out` Character. A file path to the variance raster.

---

gplot *Plot a Raster* object*

Description

This is a wrapper for terra::plot()

Usage

```
gplot(x, ...)
```

Arguments

x	a Raster* object
...	additional arguments passed to terra::plot()

Value

a plot

Examples

```
r <- terra::rast(nrows=108, ncols=21, xmin=0, xmax=10)
gplot(r)
gplot(r, col = "red")
gplot(r, col = "red", breaks = 10)
```

---

hci_diff   *Calculate difference map*

Description

This function produces a map of difference b/w mean and sum indexes in rank of cropland harvested area fraction.

Usage

```
hci_diff(x, y, global, geoscale, res = reso(), outdir = tempdir())
```

Arguments

x	SpatRaster.
y	SpatRaster.
global	Logical. TRUE if global analysis is desired, FALSE otherwise. east and west are required when TRUE.
geoscale	Numeric vector. x will be cropped to this extent.
res	Numeric. Map resolution. This value is used in aggregation and dis-aggregation operation. Default is reso().
outdir	Character. Output directory for saving raster in TIFF format. Default is tempdir().

Details

Ideally, the function is tested to yield desired results when length(which(y[] > 0)) > length(which(x[] > 0)).

Value

RiskMap. Contains result in the form of SpatRaster objects and file path of the saved maps.

---

hci_mean   *Calculate mean of raster objects*

Description

Wrapper for terra::mean(). Calculates mean of list of rasters.

Usage

```
hci_mean(
  indices,
  global = FALSE,
  east = NULL,
  west = NULL,
  geoscale = NULL,
  res = reso(),
  plt = TRUE,
  outdir = tempdir()
)
```

Arguments

indices	List of SpatRasters. This input represents the spatial raster collection for which mean is to be calculated.
global	Logical. TRUE if global analysis is desired, FALSE otherwise. east and west are required when TRUE.
east	SpatRaster. Collection of risk indices on eastern extent.
west	SpatRaster. Collection of risk indices on western extent. When TRUE, geoscale is ignored. Default is TRUE.
geoscale	Vector. geographical scale. Default is NULL.
res	Numeric. Map resolution. This value is used in aggregation and dis-aggregation operation. Default is [reso()]().
plt	TRUE if need to plot mean map, FALSE otherwise.
outdir	Character. Output directory for saving raster in TIFF format. Default is [tempdir()]().

Value

RiskMap. Contains result in the form of SpatRaster objects and file path of the saved maps.

---

hci_variance   *Calculate variance of CCRI*

Description

This function produces a map of variance of CCRI based on input parameters

Usage

```
hci_variance(
  indices,
  rast,
  global,
  east = NULL,
```

```
    west = NULL,
    geoscale,
    res = reso(),
    outdir = tempdir()
)
```

Arguments

`indices`	SpatRaster. Collection of risk indices.
`rast`	SpatRaster. Template for variance output
`global`	Logical. `TRUE` if global analysis is desired, `FALSE` otherwise. `east` and `west` are required when `TRUE`.
`east`	SpatRaster. Collection of risk indices on eastern extent.
`west`	SpatRaster. Collection of risk indices on western extent. When `TRUE`, `geoscale` is ignored. Default is `TRUE`.
`geoscale`	Vector. geographical scale. Default is `NULL`.
`res`	Numeric. Map resolution. This value is used in aggregation and dis-aggregation operation. Default is [`reso()`](reso()).
`outdir`	Character. Output directory for saving raster in TIFF format. Default is [`tempdir()`](tempdir()).

Value

`RiskMap`. Contains result in the form of `SpatRaster` objects and file path of the saved maps.

`host_density<-`	*Set the host density.*

Description

Sets the host density slot in the GeoNetwork object

Usage

```
host_density(x) <- value
```

Arguments

`x`	the GeoNetwork object.
`value`	SpatRaster.

---

inv_powerlaw   *Dispersal kernels*

Description

-[inv_powerlaw()] Get parameters and values pertaining to the inverse power law model. -[neg_exp()] Get parameters and values pertaining to the negative exponential model.

Usage

```
inv_powerlaw(
  params = load_parameters(),
  betas = NULL,
  mets = NULL,
  we = NULL,
  linkcutoff = NULL
)

neg_expo(
  params = load_parameters(),
  gammas = NULL,
  mets = NULL,
  we = NULL,
  linkcutoff = NULL
)
```

Arguments

params	Object. This function uses the parameter values defined in load_parameters() by default. If load_parameters() is not provided, the function will require the user to specify all arguments as listed below. If both load_parameters() and values for the arguments listed below are provided, load_parameters() takes precedence over the specified arguments.
betas	Numeric. Beta is the dispersal parameter used in the inverse power law to estimate a species' dispersal gradient. Please refer to Mundt et al (2009) for details on how to calculate this parameter. Any beta values should be positive. Smaller beta values indicate a higher likelihood of dispersal between nodes.
mets	Character. There are seven network metrics supported by geohabnet: "node_strength", "sum_of_nearest_neighbors", "eigenvector_centrality", "closeness", "betweeness", "degree", and "page_rank". Each specified network metric is calculated for each location in the target region, based on the link weights between each pair of locations. Run, for example, pagerank() for details of each network metric.
we	Numeric. This parameter indicates the weight(s) of each specified network metric, representing the importance of the network metric in the analysis. Since these weights represent percentages, any weight(s) should be between 0 and 100, and the sum of all specified weights should be 100.

linkcutoff   Numeric. This parameter is only used to calculate `betweeness()` and `closeness()`, and is equivalent to `cutoff` in these functions in the `igraph` package.

gammas   Numeric. Gamma is the dispersal parameter used in the negative exponential to estimate a species' dispersal gradient. Any gamma values should be positive. Smaller gamma values indicate a higher likelihood of dispersal between nodes.

Details

Refer to Esker et al (2007) for a discussion on the characteristics of each dispersal gradient or kernel model (i.e., inverse power law and negative exponential). The resulting object produced by `load_parameters()` provides the following values used when running the analysis -beta is a dispersal parameter for calculating the inverse power law model. -gamma is a dispersal parameter for calculating the negative exponential model. -metrics Each network metric is applied to the adjacency matrix produced in the intermediate step. -weights The link weights that is used in the network analysis. -cutoff Currently used as a parameter to calculate centrality in the network - `betweeness()` and `closeness()`. As defined in `igraph::betweenness()`, it's the maximum length to consider when calculating centrality. If zero or negative, then there is no such limit.

Value

List with parameters and values. See details.

References

Esker PD, Sparks AH, Antony G, Bates M, Dall' Acqua W, Frank EE, Huebel L, Segovia V, Garrett KA (2007). "Ecology and Epidemiology in R: Modeling dispersal gradients." *The Plant Health Instructor*. doi:10.1094/PHIA2008012903

Mundt CC, Sackett KE, Wallace LD, Cowger C, Dudley JP (2009). "Aerial Dispersal and Multiple-Scale Spread of Epidemic Disease." *Ecohealth*. doi:10.1007/s103930090251z

Csardi G, Nepusz T (2006). "The igraph software package for complex network research." *Inter-Journal*, *Complex Systems*, 1695. https://igraph.org.

Csárdi G, Nepusz T, Traag V, Horvát Sz, Zanini F, Noom D, Müller K (2024). *igraph: Network Analysis and Visualization in R*. doi:10.5281/zenodo.7682609, R package version 1.5.1, https://CRAN.R-project.org/package=igraph.

See Also

`supported_metrics()`

---

load_parameters   *Load Parameters from YAML File*

Description

This function loads parameters from a YAML file and stores them in an object.

Usage

```
load_parameters(filepath = .param_fp())
```

Arguments

filepath   Path to the YAML file containing the parameters. By default, it takes the value
               of `parameters.yaml` in R user's directory.

Value

object with parameters and values

Examples

```
# Load parameters from default file
load_parameters()
```

---

mapspam   *Supported sources for Mapspam*

Description

Supported sources for Mapspam

Usage

```
mapspam()
```

---

monfreda   *Supported sources for monfreda*

Description

Supported sources for monfreda

Usage

```
monfreda()
```

---

ndplot   *Network density*

Description

Calculates and plots the network density of a GeoNetwork object.

Usage

```
ndplot(x)
```

Arguments

x   A GeoNetwork object

Value

Vector. Up to two ggplot2 objects

---

ndplot,GeoNetwork-method
                                 Network density

Description

Calculates and plots the network density of a GeoNetwork object. The plot is produced as dispersal parameter vs edge density.

Usage

```
## S4 method for signature 'GeoNetwork'
ndplot(x)
```

Arguments

x   A GeoNetwork object

Value

Vector. Up to two ggplot2 objects

Description

These are functions under the igraph package adapted to calculate habitat connectivity. In the context of habitat connectivity, the functions can be interpreted as follows:

- nn_sum(): Calculates the sum of nearest neighbors igraph::knn().
- node_strength(): Calculates the sum of edge weights of adjacent nodes igraph::strength().
- betweenness(): Calculates the node betweenness based on the number of shortest paths. Because the igraph::betweenness() function in interprets link weights as distances to calculate the shortest paths, the betweeness() function in geohabnet transforms the link weights (or the relative likelihood of pathogen or pest movement) in the adjacency matrix so that higher link weight values will be the shortest (or more likely) paths for pathogen or pest movement.
- ev(): Calculates the eigenvector centrality of positions within the network igraph::eigen_centrality().
- closeness(): measures how many steps is required to access every other vertex from a given vertex igraph::closeness(). Because the igraph::closeness() function interprets link weights as distances to calculate the shortest paths, this transforms the link weights (or the relative likelihood of pathogen or pest movement) in the adjacency matrix so that higher link weight values will be the shortest (or more likely) paths for pathogen or pest movement.
- degree(): number of adjacent edges igraph::degree().
- pagerank(): page rank score for vertices igraph::page_rank().

Usage

```
nn_sum(crop_dm, ...)

node_strength(crop_dm, ...)

betweeness(crop_dm, ...)

ev(crop_dm, ...)

degree(crop_dm, ...)

closeness(crop_dm, ...)

pagerank(crop_dm, ...)
```

Arguments

crop_dm   A square adjacency matrix, in which rows and columns names represent nodes (or locations) and each entry indicate the relative likelihood of pathogen or pest movement between a pair of nodes. In the internal workflow, the adjacency matrix comes as a result of operations within sean() function. This weight represents the importance of the network metric in the habitat connectivity analysis.

| ... | arguments to corresponding funtions in `igraph` |

Value

SpatRaster. Representing connectivity of each node or location.

References

Csardi G, Nepusz T (2006). "The igraph software package for complex network research." *Inter-Journal*, *Complex Systems*, 1695. <https://igraph.org>.

See Also

Other metrics: `supported_metrics()`

---

reset_params   *Reset parameters.yaml*

Description

Resets the values in the `parameters.yaml` file to the default initial values.

Usage

```
reset_params()
```

Value

Logical. `TRUE` if function was successfully executed

Examples

```
reset_params()
```

---

reso *Get resolution value*

Description

Resolution stored in `parameter.yaml`. Here, resolution values refer to the aggregation factor or granularity. Granularity is the number of small grid cells that are aggregated into larger grid cells in each direction (horizontally and vertically). For example, the finest spatial resolution of the Monfreda and MAPSPAM datasets in geohabnet is 5 minutes, a granularity value of 6 will result in maps with a spatial resolution of 0.5 degrees. If not provided, the resolution value used for the analysis is by default 24 (or two degrees when using the Monfreda and MAPSPAM datasets). Otherwise, a single integer value for granularity equal to or greater than one should be specified.

Usage

```
reso()
```

Value

Numeric. Resolution from `parameters.yaml`. The default is 12.

RiskMap-class *RiskMap class*

Description

An S4 class representing resulting maps from the specific operation type.

Fields

`map` Character. A file path to the map.

`riid` SpatRaster. This is one of the risk maps.

`spr` SpatRaster. A spatial raster representing the risk index.

`fp` Character. A file path to the risk index raster.

---

risk_indices   *Get risk indices*

Description

Get risk indices from GeoRasters object.

Usage

```
risk_indices(ri)
```

Arguments

ri   GeoRasters object

Details

This function will unpack SpatRasters from GeoModel and thus is `future::future()` safe.

Value

List of risk indices. If the `ri` is global, the list will contain two elements, one for each hemisphere. e.g. `list(east = list(), west = list())`. If the `ri` is not global, the list will contain a single element, e.g. `list()`.

---

sa_onrasters   *Run sensitivity analysis*

Description

Same as `sensitivity_analysis()` but it takes raster object and other parameters as an input.

- `sa_onrasters()` is a wrapper around `sean()` function. Takes raster object and other parameters as an input.

- `msean_onrast()` same as `sa_onrasters()`. Use this for side effects + results. Produces and plots the maps for the outcomes and results are returned as an object. It produces and plots the maps for the outcomes and results are returned as an object.

Usage

```
sa_onrasters(rast, link_thresholds = c(0), hd_thresholds = c(0), ...)

msean_onrast(
  global = TRUE,
  geoscale = NULL,
  res = reso(),
  outdir = tempdir(),
  ...
)
```

Arguments

rast	Raster object which will be used in analysis.
link_thresholds	
	Numeric vector. link threshold values
hd_thresholds	Numeric vector. host density threshold values
...	Additional parameters to be passed to `sean()`.
global	Logical. `TRUE` if global analysis, `FALSE` otherwise. Default is `TRUE`
geoscale	Numeric vector. Geographical coordinates in the form of c(Xmin, Xmax, Ymin, Ymax) which EPSG:4326 in coordinate reference system. If `geoscale` is `NuLL`, the extent is extracted from `rast`(SpatRaster) using `terra::ext()`.
res	Numeric. Resolution of the raster. Default is `reso()`.
outdir	Character. Output directory for saving raster in TIFF format. Default is `tempdir()`.

Details

Error not handled for non-overlapping extents.

Value

A list of calculated CCRI indices after operations. An index is generated for each combination of paramters. One combination is equivalent to `sean()` function.

References

Yanru Xing, John F Hernandez Nopsa, Kelsey F Andersen, Jorge L Andrade-Piedra, Fenton D Beed, Guy Blomme, Mónica Carvajal-Yepes, Danny L Coyne, Wilmer J Cuellar, Gregory A Forbes, Jan F Kreuze, Jürgen Kroschel, P Lava Kumar, James P Legg, Monica Parker, Elmar Schulte-Geldermann, Kalpana Sharma, Karen A Garrett, *Global Cropland connectivity: A Risk Factor for Invasion and Saturation by Emerging Pathogens and Pests*, BioScience, Volume 70, Issue 9, September 2020, Pages 744–758, doi:10.1093/biosci/biaa067

Hijmans R (2023). *terra: Spatial Data Analysis*. R package version 1.7-46, https://CRAN.R-project.org/package=terra

See Also

Use get_rasters() to obtain raster object.

msean_onrast() supported_sources()

Examples

```
rr <- get_rasters(list(monfreda = c("avocado")))
res1 <- sa_onrasters(rr[[1]],
           global = FALSE,
           geoscale = c(-115, -75, 5, 32),
           c(0.0001, 0.00004),
           c(0.0001, 0.00005),
           c("sum", "mean"),
           res = 12)
res2 <- sa_onrasters(rr[[1]],
           global = TRUE,
           link_thresholds = c(0.000001),
           hd_thresholds = c(0.00015),
           agg_methods = c("sum"),
           res = 12)
res3 <- msean_onrast(rast = rr[[1]],
          link_thresholds = c(0.000001),
          hd_thresholds = c(0.00015))
```

---

sean   *Sensitivity analysis across maps of habitat connectivity*

Description

This function performs a sensitivity analysis across different values of habitat connectivity for each location in a map. For each combination of selected parameters, an index of habitat connectivity is calculated. sensitivity_analysis() is a wrapper around sean() function.

- msean() is a wrapper around sean() function. It has additional argument to specify maps which are calculated using connectivity() function. The maps are essentially the risk network.

Usage

```
sean(
  rast,
  global = TRUE,
  geoscale = NULL,
  agg_methods = c("sum", "mean"),
  dist_method = "geodesic",
  link_threshold = 0,
  hd_threshold = 0,
```

```
    res = reso(),
    inv_pl = inv_powerlaw(NULL, betas = c(0.5, 1, 1.5), mets = c("betweeness",
     "NODE_STRENGTH", "Sum_of_nearest_neighbors", "eigenVector_centrAlitY"), we = c(50,
       15, 15, 20), linkcutoff = -1),
    neg_exp = neg_expo(NULL, gammas = c(0.05, 1, 0.2, 0.3), mets = c("betweeness",
     "NODE_STRENGTH", "Sum_of_nearest_neighbors", "eigenVector_centrAlitY"), we = c(50,
       15, 15, 20), linkcutoff = -1)
  )

  msean(
    rast,
    global = TRUE,
    geoscale = NULL,
    res = reso(),
    ...,
    outdir = tempdir()
  )
```

Arguments

rast	Raster object which will be used in analysis.
global	Logical. TRUE if global analysis, FALSE otherwise. Default is TRUE
geoscale	Numeric vector. Geographical coordinates in the form of c(Xmin, Xmax, Ymin, Ymax) which EPSG:4326 in coordinate reference system. If geoscale is NuLL, the extent is extracted from rast(SpatRaster) using [terra::ext()]().
agg_methods	Character. One or both the values - SUM, MEAN. Aggregation strategy for scaling the input raster to the desired resolution.
dist_method	Character. The method to calculate the distance matrix.
link_threshold	Numeric. A threshold value for link weight. All link weights that are below this threshold will be replaced with zero for the connectivity analysis. Link weights represent the relative likelihood of pathogen, pest, or invasive species movement between a pair of host locations, which is calculated using gravity models based on host density (or availability) and dispersal kernels.
hd_threshold	Numeric. A threshold value for host density. All locations with a host density below the selected threshold will be excluded from the connectivity analysis, which focuses the analysis on the most important locations. The values for the host density threshold can range between 0 and 1; if value is 1, all locations will be excluded from the analysis and 0 will include all locations in the analysis. Selecting a threshold for host density requires at least knowing what is the maximum value in the host density map to avoid excluding all locations in the analysis. if value is 1, all locations will be excluded from the analysis and 0 will include all locations in the analysis. Selecting a threshold for host density requires at least knowing what is the maximum value in the host density map to avoid excluding all locations in the analysis.
res	Numeric. Resolution of the raster. Default is [reso()]().
inv_pl	List. A named list of parameters for inverse power law. See details.

neg_exp	List. A named list of parameters for inverse negative exponential. See details. All locations with a host density below the selected threshold will be excluded from the connectivity analysis, which focuses the analysis on the most important locations. The values for the host density threshold can range between 0 and 1;
...	arguments passed to `sean()`
outdir	Character. Output directory for saving raster in TIFF format. Default is `tempdir()`.

Details

When `global = TRUE`, geoscale is ignored and `global_scales()` is used by default.

The functions `sean()` and `msean()` perform the same sensitivity analysis, but they differ in their return value. The return value of `msean()` is GeoNetwork, which contains the result from applying the `connectivity()` function on the habitat connectivity indexes. Essentially, the risk maps.

If neither the inverse power law nor the negative exponential dispersal kernel is specified, the function will return an error.

In `msean()`, three spatRasters are produced with the following values. For each location in the area of interest, the mean in habitat connectivity across selected parameters is calculated. For each location in the area of interest, the variance in habitat connectivity across selected parameters is calculated. For each location in the area of interest, the difference between the rank of habitat connectivity and the rank of host density is calculated. By default, each of these spatRasters is plotted for visualization.

Value

GeoRasters.

GeoNetwork.

References

Yanru Xing, John F Hernandez Nopsa, Kelsey F Andersen, Jorge L Andrade-Piedra, Fenton D Beed, Guy Blomme, Mónica Carvajal-Yepes, Danny L Coyne, Wilmer J Cuellar, Gregory A Forbes, Jan F Kreuze, Jürgen Kroschel, P Lava Kumar, James P Legg, Monica Parker, Elmar Schulte-Geldermann, Kalpana Sharma, Karen A Garrett, *Global Cropland connectivity: A Risk Factor for Invasion and Saturation by Emerging Pathogens and Pests*, BioScience, Volume 70, Issue 9, September 2020, Pages 744–758, doi:10.1093/biosci/biaa067

Hijmans R (2023). *terra: Spatial Data Analysis*. R package version 1.7-46, https://CRAN.R-project.org/package=terra

See Also

Uses `connectivity()`

Uses `msean()` `inv_powerlaw()` `neg_expo()`

Examples

```
avocado <- cropharvest_rast("avocado", "monfreda")

# global
ri <- sean(avocado) # returns a list of GeoRasters
mri <- msean(rast = avocado) # returns GeoNetwork object

# non-global
# geoscale is a vector of xmin, xmax, ymin, ymax

# returns GeoRasters object
ri <- sean(avocado, global = FALSE, geoscale = c(-115, -75, 5, 32))
ri

# returns GeoNetwork object
mri <- msean(rast = avocado, global = FALSE, geoscale = c(-115, -75, 5, 32))
mri
```

---

search_crop   *Search for crop*

Description

It returns the dataset sources in which crop data is available. Essentially, a wrapper around `geodata::spamCrops()` and `geodata::monfredaCrops()`

Usage

```
search_crop(name)
```

Arguments

name   name of crop

Value

Logical. Sources iin crop data is available.

See Also

`supported_sources()`

Examples

```
search_crop("coffee")
search_crop("wheat")

search_crop("jackfruit")
```

sensitivity_analysis   *Sensitivity analysis for habitat connectivity*

Description

This function runs a sensitivity analysis on habitat connectivity calculated based on every combination of selected parameters. Parameter values in `sensitivity_analysis()` should be provided using the function `set_parameters()`. If no parameters are provided, then the `sensitivity_analysis()` function will run the sensitivity analysis using a default set of parameter values, which is accessible through the function `get_parameters()`. To customize parameter values, open the parameters.yaml that was automatically downloaded when geohabnet was installed, change, remove, or add parameter values directly in the parameters.yaml and save it. Once the values have been changed manually, run `set_parameters()` to set the new parameter values, which will return TRUE if the parameters were set successfully.

Usage

```
sensitivity_analysis(maps = TRUE, alert = TRUE)
```

Arguments

maps	logical. `TRUE` if maps of outcomes are to be plotted, `FALSE` otherwise. If `TRUE`, three maps are possible: a map of mean habitat connectivity, a map of variance of habitat connectivity, and a map of the difference between the ranks in habitat connectivity and habitat density.
alert	logical. `TRUE` if a beep sound is to be played once the analysis is completed, `FALSE` otherwise

Details

For each location in a region, sensitivity_analysis() calculates the habitat connectivity risk index (CCRI) proposed by Xing et al. (2021). By default, sensitivity_analysis() runs a sensitivity analysis on a global extent, see `global_scales()` for details. This function also plots maps of the outcomes automatically, but it will suppress maps for outcomes if maps = FALSE or `interactive()` is FALSE. The returned object is of class `GeoNetwork`, which contains two types of outcomes. One outcome type corresponds to spatRasters representing the maps of habitat connectivity. The second type corresponds to adjacency matrices used to calculate the habitat connectivity, where columns and rows represent locations in the maps and entries are the relative likelihood of pathogen or pest movement between each pair of nodes.

Value

GeoNetwork. Errors are not handled.

References

Yanru Xing, John F Hernandez Nopsa, Kelsey F Andersen, Jorge L Andrade-Piedra, Fenton D Beed, Guy Blomme, Mónica Carvajal-Yepes, Danny L Coyne, Wilmer J Cuellar, Gregory A Forbes, Jan F Kreuze, Jürgen Kroschel, P Lava Kumar, James P Legg, Monica Parker, Elmar Schulte-Geldermann, Kalpana Sharma, Karen A Garrett, *Global Cropland connectivity: A Risk Factor for Invasion and Saturation by Emerging Pathogens and Pests*, BioScience, Volume 70, Issue 9, September 2020, Pages 744–758, doi:10.1093/biosci/biaa067

Hijmans R (2023). *terra: Spatial Data Analysis*. R package version 1.7-46, `https://CRAN.R-project.org/package=terra`

See Also

`sa_onrasters()` `sean()` `global_scales()` `get_parameters()` `set_parameters()` `connectivity()`

Examples

```
# Run analysis on specified parameters.yaml
ss1 <- sensitivity_analysis()
ss2 <- sensitivity_analysis(FALSE, FALSE)
ss3 <- sensitivity_analysis(TRUE, FALSE)
```

---

setmaps,Gmap-method   *Sets the map slots in the Gmap object.*

Description

This wraps the results(SpatRasters) from the risk analysis.

Usage

```
## S4 method for signature 'Gmap'
setmaps(x, me, vari, dif)
```

Arguments

x	A Gmap object.
me	A GeoRaster object representing mean risk index.
vari	A GeoRaster object representing variance.
dif	A GeoRaster object representing difference.

Value

A Gmap object.

---

setprops   *Set properties of the GeoModel object.*

Description

Set properties of the GeoModel object.

Usage

```
setprops(x, aggregation, hdthreshold, linkthreshold)
```

Arguments

x	The GeoModel object.
aggregation	Character. A value representing the aggregation method.
hdthreshold	Numeric. A value representing the host density threshold.
linkthreshold	Numeric. A value representing the link threshold in a network.

Value

The GeoModel object with updated properties.

---

setprops,GeoModel,character,numeric,numeric-method
                          Set properties of the GeoModel object.

Description

Set properties of the GeoModel object.

Usage

```
## S4 method for signature 'GeoModel,character,numeric,numeric'
setprops(x, aggregation, hdthreshold, linkthreshold)
```

Arguments

x	The GeoModel object.
aggregation	Character. A value representing the aggregation method.
hdthreshold	Numeric. A value representing the host density threshold.
linkthreshold	Numeric. A value representing the link threshold in a network.

Value

The GeoModel object with updated properties.

---

set_global_scales   *Set global geographical extent*

Description

This function sets the geographical extents used in global analysis. See also [geoscale_param()](#) to set the geographic extent of an analysis that is not global. Each geographic extent should be in the form of c(Xmin, Xmax, Ymin, Ymax). Geographic extent must be specified by four values in degrees that represent the geographic limits of the area for analysis, following the order: minimum longitude, maximum longitude, minimum latitude, and maximum latitude. Degrees are in decimal notation and have a negative sign for the southern and western hemispheres.

Usage

```
set_global_scales(value)
```

Arguments

value   list. Named list of eastern and western hemisphere extents. See usage.

Value

List. Named list with scales for eastern and western hemispheres

See Also

[global_scales()](#) [terra::ext()](#)

Examples

```
set_global_scales(list(east = c(-24, 180, -58, 60), west = c(-140, -34, -58, 60)))
```

---

set_parameters   *Set Parameters*

Description

This function allows the user to set the parameters by replacing the existing parameters file with a new one. Use [get_parameters()](#) to modify the parameter values.

Usage

```
set_parameters(new_params, iwindow = FALSE)
```

Arguments

new_params	The path to the new parameters file.
iwindow	Logical, indicating whether to prompt the user to select the new parameters file using a file selection window. The default value of this parameter is set to FALSE.

Value

None

Examples

```
param_fp <- get_parameters()
set_parameters(param_fp)
```

---

sp_rast   *Raster for mapspam crop.*

Description

get raster for crop in mapspam dataset

Usage

```
sp_rast(crp, africa = FALSE)
```

Arguments

crp	character. name of a crop. Case-insensitive.
africa	Fetch the 2017 Mpaspam crop data of Africa. Default is FALSE.

Details

See [geodata::spamCrops()](#) for supported crops.

Value

SpatRaster

References

International Food Policy Research Institute, 2020. Spatially-Disaggregated Crop Production Statistics Data in Africa South of the Sahara for 2017. <doi: 10.7910/DVN/FSSKBW>, Harvard Dataverse, V2

See Also

geodata::spamCrops() search_crop()

Examples

```
sp_rast("potato")
sp_rast("potato", TRUE)
```

---

supported_metrics   *Returns metrics currently supported in the analysis.*

Description

Returns metrics currently supported in the analysis.

Usage

```
supported_metrics()
```

Value

vector of supported metrics.

See Also

Other metrics: [nn_sum()](nn_sum())

Examples

```
supported_metrics()
```

---

supported_sources   *Get supported sources of crops*

Description

When provided, cropharvest_rast() will look for cropland data in this specific source.

Usage

```
supported_sources()
```

Value

Vector of supported sources. Also used as a lookup to find get raster object.

Examples

```
# Get currently supported sources
supported_sources()
```

`tiff_torast`   *Get raster object from tif file*

Description

This is a wrapper of terra::rast() and generates a raster object if provided with a TIF file.

Usage

```
tiff_torast(path_to_tif)
```

Arguments

`path_to_tif`   TIFF file. This is an encoding of map in raster format.

Value

SpatRaster.

Examples

```
# Generate raster for usage
fp <- paste(tempfile(), ".tif", sep = "")
ret <- utils::download.file(
"https://geohabnet.s3.us-east-2.amazonaws.com/util-rasters/avocado_HarvestedAreaFraction.tif",
destfile = fp, method = "auto", mode = "wb")
tiff_torast(fp)
```

Index