

# Package ‘geohabnet’

May 26, 2026

**Title** Geographical Risk Analysis Based on Habitat Connectivity

**Version** 2.3

**Date** 2026-05-25

## 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](https://doi.org/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](https://doi.org/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](https://doi.org/10.7910/DVN/PRFF8V)> ``Global Spatially-Disaggregated Crop Production Statistics Data for 2010 Version 2.0, Harvard Data-verse, 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 functions are `msean()` and `sensitivity analysis()`.

Most functions in 'geohabnet' provide 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** 7.3.2

**Imports** config (>= 0.3.1), geosphere (>= 1.5.18), igraph (>= 2.0.3), terra (>= 1.7.29), yaml (>= 2.3.7), stats, stringr (>= 1.5.0), memoise (>= 2.0.1), graphics, 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, 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>

**NeedsCompilation** no

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

**Maintainer** Krishna Keshav <[krishnakeshav.pes@gmail.com](mailto:krishnakeshav.pes@gmail.com)>

**Repository** CRAN

**Date/Publication** 2026-05-26 02:10:03 UTC

## Contents

.indices	3
connectivity	4
dist_methods	6
GeoModel-class	6
GeoNetwork-class	7
GeoRasters-class	8
geoscale_param	8
get_parameters	9
get_rasters	10
GlobalRast-class	10
global_scales	11
Gmap-class	11
gplot	12
habitat_density<-	13
hci_diff	13
hci_mean	14
hci_variance	15
inv_powerlaw	16
load_parameters	18
ndplot	19
ndplot,GeoNetwork-method	19
nn_sum	20
reset_params	21
reso	22
RiskMap-class	22
risk_indices	22
sa_onrasters	23
sean	25
sensitivity_analysis	28
setmaps,Gmap-method	29
setprops	30
setprops,GeoModel,character,numeric,numeric-method	31
set_global_scales	31
set_parameters	32
supported_metrics	33
<b>Index</b>	<b>34</b>

---

.indices

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

---

### Description

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

**Usage**

```
.indices(crop_rasters)
```

**Arguments**

crop\_rasters List of SpatRaster objects.

**Value**

A list of risk indices.

---

connectivity	<i>Calculate and plot maps of habitat connectivity</i>
--------------	--

---

**Description**

Calculate the mean habitat connectivity across a set of selected parameters, variance in habitat connectivity, and the difference in ranks between mean habitat connectivity and habitat availability. The result is produced in form of maps plotted with predefined graphics 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. A SpatRaster object for the spatial distribution of habitat availability (such as host availability or cropland density). Note that a valid input data for geohabnet is a raster layer of habitat availability (such as host availability), in which each grid cell has any values between zero and one. If you are using your own dataset, please also make sure that your raster layer is in the standard coordinate reference system (i.e., EPSG:4326), the only CRS supported by geohabnet for now.

indices SpatRaster. Collection of risk indices.

global	Logical. Select TRUE if a 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. This refers to the geographical extent for the habitat connectivity analysis when global is set to FALSE. Default is NULL.
res	Numeric. This parameter refers to the spatial aggregation factor. This value is the number of cells that are grouped when aggregating a raster layer from fine to coarse spatial resolution to reduce computational costs. Setting this parameter to 1 would not aggregate the raster layers. Default is <code>reso()</code> .
pmean	Logical. TRUE if a map of mean habitat connectivity should be plotted, FALSE otherwise.
pvar	Logical. TRUE if a map of the variance in habitat connectivity should be plotted, FALSE otherwise.
pdiff	Logical. TRUE if a map of the difference in the ranks between the mean habitat connectivity and habitat availability should be plotted, FALSE otherwise.
outdir	Character. Output directory for saving raster in TIFF format. Default is <code>tempdir()</code> .

### Details

`indexes` has a list of `spatRaster` objects resulting from the unique combinations of all parameters specified in either `parameters.yaml` or `sean()`. For each unique combination of parameters, an index of habitat connectivity is calculated for each location in a landscape. Then these indices are used to calculate the mean and the variance of habitat connectivity of a location across all specified parameters.

This function 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., plot panel in Rstudio). The maps are plotted using `SpatRaster` objects. These `SpatRaster` objects are also 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\(\)](#)

---

dist\_methods

*Distance methods supported*

---

**Description**

Contains supported strategies to calculate distance between two points. Use of one of two methods in [sean\(\)](#) or [sensitivity\\_analysis\(\)](#).

**Usage**

dist\_methods()

**Value**

vector

**Examples**

dist\_methods()

---

GeoModel-class

*GeoModel class*

---

**Description**

A reference class to represent results of dispersal models.

**Slots**

**amatrix** A square adjacency matrix that represents the likely movement of a species between locations. In this adjacency matrix, rows and columns are the identification of the locations, and each entry indicates the relative likelihood of a species moving between a pair of locations. An adjacency matrix is produced for each unique value of the dispersal parameters chosen.

**index** A raster object representing the habitat connectivity index of the locations in the selected region. Note that connectivity is calculated based on a weighted sum of the network metrics chosen by the user. A raster object is produced for each unique value of the dispersal parameters chosen.

**hdthreshold** A numeric value representing the threshold for habitat availability (e.g., cropland density or host density) used in the sensitivity analysis.

**aggregation** A character value representing the spatial aggregation method used for aggregating the habitat availability map before conducting the sensitivity analysis.

**linkthreshold** A numeric value representing the threshold for the link weights used to calculate habitat connectivity of each location. Note that link weights indicate the relative likelihood of a species moving between locations (nodes) and correspond to the entries in the adjacency matrix.

**beta** A numeric value representing the beta parameter. The beta parameter is the dispersal parameter in one of two dispersal kernel models (the inverse power law model) included in `geohabnet`.

**gamma** A numeric value representing the gamma parameter. The gamma parameter is the dispersal parameter in one of two dispersal kernel models (the negative exponential model) included in `geohabnet`.

---

GeoNetwork-class      *GeoNetwork*

---

## Description

An S4 class object with the multiple components resulting from a geographical habitat network analysis. The `GeoNetwork` object will wrap all the results from the habitat connectivity analysis using `sean()` or `sensitivity_analysis()`. Specifically, this class contains the field from `Gmap` class, which has results of the habitat connectivity analysis in the form of `SpatRaster` and `TIFF` file.

## Usage

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

## Arguments

<code>x</code>	<code>GeoNetwork</code> .
<code>value</code>	<code>SpatRaster</code> .

## Value

`GeoNetwork`.

## Slots

**rasters** A list of `GeoRasters` objects.

**habitat\_density** A `SpatRaster` representing the habitat availability (or simply host density) that was used as input in the habitat connectivity analysis.

**me\_rast** A `SpatRaster` representing the mean habitat connectivity of a region. The mean is calculated based on a sensitivity analysis across the user-specified dispersal parameters.

**me\_out** Character. A file path to where the mean habitat connectivity raster is saved.

**diff\_rast** A `SpatRaster` representing the difference in ranks between the mean habitat connectivity and the user-supplied map of habitat availability.

diff\_out Character. A file path to where the difference raster is saved.

var\_rast A `SpatRaster` representing the variance in habitat connectivity in a region. The variance is calculated based on a sensitivity analysis across the user-specified dispersal parameters.

var\_out Character. A file path to where the variance raster is saved.

---

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

habitat\_density `SpatRaster`. A spatial raster representing habitat 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 geographic coordinates that delimit the extent of a region of the world.

---

get\_parameters      *Get Parameters*

---

### Description

This function retrieves the list of parameters and saves a copy of the parameter file (of type .yaml) 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 <code>tempdir()</code>
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 this configuration where the parameters are structurally listed in a yaml file is an alternative method used in the `sensitivity_analysis()` function. Once the parameter.yaml is saved in a local directory, the user can modify each parameter value, save this file with the changes, and get the new parameters back in R with `set_parameters()`.

Note that the `sean()` or `msean()` function will require to directly list the parameters within the function as it is typical in other R packages.

### Value

character. The path to the copied parameter file.

### See Also

`set_parameters()`

### Examples

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

---

get_rasters	<i>Get rasters object from parameters See host object in <a href="#">get_parameters()</a> or <a href="#">load_parameters()</a>.</i>
-------------	---

---

**Description**

Get rasters object from parameters See host object in [get\\_parameters\(\)](#) or [load\\_parameters\(\)](#).

**Usage**

```
get_rasters(host)
```

**Arguments**

host	SpatRaster. It is synonym to Hosts object in parameters. This is a wrapper to <a href="#">terra::rast()</a> and generates a raster object if provided with a TIF file.
------	--

**Value**

List of SpatRaster.

**See Also**

[load\\_parameters\(\)](#), [get\\_parameters\(\)](#)

**Examples**

```
# Get default raster
get_rasters(terra::rast())
```

---

GlobalRast-class	<i>GlobalRast class</i>
------------------	-------------------------

---

**Description**

A class to represent raster objects 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 objects for eastern hemisphere.

west A list of raster objects for western hemisphere.

---

global_scales	<i>Global geographical extent</i>
---------------	-----------------------------------

---

### 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	<i>Gmap class</i>
------------	-------------------

---

### Description

An S4 class object to organize various maps in the form of `SpatRaster` in a single object.

### Usage

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

**Arguments**

x	Gmap. Please provide a Gmap object.
me	SpatRaster. A SpatRaster used as background when plotting the map of mean habitat connectivity.
vari	SpatRaster. A Spatraster used as background when plotting the map of variance in habitat connectivity.
dif	SpatRaster. A raster used as background when plotting the map of difference in habitat connectivity and habitat availability.

**Value**

Gmap object.

**Slots**

me_rast	A SpatRaster object representing habitat connectivity of a region averaged across all selected parameters.
me_out	Character. A file path to where the mean habitat connectivity is saved.
diff_rast	A SpatRaster object representing the difference in ranks between mean habitat connectivity and habitat availability in a region.
diff_out	Character. A file path to where the difference raster is saved.
var_rast	A SpatRaster object representing the variance in habitat connectivity of a region calculated across all specified parameters.
var_out	Character. A file path to where the variance raster is saved.

---

gplot

*Plot a Raster\* object*

---

**Description**

This is a wrapper for `terra::plot()` with customized parameters for an enhanced visualization.

**Usage**

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

**Arguments**

x	a Raster* object
...	additional arguments passed to <code>terra::plot()</code>

**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)
```

---

habitat\_density<-      *Set the habitat density.*

---

**Description**

This function helps to set a SpatRaster of the habitat availability or density in a GeoNetwork object. The function is an S4 replacement method in the geohabnet package. It allows you to assign a host availability SpatRaster to a geohabnet object.

**Usage**

```
habitat_density(x) <- value
```

**Arguments**

x	the GeoNetwork object.
value	SpatRaster.

**Value**

The same object type as x, that is, GeoNetwork. This function returns the updated S4 GeoNetwork object with the new habitat availability SpatRaster assigned.

---

hci\_diff      *Calculate difference map*

---

**Description**

This function produces a map of the difference in ranks between mean habitat connectivity and habitat availability.

**Usage**

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

**Arguments**

x	SpatRaster.
y	SpatRaster.
global	Logical. Select TRUE if a 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. This parameter refers to the spatial aggregation factor. This value is the number of cells that are grouped when aggregating a raster layer from fine to coarse spatial resolution to reduce computational costs. Setting this parameter to 1 would not aggregate the raster layers. Default is <code>reso()</code> .
outdir	Character. Output directory for saving raster in TIFF format. Default is <code>tempdir()</code> .
plt	TRUE if need to plot mean map, FALSE otherwise.

**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 habitat connectivity in a region*


---

**Description**

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

**Usage**

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

**Arguments**

indices	List of SpatRasters indicating the habitat connectivity of a region. This input represents the spatial raster collection for which mean is to be calculated.
global	Logical. Select TRUE if a 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. This refers to the geographical extent for the habitat connectivity analysis when global is set to FALSE. Default is NULL.
res	Numeric. This parameter refers to the spatial aggregation factor. This value is the number of cells that are grouped when aggregating a raster layer from fine to coarse spatial resolution to reduce computational costs. Setting this parameter to 1 would not aggregate the raster layers. Default is <code>reso()</code> .
outdir	Character. Output directory for saving raster in TIFF format. Default is <code>tempdir()</code> .
plt	TRUE if need to plot mean map, FALSE otherwise.

**Value**

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

---

hci_variance	<i>Calculate variance of habitat connectivity in a region</i>
--------------	---

---

**Description**

This function produces a map of variance of habitat connectivity across selected parameter values

**Usage**

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

**Arguments**

indices	SpatRaster. Collection of risk indices.
rast	SpatRaster. Template for variance output
global	Logical. Select TRUE if a 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. This refers to the geographical extent for the habitat connectivity analysis when global is set to FALSE. Default is NULL.
res	Numeric. This parameter refers to the spatial aggregation factor. This value is the number of cells that are grouped when aggregating a raster layer from fine to coarse spatial resolution to reduce computational costs. Setting this parameter to 1 would not aggregate the raster layers. Default is <code>reso()</code> .
outdir	Character. Output directory for saving raster in TIFF format. Default is <code>tempdir()</code> .
plt	TRUE if need to plot mean map, FALSE otherwise.

**Value**

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

---

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 <code>load_parameters()</code> by default. If <code>load_parameters()</code> is not provided, the function will require the user to specify all arguments as listed below. If both <code>load_parameters()</code> and values for the arguments listed below are provided, <code>load_parameters()</code> 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", "betweenness", "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, <code>pagerank()</code> 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 <code>betweenness()</code> and <code>closeness()</code> , and is equivalent to <code>cutoff</code> in these functions in the <code>igraph</code> 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 two types of node centrality - `betweenness()` and `closeness()`. As defined in `igraph::betweenness()`, `cutoff` refers to 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()
```

---

ndplot	<i>Network density plot</i>
--------	-----------------------------

---

**Description**

This function first calculates the network density for each dispersal parameter specified by the user. Network density compares the number of available links in a network versus the total number of possible links in the same network. Network density is a measure of how well an entire network is, ranging from 0 (not connected at all) to 1 (fully connected). This function then compares visually how network density changes with changes in dispersal parameter values. In other words, it 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, one for the dispersal parameter values in the negative exponential model and one for the dispersal parameter values in the inverse power law model.

---

ndplot, GeoNetwork-method	<i>Network density</i>
---------------------------	------------------------

---

**Description**

This function first calculates the network density for each dispersal parameter specified by the user. Network density compares the number of available links in a network versus the total number of possible links in the same network. Network density is a measure of how well an entire network is, ranging from 0 (not connected at all) to 1 (fully connected). Calculates and plots the network density of a GeoNetwork object.

**Usage**

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

**Arguments**

x	A GeoNetwork object
---	---------------------

**Value**

Vector. Up to two ggplot2 objects

---

nn\_sum

*Calculation on network metrics a.k.a centralities.*

---

**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 interprets link weights as distances to calculate the shortest paths, the `betweenness()` 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, ...)
```

```
betweenness(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, pest, or other species 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	<i>Get resolution value</i>
------	-----------------------------

---

**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 dataset 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 12 (or two degrees when using the Monfreda and MAPSPAM dataset). 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	<i>RiskMap class</i>
---------------	----------------------

---

**Description**

An S4 class representing resulting maps from specific operation types. Handles automated defaults for missing or NULL spatial data.

---

risk_indices	<i>Get risk indices</i>
--------------	-------------------------

---

**Description**

Get a habitat connectivity index for each unique combination of parameters 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 habitat connectivity 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	<i>Run sensitivity analysis</i>
--------------	---------------------------------

---

**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** This is a SpatRaster object that indicates habitat suitability across locations in a landscape. Entry values in this object can include any positive real numbers, but they typically range from 0 to 1. Currently, this argument supports only raster layers in the standard global coordinate reference system -CRS-, that is, WGS 84 or World Geodetic System 1984, identified by the EPSG code 4326. Users need to project the raster layer before providing it as an input in `msean()` if it is in another CRS.

**link\_thresholds** Numeric vector. Threshold values for link weight. Based on the information on the habitat layer and dispersal kernels, adjacency matrices are created, where

entries represent the potential of an organism's movement between habitat locations. Then, the adjacency matrices are converted into graph objects to perform a network analysis, where the entries in the adjacency matrices are now the weights of the links of the network. This parameter supports any positive values, but make sure these values are smaller than the maximum link weight in the network. If `link_threshold = 0`, all links in the network will be considered in the connectivity analysis. Choosing link weight thresholds greater than zero helps to focus the analysis on the most likely dispersal connections in the landscape. All link weights that are below this threshold will be replaced with zeros for the connectivity analysis.

<code>hd_thresholds</code>	Numeric vector. host density threshold values
<code>...</code>	Additional parameters to be passed to <code>sean()</code> .
<code>global</code>	Logical. TRUE if global analysis, FALSE otherwise. Default is TRUE
<code>geoscale</code>	Numeric vector. This is a set of geographic coordinates in the form of <code>c(Xmin, Xmax, Ymin, Ymax)</code> which correspond to the positions in the standard geographic coordinate system or EPSG:4326. <code>geoscale</code> refers to the geographic extent, or bounding box, for the habitat connectivity analysis. Habitat locations outside the geographic extent will be excluded from the analysis. If <code>geoscale</code> is <code>NULL</code> , the extent is extracted from <code>rast(SpatRaster)</code> using <code>terra::ext()</code> .
<code>res</code>	Numeric. The spatial aggregation factor that will be used to aggregate the raster layer of habitat availability, from fine to coarse resolution. Default is <code>reso()</code> .
<code>outdir</code>	Character. Output directory for saving raster in TIFF format. Default is <code>tempdir()</code> .

### 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 parameters. 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 Schultegeldermann, 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](https://doi.org/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()`

**Description**

This function performs a sensitivity analysis of habitat connectivity across dispersal, geographic, and habitat parameters. For each combination of selected parameters, an index of habitat connectivity is calculated. `msean()` is a wrapper function around `sean()` function. `sean()` is the base connectivity analysis function, while `msean()` (mapped sean) is a wrappers that produces visual maps. `sean()` returns a `GeoRasters` object, whereas `msean()` produces a `GeoNetwork` object containing maps for mean connectivity, variance, and rank differences. Users can use `sean()` for raw numerical connectivity analysis data, and `msean()` to generate the three standard output maps: mean, variance, difference. It has additional argument to specify maps which are calculated using `connectivity()` function.

**Usage**

```
sean(
  rast,
  global = TRUE,
  geoscale = NULL,
  agg_methods = c("sum", "mean"),
  dist_method = "geodesic",
  link_threshold = 0,
  hd_threshold = 0,
  res = reso(),
  inv_p1 = inv_powerlaw(NULL, betas = c(0.5, 1, 1.5), mets = c("betweenness",
    "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("betweenness",
    "NODE_STRENGTH", "Sum_of_nearest_neighbors", "eigenVector_centralitY"), we = c(50,
    15, 15, 20), linkcutoff = -1)
)

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

**Arguments**

`rast` This is a `SpatRaster` object that indicates habitat suitability across locations in a landscape. Entry values in this object can include any positive real numbers, but

they typically range from 0 to 1. Currently, this argument supports only raster layers in the standard global coordinate reference system -CRS-, that is, WGS 84 or World Geodetic System 1984, identified by the EPSG code 4326. Users need to project the raster layer before providing it as an input in `msean()` if it is in another CRS.

<code>global</code>	Logical. TRUE if global analysis, FALSE otherwise. Default is TRUE
<code>geoscale</code>	Numeric vector. This is a set of geographic coordinates in the form of <code>c(Xmin, Xmax, Ymin, Ymax)</code> which correspond to the positions in the standard geographic coordinate system or EPSG:4326. <code>geoscale</code> refers to the geographic extent, or bounding box, for the habitat connectivity analysis. Habitat locations outside the geographic extent will be excluded from the analysis. If <code>geoscale</code> is <code>NULL</code> , the extent is extracted from <code>rast(SpatRaster)</code> using <code>terra::ext()</code> .
<code>agg_methods</code>	Character. One or both methods of spatial aggregation - SUM, MEAN. This is an aggregation strategy for upscaling the input raster to the desired spatial resolution. If <code>agg_methods = c("sum")</code> , then the sum of the habitat suitability of a set of small grid cells is divided by the total number of small cells within the resulting larger grid. If <code>agg_methods = c("mean")</code> , then the sum of the habitat suitability of a set of small grid cells is divided by the number of small cells containing only land within the large grid. In this strategy, small cells with water are excluded from spatial aggregation.
<code>dist_method</code>	Character. The method to calculate the distance matrix. For each pair of habitat locations in the <code>SpatRaster</code> object, <code>msean()</code> will calculate the geographic distances and use them to calculate the probability of an organism's movement between locations based on dispersal kernels. There are two options to calculate the distance between locations: "vincentyEllipsoid", "geodesic". Vincenty ellipsoid distance is highly accurate, accounting for the curvature of Earth, but more computationally expensive. Geodesic distance is less computationally expensive but may be less accurate for large distances than the option above.
<code>link_threshold</code>	Numeric. A threshold value for link weight. Based on the information on the habitat layer and dispersal kernels, adjacency matrices are created, where entries represent the potential of an organism's movement between habitat locations. Then, the adjacency matrices are converted into graph objects to perform a network analysis, where the entries in the adjacency matrices are now the weights of the links of the network. This parameter supports any positive values, but make sure these values are smaller than the maximum link weight in the network. If <code>link_threshold = 0</code> , all link weights in the network will be considered in the connectivity analysis. Choosing link weight thresholds greater than zero helps to focus the analysis on the most likely dispersal connections in the landscape. All link weights that are below this threshold will be replaced with zeros for the connectivity analysis.
<code>hd_threshold</code>	Numeric. A threshold value for habitat suitability (e.g., habitat density or climate suitability). All locations with a habitat suitability below the selected threshold will be excluded from the connectivity analysis, which focuses the analysis on the most important locations. For example, the values for the habitat suitability range between 0 and 1; if the threshold is 1, all locations will be excluded from the analysis and 0 will include all locations in the analysis.

Selecting a threshold for habitat suitability requires at least knowing the maximum value in the habitat suitability map to avoid excluding all locations from the analysis. Note that if the layer of habitat suitability has entry values above 1, `hd_threshold` can also be adjusted accordingly.

<code>res</code>	Numeric. The spatial aggregation factor that will be used to aggregate the raster layer of habitat availability, from fine to coarse resolution. Default is <code>reso()</code> .
<code>inv_pl</code>	List. A named list of parameters for inverse power law model. In <code>geohabnet</code> , two dispersal kernel models are used to calculate the probability of an organism's movement between habitat locations. In this model, the dispersal probability distribution is fat-tailed. Thus, very long-distance dispersal events are assigned a higher probability compared to the negative exponential model. Please use the <code>inv_powerlaw()</code> function as input for the <code>inv_pl</code> argument, customizing parameters as exemplified above. Refer to Xing et al 2020 for the specific mathematical formulation of this dispersal kernel. Users need to specify at least one value for each parameter in <code>inv_powerlaw()</code> . <code>msean()</code> works with either or both dispersal kernel models, <code>inv_pl</code> and/or <code>neg_exp</code> .
<code>neg_exp</code>	List. A named list of parameters for negative exponential model. See details. This is another dispersal kernel model commonly used in landscape and movement ecology to calculate the probability of an organism's movement between habitat locations. In this model, the dispersal probability distribution tail is exponentially bounded. Thus, long-distance dispersal events will be assigned a very low probability of occurrence. Please use the <code>neg_expo()</code> function as input for the <code>neg_exp</code> argument, customizing parameters as exemplified above. Refer to Xing et al 2020 for the specific mathematical formulation of this dispersal kernel. Users need to specify at least one value for each parameter in <code>inv_powerlaw()</code> . <code>msean()</code> works with either or both dispersal kernel models, <code>inv_pl</code> and/or <code>neg_exp</code> .
<code>maps</code>	Logical. True, if plots should be included in side effects. False, otherwise.
<code>...</code>	arguments passed to <code>sean()</code>
<code>outdir</code>	Character. Output directory for saving raster in TIFF format. Default is <code>tempdir()</code> .

## 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\(\)](#)

---

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). If you are providing a map of habitat availability (as opposed to simply cropland density or host availability), you could call the output of your sensitivity analysis as the habitat connectivity index, which is a broader term than CCRI. :) 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`. Check documentation of the `sean()` function for better explanation of the parameters used.

## 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\(\)](#)

---

`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 habitat connectivity.
vari	A GeoRaster object representing variance in habitat connectivity.
dif	A GeoRaster object representing the difference in ranks between habitat connectivity and habitat availability.

**Value**

A Gmap object.

---

setprops	<i>Set properties of the GeoModel object.</i>
----------	---

---

**Description**

Set properties of the GeoModel object.

**Usage**

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

**Arguments**

x	The GeoModel object.
aggregation	Character. A value representing the spatial aggregation method used for aggregating the habitat availability map before conducting the sensitivity analysis. There are two aggregation methods available in geohabnet: sum and/or mean, either excludes NaNs during aggregation.
hdthreshold	Numeric. A numeric value representing the threshold for habitat availability (e.g., cropland density or host density) used in the sensitivity analysis.
linkthreshold	Numeric. A numeric value representing the threshold for the link weights used to calculate habitat connectivity of each location. Note that link weights indicate the relative likelihood of a species moving between locations (nodes) and correspond to the entries in the adjacency matrix.

**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 spatial aggregation method used for aggregating the habitat availability map before conducting the sensitivity analysis. There are two aggregation methods available in geohabnet: sum and/or mean, either excludes NaNs during aggregation.
hdthreshold	Numeric. A numeric value representing the threshold for habitat availability (e.g., cropland density or host density) used in the sensitivity analysis.
linkthreshold	Numeric. A numeric value representing the threshold for the link weights used to calculate habitat connectivity of each location. Note that link weights indicate the relative likelihood of a species moving between locations (nodes) and correspond to the entries in the adjacency matrix.

**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. After running this function, users can use the modified parameters in [sensitivity\\_analysis\(\)](#).

**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)
```

---

`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\(\)](#)

**Examples**

`supported_metrics()`

# Index

- \* **metrics**
  - nn\_sum, 20
  - supported\_metrics, 33
- .grast\_ro (GlobalRast-class), 10
- .indices, 3
- .model\_ob (GeoModel-class), 6
- .rast\_ro (GeoRasters-class), 8
  
- betweenness (nn\_sum), 20
- betweenness(), 17, 20
  
- closeness (nn\_sum), 20
- closeness(), 17
- connectivity, 4
- connectivity(), 25, 27–29
  
- degree (nn\_sum), 20
- dist\_methods, 6
  
- ev (nn\_sum), 20
  
- future::future(), 23
  
- geohabnet, 20
- GeoModel-class, 6
- GeoNetwork-class, 7
- GeoRasters-class, 8
- geoscale\_param, 8
- geoscale\_param(), 31
- get\_parameters, 9
- get\_parameters(), 10, 28, 29, 32
- get\_rasters, 10
- get\_rasters(), 24
- global\_scales, 11
- global\_scales(), 10, 27, 29, 32
- GlobalRast-class, 10
- Gmap-class, 11
- gplot, 12
  
- habitat\_density<-, 13
  
- habitat\_density<- , GeoNetwork-method  
(GeoNetwork-class), 7
- hci\_diff, 13
- hci\_diff(), 6
- hci\_mean, 14
- hci\_mean(), 6
- hci\_variance, 15
- hci\_variance(), 6
  
- igraph::betweenness(), 17, 20
- igraph::closeness(), 20
- igraph::degree(), 20
- igraph::eigen\_centrality(), 20
- igraph::knn(), 20
- igraph::page\_rank(), 20
- igraph::strength(), 20
- interactive(), 5, 29
- inv\_powerlaw, 16
- inv\_powerlaw(), 27, 28
  
- load\_parameters, 18
- load\_parameters(), 8, 10, 17
  
- msean (sean), 25
- msean(), 9, 23, 25–28
- msean\_onrast (sa\_onrasters), 23
- msean\_onrast(), 24
  
- ndplot, 19
- ndplot, GeoNetwork-method, 19
- neg\_expo (inv\_powerlaw), 16
- neg\_expo(), 27, 28
- nn\_sum, 20, 33
- node\_strength (nn\_sum), 20
  
- pagerank (nn\_sum), 20
- pagerank(), 17
  
- reset\_params, 21
- reso, 22
- reso(), 5, 14–16, 24, 27

risk\_indices, 22  
RiskMap (RiskMap-class), 22  
RiskMap-class, 22

sa\_onrasters, 23  
sa\_onrasters(), 23, 29  
sean, 25  
sean(), 5–7, 9, 23–25, 27, 29  
sensitivity\_analysis, 28  
sensitivity\_analysis(), 6, 7, 9, 23, 28, 29,  
32

set\_global\_scales, 31  
set\_global\_scales(), 11  
set\_parameters, 32  
set\_parameters(), 9, 28, 29  
setmaps (Gmap-class), 11  
setmaps, Gmap-method, 29  
setprops, 30  
setprops, GeoModel, character, numeric, numeric-method,  
31

supported\_metrics, 21, 33  
supported\_metrics(), 18

tempdir(), 5, 9, 14–16, 24, 27  
terra::ext(), 24, 26, 32  
terra::mean(), 14  
terra::plot(), 12  
terra::rast(), 10