Package ‘NicheR’

August 25, 2025

Type Package

Title Ecological Niche Modeling and Simulation

Version 0.1.0

Description Provides a robust set of tools for researchers and modelers to construct and define virtual ecological niches using ellipsoid geometries. It enables the identification and extraction of suitable environmental areas, simulation of species occurrence points with various sampling strategies, and visualization of niche boundaries and simulated occurrences in both environmental and geographic space. Inspired by methodologies in NicheA and the virtualspecies R package, NicheR aims to streamline the process of niche conceptualization and data generation for ecological studies.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

URL `https://github.com/castanedaM/NicheR`

BugReports `https://github.com/castanedaM/NicheR/issues`

RoxygenNote 7.3.2

Imports dplyr,
ggplot2,
ggpubr,
magrittr,
plotly,
RColorBrewer,
terra,
rgl

Contents

`build_ellps` *Build a 2D or 3D Ellipsoid Object*

Description

Creates a structured object representing a rotated ellipsoid. This object contains all necessary matrices for geometric and statistical calculations, as well as surface points for plotting. This function is a core component for defining virtual niches in environmental space.

Usage

```
build_ellps(
  center = c(x = 0, y = 0),
  axes = c(1, 1),
  angles = 0,
  n_points = 100
)
```

Arguments

`center` A numeric vector of length 2 or 3 defining the ellipsoid’s center. This vector can be named (e.g., ‘c(Temp = 15, Precip = 800)‘) to link dimensions to environmental variables.

`axes` A numeric vector with the same length as ‘center‘, defining the semi-axis lengths *before* rotation. These values represent the standard deviations of the ellipsoid along its principal axes. Must be positive.

`angles` A numeric vector of rotation angles in radians. For 2D, a single angle is sufficient. For 3D, a vector of three angles ‘c(ax, ay, az)‘ is required, corresponding to rotations around the x, y, and z axes, respectively.

`n_points` An integer specifying the resolution for the plotted surface. A higher value results in a smoother surface but increases computation time and file size.

Details

The function internally calculates the covariance matrix (‘Sigma‘) and its inverse, which are fundamental for computing Mahalanobis distance to determine if a point falls within the ellipsoid. The ‘surface‘ points are generated as a set of points on a unit sphere (in 3D) or circle (in 2D) that are then scaled by ‘axes‘ and rotated by ‘R‘ before being shifted to the ‘center‘.

Value

An S3 object of class ‘ellipsoid‘, which is a list containing:

`center` The numeric vector defining the ellipsoid’s center.

`axes` The numeric vector of semi-axis lengths.

`angles` The numeric vector of rotation angles.

`dimen` The dimensionality of the ellipsoid (2 or 3).

`R` The rotation matrix used to orient the ellipsoid.

`Sigma` The covariance matrix of the ellipsoid, based on ‘axes‘ and ‘R‘.

Sigma_inv The inverse of the covariance matrix, useful for Mahalanobis distance calculations.

surface A data frame of points representing the ellipsoid’s surface, suitable for plotting.

get_sample_occ *Sample Occurrence Points from a Suitable Environment*

Description

This function samples occurrence points from a background environmental dataset based on a defined ellipsoid niche. Sampling can be biased towards the center, the edge, or be purely random.

Usage

```
get_sample_occ(
  n_occ,
  env_bg,
  niche,
  method = c("random", "center", "edge"),
  seed = NULL
)
```

Arguments

n_occ Integer; the number of occurrence points to sample.

env_bg A ‘terra::SpatRaster‘, ‘data.frame‘, or ‘matrix‘ of environmental predictor variables. It must contain the variables referenced by the ‘niche‘ object. If a ‘data.frame‘ is used, it should also contain ’x’ and ’y’ columns for spatial referencing.

niche An object of class ‘ellipsoid‘ created by ‘build_ellipsoid()‘.

method A character string specifying the sampling strategy. Can be one of ‘"random"‘, ‘"center"‘, or ‘"edge"‘.

- ‘"random"‘: Samples are drawn with uniform probability from the suitable environmental space.
- ‘"center"‘: Sampling probability is inversely proportional to the Mahalanobis distance, favoring points closer to the niche center.
- ‘"edge"‘: Sampling probability is proportional to the Mahalanobis distance, favoring points closer to the niche boundary.

seed An integer used to set the random number generator seed for reproducible results.

Details

The function first identifies all points within the ellipsoid niche using the Mahalanobis distance. It then applies a weighting scheme based on the ‘method‘ argument to probabilistically select ‘n_occ‘ points from this suitable environment pool. The Mahalanobis distance squared is defined as:

$$(x-\mu)^T\Sigma^{-1}(x-\mu)$$

where x is a point in environmental space, μ is the niche center, and Σ^{-1} is the inverse covariance matrix. Points with a distance ≤ 1 are considered suitable.

Value

A ‘data.frame‘ containing ‘n_occ‘ sampled rows from ‘env_bg‘, including the ’x’ and ’y’ coordinates and the environmental predictor values.

See Also

[build_ellipsoid()], [get_suitable_env()]

Examples

```
## Not run:
# Assuming `build_ellipsoid` and `get_suitable_env` are available
# Define a 3D niche
my_niche <- build_ellipsoid(center = c(10, 50, 20),
                            axes = c(5, 15, 8),
                            angles = c(0, 0.5, 0.2))

# Create a sample environmental background dataset
set.seed(123)
env_data <- data.frame(
  x = runif(50000), y = runif(50000),
  Var1 = rnorm(50000, 10, 10),
  Var2 = rnorm(50000, 50, 10),
  Var3 = rnorm(50000, 20, 10)
)

# Sample 1000 random occurrences from the suitable area
occ_random <- get_sample_occ(n_occ = 1000,
                             env_bg = env_data,
                             niche = my_niche,
                             method = "random")

# Sample 1000 occurrences biased towards the center
occ_center <- get_sample_occ(n_occ = 1000,
                             env_bg = env_data,
                             niche = my_niche,
                             method = "center")

## End(Not run)
```

---

get_suitable_env *Extract Suitable Environmental Area from a Niche Ellipsoid*

Description

This function identifies and extracts all environmental grid cells or data points that fall within a defined ellipsoid niche based on Mahalanobis distance.

Usage

```
get_suitable_env(
  niche,
  env_bg,
  out = c("data.frame", "spatial", "both"),
  distances = FALSE
)
```

Arguments

niche An object of class ‘ellipsoid‘ created by ‘build_ellipsoid()‘.

env_bg A ‘terra::SpatRaster‘, ‘data.frame‘, or ‘matrix‘ of environmental predictor variables. It must contain the variables referenced by the ‘niche‘ object.

out A character string specifying the desired output format. Can be ‘"data.frame"‘, ‘"spatial"‘, or ‘"both"‘.

- ‘"data.frame"‘: Returns a data frame of all suitable points.
- ‘"spatial"‘: Returns a ‘terra::SpatRaster‘ where suitable cells are marked with 1 and unsuitable with 0. Requires ‘env_bg‘ to be a ‘terra::SpatRaster‘.
- ‘"both"‘: Returns a list containing both the spatial raster and the data frame of suitable points.

distances Logical; if ‘TRUE‘, an additional column named ‘dist_sq‘ is added to the output data frame containing the squared Mahalanobis distance for each suitable point.

Details

The function converts ‘env_bg‘ to a data frame, calculates the squared Mahalanobis distance for each point, and then filters for points where this distance is less than or equal to 1. This method efficiently identifies all locations within the niche’s boundary.

Value

The suitable environmental area in the specified format, which can be a ‘data.frame‘, a ‘terra::SpatRaster‘, or a list containing both.

See Also

[build_ellipsoid()], [get_sample_occ()]

plot_e_space *Plot Environmental Space with Optional Ellipsoid Overlays*

Description

Produces pairwise views of a 3D environmental space with optional overlays of a virtual niche ellipsoid and occurrence points. In 2D mode, it returns a grid of pairwise scatterplots with projected ellipse boundaries. In 3D mode, it returns an interactive ‘plotly‘ scatterplot.

Usage

```
plot_e_space(
  env_bg,
  x,
  y,
  z,
  labels = c("ENV 1", "ENV 2", "ENV 3"),
  n_bg = 10000,
  niche = NULL,
  show.pts.in = FALSE,
  occ_pts = NULL,
  rand_seed = 1234,
  show.occ.density = FALSE,
  plot.3d = FALSE
)
```

Arguments

`env_bg` A ‘data.frame‘ of background environments with at least three numeric predictor columns. These columns must contain the variables referenced by ‘x‘, ‘y‘, and ‘z‘.

`x, y, z` Column specifications for the three predictors to display. Each may be a single column name (character string) or a single 1-based integer index into ‘env_bg‘.

`labels` Character vector of length 3 giving axis labels for the x, y, and z variables in display order. Defaults to ‘c("ENV 1", "ENV 2", "ENV 3")‘.

`n_bg` Positive integer giving the maximum number of background rows to plot. If ‘nrow(env_bg)‘ is greater than ‘n_bg‘, a random subset of size ‘n_bg‘ is drawn. Using a large ‘n_bg‘ may slow plotting.

`niche` Optional object of class ‘ellipsoid‘ describing the niche. If provided, its boundary and center will be plotted. For 2D plots, the object must contain ‘niche$angles‘.

`show.pts.in` Logical. If ‘TRUE‘ and ‘niche‘ is provided, points from ‘env_bg‘ that fall inside the ellipsoid are highlighted. A warning is issued if ‘niche‘ is not provided.

`occ_pts` Optional ‘data.frame‘ of occurrence points that includes the same predictor columns used for ‘x‘, ‘y‘, and ‘z‘. These are overplotted if supplied.

`rand_seed` Integer used to set the random number generator seed for reproducible background downsampling.

`show.occ.density` Logical. If ‘TRUE‘ and ‘occ_pts‘ is provided, this adds marginal density panels for each variable. This is only supported in 2D plots (‘plot.3d = FALSE‘).

`plot.3d` Logical. If ‘TRUE‘, returns an interactive ‘plotly‘ 3D scatter plot. If ‘FALSE‘ (the default), returns a static ‘ggpubr‘ grid of 2D panels.

Details

This function is a powerful visualization tool for understanding the relationship between a species’ niche, its occurrences, and the available environmental space. The 2D views are especially useful for publication-quality figures, while the 3D interactive plot is great for data exploration. Note that in 2D plots, points from a 3D space may appear to fall outside the projected 2D ellipse boundary. This is expected and does not indicate an error.

Value

- If ‘plot.3d = TRUE‘: A ‘plotly‘ object. - If ‘plot.3d = FALSE‘: A ‘ggpubr‘ object containing arranged ‘ggplot2‘ panels.

See Also

[validate_plot_e_space_args()], [build_ellps()], [get_suitable_env()]

`%>%` *Pipe operator*

Description

This re-exports the pipe operator from `magrittr` to make it available for use in your package’s functions and for package users. See `magrittr::%>%` for details.

Arguments

`lhs`	A value or the magrittr placeholder.
`rhs`	A function call using the magrittr semantics.

Value

The result of calling ‘rhs(lhs)‘.

Index