

Package: ecoTolerance (via r-universe)

May 21, 2026

Type Package

Title Ecological Tolerance Indices

Version 0.1.0

Description Computes the Road Tolerance Index (RTI) and the Human Footprint Tolerance Index (HFTI) for species occurrence data. It automates data cleaning and integrates spatial data (roads and human footprint) to produce reproducible tolerance metrics for biodiversity and conservation research. The HFTI calculation is based on the global human footprint dataset by Mu et al. (2022) <doi:10.1038/s41597-022-01284-8>. This package is part of a PhD thesis focused on amphibian ecology in Brazil.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Imports sf, raster, dplyr, stats, ggplot2, magrittr, methods

Suggests terra, testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Diego F. Miranda [aut, cre], Lucas R. Forti [aut]

Maintainer Diego F. Miranda <dfernandes115@gmail.com>

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://mirandafdiego.r-universe.dev>

Date/Publication 2026-02-19 20:40:02 UTC

RemoteUrl <https://github.com/cran/ecoTolerance>

RemoteRef HEAD

RemoteSha 716da03ec84a33879a175e23e90bd881fe37197b

Contents

calculate_HFTI	2
calculate_RTI	3
compute_indices	4
generate_all_reports	5
load_human_footprint	6
load_roads	7
process_occurrences	7
Index	9

calculate_HFTI	<i>Calculate Human Footprint Tolerance Index (HFTI)</i>
----------------	---

Description

This function calculates the HFTI for each species occurrence based on the average human footprint within a 1 km buffer around each occurrence.

Usage

```
calculate_HFTI(occurrences, footprint, divisor = 50, buffer_m = 1000)
```

Arguments

occurrences	An sf object containing processed occurrences.
footprint	A RasterLayer containing human footprint values.
divisor	A numeric value to divide the footprint by (default = 50).
buffer_m	Numeric. Buffer size in meters for footprint extraction (default = 1000).

Value

A list with:

data The sf object of occurrences with a new column HFTI_value.

HFTI A data frame with the median HFTI per species.

Examples

```
# Note: simple example using the internal "toy" data
footprint <- load_human_footprint()
# We need dummy occurrence data for the example to run
occ_data <- data.frame(species = "SpA", longitude = -38.4, latitude = -12.6)
processed <- process_occurrences(occ_data)
hfti_result <- calculate_HFTI(processed, footprint, divisor = 50, buffer_m = 1000)
```

calculate_RTI	<i>Calculate Road Tolerance Index (RTI)</i>
---------------	---

Description

Esta função calcula o RTI para cada ocorrência de espécie com base na distância mínima até a estrada mais próxima. A fórmula usada é:

$$RTI = \frac{ref_dist}{dist + ref_dist}$$

Usage

```
calculate_RTI(occurrences, roads, ref_dist = 3.5)
```

Arguments

occurrences	Um objeto sf contendo as ocorrências já processadas (output de <code>process_occurrences()</code>).
roads	Um objeto sf contendo a rede de estradas.
ref_dist	Distância de referência em quilômetros (padrão = 3.5).

Details

Se `dist = 0`, então `RTI = 1`. Conforme `dist` aumenta, `RTI` aproxima-se de 0.

Value

Uma lista com:

data O objeto sf de ocorrências, porém com uma nova coluna `RTI_value` com o índice calculado.

RTI Um data frame com a mediana de RTI por espécie.

Examples

```
# Carregar rodovias de exemplo (Toy data)
roads <- load_roads()

# Criar dados de ocorrencia ficticios na area do Toy Data
occ_data <- data.frame(
  species = "SpA",
  longitude = -38.4,
  latitude = -12.6
)

# Processar ocorrencias
processed <- process_occurrences(occ_data)

# Calcular RTI
rti_res <- calculate_RTI(processed, roads, ref_dist = 3.5)
```

compute_indices *Compute both RTI and HFTI*

Description

This main function processes occurrence data (removing duplicates, filtering close points), then calculates the Highway Tolerance Index (RTI) and the Human Footprint Tolerance Index (HFTI).

Usage

```
compute_indices(
  data,
  remove_duplicates = TRUE,
  buffer_km = 1,
  ref_dist = 3.5,
  divisor = 50,
  data_type = "toy"
)
```

Arguments

data	A data frame with columns "species", "longitude", and "latitude".
remove_duplicates	Logical. If TRUE, remove exact duplicates of species + coordinates.
buffer_km	Numeric. Buffer size in kilometers for removing nearby points.
ref_dist	Numeric. Reference distance in kilometers for RTI.
divisor	Numeric. Divisor for the human footprint values in HFTI.
data_type	Character. Options are "toy" (default) or "full".

Value

A list containing:

RTI A data frame with median RTI values per species.

HFTI A data frame with median HFTI values per species.

indices A data frame combining both RTI and HFTI by species.

processed_data The sf object of occurrences with RTI and HFTI values.

Examples

```
# Example usage with internal toy data
# Using coordinates that fall within the provided toy dataset (Bahia, Brazil)
sample_data <- data.frame(
  species = c("SpA", "SpA", "SpB", "SpB"),
  longitude = c(-38.40, -38.41, -38.42, -38.43),
  latitude = c(-12.60, -12.61, -12.62, -12.63)
```

```
)  
  
result <- compute_indices(  
  data = sample_data,  
  remove_duplicates = TRUE,  
  buffer_km = 1,  
  ref_dist = 3.5,  
  divisor = 50  
)  
  
print(result$RTI)  
print(result$HFTI)  
print(result$indices)
```

generate_all_reports *Generate and save maps & density plots for RTI and HFTI of all species*

Description

Esta funcao recebe o resultado de `compute_indices()`, um shapefile de area de estudo e o diretorio de saida. Para cada especie presente em `result$processed_data`, ela gera:

1. Mapa de RTI (points coloridos por RTI_value)
2. Mapa de HFTI (points coloridos por HFTI_value)
3. Grafico de densidade de RTI (com linha vertical na mediana)
4. Grafico de densidade de HFTI (com linha vertical na mediana) Alem disso, salva um arquivo CSV com RTI e HFTI medios por especie.

Usage

```
generate_all_reports(  
  result,  
  area_shapefile,  
  out_dir,  
  map_crs = 4326,  
  point_size = 1.5,  
  density_fill = "steelblue",  
  density_alpha = 0.35  
)
```

Arguments

`result` Lista retornada por `compute_indices()`, contendo:
processed_data Um objeto `sf` com as ocorrencias.
indices Data frame com RTI e HFTI medios por especie.

<code>area_shapefile</code>	Caminho (string) para o shapefile da area de estudo (por exemplo, limite do Brasil). Deve ser um arquivo .shp.
<code>out_dir</code>	Caminho (string) para o diretorio onde as imagens e o CSV serao salvos. Caso nao exista, sera criado automaticamente.
<code>map_crs</code>	CRS desejado para os mapas (padrao = 4326, WGS84).
<code>point_size</code>	Tamanho dos pontos nos mapas (padrao = 1.5).
<code>density_fill</code>	Cor de preenchimento dos graficos de densidade (padrao = "steelblue").
<code>density_alpha</code>	Transparencia do preenchimento dos graficos de densidade (padrao = 0.35).

Value

Invisivel. A funcao retorna NULL apos salvar todos os arquivos em `out_dir`.

`load_human_footprint` *Load Human Footprint GeoTIFF*

Description

This function loads the human footprint data. By default, it loads a small "toy" raster included in the package. To use the full Brazil raster, set `type = "full"`.

Usage

```
load_human_footprint(type = "toy")
```

Arguments

`type` Character. Options are "toy" (default) or "full".

Value

A RasterLayer object containing human footprint data.

load_roads	<i>Load Brazilian Roads Data</i>
------------	----------------------------------

Description

This function loads the roads data. By default, it loads a small "toy" dataset included in the package for testing and examples. To use the full dataset (all Brazilian roads), set `type = "full"`. The full dataset will be downloaded from Zenodo upon the first request and cached locally.

Usage

```
load_roads(type = "toy")
```

Arguments

`type` Character. Options are "toy" (default) or "full".

Value

An sf object containing roads data.

process_occurrences	<i>Process occurrence data</i>
---------------------	--------------------------------

Description

This function processes species occurrence data by optionally removing duplicates and filtering out points that are closer than a specified distance (buffer).

Usage

```
process_occurrences(data, remove_duplicates = TRUE, buffer_km = 1)
```

Arguments

`data` A data frame containing at least the columns "species", "longitude", and "latitude".

`remove_duplicates` Logical. If TRUE, remove exact duplicates of species + coordinates.

`buffer_km` The buffer size in kilometers used to filter out close points.

Value

An sf object with the processed occurrences.

Examples

```
# Example data
occ_data <- data.frame(
  species = c("SpA", "SpA", "SpB", "SpB"),
  longitude = c(-50, -50.0005, -51, -51.001),
  latitude = c(-10, -10.0005, -11, -11.001)
)

processed <- process_occurrences(
  data = occ_data,
  remove_duplicates = TRUE,
  buffer_km = 1
)
processed
```

Index

calculate_HFTI, [2](#)
calculate_RTI, [3](#)
compute_indices, [4](#)

generate_all_reports, [5](#)

load_human_footprint, [6](#)
load_roads, [7](#)

process_occurrences, [7](#)