

Package ‘AnimalHabitatNetwork’

July 21, 2025

Type Package

Title Networks Characterising the Physical Configurations of Animal Habitats

Version 0.1.0

Maintainer Peng He <phe@ab.mpg.de>

Description Functions for generating and visualising networks for characterising the physical attributes and spatial organisations of habitat components (i.e. habitat physical configurations). The network generating algorithm first determines the X and Y coordinates of N nodes within a rectangle with a side length of L and an area of A. Then it computes the pair-wise Euclidean distance D_{ij} between node i and j, and then a complete network with $1/D_{ij}$ as link weights is constructed. Then, the algorithm removes links from the complete network with the probability as shown in the function `ahn_prob()`. Such link removals can make the network disconnected whereas a connected network is wanted. In such cases, the algorithm rewires one network component to its spatially nearest neighbouring component and repeat doing this until the network is connected again. Finally, it outputs an undirected network (weighted or unweighted, connected or disconnected). This package came with a manuscript on modelling the physical configurations of animal habitats using networks (in preparation).

License GPL-3

Imports stats, igraph (>= 1.2.4.1), ggplot2 (>= 3.2.0)

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Author Peng He [aut, cre] (ORCID: <<https://orcid.org/0000-0002-7176-701X>>),
Damien R. Farine [ths] (ORCID: <<https://orcid.org/0000-0003-2208-7613>>)

Repository CRAN

Date/Publication 2019-11-25 19:40:02 UTC

Contents

ahn_gen	2
ahn_plot	3
ahn_prob	4

ahn_gen

*Generate networks characterising habitat physical configurations***Description**

Generate undirected networks (weighted or unweighted, connected or disconnected) characterising the physical attributes and spatial organizations (or distributions) of habitat components (i.e. habitat configurations).

Usage

```
ahn_gen(N, L, mu, lamda, Connected = TRUE, Weighted = TRUE, eta = 1,
        A = 25, X = NULL, Y = NULL, U = NULL, V = NULL)
```

Arguments

N	The number of nodes
L	A side length of the rectangle landscape within which nodes are anchored
mu	the critical D_{ij} (i.e. Euclidean distance between node i and j) at which the link removing probability curve $P(D_{ij}, \mu, \text{lamda})$ transits from concave to convex (see ahn_prob)
lamda	the steepness of the link removing probability curve $P(D_{ij}, \mu, \text{lamda})$, see ahn_prob
Connected	TRUE for connected while FALSE ignores whether the networks are connected or not
Weighted	TRUE for weighted while FALSE for unweighted networks
eta	mediates the weight, i.e. $(D_{ij})^{-\eta}$, of the link rewiring node i from one network component and node j from another network component (i and j are with an Euclidean distance of D_{ij}) when the network becomes disconnected after removing links from the initial complete network with the probability $P(D_{ij}, \mu, \text{lamda}) = [1 + \exp(-\text{lamda}(D_{ij} - \mu))]^{-1}$ when both <code>Connected = TRUE</code> and <code>Weighted = TRUE</code>
A	The area of the rectangle landscape within which the network is defined
X	A vector of X coordinates for the N nodes (sampled from $[0, L]$ uniformly at random if NULL)
Y	A vector of Y coordinates for the N nodes (sampled from $[0, A/L]$ uniformly at random if NULL)
U	A vector with N elements specifying node attributes (qualitative or quantitative), by default NULL
V	A vector with N elements specifying node attributes (qualitative or quantitative), by default NULL

Value

Return an animal habitat network (an igraph object)

Examples

```
# generate a connected and weighted network
ahn_gen(N = 10, L = 5, mu = 1, lamda = 5)

N <- 10
x <- runif(N, 0, 5)
ql <- sample(LETTERS, N, replace = TRUE)
qn <- sample(1:20, N, replace = TRUE)

# specify the X coordinates, node attributes U and V for a connected and unweighted network
ahn_gen(N, L = 5, mu = 1, lamda = 5, Weighted = FALSE, X = x, U = ql, V = qn)

# specify the Y coordinates, node attributes U and V for a weighted network, no matter if the
# network will be connected or not
ahn_gen(N, L = 5, mu = 1, lamda = 5, Weighted = TRUE, Connected = FALSE, Y = x, U = ql, V = qn)
```

ahn_plot

*Plot networks***Description**

Visualise networks generated by the function [ahn_gen](#).

Usage

```
ahn_plot(ahn, NodeLabels = unname(V(ahn)), NodeColors = unname(V(ahn)),
  NodeSizes = rep(3, length(V(ahn))))
```

Arguments

ahn	Networks returned by ahn_gen
NodeLabels	The labels of nodes in ahn (node IDs by default)
NodeColors	The colors of nodes in ahn (each node has a unique color by default)
NodeSizes	The sizes of nodes in ahn (nodes are with the identical size of 3 by default)

Value

Return a plot of the network

Examples

```
# generate a weighted and connected network and plot it by default
N <- 10
x <- runif(N, 0, 5)
ahn <- ahn_gen(N, L = 5, mu = 1, lamda = 5, X = x)
ahn_plot(ahn)

# plot the network with specified colors, labels and sizes for nodes
ahn_plot(
  ahn,
  NodeColors = sample(4, N, replace = TRUE),
  NodeLabels = letters[1:N],
  NodeSizes = seq(1, 5, length.out = N))
```

ahn_prob

Plot probability curves

Description

Plot the probability curve $P(D_{ij}, \mu, \lambda)$ for removing links from the initial complete network

Usage

```
ahn_prob(Dij = seq(0.05, 10, length.out = 30), mu = c(0.1, 2, 5, 10),
  lamda = c(1e-04, 0.15, 0.35, 0.75, 1.25, 5, 30))
```

Arguments

Dij	A vector of Euclidean distances between node i and j
mu	The concave-to-convex transition point of the probability curves $P(D_{ij}, \mu, \lambda) = [1 + \exp(-\lambda(D_{ij} - \mu))]^{-1}$, where D_{ij} is the Euclidean distance between node i and j
lamda	The steepness of the probability curves

Value

Return a plot with probability curves

Examples

```
# plot the probabilities for removing network links between node i and j with  
# Euclidean distances Dij
```

```
dis <- seq(.05, 10, length.out = 20)  
m <- c(.1, 2, 5, 10)  
l <- c(.0001, .15, .35, .75, 1.25, 5, 30)  
ahn_prob(dis, m, l)
```

Index

ahn_gen, [2](#), [3](#)
ahn_plot, [3](#)
ahn_prob, [2](#), [4](#)