Package ‘maxnodf’

October 13, 2022

Title Approximate Maximisation of Nestedness in Bipartite Graphs

Version 1.0.0

Description Functions to generate graphs that maximise the NODF (nestedness metric based on overlap and decreasing fill) metric for a given number of rows, columns and links. NODF was originally defined by Almeida-Neto et al. (2008) <doi:10.1111/j.0030-1299.2008.16644.x>. As nestedness in ecological networks depends on the size of the networks we require normalisation to make them comparable. We offer three highly optimised algorithms to find the optimising graphs so that users can choose an appropriate trade off between computation time and NODF value for the task at hand.

Depends R (>= 3.4.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

LinkingTo Rcpp

Imports Rcpp (>= 0.12.18), stats (>= 3.4.4), utils (>= 3.4.4),

Suggests testthat, covr

NeedsCompilation yes

Author Christoph Hoeppke [aut, cre],
Benno Simmons [aut]

Maintainer Christoph Hoeppke <christoph.hoeppke@googlemail.com>

Repository CRAN

Date/Publication 2020-03-13 11:10:02 UTC

R topics documented:

`maxnodf` *Calculate the maximum nestedness of a bipartite network*

Description

Calculates the maximum NODF that be achieved in a network with a given number of rows, columns and links.

Usage

```
maxnodf(web, quality = 0)
```

Arguments

`web`	Either a numeric matrix describing a bipartite network (a bipartite incidence matrix where elements are positive numbers if nodes interact, and 0 otherwise) or a numeric vector of length 3 of the form web = c(#Rows, #Columns, #Links).
`quality`	An optional quality parameter to control the tradeoff between computation time and result quality. Can be 0, 1 or 2.

Details

For a given network, `maxnodf` calculates the maximum nestedness that can be achieved in a network with a given number of rows, columns and links, subject to the constraint that all rows and columns must have at least one link (i.e. marginal totals must always be >= 1). This allows nestedness values to be normalised as $NODF/max(NODF)$ following Song et al (2017). To control for connectance and network size, Song et al. (2017) suggest an additional normalisation that can be used: $(NODF/max(NODF))/(C * log(S))$ where C is the network connectance and S is the geometric mean of the number of plants and pollinators in the network.

`maxnodf` has three algorithms for finding the maximum nestedness of a bipartite network. These can be set using the `quality` argument. Lower quality settings are faster, but find worse optima. Higher quality settings are slower, but find better optima.

- `quality` = 0, uses a greedy algorithm.
- `quality` = 1, uses a greedy algorithm plus hillclimbing.
- `quality` = 2, uses a simulated annealing algorithm, with the greedy algorithm output as the start point. Best results, but requires the most computation time.

Value

Returns a list of length 2, where the first element ('max_nodf') is the maximum nestedness of the network and the second element ('max_nodf_mtx') is the incidence matrix corresponding to this maximum nestedness.

References

Song, C., Rohr, R.P. and Saavedra, S., 2017. Why are some plant–pollinator networks more nested than others? Journal of Animal Ecology, 86(6), pp.1417-1424

Examples

```
maxnodf(matrix(1.0, 12, 10))
maxnodf(c(14, 13, 52), 2)
```

NODFc *Calculate NODF_c for a bipartite network*

Description

Calculates the NODF_c metric proposed by Song et al (2017) for a bipartite incidence matrix

Usage

```
NODFc(web, quality = 0)
```

Arguments

`web`	A numeric matrix describing a bipartite network (a bipartite incidence matrix where elements are positive numbers if nodes interact, and 0 otherwise).
`quality`	An optional quality parameter to control the tradeoff between computation time and result quality. Can be 0, 1 or 2.

Details

For a given network, `NODFc` calculates the NODF_c metric proposed by Song et al (2017), defined as $(NODF/max(NODF))/(C * log(S))$ where C is the network connectance, S is the geometric mean of the number of plants and pollinators in the network, NODF is the raw NODF of the network and max(NODF) is the maximum nestedness that can be achieved in a network with the same number of rows, columns and links as `web`, subject to the constraint that all rows and columns must have at least one link (i.e. marginal totals must always be >= 1). `NODFc` has three algorithms for finding the maximum nestedness of a bipartite network. These can be set using the `quality` argument. Lower quality settings are faster, but find worse optima. Higher quality settings are slower, but find better optima.

- `quality` = 0, uses a greedy algorithm.
- `quality` = 1, uses a greedy algorithm plus hillclimbing.
- `quality` = 2, uses a simulated annealing algorithm, with the greedy algorithm output as the start point. Best results, but requires the most computation time.

Value

Returns the value of NODF_c as a single number.

References

Song, C., Rohr, R.P. and Saavedra, S., 2017. Why are some plant–pollinator networks more nested than others? Journal of Animal Ecology, 86(6), pp.1417-1424

Examples

```
set.seed(123)
NODFc(matrix(sample(x = 0:1, size = 100, replace = TRUE),10,10), quality = 0)
```

---

nodf_cpp *Raw NODF calculation*

Description

Calculates the raw NODF of a bipartite incidence matrix

Usage

```
nodf_cpp(mtx)
```

Arguments

mtx	A numeric matrix describing a bipartite network (a bipartite incidence matrix where elements are positive numbers if nodes interact, and 0 otherwise).

Details

For a given network, `nodf_cpp` calculates the raw NODF value. Calculation is fast as the code is implemented in C++.

Value

Returns the NODF of the network.

Examples

```
set.seed(123)
nodf_cpp(matrix(sample(x = 0:1, size = 100, replace = TRUE),10,10))
```

Index