Tutorial do reproduce the case study from Pinheiro et al. (submitted)

About

Supplementary code and data from the study:

Pinheiro, R. B. P., Dormann, C. F., Felix, G. M. F., & Mello M. A. R. ***Linking null models to hypotheses to improve nestedness analysis****. Submitted.*

Ecological Synthesis Lab (SintECO)

Authors: Rafael B. P. Pinheiro, Gabriel M. Felix, Carsten F. Dormann, and Marco A. R. Mello.

E-mail: rafael-bpp@hotmail.com.

Published originaly on September 29th, 2021 (English version).

Disclaimer: You may use this script freely for commercial or non-commercial purposes at your own risk. We assume no responsibility or liability for the use of this software, convey no license or title under any patent, copyright, or mask work right to the product. We reserve the right to make changes in the software without notification. We also make no representation or warranty that such application will be suitable for the specified use without further testing or modification. If this script helps you produce any academic work (paper, book, chapter, dissertation, monograph, thesis, report, talk, keynote, lecture, and similar), please acknowledge the authors and cite the source.

See detailed instructions and further info in the README file.

General instructions

Use the following tutorial to reproduce the case study presented in our paper. The file `net.txt` represents the network analyzed by Kaiser-Bunbury et al. (2010). If you want to apply our new protocol to your own data, just replace `net.txt` with another network of your choice, following the same format.

The empirical network analyzed in our paper was downloaded from the Web of Life database (network: M_PL_060_17). It was originally analyzed by:

Kaiser-Bunbury CN, Muff S, Memmott J, Muller CB, Caflisch A. 2010. ***The robustness of pollination networks to the loss of species and interactions: a quantitative approach incorporating pollinator behaviour****. Ecol. Lett. 13, 442-452. http://dx.doi.org/10.1111/j.1461-0248.2009.01437.x.*

Step-by-step protocol

The goal of this tutorial is to help you analyze the nestedness of an example network using our new protocol. Most researchers measure a network's degree of nestedness and then compare this observed value against a randomized distribution generated with a null model, in order to state whether the networks is "nested or not". We propose to go beyond this dichotomic approach by assessing topology more thoroughly, for the reasons explained in our paper.

First, we measure the nestedness observed in the network with a metric of choice. Second, using a null model that removes all nestedness-generating mechanisms (*equiprobable*), we create a randomized distribution of those nestedness values. This distribution represents what would be expected for a *non-nested topology*, to which we compare the observed value against. Third, using a null model that maximizes nestedness in the network (*proportional*), we create another randomized distribution of nestedness values. This distribution represents what would be expected for a *fully nested topology*, to which we also compare the observed value against. This procedure leads to four possible scenarios:

1. If the observed value of nestedness is equal to or lower than 95% (or any other cut you prefer) of the randomized values that represent a non-nested topology, we conclude that the network's topology probably emerged at random, without an underlying mechanism;
2. If the observed value of nestedness is higher than 95% of the randomized values that represent a non-nested topology, we conclude that the network's topology contains at least some nestedness and probably did not emerge at random;
3. If the observed value of nestedness is equal to or lower than 95% of the randomized values that represent a fully nested topology, we conclude that the network's topology contains at least some nestedness, but that nestedness is not a good archetype to represent the observed structure;
4. If the observed value of nestedness is higher than 95% of the randomized values that represent a fully nested topology, we conclude that nestedness is a good archetype to represent the observed structure.

Computationally, our new protocol consists of the following steps:

Step 1

Create a new object with the network to be analyzed.

```
net1 <- as.matrix(read.table("data/net.txt")) #You may use another data file of your choice
```

Step 2

Load the required packages:

```
library(bipartite)
```

Step 3

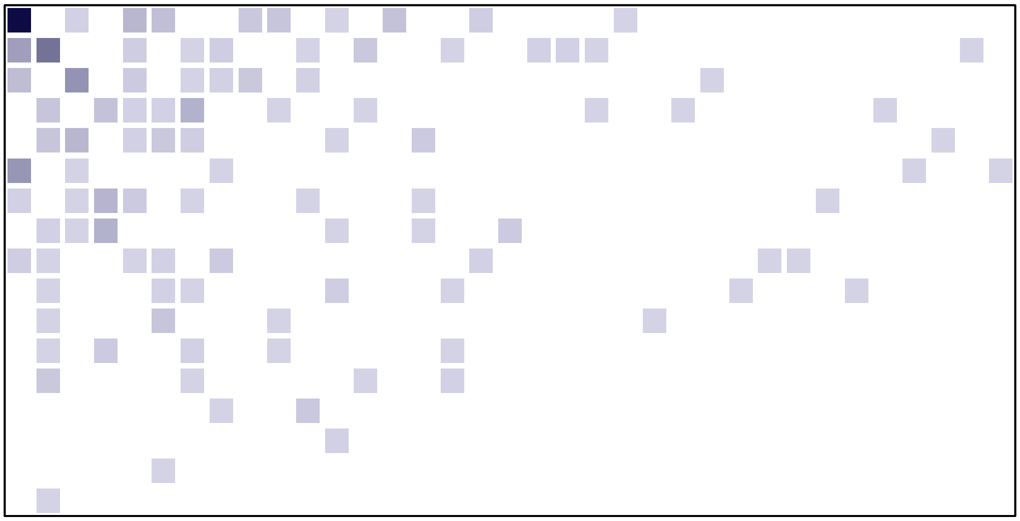
Load our custom-made function to calculate nestedness and run the null models. Check out the comments in the respective R file to obtain detailed information about the function, including its arguments and options.

```
source("functions/nestmodels.R")
```

Step 4

Plot the matrix to get a feeling about its structure. Visual inspection is a key step in all kinds of numerical analysis, and we strongly recommend it also for network analysis.

```
plotmatrix(net1, binary = F,
           within_color = "#0e0a44FF",
           base_color = "#d7d6e9ff")
```



Step 5

The simulations included in our protocol are quite resource-consuming and may require several minutes, depending on your computer's processing power. Therefore, choose the number of randomized matrices (iterations) carefully. In our paper, we ran 10,000 iterations, but we suggest experimenting with smaller numbers first.

```
matrices <- 10000 #This number will be the same for all functions
```

Step 6

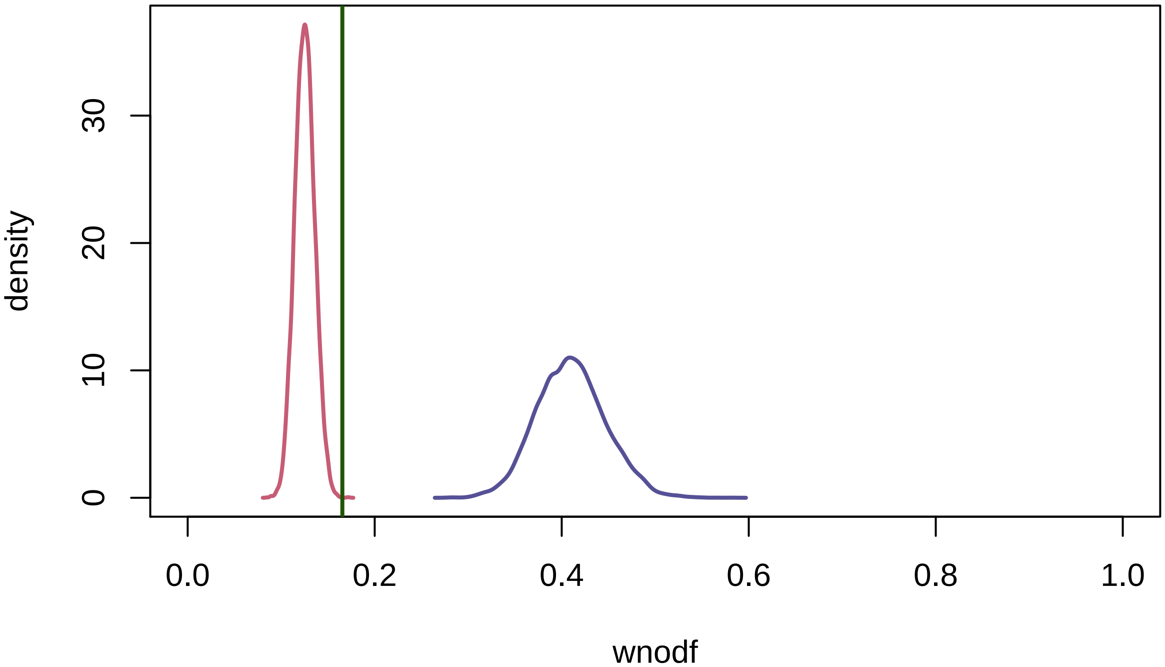
Generate randomized matrices and calculate their degrees of nestedness using different metrics.

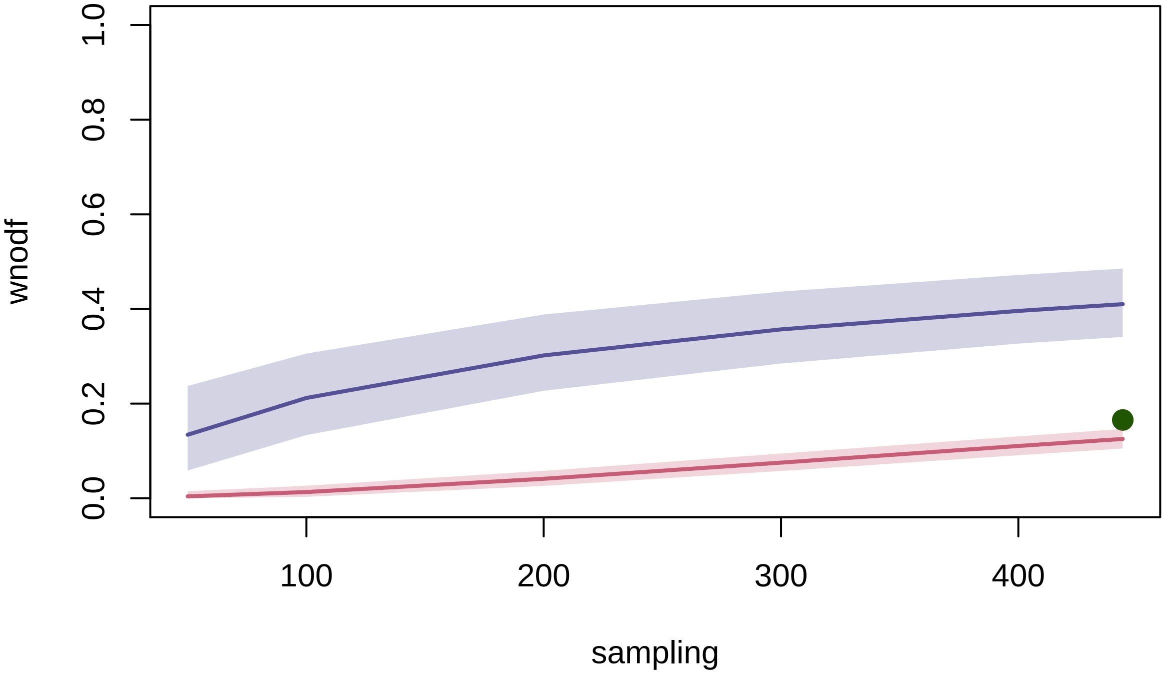
WNODF

Run the protocol.

```
WNODF=nestmodels(M=net1,
                 index="wnodf",
                 n.model = matrices,
                 prop.model = T,
                 print.at.each = 1000,
                 density.plot = T,
                 sampling.plot = T,
                 calc.at = c(50,100,200,300,400,444))
```

```
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 7000
## [1] 8000
## [1] 9000
## [1] 10000
```





The **first plot** shows the distributions of randomized nestedness values (densities) generated with the equiprobable (**red**, removes all nestedness-generating mechanisms) and proportional (**blue**, maximizes nestedness) null models. In **green** it shows the observed nestedness value.

The **second plot** shows the variation in nestedness values as a function of sampling effort (number of links in the network), for the proportional (**blue**) and the equiprobable (**red**) null models. In **green** it shows the observed nestedness value.

Check out the observed value and the P-values for each null model.

```
WNODF$observed
```

```
## [1] 0.1654143
```

```
WNODF$significance
```

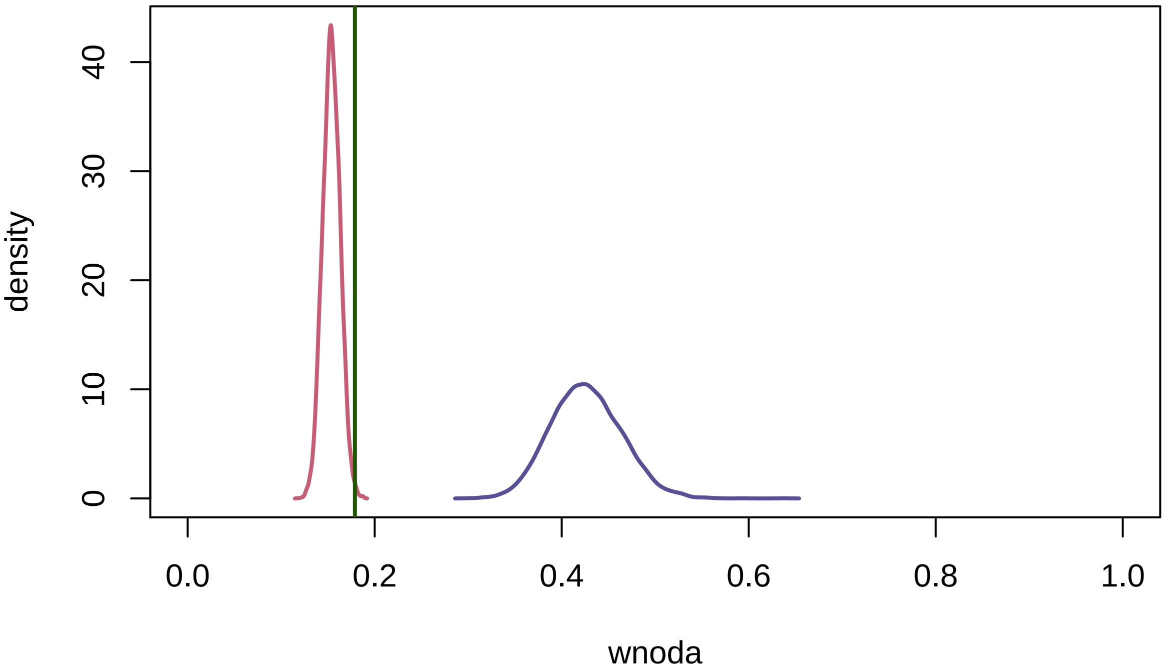
```
## $pvalue.equi
## [1] 7.302918e-05
##
## $Zscore.equi
## [1] 3.797677
##
## $pvalue.prop
## [1] 1.417305e-11
##
## $Zscore.prop
## [1] -6.654916
```

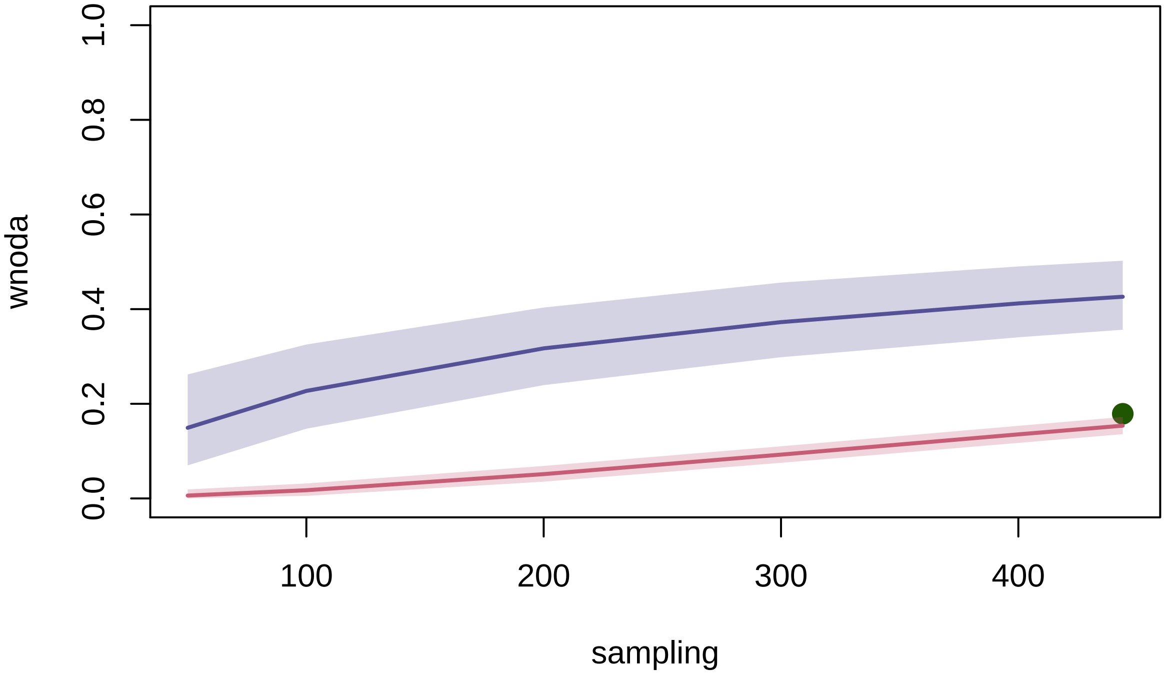
WNODA

Run the protocol.

```
WNODA=nestmodels(M=net1,
                 index="wnoda",
                 n.model = matrices,
                 prop.model = T,
                 print.at.each = 1000,
                 density.plot = T,
                 sampling.plot = T,
                 calc.at = c(50,100,200,300,400,444))
```

```
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 7000
## [1] 8000
## [1] 9000
## [1] 10000
```





The **first plot** shows the distributions of randomized nestedness values (densities) generated with the equiprobable (**red**, removes all nestedness-generating mechanisms) and proportional (**blue**, maximizes nestedness) null models. In **green** it shows the observed nestedness value.

The **second plot** shows the variation in nestedness values as a function of sampling effort (number of links in the network), for the proportional (**blue**) and the equiprobable (**red**) null models. In **green** it shows the observed nestedness value.

Check out the observed value and the P-values for each null model.

```
WNODA$observed
```

```
## [1] 0.1788271
```

```
WNODA$significance
```

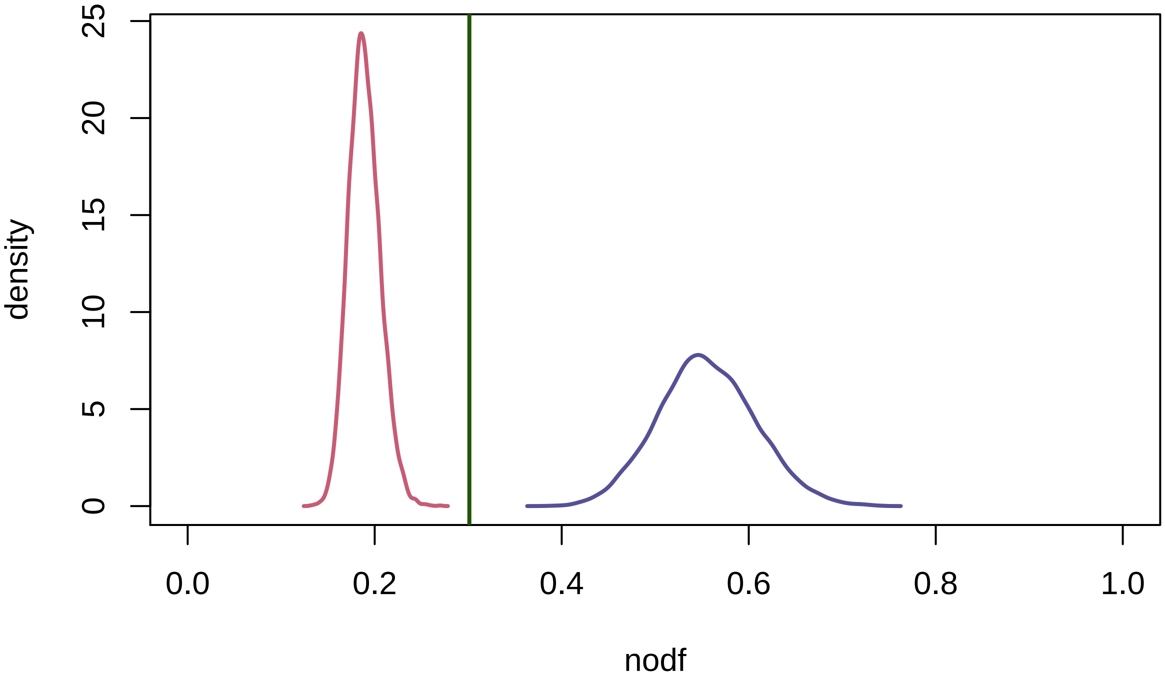
```
## $pvalue.equi
## [1] 0.003770218
##
## $Zscore.equi
## [1] 2.671983
##
## $pvalue.prop
## [1] 2.439684e-11
##
## $Zscore.prop
## [1] -6.574571
```

NODF

Accounting for weighted marginal sums Run the protocol.

```
NODF1=nestmodels(M=net1,
                 index="nodf",
                 n.model = matrices,
                 prop.model = T,
                 equi.model = T,
                 print.at.each = 1000,
                 density.plot = T,
                 wprob = T,
                 wsamp=F)
```

```
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 7000
## [1] 8000
## [1] 9000
## [1] 10000
```



The **plot** shows the distributions of randomized nestedness values (densities) generated with the equiprobable (**red**, removes all nestedness-generating mechanisms) and proportional (**yellow**, maximizes nestedness, considering link weights) null models. In **green** it shows the observed nestedness value.

Check out the observed value and the P-values for each null model.

```
NODF1$observed
```

```
## [1] 0.3012393
```

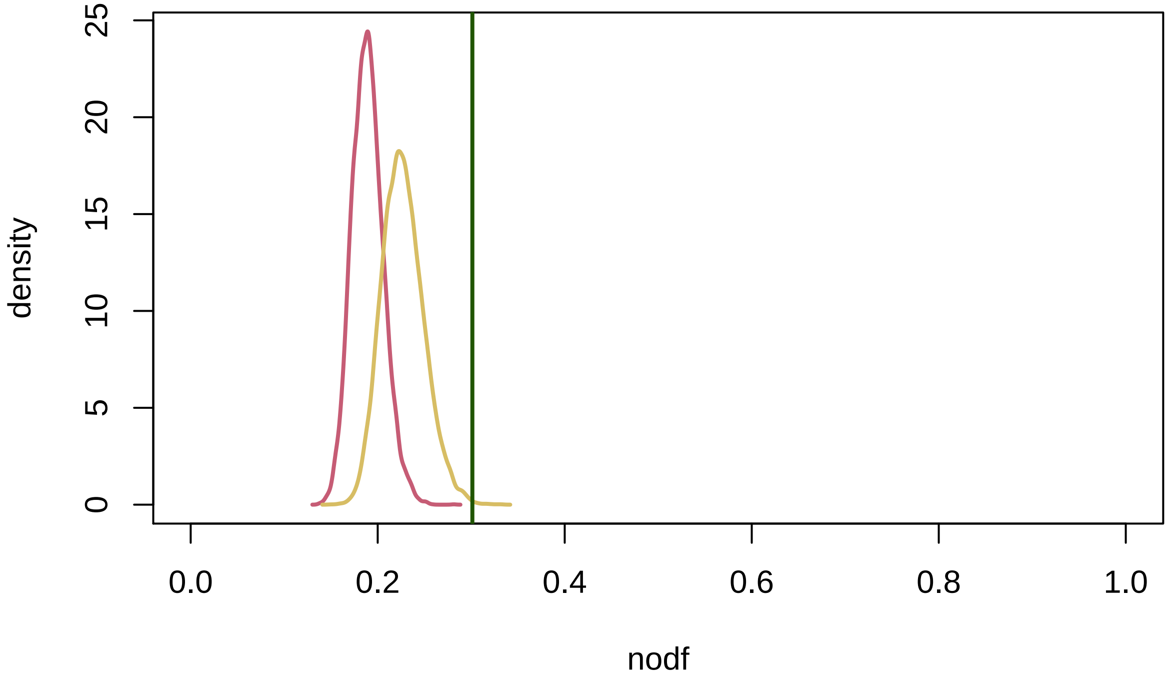
```
NODF1$significance
```

```
## $pvalue.equi
## [1] 1.383995e-11
##
## $Zscore.equi
## [1] 6.658414
##
## $pvalue.prop
## [1] 4.652947e-07
##
## $Zscore.prop
## [1] -4.905775
```

Using only binary information Run the protocol.

```
NODF2=nestmodels(M=net1,
                 index="nodf",
                 n.model = matrices,
                 prop.model = T,
                 equi.model = T,
                 print.at.each = 1000,
                 density.plot = T,
                 wprob = F,
                 wsamp=F)
```

```
## [1] 1000
## [1] 2000
## [1] 3000
## [1] 4000
## [1] 5000
## [1] 6000
## [1] 7000
## [1] 8000
## [1] 9000
## [1] 10000
```



The **plot** shows the distributions of randomized nestedness values (densities) generated with the equiprobable (**red**, removes all nestedness-generating mechanisms) and proportional (**yellow**, maximizes nestedness, but ignoring link weights) null models. In **green** it shows the observed nestedness value.

Check out the observed value and the P-values for each null model.

```
NODF2$observed
```

```
## [1] 0.3012393
```

```
NODF2$significance
```

```
## $pvalue.equi
## [1] 1.226154e-11
```

```
##
## $Zscore.equi
## [1] 6.676192
##
## $pvalue.prop
## [1] 0.0004280806
##
## $Zscore.prop
## [1] 3.333969
```