A novel perspective on the meaning of nestedness with conceptual and methodological solutions. Pinheiro R. B. P.*, Dormann C.F., Felix G.M.F., and Mello M.A.R.

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

* Webpage: https://rbppinheiro.wordpress.com

* GitHub: https://github.com/pinheirorbp

Contents

Commented scripts

To reproduce the analysis performed in our study, please keep the actual organization of folders and set as working directory the main folder containing the files “simulations.R” and “case_study.R”.

Packages required: sads, bipartite, betapart, and doParallel. All required packages are available at CRAN repository (https://CRAN.R-project.org).

simulations.R

This script runs simulations to compare the capacity of several nestedness indices in distinguishing between matrices with random topologies (equiprobable null model) and nested matrices (proportional model).

The script performs 3 types of simulations: 1) weighted matrices with fixed dimensions and total weights, 2) binary matrices with fixed connectance, 3) binary matrices with fixed connectance but with probabilities derived from binary marginal totals (see main text). Simulated matrices are produced, indices are calculated, and results are saved within the folder “simulations”. For each type of simulation, 720 files are saved, each one containing 1000 simulations.

Parallel processing is used on the script. The easiest way to run the code without parallelizing is by registering only 1 core in “registerDoParallel (cores=1)”, but be prepared because the simulations may take a LONG time.

After performing the simulations and calculating the indices, the last part of the script summarizes the results, producing the files “simulations1.RData”, “simulations2.RData”, and “simulations3.RData” within the folder “results”. Those files are used in the R markdown files to produce Appendix S1 and S2.

As this code demands significant computational resources, we saved within the folder “results” the three summary files (“simulations1.RData”, “simulations2.RData”, and “simulations3.RData”) resulting from the simulations we presented in the main text. Thus, you may inspect our analysis thought this “intermediary” data, instead of re-running everything from the beginning.

R Markdown scripts: Appendix S1.Rmd and Appendix S2.Rmd

These scripts are used to generate the pdf Appendices. They use the summaries produced at the end of simulations.R and produce plots to compare the ability of nestedness indices in distinguishing nested and non-nested matrices.

case_study.R

In this script we illustrate our new procedures and the use of our new functions in a weighted empirical plant-pollinator network. This network was downloaded from the Web of Life database (www.web-of-life.es, network: M_PL_060_17), and originally described by Kaiser-Bunbury [1].

Manual for the new functions

Function nestmodels

Description: **nestmodels** is an R function to calculate matrix nestedness, generate randomized models, and visualize results following the procedure proposed in the main text.

Usage: nestmodels (M, index="wnoda", equi.model=TRUE, prop.model=TRUE, n.model=1000, calc.at=NULL, print.at.each=NULL, wprob=FALSE, wsamp=FALSE, density.plot=TRUE, sampling.plot=TRUE, diag.rm=FALSE, lower.rm=FALSE)

Table 1 - Nestedness metrics implemented in the R function *nestmodels*.

Metric	Code	Reference
Binary		
Temperature	temperature	[2]
Manhattan distance	MD	[3]
NODF	nodf	[4]
β_{NES}	betaNES	[5]
β_{NES} / β_{SOR}	betaNES2	[5]
Binary spectral radius	binSR	[6]
Weighted		
WNODF	wnodf	[7]
WNODA	wnoda	[8]
Spectral radius	SR	[6]

Requirements:

R packages *vegan*, *bipartite*, and *betapart*. All required packages are available at CRAN repository (https://CRAN.R-project.org).

Arguments:

M = an interaction matrix (binary or weighted).

index = the nestedness (or overlap) metric to be applied. See codes in Table 1 for each metric implemented in the function.

equi.model = False or True. Indicates whether you want to generate an equiprobable null model.

prop.model = False or True. Indicates whether you want to generate a proportional model.

n.model = the number of randomized matrices in each model.

calc.at = a vector indicating the different sampling intensities at which nestedness should be calculated on the randomized models. If NULL, nestedness is only calculated at the actual sampling intensity. Only meaningful if total sum instead of connectance is conserved in the null models (weighted indices or binary indices in weighted network with wsamp=T).

print.at.each = the number of null matrices produced for the function to print the progress. If this is not necessary, set to NULL.

wprob = False or True. If a weighted matrix is provided but a binary nestedness index is selected, indicates whether the weighted marginal sums should be used to produce the randomized models. If False and a binary index is selected, probabilities follow the null model 2 of Bascompte *et al.* [9].

wsamp = False or True. If a weighted matrix is provided but a binary nestedness index is selected, indicates whether total sum instead of connectance should be fixed in the randomized matrices.

density.plot = False or True. Indicates whether a density plot for comparison between the observed and the randomized models is produced after the analysis.

sampling.plot = False or True. Indicates whether a plot for comparison between the observed and the randomized models, under different sampling intensities, is produced after the analysis.

diag.rm = False or True. Indicates whether interaction on the diagonal of the matrix should be removed and forbidden in the randomized models. Might be used for unipartite networks without self-

connections. NA's on the diagonal of the input matrix are allowed if diag.rm is True. If True, the matrix must be square.

lower.rm = False or True. Indicates whether the lower triangular of the matrix should be removed and forbidden in the randomized models. Might be used for non-directed unipartite networks. If True, the input matrix must be square and symmetrical.

Outputs:

A list with

$observed = observed nestedness for the input matrix.

$significance = significance of nestedness compared to the randomized models. Significance is calculated through a one tailed one sample Z test.

$summary.equi = mean, standard deviation, and upper and lower limits for 95% of values for the equiprobable model, for each sampling intensity defined in the argument calc.at.

$summary.prop = mean, standard deviation, and upper and lower limits for 95% of values for the proportional model, for each sampling intensity defined in the argument calc.at.

$equimodel = nestedness values for each matrix in the equiprobable model.

$propmodel = nestedness values for each matrix in the proportional model.

$nsampled = sampling intensity (total frequency of interactions) in the input matrix

$parameters = a list with the parameter values applied

Plots

if density.plot = True: a density plot for comparison between the observed and the randomized models. Red line: equiprobable model. Blue line: proportional model based on weighted marginal sums; Yellow line: proportional model based on binary marginal sums (node degrees).

if sampling.plot = True: a plot for comparison between the observed and the randomized models, under different sampling intensities (total frequency of interactions). It can only be produced when

total sum instead of connectance is fixed on random matrices (weighted index applied or wsamp = T). Red line: equiprobable model. Blue line: proportional model based on weighted marginal sums;

Notes:

The code used to calculate Spectral Radius and Manhattan Distance was adapted from the FALCON package developed by Stephen Beckett. FALCON is freely available at GitHub: https://github.com/sjbeckett/FALCON.

Function plot1.nestmodels

Description: **plot1.nestmodels** creates a density plot for comparison between the observed and the randomized models from an output of **nestmodels** function. Identical to the plot produced when the argument density.plot = T in nestmodels.

Usage: plot1.nestmodels (x)

Arguments:

x = an output of **nestmodels** function.

Function plot2.nestmodels

Description: **plot2.nestmodels** creates a plot for comparison between the observed and the randomized models, under different sampling intensities, from an output of **nestmodels** function. Identical to the plot produced when the argument sampling.plot = T in nestmodels.

Usage: plot2.nestmodels (x)

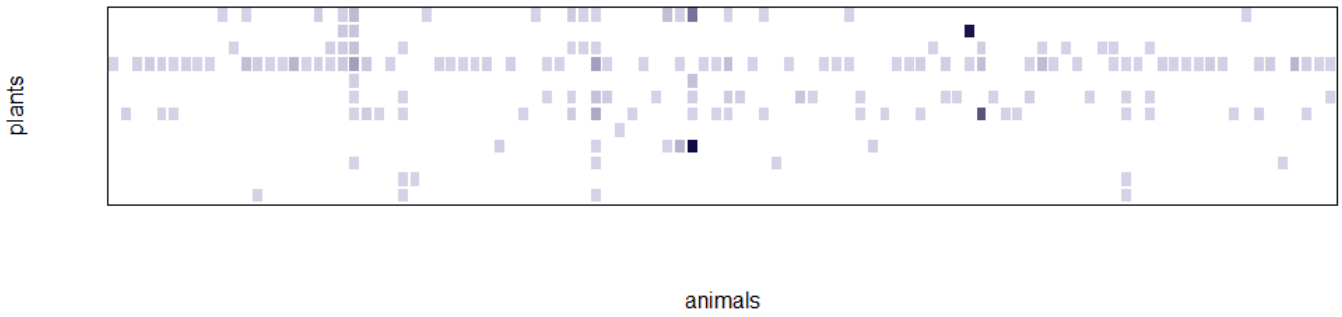
Arguments:

x = an output of **nestmodels** function.

Vignette for nestmodels function

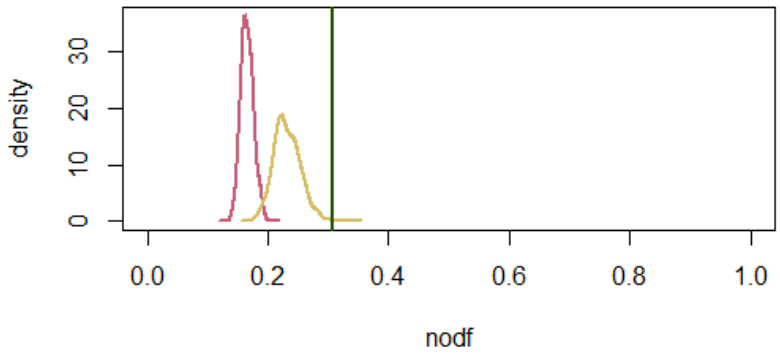
To exemplify the usage of the functions, we will use an animal-plant interaction network available at bipartite package. First, we source the functions. Then, through the function *plotmatrix* we plot the interaction matrix.

```
> source ("functions/nestmodels.R")
> source ("functions/plot1nestmodels.R")
> source ("functions/plot2nestmodels.R")

> library (bipartite)
> net= barrett1987
> plotmatrix (net, binary= F, within_color = "#0e0a44FF", base_color = "#d7d6e9ff", xlab = "animals", ylab= "plants")
```



Let's first apply the new approach using NODF. NODF is a binary nestedness index, so, by default, the function only uses the binary information of the matrix. The proportional model used is the same as the null model 2 of Bascompte *et al.* [9]. With the argument *n.model* we define the number of randomized matrices in each model (= 1000).

```
> NODFtest= nestmodels (net, index= "nodf", n.model = 1000, equi.model = TRUE, prop.model = TRUE, density.plot =TRUE)
```



As density.plot is TRUE, after running, the function automatically plots the comparison between the observed value (green vertical line), the equiprobable model (red curve), and the proportional model based on binary marginal sums (yellow curve). The object *NODFtest* includes, among other information, the observed nestedness value, and the significance obtained by one tailed Z tests against each model distribution.

```
> NODFtest$observed
```

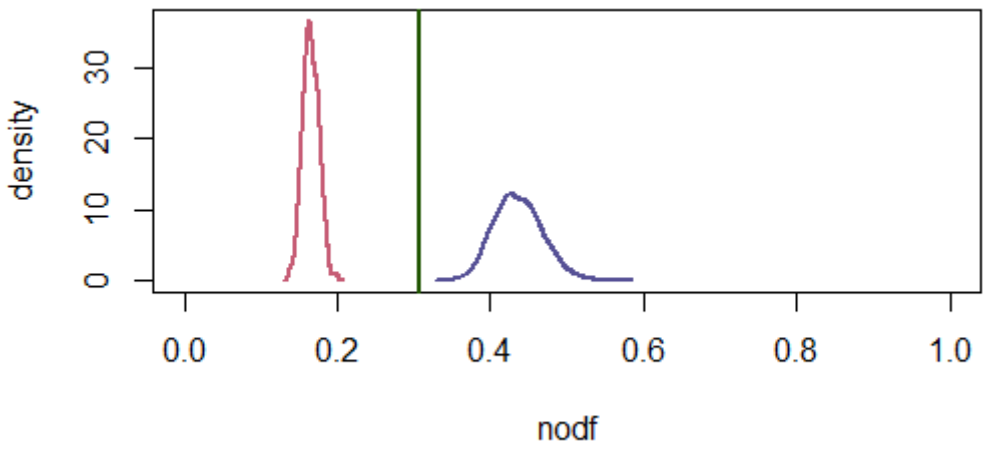
```
[1] 0.3078467
```

```
> NODFtest$significance
```

```
$pvalue.equi
[1] 5.086364e-40

$Zscore.equi
[1] 13.18885

$pvalue.prop
[1] 0.0002753847

$Zscore.prop
[1] 3.454767
```

In our example, the observed nestedness is higher than the expected by both models. As we discussed in the article, the use of binary marginal sums to build the proportional model usually results in underestimation. Here, contrary to most of the cases when binary indices are applied, we have the weighted information. So, now, we set the argument *wprob = TRUE*, so that the model is build based on weighted marginal sums.

```
> NODFtest2= nestmodels (net, index= "nodf", n.model = 1000, equi.model = TRUE, prop.model = TRUE, density.plot = TRUE, wprob = TRUE)
```




```
> NODFtest2$observed
```

[1] 0.3078467

```
> NODFtest2$significance
```

$pvalue.equi
[1] 1.723955e-40

$Zscore.equi
[1] 13.27017

$pvalue.prop
[1] 2.526637e-05

$Zscore.prop
[1] -4.053149

Now, the observed value is higher compared to the distribution of nestedness for the equiprobable model, but lower than the proportional model.

Next, we will use a weighted nestedness index: WNODA. Weighted analyses usually demand higher computation and take longer times than binary analyses. We set the argument *print.at.each = 100*, so that the program prompt the number of randomized matrices produced while running.

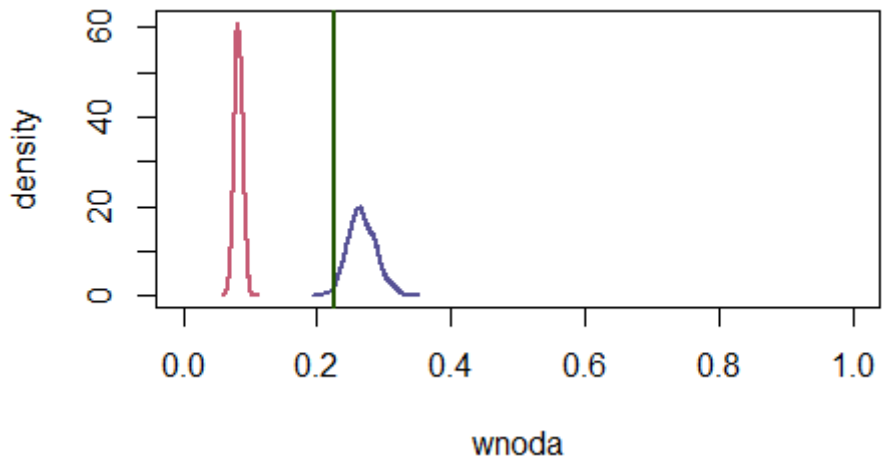
```
> WNODAtest = nestmodels (net, index="wnoda", n.model = 1000, equi.model = TRUE, prop.model = TRUE, wprob= TRUE, print.at.each = 100, density.plot = FALSE)
```

[1] 100
[1] 200

```
[1] 300
[1] 400
[1] 500
[1] 600
[1] 700
[1] 800
[1] 900
[1] 1000
```

This time, the density plot was not automatically produced, because we set *density.plot = FALSE*. If we want to produce the plot, we can always use the function *plot1.nestmodels* with the output of *nestmodels* as argument.

```
> plot1.nestmodels (WNODAtest)
```




```
> WNODAtest$observed
```

```
[1] 0.2258285
```

```
> WNODAtest$significance
```

```
$pvalue.equi
[1] 2.033186e-112

$Zscore.equi
[1] 22.50091

$pvalue.prop
[1] 0.02130943

$Zscore.prop
```

```
[1] -2.027426
```

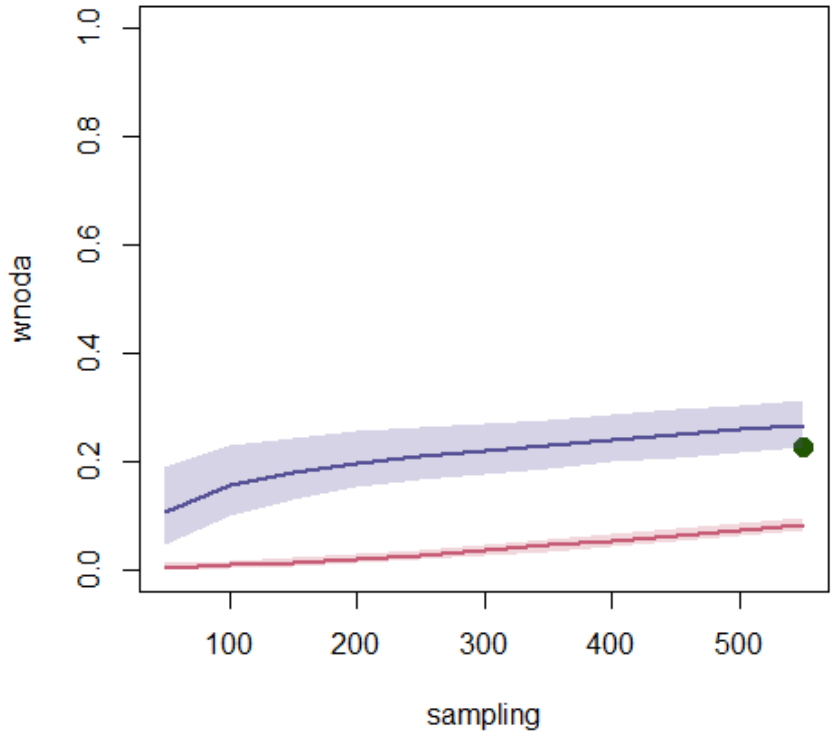
Here, we see that, when a weighted index is applied, nestedness of the network is much higher than the nestedness of the randomized matrices in the equiprobable model, and significantly lower, but close, to the expectations in the proportional model.

These results mean that the frequency of interactions between each animal-plant pair is highly driven by the differences in marginal sums (the total frequency of interactions of each animal and each plant). However, there are some small preferences shaping the network, so that it is not entirely nested.

For last, the *nestmodels* function allow us to inspect the effect of sampling (total frequency of interaction) on the separation between the distributions of the equiprobable and the proportional model. This approach might be useful to check whether the actual sampling is good enough to evaluate the nestedness of the observed network.

With the argument *calc.at*, we define intermediary samplings, during the distribution of interaction in the randomized matrices, to calculate nestedness. If *sampling.plot* is TRUE, at the end of the analysis, the function produces a plot with the mean and the confidence interval at each sampling level for each randomized model. Be aware, though, that this analysis may take much longer than the previous.

```
> WNODAtest = nestmodels (net, index="wnoda", n.model = 1000, equi.model = TRUE, prop.model = TRUE, wprob= TRUE, density.plot = FALSE, calc.at = c(50,100,150,200,250,300,350,400,450,500,550), sampling.plot = TRUE)
```



Here we see that the equiprobable and the proportional model distribution are highly distinct, even in much lower sampling effort than the total frequency of interaction represented in the network. Thus, we have strong evidence that the analysis was appropriate and that the results are not likely to modify with a higher sampling effort.

References:

1. Kaiser-Bunbury CN, Muff S, Memmott J, Müller 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. (doi:10.1111/j.1461-0248.2009.01437.x)

2. Rodriguez-Girones MA, Santamaria L. 2006 A new algorithm to calculate the nestedness temperature of presence-absence matrices. *J. Biogeogr.* **33**, 924–935. (doi:10.1111/j.1365-2699.2006.01444.x)

3. Corso G, Britton NF. 2012 Nestedness and τ-temperature in ecological networks. *Ecol. Complex.* **11**, 137–143. (doi:10.1016/j.ecocom.2012.05.003)

4. Almeida-Neto M, Guimarães P, Guimarães PR, Loyola RD, Ulrich W. 2008 A consistent metric for nestedness analysis in ecological systems: reconciling concept and measurement. *Oikos* **117**, 1227–1239. (doi:10.1111/j.0030-1299.2008.16644.x)

5. Baselga A. 2010 Partitioning the turnover and nestedness components of beta diversity. *Glob. Ecol. Biogeogr.* **19**, 134–143. (doi:10.1111/j.1466-8238.2009.00490.x)

6. Staniczenko PPA, Kopp JC, Allesina S. 2013 The ghost of nestedness in ecological networks. *Nat. Commun.* **4**, 1391. (doi:10.1038/ncomms2422)

7. Almeida-Neto M, Ulrich W. 2011 A straightforward computational approach for measuring nestedness using quantitative matrices. *Environ. Model. Softw.* **26**, 173–178. (doi:10.1016/j.envsoft.2010.08.003)

8. Pinheiro RBP, Félix GMF, Dormann CF, Marco AR. 2018 A new model explaining the origin of different topologies in interaction networks. *bioRxiv* , 1–30. (doi:10.1101/362871)

9. Bascompte J, Jordano P, Melian CJ, Olesen JM. 2003 The nested assembly of plant-animal mutualistic networks. *Proc. Natl. Acad. Sci.* **100**, 9383–9387. (doi:10.1073/pnas.1633576100)