CompoundTest

Gabriel Felix, Rafael Pinheiro & Marco Mello

10/09/2020

How to test for a compound topology using the restricted null model

Ecological Synthesis Lab (SintECO)

See README for further info.

Follow the instructions in the sequence proposed here and will be able to run a compound topology test.

You can also replace the example network (net1.txt) with your own network. Just follow the same formatting and remember to keep the names consistent.

Summary

1. PREPARING THE DATA

Set the working directory:

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
```

Delete all previous objects:

```
rm(list= ls())
```

Clear the console:

```
cat("\014")
```

Load the required packages and functions:

```
library(bipartite)
```

```
## Loading required package: vegan

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.5-6

## Loading required package: sna

## Loading required package: statnet.common

##
## Attaching package: 'statnet.common'

## The following object is masked from 'package:base':
##
##     order

## Loading required package: network

## network: Classes for Relational Data
## Version 1.16.0 created on 2019-11-30.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##                     Mark S. Handcock, University of California -- Los Angeles
##                     David R. Hunter, Penn State University
##                     Martina Morris, University of Washington
##                     Skye Bender-deMoll, University of Washington
##  For citation information, type citation("network").
##  Type help("network-package") to get started.

## sna: Tools for Social Network Analysis
## Version 2.5 created on 2019-12-09.
## copyright (c) 2005, Carter T. Butts, University of California-Irvine
##  For citation information, type citation("sna").
##  Type help(package="sna") to get started.

##  This is bipartite 2.15.
##  For latest changes see versionlog in ?"bipartite-package". For citation see: citation("bipartite").
##  Have a nice time plotting and analysing two-mode networks.

##
## Attaching package: 'bipartite'

## The following object is masked from 'package:vegan':
##
##     nullmodel
```

```
source("RestNullModel.R")
source("PosteriorProb.R")
```

Load the data:

```
data<-as.matrix(read.table("net1.txt", head=TRUE))
```

Taka a look at the data:

```
head(data)
```

```
##            consumer_1 consumer_2 consumer_3 consumer_4 consumer_5 consumer_6
## resource_1          3          3          6          2          2          1
## resource_2          7          1          4          2          1          6
## resource_3          1          1          2          1          3          6
## resource_4          2          2          4          2          0          0
## resource_5          3          1          0          0          1          2
## resource_6          1          2          1          1          0          2
##            consumer_7 consumer_8 consumer_9 consumer_10 consumer_11 consumer_12
## resource_1          1          5          1           0           0           0
## resource_2          0          0          0           1           0           0
## resource_3          1          1          1           0           0           0
## resource_4          0          0          0           0           0           0
## resource_5          1          0          0           0           0           0
## resource_6          0          0          0           0           0           0
##            consumer_13 consumer_14 consumer_15 consumer_16 consumer_17
## resource_1           0           0           0           0           0
## resource_2           0           0           0           0           0
## resource_3           0           0           0           0           0
## resource_4           0           0           0           0           0
## resource_5           0           0           0           0           0
## resource_6           1           0           0           0           0
##            consumer_18 consumer_19 consumer_20 consumer_21 consumer_22
## resource_1           0           0           0           0           0
## resource_2           0           0           0           0           0
## resource_3           0           0           0           0           0
## resource_4           0           0           0           1           0
## resource_5           0           0           1           0           0
## resource_6           0           0           0           0           1
##            consumer_23 consumer_24 consumer_25 consumer_26 consumer_27
## resource_1           0           0           0           1           0
## resource_2           0           0           0           0           0
## resource_3           0           0           0           0           0
## resource_4           0           0           0           0           0
## resource_5           0           0           0           0           0
## resource_6           0           0           0           0           0
##            consumer_28 consumer_29 consumer_30
## resource_1           0           0           0
## resource_2           0           0           0
## resource_3           1           0           0
## resource_4           0           0           0
## resource_5           0           1           0
## resource_6           0           0           0
```

Inspect the data:

```
dim(data)
```

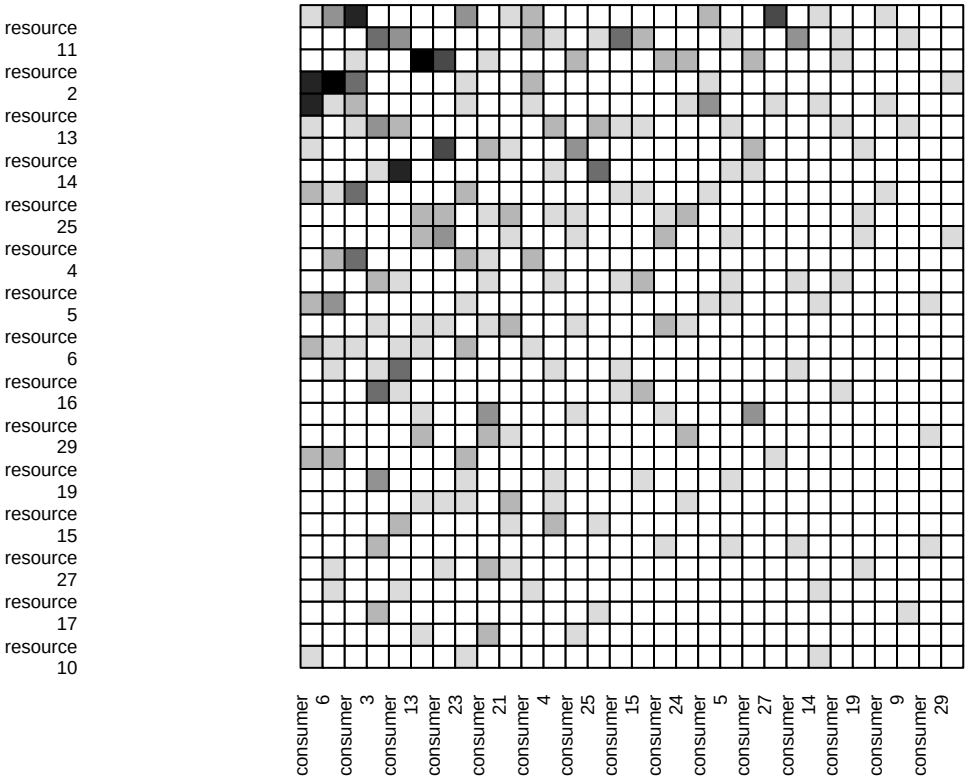
```
## [1] 30 30
```

```
class(data)
```

```
## [1] "matrix" "array"
```

Visualize the raw matrix:

```
visweb(data)
```



2. MODULARITY ANALYSIS

Compute modularity:

```
Mod <- bipartite::computeModules(data)
```

Recover the partitions:

```
Part <- bipartite::module2constraints(Mod)
row.Part <- Part[1:nrow(data)]
col.Part <- Part[(nrow(data)+1):(nrow(data)+ncol(data))]
```

Test for the significance of modularity with a Monte Carlo procedure:

Generate randomized matrices:

```
nulls <- nullmodel(data, N=9, method="r2d")
```

Calculate the modularity of the randomized matrices:

```
mod.nulls <- sapply(nulls, computeModules)
like.nulls <- sapply(mod.nulls, function(x) x@likelihood)
```

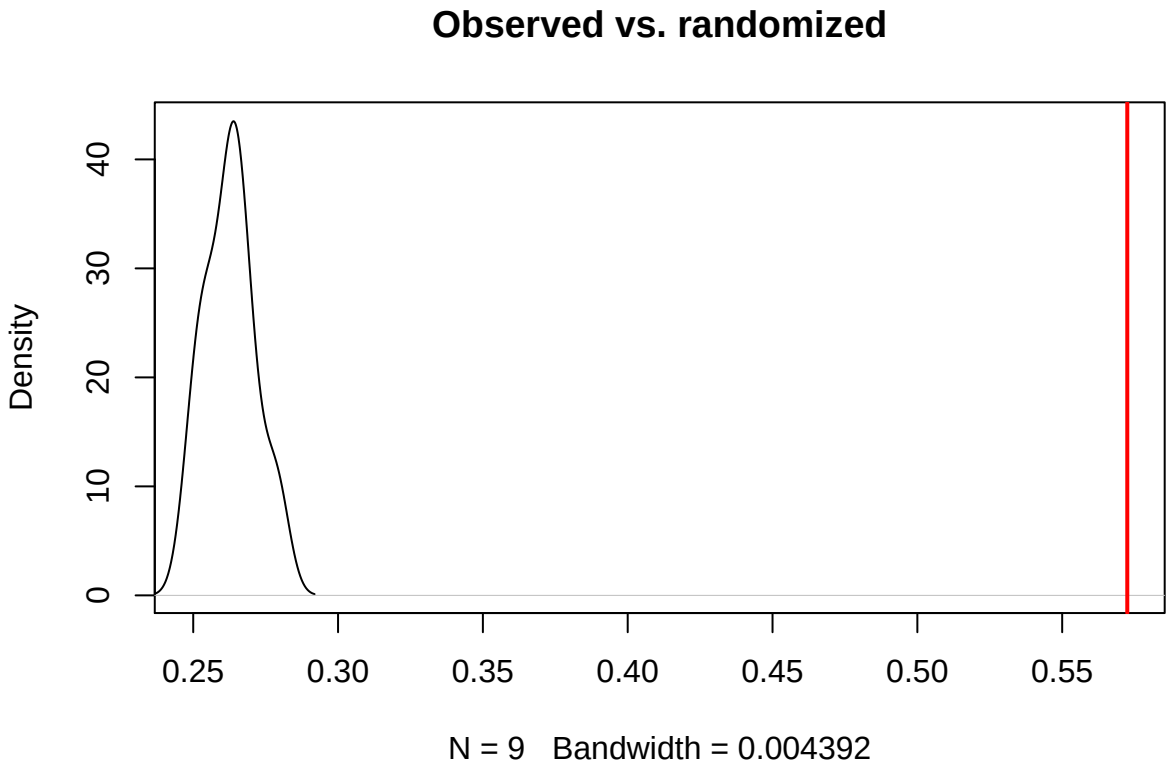
Calculate the z-score of the randomized distribution:

```
(z <- (Mod@likelihood - mean(like.nulls))/sd(like.nulls))
```

```
## [1] 34.74051
```

Plot the observed modularity value against the distribution of randomized values:

```
plot(density(like.nulls), xlim=c(min((Mod@likelihood), min(like.nulls)), max((Mod@likelihood), max(like
     main="Observed vs. randomized")
abline(v=(Mod@likelihood), col="red", lwd=2)
```



Estimate the P-value:

```
mean(like.nulls)
```

```
## [1] 0.262731
```

```r
sd(like.nulls)
```

```
## [1] 0.008915926
```

```r
Mod@likelihood
```

```
## [1] 0.5724748
```

```r
praw <- sum(like.nulls>(Mod@likelihood)) / length(like.nulls)
ifelse(praw > 0.5, 1-praw, praw)
```

```
## [1] 0
```

3. NESTEDNESS ANALYSIS

Calculate the desired nestedness metric (here WNODA) for the original network:

```r
obs <- unlist(bipartite::nest.smdm(x = data,
                                   constraints = Part, #Input the modular structured recovered from ste
                                   weighted = T, #By considering the edge weights, you are choosing WNO
                                   decreasing = "abund"))
```

Check the scores:

```r
obs
```

```
##        WNODArow        WNODAcol     WNODAmatrix    WNODA_SM_row    WNODA_DM_row
##       10.040578       11.261474       10.651026       25.255251        3.193975
##    WNODA_SM_col    WNODA_DM_col WNODA_SM_matrix WNODA_DM_matrix
##       28.492408        3.423991       26.879802        3.308791
```

4. RESTRICTED NULL MODEL ANALYSIS

Calculate constrained interaction probabilities considering the network's modular structure:

```r
Pij <- PosteriorProb(M = data,
                     R.partitions = row.Part, C.partitions = col.Part, #Input the modular structured re
                     Prior.Pij = "degreeprob", #Choose the null model
                     Conditional.level = "modules") #Choose the kind of constraints
```

#Take a look at the probabilities:

```r
head(Pij)
```

```
##             [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
## [1,] 0.013970158 0.010325769 0.013970158 0.006681380 0.004859185 0.014577556
## [2,] 0.012293739 0.009086677 0.012293739 0.005879614 0.004276083 0.012828250
## [3,] 0.010058514 0.007434554 0.010058514 0.004810594 0.003498614 0.010495841
```

```
## [4,] 0.006146870 0.004543338 0.006146870 0.002939807 0.002138042 0.006414125
## [5,] 0.005588063 0.004130308 0.005588063 0.002672552 0.001943674 0.005831023
## [6,] 0.005029257 0.003717277 0.005029257 0.002405297 0.001749307 0.005247920
##             [,7]        [,8]         [,9]        [,10]        [,11]
## [1,] 0.003036991 0.004251787 0.0018221945 6.252905e-05 0.0007190841
## [2,] 0.002672552 0.003741573 0.0016035312 5.502556e-05 0.0006327940
## [3,] 0.002186633 0.003061287 0.0013119801 4.502092e-05 0.0005177405
## [4,] 0.001336276 0.001870786 0.0008017656 2.751278e-05 0.0003163970
## [5,] 0.001214796 0.001700715 0.0007288778 2.501162e-05 0.0002876336
## [6,] 0.001093317 0.001530643 0.0006559900 2.251046e-05 0.0002588703
##             [,12]        [,13]        [,14]        [,15]        [,16]
## [1,] 0.0002813807 0.0006565550 1.875872e-04 0.0002813807 0.0003439098
## [2,] 0.0002476150 0.0005777684 1.650767e-04 0.0002476150 0.0003026406
## [3,] 0.0002025941 0.0004727196 1.350627e-04 0.0002025941 0.0002476150
## [4,] 0.0001238075 0.0002888842 8.253835e-05 0.0001238075 0.0001513203
## [5,] 0.0001125523 0.0002626220 7.503486e-05 0.0001125523 0.0001375639
## [6,] 0.0001012971 0.0002363598 6.753137e-05 0.0001012971 0.0001238075
##             [,17]        [,18]        [,19]        [,20]        [,21]
## [1,] 0.0002813807 9.379358e-05 1.563226e-04 2.501162e-04 0.0005002324
## [2,] 0.0002476150 8.253835e-05 1.375639e-04 2.201023e-04 0.0004402045
## [3,] 0.0002025941 6.753137e-05 1.125523e-04 1.800837e-04 0.0003601673
## [4,] 0.0001238075 4.126917e-05 6.878196e-05 1.100511e-04 0.0002201023
## [5,] 0.0001125523 3.751743e-05 6.252905e-05 1.000465e-04 0.0002000930
## [6,] 0.0001012971 3.376569e-05 5.627615e-05 9.004183e-05 0.0001800837
##             [,22]        [,23]        [,24]        [,25]        [,26]
## [1,] 0.0005627615 0.0005627615 0.0002813807 0.0003126453 0.0003751743
## [2,] 0.0004952301 0.0004952301 0.0002476150 0.0002751278 0.0003301534
## [3,] 0.0004051882 0.0004051882 0.0002025941 0.0002251046 0.0002701255
## [4,] 0.0002476150 0.0002476150 0.0001238075 0.0001375639 0.0001650767
## [5,] 0.0002251046 0.0002251046 0.0001125523 0.0001250581 0.0001500697
## [6,] 0.0002025941 0.0002025941 0.0001012971 0.0001125523 0.0001350627
##             [,27]        [,28]        [,29]        [,30]
## [1,] 2.501162e-04 0.0002813807 9.379358e-05 1.250581e-04
## [2,] 2.201023e-04 0.0002476150 8.253835e-05 1.100511e-04
## [3,] 1.800837e-04 0.0002025941 6.753137e-05 9.004183e-05
## [4,] 1.100511e-04 0.0001238075 4.126917e-05 5.502556e-05
## [5,] 1.000465e-04 0.0001125523 3.751743e-05 5.002324e-05
## [6,] 9.004183e-05 0.0001012971 3.376569e-05 4.502092e-05
```

Generate randomized networks with the null model of your choice, considering the interaction probabilities calculated before:

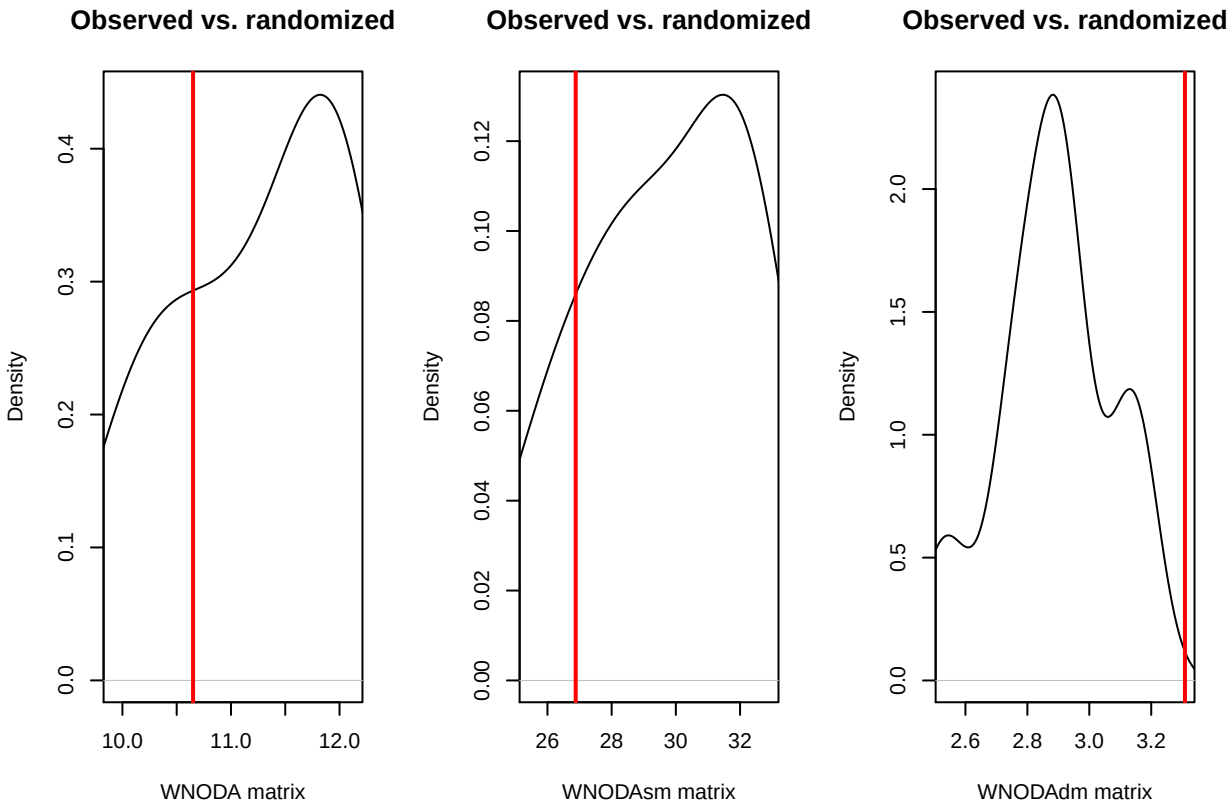
```
nulls <- RestNullModel(M = data,
                       Pij.Prob = Pij, #Recover the probabilities calculated in the previous command
                       Numbernulls = 9, #This step may take long, so start experimenting with low value
                       Print.null = F,
                       allow.degeneration = F, #Choose whether you allow orphan rows and columns to be
                       return.nonrm.species = F,
                       connectance = T, byarea = T,
                       R.partitions = row.Part, C.partitions = col.Part)
```

Calculate the same nestedness metric for all randomized networks:

```
null <- sapply(nulls, function(x) bipartite::nest.smdm(x = x, constraints = Part, weighted = T, decreas
WNODA.null <- unlist(null[3,])
WNODAsm.null <- unlist(null[8,])
WNODAdm.null <- unlist(null[9,])
```

Plot the observed nestedness value against the distribution of randomized values:

```
par(mfrow = c(1,3))
plot(density(WNODA.null), xlim=c(min(obs[3], min(WNODA.null)), max(obs[3], max(WNODA.null))),
     main="Observed vs. randomized", xlab = "WNODA matrix")
abline(v=obs[3], col="red", lwd=2)
plot(density(WNODAsm.null), xlim=c(min(obs[8], min(WNODAsm.null)), max(obs[8], max(WNODAsm.null))),
     main="Observed vs. randomized", xlab = "WNODAsm matrix")
abline(v=obs[8], col="red", lwd=2)
plot(density(WNODAdm.null), xlim=c(min(obs[9], min(WNODAdm.null)), max(obs[9], max(WNODAdm.null))),
     main="Observed vs. randomized", xlab = "WNODAdm matrix")
abline(v=obs[9], col="red", lwd=2)
```



Estimate the P-values:

Nestedness in th entire network:

```
praw.WNODA <- sum(WNODA.null>obs[3]) / length(WNODA.null)
p.WNODA <- ifelse(praw.WNODA > 0.5, 1- praw.WNODA, praw.WNODA)    # P-value
p.WNODA
```

```
## [1] 0.3333333
```

Nestedness within the modules:

```
praw.WNODAsm <- sum(WNODAsm.null>obs[8]) / length(WNODAsm.null)
p.WNODAsm <- ifelse(praw.WNODAsm > 0.5, 1- praw.WNODAsm, praw.WNODAsm)    # P-value
p.WNODAsm
```

```
## [1] 0.1111111
```

Nestedness between the modules:

```
praw.WNODAdm <- sum(WNODAdm.null>obs[9]) / length(WNODAdm.null)
p.WNODAdm <- ifelse(praw.WNODAdm > 0.5, 1- praw.WNODAdm, praw.WNODAdm)    # P-value
p.WNODAdm
```

```
## [1] 0
```

5. PLOTTING THE NETWORK

Sort the matrix in a way that facilitates visualizing the compound topology:

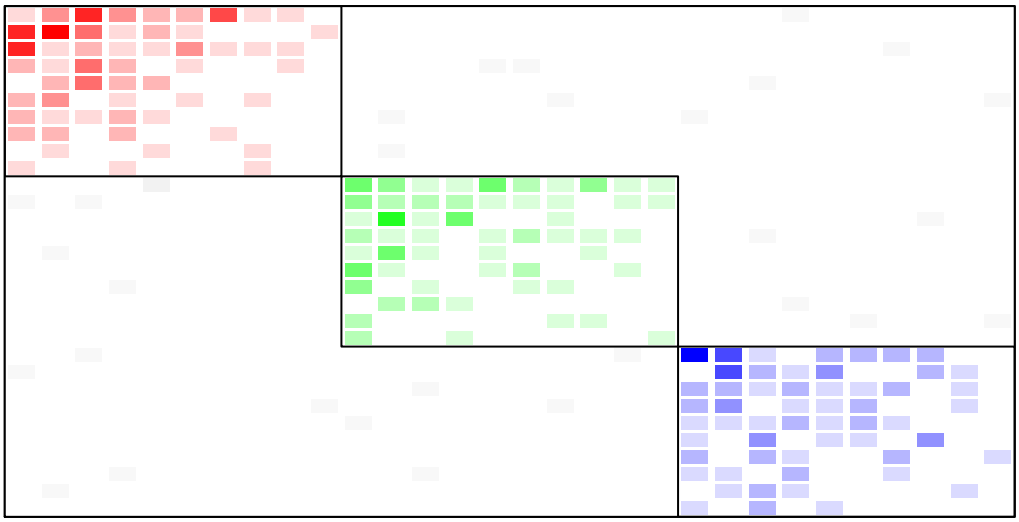
```
data.comp <- bipartite::sortmatrix(matrix = data, topology = "compound", sort_by = "weights", row_parti
```

Assign colors for the modules:

```
modcol <- rainbow((length(unique(Part))), alpha=1)
```

Plot the matrix:

```
par(mfrow = c(1,1))
plotmatrix(data.comp$matrix,
           row_partitions = data.comp$row_partitions,
           col_partitions = data.comp$col_partitions,
           border = T,
           binary = F,
           modules_colors = modcol,
           within_color = modcol,
           between_color = "lightgrey")
```



6. SOURCE STUDIES

If you want to understand the background of those new analyses before using them, read the following studies. The first three paved the ground for the analysis of compound topologies developed later by our lab.

1. Lewinsohn, T. M., P. Inácio Prado, P. Jordano, J. Bascompte, and J. M. Olesen. 2006. Structure in plant-animal interaction assemblages. Oikos 113: 174–184. Available at: http://doi.wiley.com/10.1111/j.0030-1299.2006.14583.x.

2. Bezerra, E. L. S., I. C. Machado, and M. A. R. Mello. 2009. Pollination networks of oil-flowers: a tiny world within the smallest of all worlds. J. Anim. Ecol. 78: 1096–1101. Available at: http://www.ncbi.nlm.nih.gov/pubmed/19515098.

3. Flores, C. O., S. Valverde, and J. S. Weitz. 2013. Multi-scale structure and geographic drivers of cross-infection within marine bacteria and phages. ISME J. 7: 520–532. Available at: http://www.nature.com/doifinder/10.1038/ismej.2012.135 [Accessed June 9, 2016].

4. Pinheiro, R. B. P., G. M. F. Félix, A. V Chaves, G. A. Lacorte, F. R. Santos, É. M. Braga, and M. A. R. Mello. 2016. Trade-offs and resource breadth processes as drivers of performance and specificity in a host–parasite system: a new integrative hypothesis. Int. J. Parasitol. 46: 115–121. Available at: http://www.sciencedirect.com/science/article/pii/S0020751915002933.

5. Felix, G. M., R. B. P. Pinheiro, R. Poulin, B. R. Krasnov, and M. A. R. Mello. 2017. The compound topology of a continent-wide interaction network explained by an integrative hypothesis of specialization. bioRxiv 236687. Available at: https://doi.org/10.1101/236687.

6. Pinheiro, R. B. P. 2019. As topologias de redes de interações ecológicas e suas origens. PhD Thesis, Federal Univesity of Minas Gerais. URL: http://hdl.handle.net/1843/33333.

7. Pinheiro, R. B. P., G. M. F. Felix, C. F. Dormann, and M. A. R. Mello. 2019. A new model explaining the origin of different topologies in interaction networks. Ecology 100: e02796. Available at: https://doi.org/10.1002/ecy.2796.

8. Mello, M. A. R., G. M. Felix, R. B. P. Pinheiro, R. L. Muylaert, C. Geiselman, S. E. Santana, M. Tschapka, N. Lotfi, F. A. Rodrigues, and R. D. Stevens. 2019. Insights into the assembly rules of a continent-wide multilayer network. Nat. Ecol. Evol. 3: 1525–1532. Available at: https://doi.org/10.1038/s41559-019-1002-3.