Case study 2 - permutations of networks

Timothée Poisot

11th April 2018

```
using EcologicalNetwork
```

The purpose of this case study is to illustrate how we can use the package to perform significance testing. We will see how we can compare the measured value of nestedness to random values derived from network permutations.

We start by loading a network, in this case the one by Fonseca & Ganade (1996). This is a bipartite network.

```
N = fonseca_ganade_1996()
```

We will start by getting rid of the quantitative information. This is done by converting our network to another type:

```
M = convert(BipartiteNetwork, N)
```

We will now measure the nestedness of the network, using the η measure:

```
n0 =  (M)
```

This returns a dictionary with a value for the rows, the columns, and the entire network.

```
n0
```

```
Dict{Symbol,Float64} with 3 entries:
  :rows    => 0.187648
  :network => 0.168149
  :columns => 0.148649
```

At this point, we will need to generate a few random matrices to test our empirical measure against. We will focus on the Type II null model, in which the probability of an interaction between two species is proportional to their relative degrees. We will call this probabilistic template `T`:

```
T = null2(M)
```

This function will return another network, this time a probabilistic one.

```
typeof(T)
```

```
EcologicalNetwork.BipartiteProbabilisticNetwork{Float64,String}
```

We can draw a random sample based on this template:

```
rand(T)
```

```
EcologicalNetwork.BipartiteNetwork{String}(Bool[false false … false false
; false true … false false; … ; false false … false false; false fals
e … false false], String["Camponotus balzanii", "Azteca alfari", "Azteca
isthmica", "Azteca aff. Isthmica", "Allomerus D", "Allomerus prancei", "All
omerus aff. Octoarticulata", "Solenops A", "Allomerus auripunctata", "Crema
togaster B"  …  "Crematogaster A", "Azteca TO", "Crematogaster C", "Aztec
a schummani", "Pseudomyrmex nigrescens", "Pseudomyrmex concolor", "Azteca D
", "Azteca polymorpha", "Crematogaster E", "Azteca Q"], String["Cecropia pu
rpuracens", "Cecropia concolor", "Cecropia distachya", "Cecropia ficifolia"
, "Pouruma heterophylla", "Hirtella myrmecophila", "Hirtella physophora", "
Duroia saccifera", "Cordia nodosa", "Cordia aff. Nodosa", "Tococa bullifera
", "Maieta guianensis", "Maieta poeppiggi", "Tachigali polyphylla", "Tachig
ali myrmecophila", "Amaioua aff. Guianensis"])
```

Of course, we would like to generate a larger sample – so we can draw many replicates at once:

```
random_draws = rand(T, 10000)
```

Some of these networks may be *degenerate*, *i.e.* they have species without interactions. It is safer to remove them from our sample:

```
valid_draws = filter(x -> !isdegenerate(x), random_draws)
```

We are left with a smaller number of networks, but all of them have species with at least one interaction. At this point, we can measure the nestedness on all of these networks:

```
n_prime = map(x ->  (x)[:network], valid_draws)
```

We can now figure out the quantile of the original network in the random distribution – in permutation testing, if this is lower than 0.05 or higher than 0.95, the network is significantly different than expected by chance under the specified null model:

```
quantile(n_prime, n0[:network])
```

```
0.19104128010438878
```

We can also express much of this analysis as a single pipeline:

```
n_prime = N |>
    # Step 1 - remove interaction strength
    n -> convert(BipartiteNetwork, n) |>
    # Step 2 - generate the probability template
    null2 |>
    # Step 3 - draw random networks
    n -> rand(n, 10000) |>
    # Step 4 - remove degenerate networks
    n -> filter(x -> !isdegenerate(x), n) .|>
    # Step 5 - measure the nestedness
    n ->  (n)[:network]
```

This notation is more compact, and can help understand the flow of the analysis better.