Appendix S1 – Instructions for using the code for the simulations

Basic function: `netcascade`

We provide a function written in *R*, `netcascade`, that simulates a single primary extinction and possibly an extinction cascade resulting from it under the Stochastic Coextinction Model:

```
netcascade(imatrix, ranim, rplants, deadPlants = NULL,
deadAnimals = NULL, targetGuild, target, return.matrix = F)
```

Arguments

`imatrix` – an object of class `matrix` describing plant-animal mutualistic interactions and used to calculate dependence of species on each of their partners. Either quantitave or binary matrices can be used. `netcascade` assumes that animal species are represented by the rows and plant species are represented by the columns, and requires that at least three species are contained in the matrix.

`ranim` – a numeric vector containing *R* values (ranging from 0 to 1) for the animal species in the matrix. Vector `ranim` cannot be recycled, so its number of elements must be equal to the number of rows in `imatrix`.

`rplants` – similar to `ranim`, but specifying values for plant species in the columns of the interaction matrix.

`deadAnimals`- An integer vector containing row numbers of animal species which, although represented in the interaction matrix, are considered extinct. When `netcascade` is applied sequentially to an initial interaction matrix, it may be important to separate those species that have suffered extinction in previous simulated cascades from those that suffered extinction in the current simulation. In topological models, that might be done by checking which species have zero row sums before the current simulation occurs. However, under the stochastic coextinction model on which `netcascade` is based, some species may be alive even if they have zero interactions. `netcascade` uses this argument to distinguish "dead" species with zero interactions from "living" species with zero interactions. It should NOT be used to specify species with interactions in the matrix but which are considered extinct. If the user considers any species in the original interaction matrix to be extinct, it should remove its corresponding row from the matrix before using `netcascade`, or else set its corresponding row to zeros and inform the row number in `deadAnimals`.

`deadPlants` – similar to `deadAnimals`, but specifies column numbers of plant species.

`targetGuild` – either "animal" or "plant". Specifies the guild from which the target species for primary extinction will be taken.

`target` – must be either a single number, specifying the target species for the primary extinction by its row (if `targetGuild == "animal"`) or column (if

`targetGuild == "plant"`); or a numeric vector ascribing relative risks of primary extinction for all species in the chosen guild.

`return.matrix` – see *Output* below.

Output

By default, `netcascade` returns a list containing 1) a table showing the number of extinctions in each degree of the extinction cascade, 2) a table containing the row numbers of animal species that have gone extinct, and their corresponding extinction degree, and 3) a table containing the column numbers of plant species that have gone extinct, and their corresponding extinction degree.

If `return.matrix` is `TRUE`, however, `netcascade` returns a list containing 1) the post-simulation interaction matrix, where rows and columns corresponding to extinct species have been set to zero (note, however, that having zero interactions does not imply that a species has suffered extinction); 2) a vector containing the row numbers of animal species that suffered extinction and 3) a vector containing the column numbers of plant species that suffered extinction. The post-simulation interaction matrix can be used as a starting point for an additional application of `netcascade`. In that case, the user may use the rows and columns numbers of extinct species as values for the `deadAnimals` and `deadPlants` arguments in the next use of `netcascade`.

Additional functions

In addition to `netcascade`, we provide two functions that perform replicated simulations on a fixed interaction matrix in order to construct frequency distributions for the total number of extinctions following a primary extinction (`coextNumber`) and for the degree of the extinction cascade (`coextDeg`). Those functions are less general than `netcascade` and were constructed having in mind the specific analyses in the present paper. Both of them randomly choose a species as a target for primary extinction (all species have the same probability of being chosen). Both assume that, in each simulation, all species have the same intrinsic dependence on the mutualism (*R* value), but the *R* value varies across simulations within the interval specified by `rlow` and `rup`.

```
coextNumber(imatrix, rlow, rup, nsims)
coextDeg(imatrix, rlow, rup, nsims)
```

Arguments

`imatrix` – the same as for `netcascade`.

`rlow` – a number between 0 and 1 specifying the lower limit of the interval from which the community-wide *R* value is to be uniformly sampled.

`rlow` – similar do `rlow` but specifies the upper limit of the interval.

`nsims` – the number of simulations to be performed

Output

`coextNumber` returns a vector with `nsims` elements, where each element is the total number of extinctions resulting from a primary extinction simulated according to `netcascade`. A similar vector where each element contains the degree of the extinction cascade is returned by `coextDeg`.