README – Code for the IHS model

The code to run the model based on the integrative hypothesis of specialization consists of 6 R files (R Core Team 2017). Each file contains the code for some step of the simulation. All files should be placed in the same folder, and the folder needs to be set as the working directory for R before running the model.

The code in "IHS_model.R" runs once the IHS model. If you want to perform one simulation, just set the parameter values (on the heading of the script) and run the code.

However, if you want to run a set of simulations, it is easier to use the script "simulations.R". In this case you have to define the parameter values for each simulation in a separate table, and then through "simulations.R" you can run all at once (see details below).

Here follows a description of each file and its parameters:

availabilities.R

Description

Creates a vector with the availabilities of the resource species.

Parameters

Sres Resource species richness.

k_method "all100": availability of all resource species is 100.

"rnorm200-50": availability of each resource species is a randomly drawn value from a normal distribution with mean = 200 and standard deviation = 50.

Outputs

k A vector with the availabilities of the resource species.

initial_matrix.R

Description

Creates the initial match matrix.

Parameters

Scon Consumer species richness.

Sres Resource species richness.

initial_matrix

"all0": all consumers score 0 (zero) in the match with all resources.

"all1": all consumers score 1 (one) in the match with all resources.

"rnorm": the match between each consumer and each resource is randomly drawn from a normal distribution with mean = 1 and standard deviation = 1.

Outputs

match00 A matrix with the initial match between each consumer species (rows) and each resource species (columns).

dissimilarity_matrix.R

Description

Creates the resource species pairwise dissimilarity matrix.

Parameters

Sres Resource species richness.

nclust Number of clusters in the structure of dissimilarities between resource species.

clustprob This parameter defines the probability of each resource species being assigned to each cluster, and provides a probabilistic way to unbalance cluster sizes. Parameter values need to be the probabilities for each cluster separated by "-". e.g.: "0.25-0.25-0.5" for 3 clusters, the latter having twice the probability of including each resource species than the others. If probabilities do not sum to 1, they are coerced to this value (by dividing each value by the sum of all values).

maxdis The maximum dissimilarity in the generated matrix. After dissimilarities are calculated based on simulated dimensions, the maximum value is set to *maxdis* and all other values are proportionally recalculated.

supdim Defines the overlap of the range in each dimension for resource species of different clusters. When *supdim* = 1, clusters are highly discontinuous, since there is no overlap in the values of resources of different clusters in any dimension. On the other side, higher values of *supdim* results in less separated clusters.

Details

First, each resource species is assigned to a cluster using the probabilities defined in the parameters *clustprob*. Notice that, because of the probabilistic nature of that procedure, some clusters may have no resource species. Therefore, the matrix may end up with fewer clusters than defined in *nclust*.

After the clusters are defined, the model creates simulated dimensions. The number of dimensions is the number of clusters times two. In each dimension is defined a range of values for each cluster. The *supdim* will define the superposition of these ranges. Then, resource species values in each dimension are drawn from a uniform distribution within the ranges of their clusters.

The pairwise Euclidian dissimilarities between all resource species are calculated using the values in each dimension. Finally, the maximum value in the matrix is set as *maxdis* and the others are redefined proportionally to it.

Outputs

dissimilarity_matrix A matrix with the pairwise dissimilarities of resource species.

performance.R

Description

A function that takes as input the match matrix and the resource species' availabilities and returns the performance matrix.

Usage

perf_calc = function (*match_matrix*, *k*)

Arguments

match_matrix An match matrix (rows = consumer species, columns = resource species).

k The availability vector of the resource species in *match_matrix* (must follow the same order).

Details

This function is sourced and used during the simulation. It uses the equation 1 to create the performance matrix:

$$P_{ij} = \begin{cases} \frac{M_{ij}}{\sum_{i=1}^{C_j} M_{ij}} K_j, & if \quad M_{ij} > 0 \\ 0, & if \quad M_{ij} \leq 0 \end{cases} \qquad (1)$$

where M_{ij} is the match of consumer i with resource j, P_{ij} is the performance of consumer i on resource j, C_j is the number of consumers that exploits resource j, and K_j is the availability of resource j.

Value

A matrix with the same size and names as *match_matrix*, with the performance of consumer species (rows) in resource species (columns).

IHS_model.R

Description

Runs a single simulation using the IHS model.

Parameters

iterations The number of iterations of the model before ending the simulation.

* Parameter values for the other files (“availabilities.R”, “initial_matrix.R”, “dissimilarity_matrix.R”) are defined on the first lines of “IHS_model.R”.

Details

In the first lines of “IHS_model.R” are indicated all parameters values for the simulation. Change those default values as you please.

Then the code sources the files “dissimilarity_matrix.R”, “initial_matrix.R”, and “availabilities.R”, using the defined parameter values and generating the inputs of the model (R objects: “dissimilarity_matrix”, “match00” and “k”).

Those inputs are used to run the simulation for many times as defined in *iterations*. At every 100 iterations the code prints the current progress.

Output

Log A dataframe with information of the selected consumer mutant in each iteration of the simulation. Rows are each iteration. Columns are: *mutcon*) the consumer species assigned to mutate; *focalres*) the focal resource species of the selected mutant; *focal mutation*) the increase in the match of the consumer with the focal resource species, and *performancegain*) the total gain in performance by the consumer species in the iteration.

match_matrix A matrix with the match between all consumer species (rows) and all resource species (columns) after the simulation.

performance_matrix A matrix with the performance of all parasite species (rows) in all host species (columns) after the simulation. This is the simulated network.

simulations.R

Description

Runs the basic model several times with parameters previously defined in a TXT file, and stores the results. Requires a table of parameters named “parsimulations.txt” in the same folder.

Details

The code loads the table in “parsimulations.txt”. In this table, each row is a simulation and columns present all parameters needed to perform it. Columns must be organized in the sequence: 1) an ID for the simulation, 2) *Scon*, 3) *Sres*, 4) *nclust*, 5) *supdim*, 6) *maxdis*, 7) *clustprob*, 8) *initial_matrix*, 9) *k_method*, 10) *iterations*.

Example:

ID	*Scon*	*Sres*	*nclust*	*supdim*	*maxdis*	*clustprob*	*initial_matrix*	*k_method*	*iterations*
Sim1	15	50	3	1.5	1.25	0.3-0.3-0.4	rep0	rnorm200-50	200
Sim2	30	60	1	1	0.75	1	rep1	rep100	1000
Sim3	5	200	6	2	1.5	0.2-0.2-0.2-0.2-0.1-0.1	rnorm1-1	rnorm200-50	4000

If “parsimulation.txt” has the above table, “simulations.R” would perform 3 simulations: Sim1, Sim2, and Sim3, each one using the parameter values indicated in the respective lines. At the end, the code stores the results in a file “results.RData”.

In the first 3 lines of the “simulations.R”, the program creates an empty list to store the results from all simulations (*results*) and saves it as “results.RData”. In each simulation, it loads “results.RData”, includes the result of the new simulation on the list, and overwrites the file.

CAUTION: the first 3 lines of “simulations.R” creates an empty list and saves as “results.RData” on the R working directory. If the file already exists, from previously performed simulations, it will be overwritten and any result stored in it, will be lost.

In the beginning of each simulation the program prints the simulation ID.

Output

results.RData An R.Data file containing a list called *results*. Each element of *results* corresponds to a list with the results of one simulation, and presents: 1) simulation ID, 2) parameters: a list with all the parameter values used on the simulation, 3) *dissimilarity_matrix*, 4) *match_matrix*, 5) *log*, 6) *performance_matrix*, 7) *match00*, 8) *k*.

References

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.