# README

**The hypermodular structure of tripartite ecological networks**

Rafael Pinheiro1, Luciano Cagnolo2, Guadalupe Peralta2 and Thomas Lewinsohn1

1 Departamento de Biologia Animal, Universidade Estadual de Campinas, Campinas, São Paulo, Brazil

2 Instituto Multidisciplinario de Biología Vegetal, CONICET - Universidad Nacional de Córdoba, Córdoba, Argentina

**This repository:** <https://github.com/pinheirorbp/tripartite_hypermodules>

**Rafael Pinheiro**

**E-mail:** [rafael-bpp@hotmail.com](mailto:rafael-bpp@hotmail.com)

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

**Website:** <https://rbppinheiro.wordpress.com/>

-----------------------------------------------------------------------------------------------------------------

[README 1](#_Toc183706239)

[Repository description 2](#_Toc183706240)

[Functions 2](#_Toc183706241)

[Codes for the study 5](#_Toc183706242)

## Repository description

In this repository we provide:

(1) a set of functions for hypermodularity analysis, including functions for applying the *HyperMod* algorithm, generating null models and plotting the tripartite networks,

(2) R scripts, data and RMarkdowns to reproduce step-by-step the analysis performed in the study.

Those materials, (1) and (2), are respectively in the folders *functions* and *codes\_study*.

If you want to apply in your own data the methods we developed, you should probably focus on the functions.

However, if you want to reproduce or check all the analyses we performed in our study, then you may check the scripts in *codes\_study*. The *results.pdf* is an output of an RMarkdown *results.Rmd* compiling and reporting all the analyses and results obtained in our study (see Suplementary Note 2).

## Functions

***module\_congruence.R***

Function for calculating module congruence and detecting hypermodules. Runs the HyperMod algorithm which is described in the study.

Arguments:

*Neta / Netb* = Bipartite networks (in matrix format). The connector partition must be represented by rows.

*MODNETa / MODNETb* = Vector indicating the modules for the species of the connector partition (rows) on each bipartite network. Must follow the order of rows in NETa and NETb.

*LAMBDA* = Rate of temperature decrease (λ). Default= 0.9995.

*TEMP0* = Initial temperature for the simulated annealing.

*NSTEPS\_MAX* = Maximum number of iterations. Default = 50000.

*NSTEPS\_STOP* = Maximum number of iterations without improvements. Default = 5000.

*plot\_opt* = Prompt a plot of the optimization process (adjusted congruence per iteration). Default = T.

*printR* = Periodically print the iterations during the process of optimization. Default = T.

*printR\_each* = Number of iterations between each printing (*printR*). Default= 1000.

*MODa / MODb* = A vector indicating the modules of only the connector species of the bipartite networks. Must represent species in the exact same order. If provided, *NETa* / *Netb* and *MODNETa* / *MODNETb* are not used.

The function returns a list with:

*$finalcongruence* = Final realized congruence.

*$finalOPT* = Final adjusted congruence (optimized by HyperMod).

*$M1M2* = A matrix with the optimized correspondence of modules between *NETa* and *NETb*.

*$OPT* = Vector with the adjusted congruence in each iteration of Hypermod.

*$congruence* = Vector with the realized congruence in each iteration of Hypermod.

*$BASE* = Vector with the base congruence in each iteration of Hypermod.

***hypermodularity.R***

Calculate the index of hypermodularity following the function described in the study.

Arguments:

*M1M2* = A matrix with the optimized correspondence of modules between *NETa* and *NETb* (output of the function *module\_congruence*).

*Neta / Netb* = Bipartite networks (in matrix format). The connector partition must be represented by rows.

*MODNETa / MODNETb* = Vector indicating the modules for the species of the connector partition (rows) on each bipartite network. Must follow the order of rows in NETa and NETb.

The function returns a list with:

*$overall* = hypermodularity for the tripartite network.

*$NETa* = component of hypermodularity from *NETa*.

*$NETb* = component of hypermodularity from *NETb*.

***null\_module\_congruence.R***

Function for generating the null model 1. Requires the package *doSNOW*.

Arguments:

*Neta / Netb* = Bipartite networks (in matrix format). The connector partition must be represented by rows.

*MODNETa / MODNETb* = Vector indicating the modules for the species of the connector partition (rows) on each bipartite network. Must follow the order of rows in NETa and NETb.

*M1M2* = A matrix with the optimized correspondence of modules between *NETa* and *NETb* (output of the function *module\_congruence*).

*NCores* = Number of cores used in parallel computing. Set as 1 for non-parallel analysis. Default = 1.

*Nulls* = Number of randomizations. Default = 100.

The function returns a list with:

*$Nullcongruence* = A vector with the realized congruence on each randomization.

*$NullOPT* = A vector with the adjusted congruence on each randomization.

***null\_module\_congruence.R***

Function for generating the null model 2. Requires the package *doSNOW*.

Arguments:

*Neta / Netb* = Bipartite networks (in matrix format). The connector partition must be represented by rows.

*MODNETa / MODNETb* = Vector indicating the modules for the species of the connector partition (rows) on each bipartite network. Must follow the order of rows in NETa and NETb.

*M1M2* = A matrix with the optimized correspondence of modules between *NETa* and *NETb* (output of the function *module\_congruence*).

*LAMBDA* = Rate of temperature decrease (λ). Default= 0.9995.

*TEMP0* = Initial temperature for the simulated annealing.

*NSTEPS\_MAX* = Maximum number of iterations. Default = 50000.

*NSTEPS\_STOP* = Maximum number of iterations without improvements. Default = 5000.

*NCores* = Number of cores used in parallel computing. Set as 1 for non-parallel analysis. Default = 1.

*Nulls* = Number of randomizations. Default = 100.

The function returns a list with:

*$Nullcongruence* = A vector with the realized congruence on each randomization.

*$NullOPT* = A vector with the adjusted congruence on each randomization.

## Codes for the study

The codes to reproduce our study are within the main folder *codes\_study*. It includes 3 files and 5 subfolders.

***base\_functions/***

Functions that were used in our analysis and are necessary to run the scripts.

*base\_functions/bipartite\_functions.R*

Functions for analyzing BN topology, specialization and to produce null models. These functions are similar or equal to those available in the package bipartite and were included in this script because bipartite package was not available in the computing cluster.

*base\_functions/hypermodularity.R*

*base\_functions/module\_congruence.R*

*base\_functions/null\_module\_congruence.R*

*base\_functions/null\_module\_congruence\_2.R*

***data/***

Database of species interaction networks. Each file is the adjacency matrix for one bipartite network.

Also includes a table *FINAL\_DATABASE.xlsx* listing the datasets. This file is necessary for running *script\_preparation.R*.

***spatial\_taxonomic\_data/***

Database with information of species taxonomy and distribution among localities. Applied for testing the mechanisms underlying hypermodules, through Mantel and Partial Mantel Tests.

***scripts/***

*scripts/dataset\*.R*

There is one script to load and prepare the data, and to run the hypermodularity analysis for each dataset (*dataset1.R* to *dataset47.R*). Each script sources *script\_preparation.R* and *script\_analysis.R*.

Within these scripts some corrections are made (whenever errors have been identified) and transformation of values is performed when adequate. The parameters for the analysis of each tripartite network are set:

*NCores* = number of cores in the parallel computing.

*N\_null\_topology* = number of randomizations in the null models for BN topology analysis.

*N\_null\_congruence* e *N\_null\_congruence2* = number of randomizations in the null models for tripartite hypermodularity (Null 1 and Null2)

* The codes applied were not always exactly these, because some were run in a computer cluster and had to be modified accordingly (e.g., changing addresses of files to load or functions to source and the number of cores – *NCores*).
* For most datasets, the time for running in a normal computer is really high.

*scripts/ script\_preparation.R*

This script is sourced at the beginning of each *dataset\*.R*. It loads the functions in *base\_functions* and retrieves some information from *FINAL\_DATABASE.xlsx* for the dataset being analyzed (e.g., name and partitions).

*scripts/ script\_analysis.R*

Effectively run, for the given dataset, the analyses of BN topology and TN hypermodularity, including indices, the HyperMod algorithm and null models. Use the functions sourced by *script\_preparation.R*.

After the analysis is completed, the results are saved in the folder *files\_results* as a RData file, named as *dataset\*.RData*.

*scripts/script\_Mantel.R*

Perform the Mantel and Partial Mantel tests correlating hypermodule sharing with taxonomic similarity and co-occurrence. Analyses are performed for each partition of a subset of datasets. The results for all analyzed datasets are saved in the folder *files\_results* as *Mantel\_results.RData*.

***files\_results/***

Stores RData files with the raw results of the analyses performed (e.g., BN topology, TN hypermodularity, Null Models, Mantel tests). These files are the main inputs for the RMarkdown, *results.Rmd*, that compiles and presents all the analyses and results performed in the study.

*datasets.RData* is the only file in the folder that is not an input but rather an output of *results.Rmd*. It stores a dataframe of results for the datasets.

***results.Rmd***

This is an RMarkdown that compiles all the analyses performed and results obtained, and reports in an organized format. The main output is a pdf file, but it also produces *results.xlsx* and *files\_results/datasets.RData*.

***results.PDF***

Output of the RMarkdown, reporting all the analyses performed and results obtained in the study.

***results.xlsx***

Output of the RMarkdown. It is a table with results for the dataset.