Mangal.jl and EcologicalNetworks.jl: Two complementary packages for analyzing ecological networks in Julia

Francis Banville∗1, 2, Steve Vissault2, and Timothée Poisot1

1 Department of Biology, Université de Montréal, Canada 2 Department of Biology, Université de Sherbrooke, Canada

Summary

Network ecology is an emerging field of study describing species interactions (e.g. predation, pollination) in a biological community. Ecological networks, in which two species are connected if they can interact, are a mathematical representation of all interactions encountered in a given ecosystem. Anchored in graph theory, the methods developed in network ecology are remarkably rigorous and biologically insightful. Indeed, many ecological and evolutionary processes are driven by species interactions and network structure (i.e. the arrangement of links in ecological networks). The study of ecological networks, from data importation and simulation to data analysis and visualization, requires a coherent and efficient set of numerical tools. With its powerful and dynamic type system, the Julia programming language provides a very good and fast computing environment suitable for ecological research. Julia’s typing system notably ensures that appropriate methods are used when a function is applied to different types of ecological data, which therefore minimizes the risk of committing errors of analysis and interpretation in network ecology. Mangal.jl and EcologicalNetworks.jl are two novel and complementary Julia packages designed to conduct research in network ecology efficiently. Researchers studying a broad range of ecological networks can read the data and metadata of numerous well-documented networks using Mangal.jl, and analyze their properties using EcologicalNetworks.jl.

Statement of need

Ecological networks are a gold mine of biological information. From defining species’ roles within the context of their community to describing the functioning of ecosystems, the analysis of ecological networks can help us predict how species or entire ecosystems will respond to environmental changes (Delmas et al., 2019; Guimarães, 2020). Analyzing ecological networks is however a very challenging task. First, data retrieval and manipulation can be difficult, as authors typically store their data in different repositories and in different formats. Further, diverse measures are commonly used to characterize the structure of ecological networks (Delmas et al., 2019), and their implementation can sometimes be arduous or computationally expensive. The Julia programming language was designed for data science and machine learning, and can thus provide the algorithmic efficiency needed in ecology. EcoJulia, an integrated environment for the conduction of ecological research in Julia, contains a set of packages that can be used for a variety of ecological topics, from spatial analysis to phylogenetics.

Together, two interrelated packages in EcoJulia provide a useful methodological framework for the analysis of ecological networks: Mangal.jl and EcologicalNetworks.jl. The Mangal project was set up to meet the need for an extensive ecological interaction database with an

*corresponding author
easy data retrieval system. In Julia, Mangal.jl is used to read data from mangal.io. Once imported, these networks can then be analyzed using EcologicalNetworks.jl, which could also be used to simulate ecological networks under different statistical and ecological models. Our suggested methodological framework is described in more detail in Figure 1. Mangal.jl and EcologicalNetworks.jl are designed to work with the most recent versions of Julia (Julia 1.3+). Here we provide an overview of the functionalities of these two packages, and illustrate how they can be used to answer fundamental ecological questions.

Figure 1: Proposed methodological framework for the analysis of ecological networks in Julia. Inputs: More than 1300 ecological networks of various types are archived on the mangal.io database. These networks are read in Julia using Mangal.jl. Outputs from Mangal.jl: Objects (e.g. species, interactions, some metadata) can be counted directly using Mangal.jl. Mangal.jl can also perform metadata (e.g. location, collection date, reference) and taxonomic analysis. Outputs from EcologicalNetworks.jl: All types of networks, including probabilistic networks, can be further analyzed using EcologicalNetworks.jl. The main measures and analysis supported by this package include species richness, connectance, linkage density, species’ degree, centrality measures, trophic level analysis, shortest paths, nestedness, motif counting, modularity, beta diversity measures, and information theoretic analysis. Empirical networks can then be compared to a random network distribution simulated using null models, constrained network permutations, and structural random models. EcologicalNetworks.jl is well integrated with BioEnergeticFoodWebs.jl, for the simulation of biomass flows in food webs, and with EcologicalNetworksPlots.jl, for network visualization. Backend methods: These measures and analysis methods are described in further depth in Delmas et al. (2019), Poisot et al. (2016), and Poisot et al. (2019).

State of the field

The main software packages available for analyzing ecological networks include igraph [Csardi & Nepusz (2005); Python, R, and C], networkx [Hagberg et al. (2008); Python], bipartite [Dormann et al. (2008); R], cheddar [Hudson et al. (2013); R], and BiMat [Flores et al. (2016); Octave]. To the best of our knowledge, EcologicalNetworks.jl is the only software that supports all types of ecological networks, and the only one that supports probabilistic ecological networks.

In Julia, the JuliaGraphs ecosystem contains a set of performant packages for modeling and analyzing different types of graphs and networks from many disciplines. The main JuliaGraphs package is LightGraphs.jl (Bromberger, 2017), which implements functions for the analysis of simple undirected and directed graphs.
The added-value of using EcologicalNetworks.jl to analyze ecological networks, instead of a more general-purpose package like LightGraphs.jl, is two-fold: (1) its integration with other EcoJulia packages, including Mangal.jl, and with ecological databases, greatly simplifies the conduction of comprehensive ecological research in Julia, and (2) most functions of a general-purpose graph analysis package may not be suitable to adequately address many ecological questions. Indeed, functions provided by EcologicalNetworks.jl are based on state-of-the-art practices in network ecology (Delmas et al., 2019; Poisot et al., 2016), whereas general-purpose functions might be misleading when it comes to interpreting them ecologically.

**Mangal.jl**

Mangal.jl uses a RESTful API to query data directly from the mangal.io database. To this date, 172 datasets for a total of more than 1,300 ecological networks of various types are archived on mangal.io. This makes it one of the most extensive database of empirical ecological networks.

In order to facilitate data retrieval, Mangal.jl is based on a custom hierarchical type system outlined in Table 1. Objects of type MangalInteraction represent the interaction from one MangalNode to another, and are associated to a MangalNetwork in a MangalDataset. Interactions can be Boolean, probabilistic, or quantitative. Types of interactions (e.g. predation, mutualism, parasitism, competition) are encoded in MangalAttribute objects, and different types can be represented in the same network. Objects also contain various metadata, such as the date of data collection, the geographical position, a description, a name and an ID number.

Parameters can be used to filter or sort queries. For example, one could retrieve data according to a dataset ID or according to a given type of interaction. The total number of entries in the database can also be counted, each type of object having its own count method.

| Table 1: Description of all custom types in Mangal.jl. Related custom types are used in the fields of a given Mangal type. |
|---------------------------------|----------------|--------------------------------|
| Type                            | Description              | Related custom types           |
| MangalDataset                   | Dataset metadata         | MangalReference none           |
| MangalReference                 | Dataset reference        | none                           |
| MangalNetwork                   | Network metadata         | MangalDataset                  |
| MangalInteraction               | Pairwise interaction data and metadata | MangalNetwork, MangalNode, MangalAttribute |
| MangalAttribute                 | Type of interaction      | none                           |
| MangalNode                      | Species observation metadata | MangalReferenceTaxon           |
| MangalReferenceTaxon            | Species taxonomic name   | none                           |

**Use case 1: Association between the number of species and the number of links**

Understanding how the number of interactions scales with the number of species is fundamental in ecology. This relationship has recently been revisited using all food webs archived on mangal.io (MacDonald et al., 2020), making it, to the best of our knowledge, the most extensive study of such relationship so far. The block of code below read relevant metadata from mangal.io to conduct this analysis on all types of ecological networks.
We first retrieve all networks archived on the database, which returns objects of type `MangalNetwork`. We then count the number of species $S$ and the total number of interactions $L$ in each network, as well as their numbers of interactions of predation, of herbivory, of mutualism, and of parasitism. We store these information in a data frame along with the ID numbers of the networks.

```julia
using Mangal
using DataFrames

number_of_networks = count(MangalNetwork)
count_per_page = 100
number_of_pages = convert(Int,
    ceil(number_of_networks/count_per_page))
mangal_networks = DataFrame(fill(Int64, 7),
    [:id, :S, :L, :pred, :herb, :mutu, :para],
    number_of_networks)

global cursor = 1
@progress "Paging networks" for page in 1:number_of_pages
    global cursor
    networks_in_page = Mangal.networks("count"=>count_per_page,
        "page"=>page-1)
    @progress "Counting items" for current_network
        in networks_in_page
            S = count(MangalNode, current_network)
            L = count(MangalInteraction, current_network)
            pred = count(MangalInteraction, current_network,
                "type"=>"predation")
            herb = count(MangalInteraction, current_network,
                "type"=>"herbivory")
            mutu = count(MangalInteraction, current_network,
                "type"=>"mutualism")
            para = count(MangalInteraction, current_network,
                "type"=>"parasitism")
            mangal_networks[cursor,:] .= (current_network.id,
                S, L, pred, herb, mutu, para)
        cursor = cursor + 1
end

The association between the number of species $S$ and the total number of links $L$ is plotted in Figure 2. We classify all networks according to their most frequent type of links, and consider interactions of predation and of herbivory as food-web interactions. Networks in the “other types” category include interactions between competitors, symbiotes, scavengers, and detritivores, among others. Very small networks (i.e. with less than 5 interactions) are discarded.

Figure 2: Association between the number of species (nodes) and the number of links (edges) in ecological networks archived on mangal.io. Networks with less than 5 interactions are discarded. Different types of interactions can be listed within the same network. We classify all networks according to their most frequent type of interactions. The code to reproduce the figure is included in this paper’s repository.

**EcologicalNetworks.jl**

For a deeper numerical analysis, Mangal networks can be converted into one of the many custom types of EcologicalNetworks.jl. Given that each type of network is measured differently, EcologicalNetworks.jl is based upon a hierarchical type system which ensures that measures are applied to appropriate networks (Poisot et al., 2019). The most common measures used to analyze deterministic (Delmas et al., 2019) and probabilistic ecological networks (Poisot et al., 2016) are implemented in our package. Moreover, a set of functions in EcologicalNetworks.jl simulate networks under null models (i.e. models without given processes), a common practice among ecologists to investigate the deviation of empirical measures from their expected values (see for example Bascompte et al., 2003; Fortuna & Bascompte, 2006; and Delmas et al., 2019).

The package EcologicalNetworksPlots.jl, for network visualization, uses the same type system as EcologicalNetworks.jl.

**Use case 2: Association between meaningful network measures**

Connectance (i.e. the proportion of all possible links that are realized) is undoubtedly one of the most studied and important measure of ecological networks (Poisot & Gravel, 2014). A network’s connectance is the result of many ecological processes, and predicts how a biological community functions and responds to changes (Dunne et al., 2002a, 2002b). Connectance is furthermore associated with other network measures, including nestedness and modularity (Delmas et al., 2019). A network is nested when species that interact with specialists (i.e. species with few interactions) are a subset of the species that interact with
generalists (i.e. species with many interactions). On the other hand, a network is modular when species are organized in groups of highly interacting species. Fortuna et al. (2010) showed how these two quantities were associated in ecological networks. Here we show how EcologicalNetworks.jl can be used in conjunction with Mangal.jl to retrieve these associations in food webs.

We read networks metadata from mangal.io using the code in the previous section, and select networks we classified as food webs. We then use their ID numbers to import all their data from mangal.io, which again returns objects of type MangalNetwork. However, food-web measures are computed on objects of type UnipartiteNetwork. We thus need to use EcologicalNetworks.jl for type conversion.

```julia
using EcologicalNetworks

mangal_foodwebs = network.(foodwebs.id)
    # the data frame "foodwebs" is a subset of
    # mangal_networks for networks classified as food webs

unipartite_foodwebs = convert.(UnipartiteNetwork,
    mangal_foodwebs)
```

Next, we compute food-web richness (i.e. the number of species), connectance, nestedness, and modularity using functions from EcologicalNetworks.jl. Nestedness is computed using the spectral radius of the matrices of interactions (i.e. their largest absolute eigenvalue, Staniczenko et al., 2013). To compute network modularity, we need a starting point, an optimizer, and a measure of modularity (Barber, 2007; Newman, 2006; Thébault, 2013). We use 100 random species assignments in 3 to 15 groups as our starters. We use the BRIM algorithm to optimize the modularity for each of these random partitions, and compute modularity following Newman (2006). The maximum value is retained for each food web. Associations between these measures are plotted in Figure 3.

```julia
foodweb_measures = DataFrame(fill(Float64, 4),
    [:rich, :connect, :nested, :modul],
    length(unipartite_foodwebs))

foodweb_measures.rich = richness.(unipartite_foodwebs)

foodweb_measures.connect = connectance.(unipartite_foodwebs)

# compute nestedness
foodweb_measures.nested = rho.(unipartite_foodwebs)

# compute modularity
number_of_modules = repeat(3:15, outer=100)
modules = Array{Dict}(undef, length(number_of_modules))

for i in eachindex(unipartite_foodwebs)
    current_network = unipartite_foodwebs[i]
    for j in eachindex(number_of_modules)
        modules[j] = n_random_modules
        partition_modularity = map(x -> Q(current_network, x),
```

Figure 3: Association between (1) connectance and nestedness, (2) connectance and modularity, and (3) modularity and nestedness in food webs archived on mangal.io. We compute nestedness as the spectral radius of a network (i.e. the largest absolute eigenvalue of its matrix of interactions). We optimize network modularity using the BRIM algorithm (best partition out of 100 random runs for 3 to 15 modules). The marker size is proportional to the number of species in a network, which varies between 5 and 714 species. All measures are computed using EcologicalNetworks.jl. The code to reproduce the figure is included in this paper's repository.

Acknowledgements

We would like to thank all contributors to EcoJulia, EcologicalNetworks.jl and EcologicalNetworksPlots.jl for their help in developing this integrated environment for ecological research in Julia. Funding was provided by the Institute for Data Valorization (IVADO), the Canadian Foundation for Innovation and the Natural Sciences Engineering Research Council of Canada (NSERC).

References


