`mangal` – making complex ecological network analysis simpler

T. Poisot, D. Gravel

Jan. 2014

The study of ecological networks is severaly limited by (i) the difficulty to access data, (ii) the lack of a standardized way to link metadata with interactions, and (iii) the disparity of formats in which ecological networks themselves are represented. To overcome these limitations, we conceived a data specification for ecological networks. We implemented a database respecting this standard, and released a R package (`rmangal`) allowing users to programmatically access, curate, and deposit data on ecological interactions. In this article, we show how these tools, in conjunctions with other frameworks for the programmatic manipulation of open ecological data, will streamline the analysis process, and improve reproducibility in studies of ecological networks.

```
## Error: impossible de trouver la fonction "as"
```

Introduction

Ecological networks enable ecologists to accommodate the complexity of natural communities, and to discover mechanisms contributing to their persistence, stability, resilience, and functioning. Most of the "early" studies of ecological networks were focused on understanding how the structure of interactions within one location affected the ecological properties of this local community. This led to classical results, such as the buffering impact of modularity on species loss {ref}, the increase in robustness along with increases in connectance {ref}, and {missing}. More recently, there have been new studies introducing the idea that different networks can be meaningfully compared, either to understand the importance of environmental gradients on the realisation of ecological interactions {ref}, or to understand the mechanisms behind variation in the structure of ecological networks {refs}. Yet, meta-analyses of a large number of ecological networks are still extremely rare, and most of the studies comparing several networks do so within the limit of particular systems {refs}. In part, this can be attributed to the limited methods allowing to compare networks in which no species are in common {ref}. However, the severe shortage of data in the field also restricts the power of large-scale analyses. Indeed, most of the studies working on several types of interactions focused on comparing emerging properties {refs}.

An increasing number of approachs are being put forth to *predict* the structure of ecological networks, either relying on latent variables {ref} or actual traits {ref}. These approaches, so as to be adequately calibrated, require easily accessible data. Comparing the efficiency of different methods will also be facilitated if there is an homogeneous way of representing ecological interactions, and the associated metadata. In this paper, we (i) establish the need of a data specification serving as a *lingua franca* among network ecologists, (ii) describe this data specification. Finally, we (iii) describe `mangal`, a `R` package and compagnon database, relying on this data specification. We provide some use cases showing how this new approach makes complex analyzes simpler, and allows for the integration of new tools to manipulate biodiversity resources.

Why do we need a data specification?

Ecological networks are (often) stored as their *adjacency matrix* (or as the quantitative link matrix), that is a series of `0` and `1` indicating, respectively, the absence and presence of an interaction. This format is extremely convenient for *use* (as most network analysis packages, *e.g.* `bipartite`, `betalink`, `foodweb`, require data to be presented this way), but is extremely inefficient at storing *meta-data*. In most cases, an adjacency matrix will inform on the identity of species (in cases where rows and columns headers are present), and the presence or absence of interactions. If other data about the environment (*e.g.* where the network wassampled) or the species (*e.g.* the population size, trait distribution, or other observations) are available, they are most either given in other files, or as accompanying text. In both cases, making a programmatic link between interaction data and relevant meta-data is difficult and error-prone.

By contrast, a data specification provides a common language for network ecologists to interact, and ensure that, regardless of their source, data can be used in a shared workflow. Most importantly, a data specification describes how data are *exchanged*. Each group retains the ability to store the data in the format that is most convenient for in-house use, and only needs to provide export options (*e.g.* through an API) respecting the data specification. This approach ensures that *all* data can be used in meta-analyses, and will in time increase the impact of data {ref}.

Elements of the data specification

{complete}The data specification is built around the idea that (ecological) networks are collections of relationships between ecological objects, each element having particular meta-data associated. In this section, we detail highlight the way networks are represented in the `mangal` specification. An interactive webpage with the elements of the data specification can be found online at `http://mangal.uqar.ca./doc/spec/`. The data specification is implemented as a series of `JSON` schemes, *i.e.* documents describing how the data should be formatted, and what each element represent. The schemes can be downloaded from `https://github.com/mangal-wg/mangal-schemes/releases/tag/1.0` {todo}. Rather than giving an exhaustive list of the data specification (which is available online at the aforementionned URL), this section will propose an overview of each element, and of how they interact. Within the `R` package, information about the data format can be viewed using the `whatIs` function (*e.g.* `whatIs(api, 'taxa')` will return a table with information about how `taxa` objects are formated.

We propose `JSON` as the most efficient data format for the following reasons. First, it has emerged as a *de facto* standard for web platform serving data, and accepting data from users. Second, it allows *validation* of the data: a `JSON` file can be matched against a scheme, and one can verify that it is correctly formatted. Finally, `JSON` objects are easily and cheaply (memory-wise) parsed in the most common programming languages, notably `R` (equivalent to `list`) and `python` (equivalent to `dict`).

Node informations

Taxa

Taxa are a taxonomic entity of any level, identified by their name, vernacular name, and their identifiers in a variety of taxonomic services. Associating the identifiers of each taxa is important to leverage the power of the new generation of open data tools, such as `taxize` {ref}. For example, a taxa with an associated *NCBI Taxonomy* identifier can be represented this way:

```
{
    "name": "Lamellodiscus ignoratus",
    "vernacular": "Lamellodiscus ignoratus",
    "ncbi": "142934"
}
```

The data specification currently accomodates `ncbi`, `gbif`, `itis` and `bold` identifiers. Correspondances between these and other services can be made through

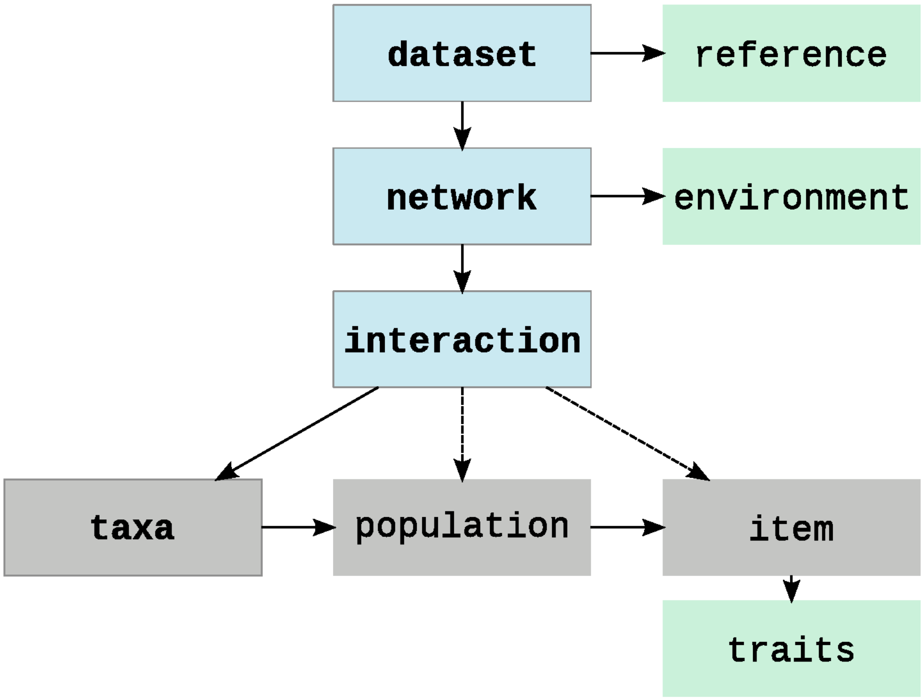

Figure 1: An overview of the data specification, and the hierarchy between objects. Each box correspond to a level of the data specification. Grey boxes are nodes, blue boxes are interactions and networks, and green boxes are metadata. The **bold** boxes (`dataset`, `network`, `interaction`, `taxa`) are the minimal elements needed to represent a network.

other tools, such as *e.g.* taxize. The structure of taxa objects can be viewed from within the R package (we present an abbreviated view):

```
whatIs(api, "taxa")[, c("field", "help", "type")]
```

```
##         field                                        help    type
## 1        bold             The BOLD identifier of the taxa integer
## 2 description             A short description of the taxa  string
## 3        gbif             The GBIF identifier of the taxa integer
## 5        itis             The ITIS identifier of the taxa integer
## 6        name             The scientific name of the taxa  string
## 7        ncbi    The NCBI Taxonomy identifier of the taxa integer
## 9  vernacular The vernacular name of the taxa, in English  string
```

Population

A population is one observed instance of a taxa object. If your experimental design is replicated through space, then each taxa will have a population object corresponding to each locality.

Item

An item is an instance of a population. Items have a level argument, which can be either individual or population; this allows to represent both individual-level networks (*i.e.* there are as many items attached to a population than there were individuals of this population sampled), and population-level networks. When item represents a population, it is possible to give a measure of the size of this population.

The notion of item is particularly useful for time-replicated designs: each observation of a population at a time-point is an item with associated trait values, and possibly population size.

Network informations

Interaction

Network

Dataset

Meta-data

Trait value

Environmental condition

User

paternity {ref}

References

Use cases

{edit}In this section, we present use cases using the `rmangal` package for `R`, to interact with a database implementing this data specification, and serving data through a `REST`ful API (`http://mangal.uqar.ca/api/v1/`). It is possible for users to deposit data into this database, through the `R` package. Data are made available under a *CC-0 Waiver*.

```
library(rmangal)
api <- mangalapi()
```

Plotting a network

```
graph <- network_as_graph(api, 2)
plot(graph, layout = layout.circle)
```

Network beta-diversity

Connectance and richness relationships

References

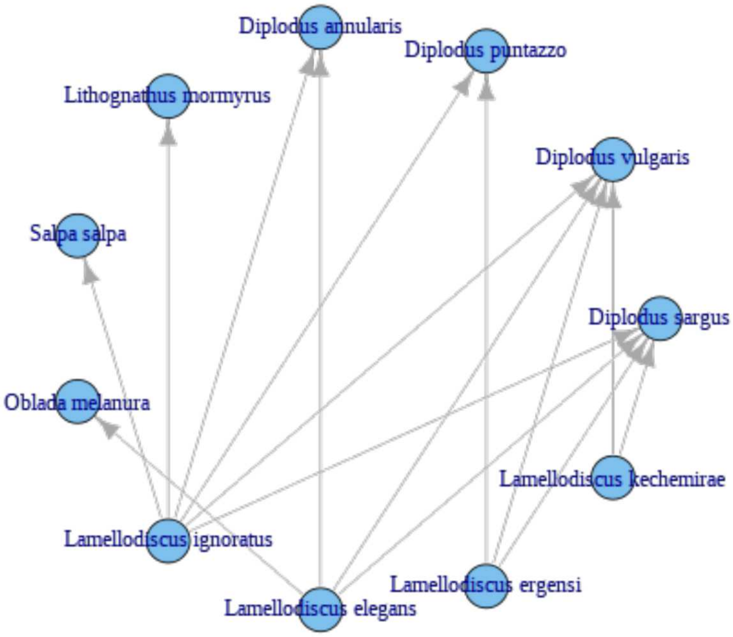

Figure 2: Example of network plotting, using the `network_as_graph` function.