





Practical guidelines and the EMLN R package for handling ecological multilayer networks

Noa Frydman ¹, Shirly Freilikhman ¹, Itamar Talpaz ¹, and Shai Pilosof ^{1,*}

¹*Department of Life Sciences, Ben-Gurion University of the Negev, Be'er-Sheva, Israel*

**Corresponding author: pilos@bgu.ac.il*

August 1, 2023

Abstract

Network analysis provides a powerful framework to study the complexity underlying the structure, dynamics, stability, and function of ecological systems. By now, analysing single-layered networks is a common practice with clear guidelines and well-established computational tools. Nevertheless, ecological communities are multilayered because they vary over space and time and contain multiple types of interactions. In recent years, the analysis of ecological multilayer networks (EMLNs) has allowed researchers to include such multilayered complexity. However, there is a paucity of practical guidelines and standardised tools to handle EMLN data, even before downstream analysis. In this article, we accomplish three objectives: (1) We provide practical guidelines for handling EMLN data. (2) We developed the EMLN R package to standardise the workflow of creating, storing, and working with EMLN objects. The package also enables conversion between multiple formats for downstream analysis with other standard packages. (3) We provide EMLN data sets for research and training purposes. A dedicated website with explanations, detailed examples, and code accompanies our paper. This website is a gateway for novice and experienced network ecologists who want to include EMLNs in their research. By simplifying the analysis of multilayer networks and promoting standardised approaches we facilitate the analysis of EMLNs. This paves the road towards gaining deeper insights into functioning and resilience of natural ecosystems.

Key words: data structures | network analysis | network ecology | network science | open source | workflow

1 Introduction

Network theory provides a powerful mathematical framework to describe interactions between entities, facilitating research into the challenging complexity of natural and experimental ecological systems [1,2]. Network analysis provided ecologists with an incredible toolkit, whose application has led to a myriad of insights into the structure, function and stability of ecosystems [1–4]. By now, ecological network analysis is common, largely due to the availability and ease of use of software packages such as igraph and bipartite [5]. However, until recently, ecological networks have been studied in isolation although ecological communities and systems are inherently multi-dimensional: they vary in space [6,7] and time [8–10], and they often contain multiple kinds of interactions (e.g., predation and mutualism) [11–15]. The recent development of multilayer networks in the broader network science [16,17] and specifically of ecological multilayer networks (EMLNs) [18,19] addresses this problem. A multilayer network has two or more layers, each of which is a monolayer network. Multilayer networks can also contain interlayer links connecting nodes from different layers (Fig. 1). Encoding processes within and between layers in a single mathematical/computational object provides a more realistic representation of complex ecological systems. Hence, in the past six years, there has been a surge in the study of EMLNs, which have proven highly insightful for understanding the structure and dynamics of ecological communities [10,15,20–22]

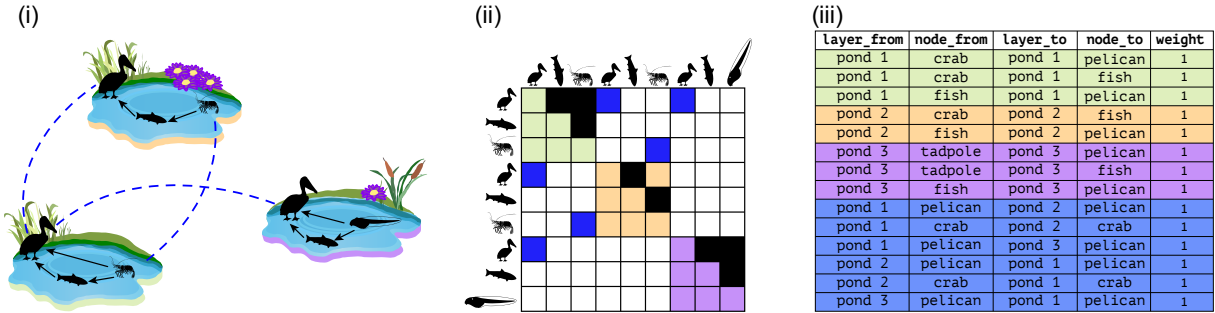
On the technical side, however, EMLNs are challenging to use. An EMLN object can contain many kinds of data because the identities and attributes of nodes and links can vary between layers. The need to computationally store multiple layers and data types under a single object complicates the workflow of creating and managing EMLN objects. This is exacerbated when including node and link attributes, which can vary between layers (e.g., the abundance of a given species varies in time). Furthermore, EMLNs can contain interlayer links, which do not exist in monolayer network analysis. From a technical perspective, different software packages require different formats for working with multilayer networks. These challenges hinder the application of EMLNs, especially for newcomers. In addition, the mathematical structure of these networks is not trivial, especially when dealing with a bipartite layer structure, which is common in ecology (e.g., pollination, parasitism).

Although there are various sources that define multilayer networks in ecology [18] and network science [16,17,23], their application in practice is still challenging. Importing and handling multilayer data is the primary hurdle of network ecologists working with EMLNs. Therefore, the goal of this paper is threefold. (1) To provide practical guidelines for handling EMLN data. (2) To standardise the workflow of creating, storing, and working with EMLN objects. This goal includes facilitating conversion between multiple computational formats to enable downstream network analysis with packages such as igraph [24], bipartite [5], Muxviz [25], and infomapecology [26]. (3) To provide resources for research and training purposes.

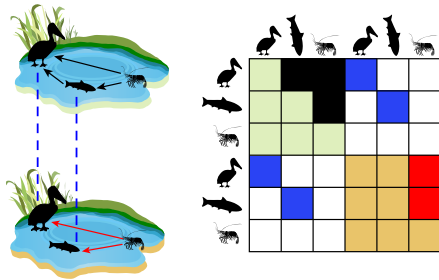
To fulfill these goals, we explain the workflow of working with EMLNs, and define the necessary data structures. In addition, we developed the EMLN package, which we include with this paper. The package contains functions to import, store, and convert EMLN data. It also contains an

extensive curated data set of multilayer networks, which can be used to perform comparative analyses, teach, and practice. Per the paper’s goal, we focus on pre-analysis workflows, which involve the formatting and organisation of EMLNs. Therefore, we do not provide explanations of multilayer network analysis methods (e.g., modularity, centrality, motifs), for which there are many resources available (e.g., [23,26–28]). Given that EMLN analysis is an advanced framework, we assume that readers have some basic experience with monolayer network analysis and are familiar with the basic concepts of EMLNs [18,19]. An overview is presented in Fig. 1 (also see an additional visual guide in [29]).

(A) Spatial



(B) Multiplex



(C) Temporal

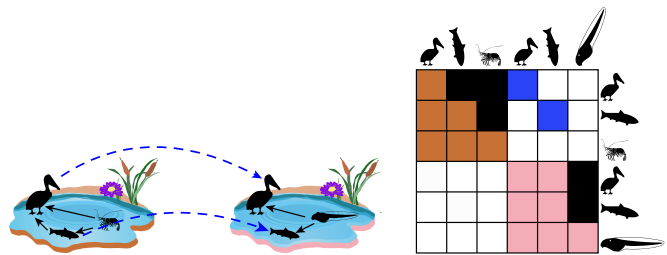


Fig. 1: Definitions and examples of ecological multilayer networks (EMLNs) and their underlying data. (A) A spatial network; (B) A node-aligned multiplex network with two interaction types; (C) A directed temporal network. An EMLN has four components: (1) *Layers* (e.g., patches, interaction types, time points). (2) *Nodes*. *Physical nodes* are the entities (e.g., the pelican), and *state nodes* are realisations of the same physical node in different layers (e.g., the pelican in different patches). Note that physical nodes do not necessarily appear in all layers. (3) *Intralayer links*, which encode interactions within each layer. Trophic (black arrows) and parasitic (red arrows) interactions are presented as an example. (4) *Interlayer links* (blue dashed lines) encoding processes that operate between layers. For example, the relative change in abundance between populations of state nodes. In (C), the interlayer links are directed to represent the temporal flow. See [17,18] for a complete definition of EMLNs, including mathematical notations. For the network in (A) there are three representations: (i) a network; (ii) a supra-adjacency matrix (SAM); and (iii) an extended link list (ELL). In these representations the green, orange and purple colours are patches. In (ii) the diagonal blocks represent intralayer links and off-diagonal blue cells are interlayer links. in (iii) coloured rows are intralayer links corresponding to the colour-corresponding layers, and the blue rows are interlayer links.

2 Practical guidelines for constructing multilayer networks

The workflow of multilayer network analysis is similar to that of monolayer networks. Given a research question and corresponding methodology, one collects data, imports them into programming languages like R or Python, and analyses them using dedicated packages (Fig. 2). However, as mentioned above, EMLNs are more complicated than monolayer ones, which requires considering the following:

- What do layers represent (e.g., space, time, interaction types).
- Layer structure (e.g., unipartite, bipartite).
- Interlayer links. While it is possible to analyse EMLNs without interlayer links, these form a distinctive component of EMLNs. What interlayer links can encode and how to measure them is an issue already discussed in detail [19], but that still remains challenging. Interlayer links should form part of the research question and be measured as part of the data collection method. In practice, however, this is not easy to do. To date, only a handful of papers have done so [10,15,21]. A different approach is to infer interlayer links from existing data. For instance, using species abundance values to encode population dynamics [18,20]. Despite the surge in EMLN studies, only sporadic examples exist of fully detailed EMLNs. Another approach is to set a uniform value to all interlayer links (e.g., [30]). However, this should be avoided because setting an arbitrary value does not encode hypotheses.
- Node or link attributes, and do they vary across layers.
- The kind of analysis that will be applied, to choose adequate data structures.

With these considerations in mind, in the following subsections we describe the data structures used to define EMLNs and provide guidelines on how to prepare the data for downstream analysis.

2.1 Multilayer data structures

There are two common data structures to store a multilayer network in software: *Supra-adjacency matrix* (SAM) and *extended link list* (ELL) (Fig. 1). We describe them in this section. Mathematically, a multilayer network is a tensor (a matrix with more than two dimensions) [16]. Specifically, the tensor $\mathbf{M}_{j\beta}^{i\alpha}$ encodes the link between node i in layer α to node j in layer β . When $\alpha = \beta$ the link is intralayer. A tensor can be “flattened” to a rank-2 tensor, also called SAM, with dimensions $NL \times NL$, where N and L are the total number of physical nodes and layers, respectively [17] (see visual guide in [29]). In a SAM, each block of size $N \times N$ is a layer and off-diagonal blocks are interlayer links. Hence, all physical nodes are present in all blocks, even if they were not observed (Fig. 1).

A SAM representation is used by many algorithms (e.g, calculation of centrality measures). However, a SAM cannot be used to store link attributes beyond weight. In addition, using SAM

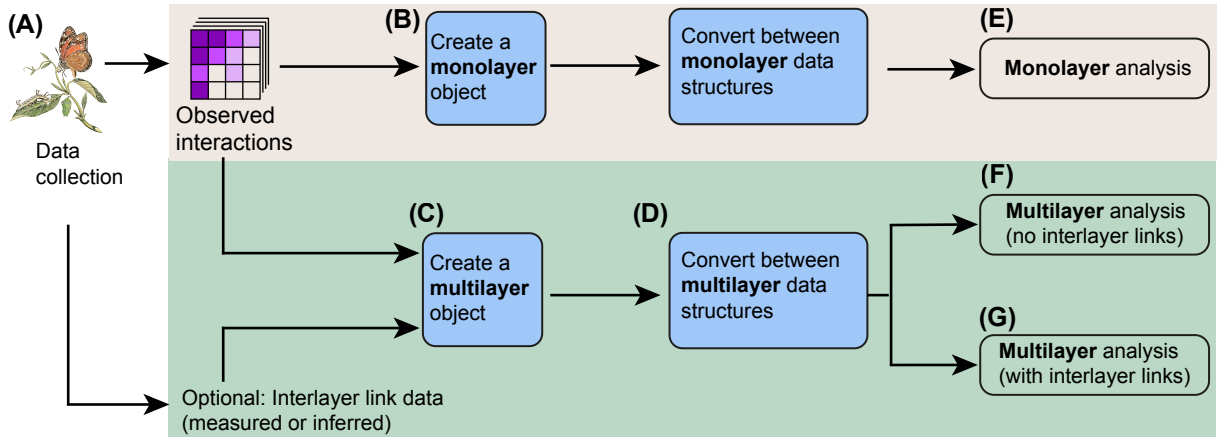


Fig. 2: Typical workflow of working with multilayer network data. Blue rectangles indicate steps that can be performed using the EMLN package. **(A)** Data is collected for intralayer and, possibly, interlayer links. The data is typically stored in multiple Excel sheets or text files (e.g., csv). **(B)** For monolayer analysis, each file can be used to construct a monolayer network using the `monolayer` class, which contains three data structure types (link list, matrix and igraph). This conversion facilitates downstream monolayer network analysis. **(C)** The files can be used to create a single, standardised object of class `multilayer` in the EMLN package (see main text for a detailed description). **(D)** The `multilayer` object can be converted to other formats (igraph or a supra-adjacency matrix) for downstream analysis. We define three levels of multilayer analysis: **(E)** Monolayer analysis of each layer by itself (e.g., with R packages `igraph` or `bipartite`); **(F)** Multilayer analysis without interlayer links (e.g., species/interaction turnover [31], or reducibility with `Muxviz` [25]). **(G)** Multilayer analysis with interlayer links (e.g., community detection with `infompaecology` [26,32]). Discussion on network analysis (E, F, G) is beyond the scope of this work.

with bipartite networks is not straightforward. A bipartite network is stored as an incidence matrix \mathbf{B} with dimensions $P \times A$, where P and A are the number of species in rows (e.g., plants) and columns (e.g., pollinators). To use a SAM, one needs to first transform each layer to a square matrix with dimensions $N = A + P$ such that the adjacency matrix of layer α is:

$$\mathbf{A}^{(\alpha)} = \begin{pmatrix} 0 & B^T \\ B & 0 \end{pmatrix} \quad (1)$$

Then, these square matrices are put together as a SAM. We provide a visual example of this approach in (Fig. 3). A similar approach was taken in a seminal paper that linked food webs and host-parasite networks [11], although without creating a square matrix for each layer and without interlayer links.

A more convenient computational object is an ELL (Fig. 1E). In ELLs, each row encodes an intralayer link (`layer_from=layer_to`) or interlayer link (otherwise) between state nodes. In unipartite networks, the nodes in the `node_from` and `node_to` fields are identical, while they do not overlap in bipartite networks. Because of the data frame format, additional link information (e.g., collection methodology, color) can be added as additional columns. It is then straightforward to construct relations between node and layer IDs that appear in an ELL and other data frames that contain node and layer attributes (Fig. 4). Moreover, unlike matrices, which store links and no links (0 values), link lists only store the links and are more efficient in

memory. This difference is especially relevant for sparse networks (in which a small proportion of the possible links are realized). However, working with sparse matrix objects in R (e.g., package Matrix) is also possible.

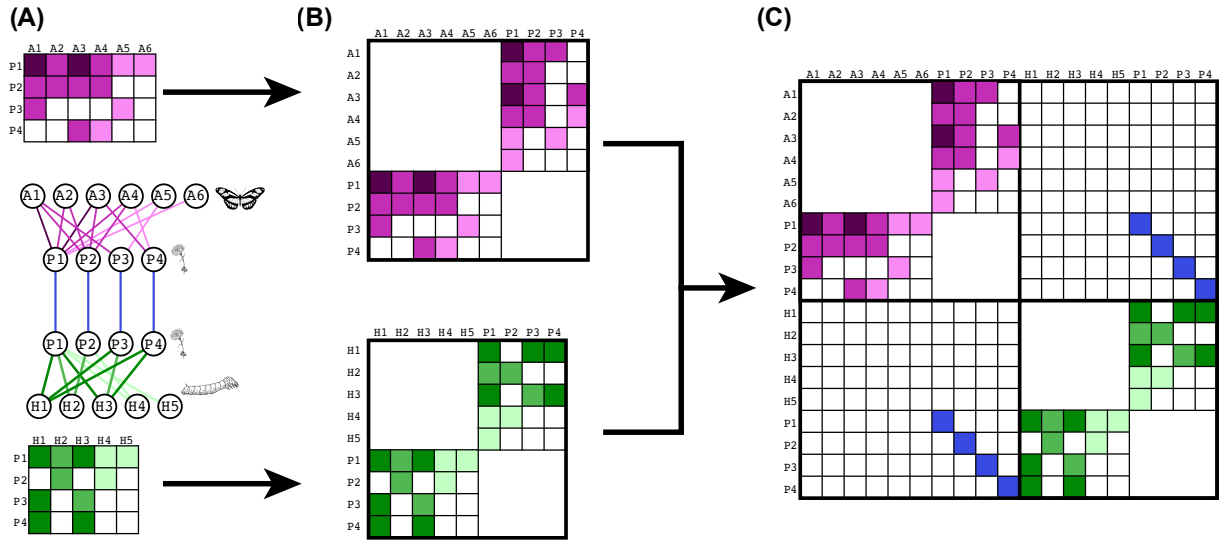


Fig. 3: Toy example for creating a supra-adjacency matrix from bipartite layers. (A) A bipartite network (middle) with plant-pollinator (purple-colored links) and plant-herbivore (green-colored links) layers. Plants are connected via blue interlayer links to form a diagonally-coupled multilayer network [18]. The matrix above each layer is the incidence matrix representation of each layer (without interlayer links). (B) Each incidence matrix is transformed to a symmetric square adjacency matrix as in equation 1. (C) The two adjacency matrices are joined to a supra-adjacency matrix. The white blocks are 0-diagonal blocks where interactions cannot occur due to the bipartite structure of the layers. The blue cells are the interlayer links between plants (e.g., [22]).

2.2 Levels of multilayer analysis

Once a multilayer object is created, network analysis is applied. We identify three levels of analysis, in increasing complexity (Fig. 2E,F,G). (1) *Monolayer analysis*, where each layer is analysed separately and independently. Such analysis can be used to observe how particular network properties vary across layers [33]. (2) *Multilayer analysis without interlayer links*. A typical analysis at this level is that of turnover in species and interactions [7,31,34] across a set of disconnected layers. In multilayer network science there are by now plenty of methods for such analysis [23,26,35]. For instance, Kefi et al [36] grouped species in a multiplex food web according to common interaction patterns of trophic and non-trophic interactions across layers. (3) *Multilayer analysis with interlayer links*. While the two former approaches can provide insights into cross-layer variation (e.g., spatio-temporal variation), they do not explicitly consider interdependency across layers. The main strength of multilayer networks lies in the interlayer links (Fig. 1B,D), which explicitly encode processes that operate between layers [15,18–21]. Interlayer links can be used in structural analysis, such as with community detection [26] and with dynamical analysis. For instance, that of biomass flow in spatial food webs.

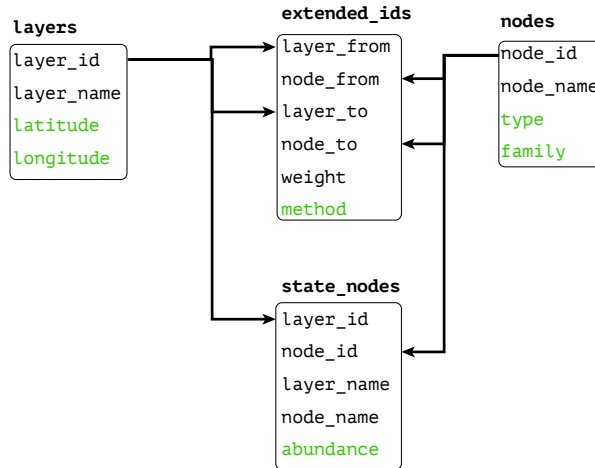


Fig. 4: Data structures. Each rounded rectangle contains the field names of a tibble that is included in the `multilayer` class. The name of the tibble is in bold. The arrows indicate which field names are linked across the tibbles, as in a relational data base (e.g., SQL). Black fields are common to all networks. Green fields are examples for field names that are idiosyncratic. For instance, not all data sets will have coordinates or species abundance data.

3 Ecological multilayer network analysis in practice

In this section we provide practical guidelines and code examples to creating multilayer networks from data. Storing and analysing monolayer networks is by now a common practice and there are many software packages and tools to do so. The commonly used ones in ecology are the R packages `igraph` [24] and `bipartite` [5], though there are implementations for Python (e.g., `NetworkX`, and `Graph-tool`) and Julia (e.g., `EcologicalNetworks.jl` [37] and `LightGraphs.jl`). However, this is not the case for multilayer networks.

Ecologists usually store topology and attribute data as matrices in Excel sheets or csv files (Fig. 2A,C). The manipulation of multiple files can result in errors and generating an ELL or SAM is not intuitive. A smooth EMLN workflow requires the ability to import network data—including intralayer links, interlayer links, and node, link and layer attributes—and store them within a single computational object. It further requires converting between multilayer data structures for downstream analysis. For instance, `MuxViz` [25] uses SAM data structures, while `infomapecology`, an R package for community detection, works with ELL [26]. Hence, converting between SAM, ELL and `igraph` data structures is essential for EMLN analysis. This entire workflow can be facilitated with the EMLN package, which we present here.

3.1 The EMLN package: implementation, availability and code

The R package EMLN consists of ten functions (Table 1). We introduce the package using toy and real data, accompanied by code examples in every step of the way. Considering that software packages are updated more flexibly and constantly than papers, the code is provided on an external website hosted on GitHub rather than within the text: https://ecological-complexity-lab.github.io/emln_package. For the same reason, we provide the full documentation of the functions in the package and an overview here. The package code can be found in <https://github.com/Ecological-Complexity-Lab/emln>.

We start with monolayer networks because importing monolayer network data and converting between data structures (matrix, link list and igraph objects) is the basis for working with layers in EMLNs.

3.2 Monolayer networks

Monolayer networks are typically stored using matrices and link lists. In R, these correspond to classes `matrix` and `data.frame` (or `tibble`), respectively. In EMLN, three functions are used for monolayer network conversion to facilitate monolayer network analysis (Fig. 2B) and to create the layers of EMLNs. The functions `matrix_to_list_bipartite` and `matrix_to_list_unipartite` take as input an incidence or adjacency matrix, respectively and create a `monolayer` object. This object is an R list that contains information on the network (e.g., if it's directed), a data frame with node attributes, and three formats for the network: matrix, link list (as a `tibble`) and an igraph object. The function `list_to_matrix` has the same functionality but takes as input a link list (as a `data.frame`), which can represent either a bipartite or an unipartite network. When using link lists, it is possible to include node and link attributes. We also include in the package a wrapper function for these three, called `create_monolayer_network`, which automatically identifies if the input is a matrix or a link list and creates a monolayer object. Examples for handling monolayer data with these functions are [online](#).

Table 1: The functions of the package EMLN.

Name	Description
<code>matrix_to_list_bipartite</code>	Convert a bipartite monolayer incidence matrix to a <code>monolayer</code> class.
<code>matrix_to_list_unipartite</code>	Convert a unipartite monolayer adjacency matrix to a <code>monolayer</code> class.
<code>list_to_matrix</code>	Convert a monolayer link list to a <code>monolayer</code> class.
<code>create_monolayer_network</code>	A wrapper for the three functions above, which automatically identifies the input and creates a monolayer object.
<code>create_multilayer_network</code>	Create a <code>multilayer</code> class from multiple matrices or link lists.
<code>get_igraph</code>	Convert a <code>multilayer</code> object to a list of igraph objects.
<code>get_sam</code>	Convert a <code>multilayer</code> object to a SAM.
<code>view_emln</code>	View the multilayer networks included in the package.
<code>search_emln</code>	Search for multilayer networks included in the package.
<code>load_emln</code>	Load the multilayer networks included in the package.

3.3 Multilayer networks

In multilayer networks, the SAM and ELL data structures correspond to classes `matrix` and `data.frame` (or `tibble`), respectively. EMLNs are created using the function `create_multilayer_network`. The function takes as input an R list of either matrices or link lists and uses the function `create_monolayer_network` to handle each of those. Layer structure can be either unipartite or bipartite, but it is recommended that all layers are of the same network type, which is typically the case. If interlayer links are present, they can be included as an ELL in a data frame ([online example](#)). Layer attributes can also be included and are obligatory if interlayer links are provided because the layer names in the `layer_name` column in the layer attributes table must correspond

to those in the interlayer links. Link and state node attributes can also be included. This can be done by using link lists as input for layers ([online example](#)). The resulting `multilayer` class is an R that includes the following tibbles ([online example](#)).

- `nodes`: A tibble with the physical nodes. Nodes have a unique id (generated automatically), names (original names provided by the user), and optionally, attributes of the physical nodes, such as taxonomy (e.g., family) or traits.
- `layers`: A tibble with layer attributes. If not provided by the user it will have a `layer_id` and `layer_name` columns generated automatically.
- `state nodes`: A tibble with columns `layer_id`, `node_id`, `layer_name`, `node_name`. Also includes state node attributes if provided as input.
- `extended`: A tibble with columns `layer_from`, `node_from`, `layer_to`, `node_to`, `weight`. If link attributes were provided they will come after the `weight` column. This is an ELL with node and layer names.
- `extended_ids`: The same as `extended` but with node and layer unique IDs.

These tibbles have common field names that can be used to link them, as in a relational data base (Fig. 4).

4 Data included with the EMLN package

We curated a database of 78 ecological multilayer networks collected from different sources (Fig. 5). Details on data preparation are in the EMLN package's wiki: <https://github.com/Ecological-Complexity-Lab/emln/wiki>. The data set includes 12 networks with state node attributes (obtained by running: `search_emln(state_nodes = TRUE)`), and one data set with interlayer links (obtained by running: `search_emln(interlayer = TRUE)`). This collection of data is highly useful for comparative analysis and for practising workflows. The data includes five types of multilayer networks, which are divided into seven types of ecological interaction networks (e.g., pollination, food web) (Fig. 5C).

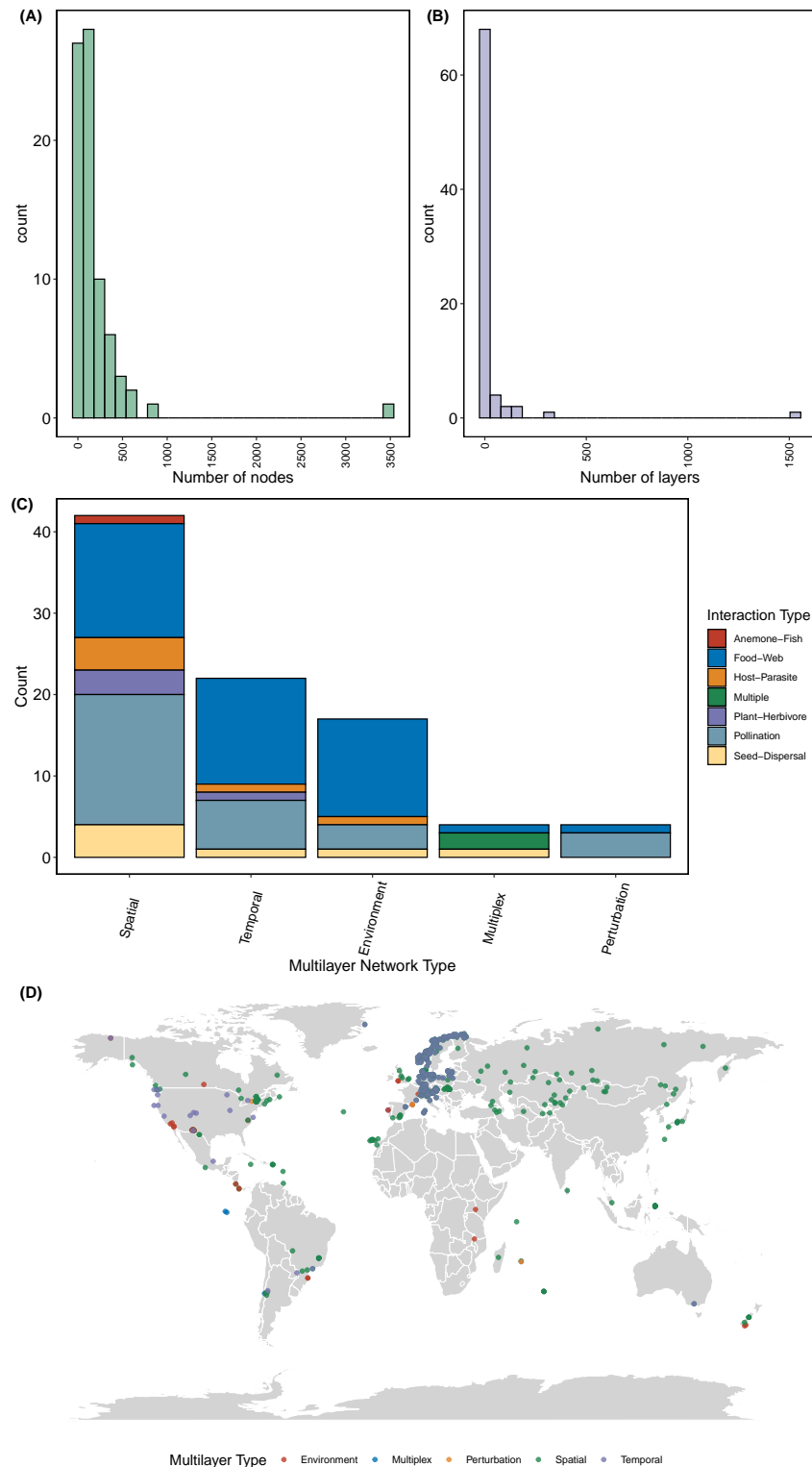


Fig. 5: Summary of the 78 ecological multilayer networks included in the data set. (A) and **(B)** are distributions of the number of nodes and layers in a network, respectively. **(C)** Distribution of ecological interaction types (colours) across multilayer network types. Multiple refers to networks with multiple kinds of layers (e.g., competition and trophic layers). Layer types (x-axis) are spatial (layers are locations), temporal (layers are time points), multiplex (layers are different interaction types), perturbation (layers represent different levels of disturbance or stress on the ecosystem), and environment (layers represent different environments in the same spatial coordinates). **(D)** Location of networks in the world.

Interacting with the data sets is straightforward as can be seen in the example code (https://ecological-complexity-lab.github.io/emln_package/data.html). The function `view_emln` provides a quick way to browse the network database. It is also possible to search for EMLNs with specific criteria using the `search_emln`. This function returns the id(s) of the EMLNs. Then, it is possible to load the data using `load_emln`. This function returns a multilayer class as described before. Hence, it is possible to convert it to SAM or igraph as with user-created EMLNs.

5 Using EMLN with other packages for network analysis

While this paper is not meant to provide guidelines for analysis, we still opted to include a few examples, to illustrate the complete workflow using EMLN in combination with other packages. These analyses and the results are provided in the accompanying website (https://ecological-complexity-lab.github.io/emln_package/analysis_example.html). To facilitate the analysis, the `multilayer` object can be converted using dedicated functions to other formats such as SAM or a list of igraph objects ([online example](#)).

In the node-level analysis, we import a real data set [36] and compare the eigenvector centrality of nodes calculated in each layer separately (monolayer) to that calculated in the SAM (multilayer). To perform monolayer analyses we convert the layers to igraph using the `get_igraph` function (Fig. 2B). This creates a list of igraph objects from the network layers. For a multilayer analysis, we convert the multilayer object to a SAM using `get_sam` (Fig. 2D). For the mesoscale-level analysis we use a temporal EMLN included in the package. After loading the network, we run `infomap` to unravel the temporal modular structure of the network. We leverage the fact that the `multilayer` object includes a built-in table called `extended_ids`, which is generated by `create_multilayer_network` automatically, because `infomap` receives as input node ids rather than node names.

6 Conclusions

The analysis of EMLNs can provide novel insights into the organisation and stability of ecological communities. However, because multilayer networks are, by definition, highly complex and multi-dimensional, handling data and subsequent analysis are challenging. Providing practical guidelines and dedicated tools is a necessary step to allow more researchers to take part in this cutting-edge aspect of network ecology. Here, we described the workflow of EMLN analysis and provided practical and visual guidelines, as well as detailed code examples in a dedicated website. The R package we provide standardises data structures and also enables an easy conversion for downstream analysis with common packages and includes practice data sets. We hope that this paper provides a springboard for ecologists interested in expanding their toolkit to include the multilayer nature of ecological networks in their research.

Acknowledgements

This work was supported by an Israel Science Foundation research grant 1281/20 to SP. We thank Ofir Segev for help with data entry and members of the Ecological Complexity Lab for valuable discussions and comments on the draft.

Author contributions

S.P. conceived the study; N.F. and S.F. collected and curated the data; N.F., S.F. and S.P. wrote the R package; I.T. analysed the data; N.F. and S.P. led the paper writing. All authors contributed to paper writing.

Conflict of interest

None of the authors have any conflict of interest to report.

References

1. Proulx, S. R., Promislow, D. E. L. & Phillips, P. C. Network thinking in ecology and evolution. *Trends Ecol. Evol.* **20**, 345–353. doi:[10.1016/j.tree.2005.04.004](https://doi.org/10.1016/j.tree.2005.04.004) (2005).
2. Bascompte, J. Networks in ecology. *Basic Appl. Ecol.* **8**, 485–490. doi:[10.1016/j.baae.2007.06.003](https://doi.org/10.1016/j.baae.2007.06.003) (2007).
3. Poisot, T., Stouffer, D. B. & Kéfi, S. Describe, understand and predict: why do we need networks in ecology? *Funct. Ecol.* **30**, 1878–1882. doi:[10.1111/1365-2435.12799](https://doi.org/10.1111/1365-2435.12799) (2016).
4. Allesina, S. *Going Big* 2020. doi:[10.2307/j.ctvs9fh2n.31](https://doi.org/10.2307/j.ctvs9fh2n.31).
5. Dormann, C. F., Fründ, J., Blüthgen, N. & Gruber, B. Indices, graphs and null models: analyzing bipartite ecological networks. *The Open Ecology Journal* **2**, 7–24. doi:[10.2174/1874213000902010007](https://doi.org/10.2174/1874213000902010007) (2009).
6. Hagen, M. *et al.* in *Advances in Ecological Research* (ed Ute Jacob And) 89–210 (Academic Press, 2012). doi:[10.1016/B978-0-12-396992-7.00002-2](https://doi.org/10.1016/B978-0-12-396992-7.00002-2).
7. Trøjelsgaard, K., Jordano, P., Carstensen, D. W. & Olesen, J. M. Geographical variation in mutualistic networks: similarity, turnover and partner fidelity. *Proc. Biol. Sci.* **282**, 20142925. doi:[10.1098/rspb.2014.2925](https://doi.org/10.1098/rspb.2014.2925) (2015).
8. Rasmussen, C., Dupont, Y. L., Mosbacher, J. B., Trøjelsgaard, K. & Olesen, J. M. Strong impact of temporal resolution on the structure of an ecological network. *PLoS One* **8**, e81694. doi:[10.1371/journal.pone.0081694](https://doi.org/10.1371/journal.pone.0081694) (2013).
9. Chacoff, N. P., Resasco, J. & Vázquez, D. P. Interaction frequency, network position, and the temporal persistence of interactions in a plant-pollinator network. *Ecology* **99**, 21–28. doi:[10.1002/ecy.2063](https://doi.org/10.1002/ecy.2063) (2018).
10. Pilosof, S. *et al.* Competition for hosts modulates vast antigenic diversity to generate persistent strain structure in *Plasmodium falciparum*. *PLoS Biol.* **17**, e3000336. doi:[10.1371/journal.pbio.3000336](https://doi.org/10.1371/journal.pbio.3000336) (2019).
11. Lafferty, K. D., Dobson, A. P. & Kuris, A. M. Parasites dominate food web links. *Proc. Natl. Acad. Sci. U. S. A.* **103**, 11211–11216. doi:[10.1073/pnas.0604755103](https://doi.org/10.1073/pnas.0604755103) (2006).
12. Melián, C. J., Bascompte, J., Jordano, P. & Krivan, V. Diversity in a complex ecological network with two interaction types. *Oikos* **118**, 122–130. doi:[10.1111/j.1600-0706.2008.16751.x](https://doi.org/10.1111/j.1600-0706.2008.16751.x) (2009).

13. Fontaine, C. *et al.* The ecological and evolutionary implications of merging different types of networks. *Ecol. Lett.* **14**, 1170–1181. doi:[10.1111/j.1461-0248.2011.01688.x](https://doi.org/10.1111/j.1461-0248.2011.01688.x) (2011).
14. Kéfi, S. *et al.* Network structure beyond food webs: mapping non-trophic and trophic interactions on Chilean rocky shores. *Ecology* **96**, 291–303. doi:[10.1890/13-1424.1](https://doi.org/10.1890/13-1424.1) (2015).
15. Hervías-Parejo, S. *et al.* Species functional traits and abundance as drivers of multiplex ecological networks: first empirical quantification of inter-layer edge weights. *Proc. Biol. Sci.* **287**, 20202127. doi:[10.1098/rspb.2020.2127](https://doi.org/10.1098/rspb.2020.2127) (2020).
16. De Domenico, M. *et al.* Mathematical formulation of multilayer networks. *Phys. Rev. X* **3**, 041022. doi:[10.1103/PhysRevX.3.041022](https://doi.org/10.1103/PhysRevX.3.041022) (2013).
17. Kivela, M. *et al.* Multilayer networks. *J. Complex Networks* **2**, 203–271. doi:[10.1093/comnet/cnu016](https://doi.org/10.1093/comnet/cnu016) (2014).
18. Pilosof, S., Porter, M. A., Pascual, M. & Kéfi, S. The multilayer nature of ecological networks. *Nat Ecol Evol* **1**, 0101. doi:[10.1038/s41559-017-0101](https://doi.org/10.1038/s41559-017-0101) (2017).
19. Hutchinson, M. C. *et al.* Seeing the forest for the trees: Putting multilayer networks to work for community ecology. *Funct. Ecol.* **33** (ed Godoy, O.) 206–217. doi:[10.1111/1365-2435.13237](https://doi.org/10.1111/1365-2435.13237) (2019).
20. Costa, J. M. *et al.* Species temporal persistence promotes the stability of fruit-frugivore interactions across a 5-year multilayer network. *J. Ecol.* **10** (ed Bartomeus, I.) 45. doi:[10.1111/1365-2745.13391](https://doi.org/10.1111/1365-2745.13391) (2020).
21. Shapiro, J. T. *et al.* Multilayer networks of plasmid genetic similarity reveal potential pathways of gene transmission. *ISME J.*, 1–11. doi:[10.1038/s41396-023-01373-5](https://doi.org/10.1038/s41396-023-01373-5) (2023).
22. Vitali, A. *et al.* Invasive species modulate the structure and stability of a multilayer mutualistic network. *Proc. Biol. Sci.* **290**, 20230132. doi:[10.1098/rspb.2023.0132](https://doi.org/10.1098/rspb.2023.0132) (2023).
23. Bianconi, G. *Multilayer Networks : Structure and Function* (Oxford University Press USA - OSO, Oxford, UNITED KINGDOM, 2018).
24. Csárdi, G & Nepusz, T. The igraph software package for complex network research. *International Journal Complex Systems* **1695**, 1695 (2006).
25. De Domenico, M. D., Porter, M. A. & Arenas, A. MuxViz: a tool for multilayer analysis and visualization of networks. *J. Complex Networks* **3**, 159–176. doi:[10.1093/comnet/cnu038](https://doi.org/10.1093/comnet/cnu038) (2015).
26. Farage, C., Edler, D., Eklöf, A., Rosvall, M. & Pilosof, S. Identifying flow modules in ecological networks using Infomap. *Methods Ecol. Evol.* **12**, 778–786. doi:[10.1111/2041-210x.13569](https://doi.org/10.1111/2041-210x.13569) (2021).
27. De Domenico, M. *Multilayer Networks: Analysis and Visualization: Introduction to muxViz with R* (Springer Nature, 2022).
28. Boccaletti, S. *et al.* The structure and dynamics of multilayer networks. *Phys. Rep.* **544**, 1–122. doi:[10.1016/j.physrep.2014.07.001](https://doi.org/10.1016/j.physrep.2014.07.001) (2014).

29. De Domenico, M. *Multilayer Networks Illustrated* <https://github.com/manlius/muxViz/blob/master/gui-old/theory/README.md>. 2022. doi:10.17605/OSF.IO/GY53K.
30. Timóteo, S., Correia, M., Rodríguez-Echeverría, S., Freitas, H. & Heleno, R. Multilayer networks reveal the spatial structure of seed-dispersal interactions across the Great Rift landscapes. *Nat. Commun.* **9**, 140. doi:10.1038/s41467-017-02658-y (2018).
31. Poisot, T. *et al.* The dissimilarity of species interaction networks. *Ecol. Lett.* **15**, 1353–1361. doi:10.1111/ele.12002 (2012).
32. De Domenico, M., Lancichinetti, A., Arenas, A. & Rosvall, M. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Phys. Rev. X* **5**, 011027. doi:10.1103/PhysRevX.5.011027 (2015).
33. Olesen, J. M., Stefanescu, C. & Traveset, A. Strong, long-term temporal dynamics of an ecological network. *PLoS One* **6**, e26455. doi:10.1371/journal.pone.0026455 (2011).
34. CaraDonna, P. J. *et al.* Interaction rewiring and the rapid turnover of plant-pollinator networks. *Ecol. Lett.* doi:10.1111/ele.12740 (2017).
35. De Bacco, C., Power, E. A., Larremore, D. B. & Moore, C. Community detection, link prediction, and layer interdependence in multilayer networks. *Phys Rev E* **95**, 042317. doi:10.1103/PhysRevE.95.042317 (2017).
36. Kéfi, S., Miele, V., Wieters, E. A., Navarrete, S. A. & Berlow, E. L. How structured is the entangled bank? The surprisingly simple organization of multiplex ecological networks leads to increased persistence and resilience. *PLoS Biol.* **14**, e1002527. doi:10.1371/journal.pbio.1002527 (2016).
37. Poisot, T., Bélisle, Z., Hoebeke, L., Stock, M. & Szefer, P. EcologicalNetworks.jl: analysing ecological networks of species interactions. *Ecography* **42**, 1850–1861. doi:10.1111/ecog.04310 (2019).