A dynamical perspective to community detection in ecological networks

Carmel Farage [1], Daniel Edler [2,4,5], Anna Eklöf [3], Martin Rosvall [2], and Shai Pilosof* [1]

[1] *Department of Life Sciences, Ben-Gurion University of the Negev, Beer-Sheva, Israel*
[2] *Integrated Science Lab, Department of Physics, Umeå University, Umeå, Sweden*
[3] *Division of Theoretical Biology, Department of Physics, Chemistry and Biology, Linköping University, Linköping, Sweden*
[4] *Gothenburg Global Biodiversity Centre, Gothenburg, Sweden*
[5] *Department of Biological and Environmental Sciences, University of Gothenburg, Gothenburg, Sweden*
[*] *pilos@post.bgu.ac.il*

April 14, 2020

Abstract

1. Ecological networks often have a modular structure with groups of nodes that interact strongly within the group and weakly with other nodes in the network. A modular structure can reveal underlying dynamic ecological or evolutionary processes, influence dynamics that operate on the network, and affect network stability. Consequently, detecting modular structures is a fundamental analysis in network ecology.

2. Although many ecological networks describe flows, such as biomass flows in food webs, disease transmission or temporal interaction networks, most modularity analyses have ignored these dynamics, which can hinder our understanding of the interplay between structure and dynamics. Here we present Infomap, an established community-detection method that maps flows on networks into modules based on random walks. To illustrate and facilitate the usage of Infomap on ecological networks, we also provide a fully documented repository with technical explanations and examples, and an open-source R package, infomapecology.

3. Infomap is a flexible tool that works on many network types, including directed and undirected networks, bipartite (e.g., plant-pollinator) and unipartite (e.g., food webs) networks, and multilayer networks (e.g., spatial, temporal or multiplex networks with several interaction types). Infomap can also use additional relevant ecological data, which we illustrate by investigating how taxonomic affiliation as node metadata influences the modular structure.

4. Flow-based modularity analysis is relevant across all of ecology and transcends to other biological and non-biological disciplines. It includes natural ways to analyse directed networks, take advantage of metadata, and analyse multilayer networks. A dynamical approach to detecting modular structure has strong potential to provide new insights into the organisation of ecological networks, that are more relevant to the system/question at hand, compared to methods that ignore dynamics.

Key-words: dynamics, ecological networks, flow, Infomap, modules, multilayer.

1 Introduction

Ecological networks of nodes and links represent interactions between entities in ecological systems. The nodes can represent, for example, species, habitat patches or individuals. The links can represent, for example, feeding interactions between predators and prey, dispersal between different habitats or encounter rates. Examples of ecological networks include consumer-resource networks such as predation and pollination networks, meta-population and community networks where links encode dispersal between patches, and contact networks between individuals. Regardless of network type, a long-lasting stance is that network structure is linked to its dynamics (Pascual and Dunne, 2006), including dynamics of network assembly (Bascompte and Stouffer, 2009; Maynard et al., 2018; Pilosof et al., 2020), dynamics of populations related to the stability of the network (May, 1973; Allesina and Tang, 2015) and dynamics of flow across the network (Brose et al., 2006; Urban et al., 2009; Tur et al., 2016). Characterising network structure has therefore been a major endeavour of network ecology since its inception.

One of the most commonly explored structures in ecological networks is the partitioning of a network into groups such as clusters of organisms with similar traits, patches tightly linked by movement of individuals or groups linked by feeding interactions in food webs. Revealing these groups helps to understand the interplay between the structure and function of ecosystems (Olesen et al., 2007; Baskerville et al., 2011; Stouffer and Bascompte, 2011; Fletcher et al., 2013; Thébault, 2013). Loosely defined, these groups – also known as 'compartments', 'communities' or 'modules' in network science – are groups of nodes that are more tightly connected among each other than to nodes in other groups. To avoid confusion in terminology between network science and ecology, we use the term 'modules' (in ecology communities are assemblages of species). Compartmentalisation, or 'modularity', is an important example of the structure-dynamics nexus. For example, modularity can make ecological communities locally stable (Grilli et al., 2016), increase species persistence in food webs (Stouffer and Bascompte, 2011) and slow down the spread of disease (Fortuna et al., 2009). Common to all these different kinds of dynamics and networks is the fact that high connectivity within modules tends to locally limit perturbations and network-wide flows, preventing effects from spreading to other parts of the network.

There are three main ways to detect modules in networks (Rosvall et al., 2018): (i) By maximising the internal density of links within groups of nodes (Newman and Girvan, 2004; Olesen et al., 2007; Thébault, 2013); (ii) by identifying structurally equivalent groups in which nodes connect to others with equal probability, typically studied using stochastic-block models (Holland et al., 1983), known as the 'group model' in ecology (Allesina and Pascual, 2009)); and (iii) by optimally describing modular flows on networks (Rosvall and Bergstrom, 2008; Rosvall et al., 2010). Because these approaches are developed for different purposes, they use different mathematical functions and algorithms to detect an 'optimal' partition of a network. Therefore, there is no single 'true' network partition (Peel et al., 2017). Instead, the method one applies should match the question one aims to answer (Ghasemian et al., 2019; Rosvall et al., 2018). For example, community detection using the group model (Allesina and Pascual, 2009) can provide insights into the roles species play in the network with groups of species that feed on, or are fed by, similar species – insights that cannot be obtained by identifying flow modules or maximising internal density. In contrast, if one is interested in understanding the effect of dispersal on metapopulation dynamics, methods based on flow may be more relevant.

In ecology, the vast majority of studies have taken the first approach and used community-detection algorithms that aim to maximise the objective function Q, called modularity (Newman, 2004; Olesen et al., 2007). Although most approaches for analysing modularity use unweighted networks, there are several generalisations of modularity, including Q for bipartite (Barber, 2007), directed (Guimerà et al., 2007), weighted (Dormann and Strauss, 2014; Beckett, 2016), and multilayer (Mucha et al., 2010) networks. Because of modularity's popularity, there are are also several optimisation algorithms (Blondel et al., 2008; Traag et al., 2019). While modularity analysis using Q or its generalisations or optimisation algorithms has provided many insights into ecological systems, none of these seek to capture network flows. By flows we do not refer to simply weighted links, but to the actual ecological meaning of the links. For example, links in a food web describe a flow of energy but due to challenges in sampling are rarely weighted. Many ecological systems describe flows on networks, including biomass flow in food webs, movement of individuals between patches, gene flow among individuals in social networks or space (Fletcher et al., 2013), and disease transmission. Therefore, understanding how the network structure organises into flow modules can provide novel insights, which are more relevant to the system at hand, compared to approaches that either treat the network as

undirected or maximise internal density.

To fill this gap, here we focus on detecting modules by describing flows on networks. We present and use a well-established method for detecting flow-based modules called Infomap. Infomap minimises an objective function known as the 'map equation', which measures how much a modular partition of a network can compress a description of flows on the network (Rosvall and Bergstrom, 2008). Infomap has been thoroughly described mathematically and computationally (Rosvall and Bergstrom, 2008; Rosvall et al., 2010; Rosvall and Bergstrom, 2011; Rosvall et al., 2014) and is widely used in non-ecological disciplines, but only in a handful of papers in ecology (Pilosof et al., 2019; Bernardo-Madrid et al., 2019; Pilosof et al., 2020). Instead of comparing or benchmarking Infomap against other methods, which has already been done (Lancichinetti and Fortunato, 2009; Aldecoa and Marín, 2013), we explain how Infomap works on ecological networks and provide several hands-on examples using empirical data. In this way, we illustrate in which cases Infomap can be applied to ecological networks, demonstrate the insights this method can provide, and provide technical guidelines for using it. To facilitate analyses in future studies, we provide a fully-documented GitHub repository, as well as a dedicated R package called *infomapecology*.

2 Infomap and the map equation objective function

Infomap has several advantages for ecological research. First, it can be applied to networks with any kind of structure, including directed/undirected, weighted/unweighted, unipartite/bipartite, and multilayer networks. Second, it is computationally effective, facilitating studies of large networks or comparing observed networks with many randomised networks. Third, it can incorporate node metadata, explicitly considering information such as taxonomy or gender in the partitioning to modules. Fourth, it can detect hierarchical structures (modules within modules). Finally, it has online documentation and an active development team that has made Infomap user-friendly and flexible with standalone code, R and Python integration, and online interactive visualisations. These advantages make Infomap a highly accessible tool for community detection that can be applied to virtually any kind of ecological system.

2.1 The mechanics of Infomap: duality between structure and compressing information

To understand how Infomap works, it is helpful to first understand the general approach for community detection (see details in Supplementary Information S1). Community detection refers to the procedure of identifying groups of interacting nodes based on the structural properties of the network. A particular assignment of the nodes into modules is referred to as a network partition. As even very small networks can have an enormous number of possible partitions, optimisation techniques are needed. Search algorithms calculate an objective function for a given partition, then make a small change in the partition, such as moving a node from one module to another, and test if this change has improved the value of the objective function. Community-detection algorithms, including Infomap, Louvain (Blondel et al., 2008), MODULAR (Marquitti et al., 2014), QuanBiMo (Dormann and Strauss, 2014) or DIRTLPawb+ (Beckett, 2016), differ in the search algorithms and objective functions they apply. Infomap optimises the objective function known as the map

equation using a modified Louvain (Blondel et al., 2008) search algorithm.

Mechanistically, Infomap operates by applying a Louvain-type search algorithm to find the partition that best compresses a description of flows modelled by a random walker on the network. The random walker moves across nodes in a way that depends on the direction and weight of the links, and tends to stay longer in dense areas representing modules of highly connected nodes. For a given partition of the network, the map equation converts the time spent in and between modules to an information-theoretic measure of the shortest possible modular description of the random walker's movements on the network. The partition that requires the *least* amount of information to describe modular flows on the network *minimises* the map equation and corresponds to the optimal partition. Understanding the mechanics of the map equation is important for an adequate application to ecological systems and we explain it in the next sections.

2.2 Undirected networks

The map equation operates by exploiting the duality between network structure and the compression of information. We illustrate this approach and derive the map equation using a simple example. For an extensive description of the mechanics, see ref. (Rosvall and Bergstrom, 2008; Rosvall et al., 2010; Bohlin et al., 2014). In our example, we consider an undirected network with six nodes and compare a single-module with a two-module solution (Fig. 1).

The map equation takes the flows on nodes and links as inputs to measure the amount of information required to describe random-walker movements on the nodes within and between modules. Hence, we first derive the node and link visit rates. First, each node receives a unique identifier called 'code word' (see below for details how). *Node visit rates* give the use rate of code words that specify the random walker's position within modules. *Link visit rates* of the links crossing module boundaries give the use rate of code words that specify transitions between modules. In undirected networks, normalising the link weights by twice the total link weights gives the link visit rates in each direction. Aggregating the incoming flows on each node gives the node visit rates. In the undirected schematic network in Fig. 1a, there are seven links. Assuming that each undirected link has a weight of 1 in each direction, the total incoming link weight is 14. Therefore, each link carries flows of link visit rate 1/14 in each direction. Nodes with two links have a node visit rate of 2/14, and nodes with three links have a node visit rate of 3/14. Based on these rates, the map equation measures the amount of information required to describe flows between and within modules.

While it is possible to compute the modular description length and compression rate without deriving the actual code words given to nodes, for a thorough understanding, we first derive them using Huffman codes (Huffman, 1952). Huffman codes map symbols or events to binary codes with variable lengths such that frequent events receive short codes and infrequent events receive long code words. In this way, Huffman codes can compress event-sequence descriptions. We use the node visit rates to build a Huffman tree that minimises the average code length by repeatedly combining the remaining least infrequent events (Fig. 1c). In a Huffman tree, left and right branches are assigned binary letters of 0 and 1, respectively. The map equation uses these binary letters to assign each node a code word based on the path from the tip of the tree towards the node. Together they form a 'module codebook' with unique code words. In the single-module solution in Fig. 1a and c, the unique code words enable an unambiguous description of a walk on the network. For example, the walk C-A-B is encoded by the eight-bit sequence 10000001, 10 for C, 000 for A, and 001 for B. The code words can be concatenated because Huffman codes are prefix-free, that is no code word is

the beginning of any other code word. The average code length, $L^{\text{Huffman}} = \sum_j p_j l_j$, with l_j for the code word length of node j weighted by its visit rate p_j, gives the average amount of information in bits required to describe a step on the network (Fig. 1c).

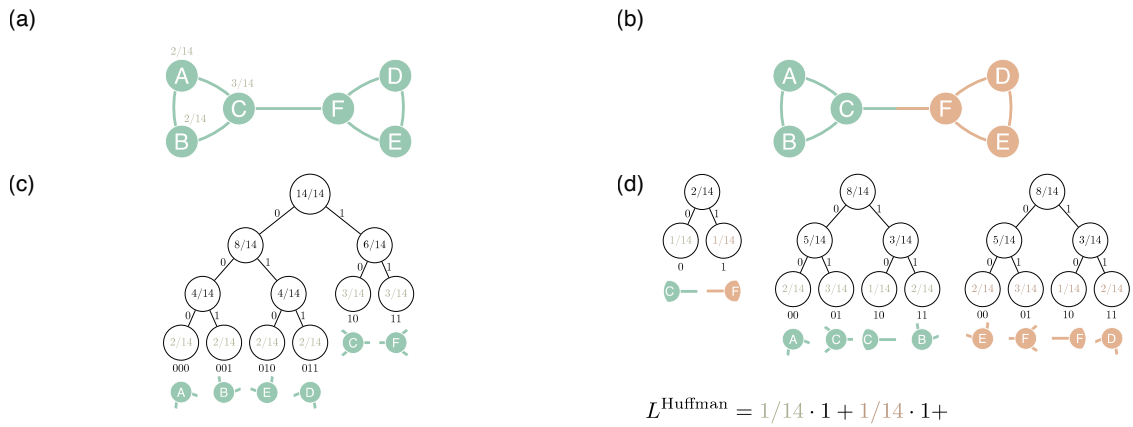
In the two-module solution, we can use the same code words for different nodes in different modules, enabling shorter code words that save information (Fig. 1b and d). However, this description is unambiguous only if we know in which module the walk is taking place. To describe movements within and also between modules, the map equation therefore uses two types of codebooks: one module codebook for each module to describe movements within modules and an 'index codebook' to describe movements between modules. For each module i, an additional exit code word in the module's codebook signals that the walker exits the module. In the index codebook, a code word for each module i signals that the walker enters module i. With these two types of codebooks, we can describe uniquely decodable walks.

The lengths of the exit and entry code words depend on the *module exit rate* and *module entry rate*, respectively, which are equal for undirected networks. In the two-module solution, the index codebook has code words for "enter green" and "enter orange", which both occur at rate 1/14 (Fig. 1d). With two modules, the average code length is the sum of all code word lengths in the index and module codebooks weighted by their use rates (Fig. 1d). With the two-module coding scheme, the walk C-A-B is now encoded by the six-bit sequence 010011, 01 for C, 00 for A, and 11 for B. The module-changing walk A-F-D is encoded by 011010111, 01 for C, 10 for exiting the green module, 1 for entering the orange module, 01 for F, and 11 for D. On average, the two-module solution requires $2.57 - 2.43 = 0.14$ fewer bits (Fig. 1c and d).

Derived from the node visit, module entry, and module exit rates from the random-walk model, the Huffman coding provides the shortest decodable description of flows in the network. However, to take advantage of the duality between compression and finding network structure, we are not interested in an actual description but only in estimating the description length. **The map equation allows us to bypass the Huffman encoding to directly calculate the theoretical limit L of the description length.** From Shannon's source coding theorem, the per-step average description length of each codebook cannot be shorter than the entropy H of its events $\mathcal{P}$, $L \geq H(\mathcal{P})$ (Shannon, 1948). Using this theorem, the map equation allows us to calculate the average length of the code describing a step of the random walk by weighing the average length of code words from the index codebook and the module codebooks by their rates of use. Summing the terms for the index codebook and the module codebooks, we obtain the map equation, which measures the theoretical minimum description length L in bits given a network partition M (Rosvall and Bergstrom, 2008; Rosvall et al., 2010),

$$L(\mathsf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p^i_{\circlearrowright} H(\mathcal{P}^i), \tag{1}$$

where $H(\mathcal{Q})$ and $H(\mathcal{P}^i)$ are the frequency-weighted average length of code words in the index codebook and the codebook of module i, respectively. These entropy terms are weighted by the rate at which the codebooks are used, the rate of entering any module, $q_{\curvearrowright}$, for the index codebook and the sum of the node-visit rates and the exit rate in module i, $p^i_{\circlearrowright}$, for module codebook i. For the examples in Fig. 1, $L(\mathsf{M}_1) \approx 2.56$ for the one-module solution and $L(\mathsf{M}_2) \approx 2.32$ for the two-module solution. In both cases, $L < L^{\text{Huffman}}$, and again the two-module solution requires fewer bits and better represents the network flows.



$$L^{\text{Huffman}} = 2/14 \cdot 3 + 2/14 \cdot 3 + 3/14 \cdot 2 + 2/14 \cdot 3 + 2/14 \cdot 3 + 3/14 \cdot 2 \approx 2.57 \text{ bits}$$

$$L = H(2/14, 2/14, 3/14, 2/14, 2/14, 3/14) \approx 2.56 \text{ bits}$$

$$L^{\text{Huffman}} = 1/14 \cdot 1 + 1/14 \cdot 1 + 2/14 \cdot 2 + 2/14 \cdot 2 + 3/14 \cdot 2 + 1/14 \cdot 2 + 2/14 \cdot 2 + 2/14 \cdot 2 + 3/14 \cdot 2 + 1/14 \cdot 2 \approx 2.43 \text{ bits}$$

$$L = 2/14 H(1/14, 1/14) + 8/14 H(2/14, 2/14, 3/14, 1/14) + 8/14 H(2/14, 2/14, 3/14, 1/14) \approx 2.32 \text{ bits}$$

Figure 1: Schematic undirected network. (a) A one-module solution with node visit rates on nodes A-C. (b) A two-module solution. (c) A Huffman tree for the one-module solution. Leaf nodes correspond to nodes in the network. Internal nodes indicate the sum of the node visit rates. Each edge of the tree is assigned a 0 or 1. The code word of each leaf in the Huffman tree is the concatenated 0s and 1s from the top and along the branch of the tree. L^{Huffman} is the node-visit weighted average description length required to describe a step on the network using a binary code. (d) The two-module solution uses an index codebook for steps between modules and two module codebooks for steps within modules. In (c) and (d) L is the sum of the average code length of each codebook weighted by their use rates and is the theoretical lower limit of the description length.

2.3 Directed networks

In directed networks, the procedure for deriving node and link visit rates depends on whether links represent real flows or constraints on flows. *Real flows* are, for example, measured biomass or energy transfer between taxa in food webs or dispersal rates between patches. In contrast, *constraints on flows* are, for example, measured contact rates between individuals that can constrain the dispersal of a pathogen where no actual disease flow has been observed. Another example is a network of plant specimens projected from a bipartite plant-pollinator network, which represents the possible flows of pollen between plant individuals. Again, actual pollen movement has not been measured but it will be constrained by the weighted structure.

When link directions and weights represent real directed flows, normalising and aggregating link weights gives link visit, node visit, module entry, and module exit rates in the same way as for undirected networks. However, for directed networks module entry and exit rates need not be the same. For example, assuming that the 20 links in the schematic directed network in Fig. 2 represent real flows, the total outgoing link weight is 20. Therefore, each link carries flows at rate 1/20, nodes with one incoming link have visit rate 1/20, and nodes with two incoming links have visit rate 2/20. For a one-module solution, the entropy of the node visit rates gives the description length 3.92 bits (Fig. 2a). For the optimal four-module solution, the map equation also uses module entry and exit rates. Since both the incoming link to and the outgoing link from a module carry flows at rates 1/20, both the module entry rate and the module exit rate is 1/20 for

each module. With one index codebook and four module codebooks, each weighted by their use rate, the description length is 3.10 bits (Fig. 2b). Since the four-module solution requires fewer bits, it provides the better modular representation of the network flows.

When the directed and possibly weighted links represent constraints on flows like floodgates between the nodes, we need to model what system-wide network flows the constraints induce. The simplest approach is to derive link and node visit rates from a random-walk model, which requires an iterative process akin to the PageRank algorithm (Brin and Page, 1998). First, each node receives an equal amount of flow volume. Then, iteratively until all node visit rates are stable, each node distributes all its flow volume to its neighbours proportionally to the outgoing link weights. To guarantee a stable and unique solution, each node teleports some proportion of its flow volume to any nodes proportional to their incoming link weights (see dynamic visualisation available on `https://www.mapequation.org/apps/MapDemo.html`). This flow redistribution prevents biased solutions from, for example, accumulation of flows in areas with no outgoing links. The teleportation rate is adjustable but typically 15 percent. Because the teleportation steps are not encoded, variations in the teleportation rate have an insignificant effect on the results (Lambiotte and Rosvall, 2012). When the nodes' visit rates are stable, a node's visit rate times the relative weight of an outgoing link gives the link visit rate. Aggregating flows on links that exit and enter a module gives the module entry and exit rates as for undirected networks. Because teleportation flows are excluded when deriving link visit rates, module entry and exit rates need not be the same.

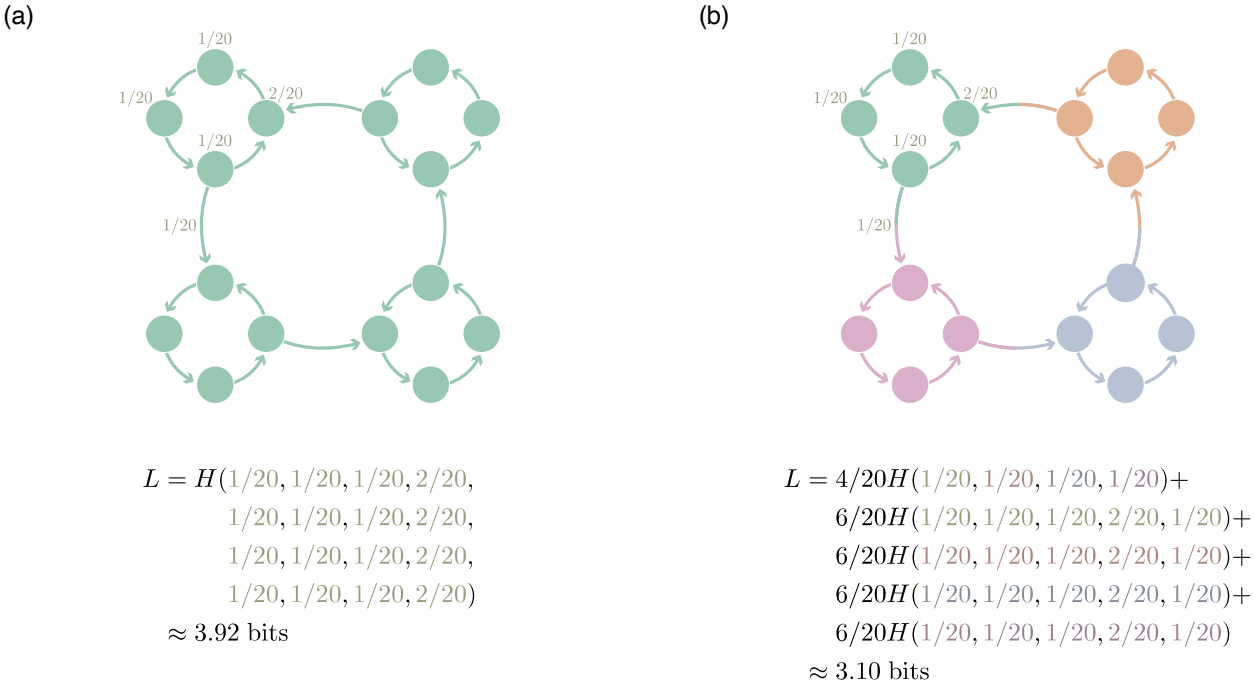
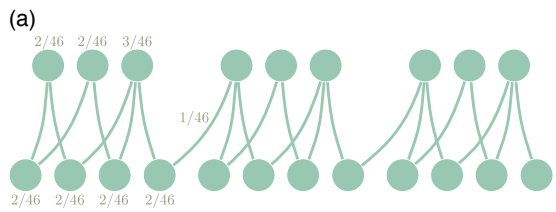


Figure 2: Schematic directed network with real flows. (a) A one-module solution. (b) A four-module solution. The one-module solution has a single module codebook and no index codebook. Therefore, the rates for calculating the entropy are the incoming rates of all the 16 nodes. The one-module solution has one index codebook with four entry rates of 1/20 and four identical module codebooks with three node visit rates of 1/20, one of 2/20, and one exit rate of 1/20. All rates have denominator 20 because each of the 20 links carries an equal amount of flow. The rates of using the module and index codebooks are $p^i_{\circlearrowleft} = 6/20$ and $q_{\curvearrowright} = 4/20$, respectively.

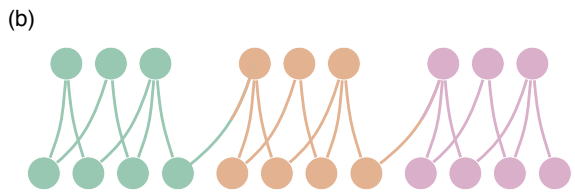
2.4 Bipartite networks

Bipartite networks consist of two sets of nodes with links only between the two sets as in plant-pollinator and host-parasite networks. Bipartite networks are generally treated as unipartite networks. For example, a plant-pollinator network represents constraints on pollen flow. Modules are then defined using nodes from both sets, for example, modules that contain both plants and pollinators. This approach is conceptually different to the calculation of modularity using Q, where a specific modification to the objective function is necessary (Barber, 2007).

In some networks, however, links do not represent real or constrained flows. Imagine for example a bipartite network in which plants are connected to patches in which they occur. In that case treating the network as unipartite can still be a valid method to detect modules composed of plants and patches that "interact" densely. Yet for such cases Infomap offers a second approach. If the objective is to find modules of plants that tend to cooccur in the same patches, Infomap can encode one type of nodes (plants) by starting the dynamics on that node set and always taking two steps between encodings of the walker's position. This approach will still generate modules with plants and patches, but the patches do not contribute information to that structure. Because Infomap "projects" the bipartite network into a unipartite network with twice as long between two encodings, the two-step projection typically results in fewer and larger modules compared to treating the network as an unipartite network.



$$L = H(2/46, 2/46, 3/46, 2/46, 2/46, 2/46, 2/46, 3/46, 2/46, 3/46, 2/46, 2/46, 2/46, 2/46, 3/46, 2/46, 3/46, 2/46, 2/46, 2/46, 2/46) \approx 4.37 \text{ bits}$$



$$L = 4/46H(1/46, 2/46, 1/46) + 16/46H(2/46, 2/46, 3/46, 2/46, 2/46, 2/46, 2/46, 1/46) + 18/46H(3/46, 2/46, 3/46, 2/46, 2/46, 2/46, 2/46, 2/46) + 17/46H(3/46, 2/46, 3/46, 2/46, 2/46, 2/46, 2/46, 1/46) \approx 3.41 \text{ bits}$$

Figure 3: Schematic undirected bipartite network. The description length according to the map equation for (a) a one-module solution and (b) a three-module solution. The bipartite network is treated as unipartite and the modules contain nodes of both types. There are 23 undirected links of weight 1, rendering a total link weight of 46.

2.5 Multilayer networks

Multilayer networks enable the representation of multiple layers of data as a single network object (De Domenico et al., 2013; Kivelä et al., 2014). Layers are themselves networks that can represent different points in time or space, or different interaction types such as parasitism and predation. *Interlayer links* encode ecological processes that operate across layers, such as dispersal, gene flow or population dynamics and represent dependencies between layers. The relative weight between inter- and intralayer links therefore indicates the relative contribution of the processes that operate within and between layers to network structure and dynamics. For example, the relative contribution of movement of individuals within patches and between patches to the structure of dispersal. While the application of multilayer networks to ecology is relatively

new (Pilosof et al., 2017), the few studies that investigated multilayer modular structures (not necessarily with Infomap) have revealed new insights into the organisation and dynamics of communities and populations (Pilosof et al., 2017; Timóteo et al., 2018; Pilosof et al., 2019). Applying Infomap to multilayer networks can capture changes in module evolution and composition over time/space, or capture changing dynamics between different states of the system (De Domenico et al., 2015).

In multilayer networks, nodes representing observable entities such as species are called *physical nodes* and nodes describing dynamics in the layers for different time points, patches or interaction types, are called *state nodes*. The random walker moves from state node to state node within and across the layers. However, the encoded position always refers to the physical node (see dynamic visualisation available on `https://www.mapequation.org/apps/multilayer-network/index.html`). Therefore, the multilayer network representation is not merely an extended network with unique nodes in all layers. Instead, assigning the same code word to all state nodes of the same physical node in the same module acknowledges that only physical nodes represent observable entities and that two state nodes of the same physical node are coupled even if they are not connected. This separation between dynamics and coding provides two advantages. First, it enables a physical node to be assigned to different modules in different layers. From an ecological perspective, this degree of freedom is crucial as, for example, a certain species can have different functions in different layers. For example, in a multilayer network where each layer represents interactions in a different year, a host can interact with different parasites in different years, resulting in temporal variability in parasite spread (Pilosof et al., 2013, 2017). Second, dynamics-coding separation enables to model coupling between layers without interlayer links.

Modelling coupling without provided interlayer links is particularly useful in ecology because interlayer links are often challenging to measure empirically (Hutchinson et al., 2018). If interlayer links are not provided, a random walker moves within a layer and, with a given 'relax rate' r, jumps to the current physical node in a random layer without recording this movement, such that the constraint to move only within layers can be gradually relaxed (Fig. 4). By tuning the relax rate, it is possible to explore the relative contribution of intra- and interlayer links to the structure. To relax between layers, Infomap has two options: (1) a rate proportional to the weighted degree of the physical node within the different layers (argument `--multilayer-relax-rate`); or (2) variable coupling based on how similar the connectivity of nodes are between layers (argument `--multilayer-js-relax-rate`) (Aslak et al., 2018). It is also possible to set the number of layers to relax to (argument `--multilayer-relax-limit`). For example, with a relax limit of 2, Infomap can relax from layer 5 to any of layers 3 to 7 (including 5). This flexibility allows Infomap to capture communities in everything from disconnected to fully aggregated layers and from neighbouring to separated layers. Despite the local relaxation, a tightly interconnected group that exists across layers can still form a multilayer community (see examples). For example, plants and pollinators that interact consistently across time can form a module that persists for several layers.

Alternatively, Infomap can handle interlayer dynamics in which interlayer links are provided and describe the flow or constraints on flow (using an extended link list input). The system under study, availability of interlayer link data and research question at hand should determine the choice of multilayer dynamics model. This is particularly crucial in case interlayer links are not provided. For example, in a spatial network the natural constraint is free relaxation between layers but in a temporal network relaxation typically occurs towards the next layer because time flows one way.

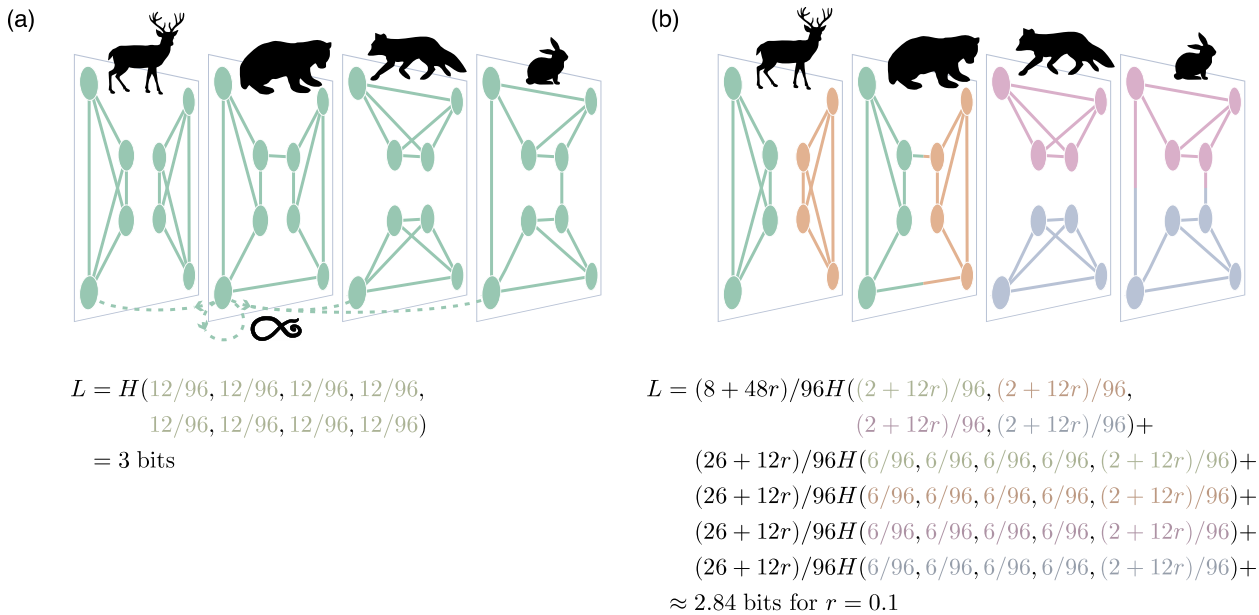


Figure 4: Schematic multilayer network. Within each layer, links represent the dispersal of a species between patches. Each layer is a different species. In the absence of empirical interlayer links, uniform coupling between all layers at relax rate r models flows between layers (depicted with dashed arrows for one layer), which is a proxy for the dispersal of a parasite between patches of different species. Note that the random walker can relax to the current layer, as indicated by a self-loop. (a) A one-module solution. (b) The optimal four-module solution. Modules represent patches tightly connected by the parasite's dispersal on different species. See Supplementary Information for detailed explanations of the codebooks.

2.6 Community detection with metadata

Often ecologists have access to additional information on nodes, so-called metadata, which can guide explanation of an observed modular structure based solely on network topology. For example, correlate module composition to phylogenetic distance (Rezende et al., 2009; Poulin et al., 2013) or habitat type (Krause et al., 2003). The opposite approach – to use additional information to guide the community detection – can also be insightful (Newman and Clauset, 2016). This can be particularly important if the interaction network is binary but expert knowledge infer that some interactions are stronger or more relevant. For example, in a seed dispersal network a certain group of species may play a significant role in the actual dispersal of seeds, whereas another group of species interact with the seed (e.g. predation) but do not contribute to successful dispersal (Howe, 2016). A partitioning of the network taking metadata into account may then give a more accurate description of the true modules of importance for seed dispersal.

A recent generalisation of the map equation enables incorporating discrete metadata in a metadata codebook for each module, which accounts for the entropy of the node metadata at each step of the random walk (Emmons and Mucha, 2019). The map equation with metadata takes the form

$$L(\mathsf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{M} p^i_{\circlearrowright} H(\mathcal{P}^i) + \eta \sum_{i=1}^{M} r^i_{\circlearrowright} H(\mathcal{R}^i), \tag{2}$$

where $r^i_{\circlearrowright}$ is the total metadata weight of all nodes in module i, $H(\mathcal{R}^i)$ is the module's metadata codebook, and the parameter η controls the relative contribution of the metadata to the map equation, with values

ranging between 0 for completely ignoring metadata to infinity for completely ignoring the network structure. Adding metadata can reveal modules that are not detected by using network flows alone (Emmons and Mucha, 2019).

2.7 Hierarchical module structure

Hierarchy is pervasive in all biology, yet hierarchical structures in ecological networks remain little studied. In an ecological network modules can represent, for instance, a division related to species distribution while submodules represent partitioning according to functional groups (see Olesen et al. (2010) for a discussion). Without limiting the map equation to identify two-level modules, the hierarchical map equation can identify nested multilevel modules that describe hierarchical organisation when such structures enable more compression (Rosvall and Bergstrom, 2011). In the hierarchical map equation, the code length from every module at all levels gives the total description length. For example, for a three-level map M of n nodes partitioned into M modules, for which each module i has M_m submodules mn, the three-level map equation takes the form

$$L(\mathsf{M}) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{m=1}^{M} \left(q_{\circlearrowright}^{m} H(\mathcal{Q}^{m}) + \sum_{n=1}^{M_m} p_{\circlearrowright}^{mn} H(\mathcal{P}^{mn}) \right), \tag{3}$$

with $\mathcal{Q}_m$ for the exit rate and submodule entry rates in module m at total rate $q_{\circlearrowright}^{m}$, and $\mathcal{P}^{mn}$ for the exit rate and node visit rates in submodule mn at total rate $p_{\circlearrowright}^{mn}$. Deeper recursive nesting gives higher hierarchical levels (Fig. S3).

3 Working Examples

A complete set of instructions for using Infomap is available here. In addition, all the scripts used to produce the analyses shown here, and in-depth technical instructions for how to use the code can be found in the dedicated Wiki and GitHub repository that accompany this paper. The code uses the R package infomapecology, which we have developed. In this section we exemplify how Infomap can be used for different types of ecological networks. In the text we focus on the logic behind the analysis of different kinds of networks and present briefly the most relevant input, output and arguments of Infomap. Our aim here is not to present full interpretations on the analysed networks, which is beyond the focus of this paper, but rather show the capacity and flexibility of the framework.

3.1 Monolayer bipartite network

We begin with a simple instructive example using the bipartite network data set `memmott1999` from package bipartite (Dormann et al., 2009; Memmott, 1999). First, we describe how the input to and the output from Infomap is organised. This is a weighted bipartite network describing a plant-flower visitor interaction web (25 plant species and 79 flower visitor species) in the vicinity of Bristol, U.K. We use Infomap to detect modules that contain both plants and pollinators. Because number of visits is a constraint for flow of elements such as pathogens (Truitt et al., 2019), a relevant research question would be to detect modules of plants and pollinators between which pathogens are more likely to transmit. To distinguish between the

two node sets we number the pollinator species from 1-79 and the plants from 80-104. The input to Infomap is described in Table 1. We run Infomap using a flow model for undirected networks and avoid hierarchical modules (see technical details here). Infomap's native output includes module affiliation and the amount of flow that goes through each node as in Table 2 but infomapecology adds the affiliation of modules to the node table (Table 3).

Table 1: Link list input for a bipartite network. The 'from' column is the ID of pollinators (can only contain IDs 1-79), the 'to' column is the ID of the plants (can only contain IDs 80-104), and the right-most column is the frequency visit (link weight).

from	to	weight
9	92	1
9	96	1
9	104	1
9	95	1
10	95	4

Table 2: The output of a tree file (first and last three rows). Each row begins with the multilevel module assignments of a node. The module assignments are separated by colons from coarse to fine levels, and all modules within each level are sorted by the total flow (PageRank) of the nodes they contain. The integer after the last colon is the ID of the leaf (not the node) within the finest-level module, ordered by flow (in this example, of a two-level analysis, there is only a single colon). The flow column is the amount of flow in that node, i.e. the steady state population of random walkers. The content name column is the node name. In our coding scheme in infomapecology the real node names are incorporated when parsing Infomap's output to the final result table (Table 3). Finally, node_id is the index of the node in the original network file.

path	flow	name	node_id
1:1	0.429226	"88"	88
1:2	0.0984883	"103"	103
1:3	0.0641319	"89"	89
	...		
25:1	0	"69"	69
26:1	0	"72"	72
27:1	0	"73"	73

Table 3: Parsing Infomap's output. Our code reads the tree output of infomap and integrates it with the metadata of nodes provided by the user. Because this is a two-level solution, there is only one level of modules (the second level are the nodes themselves).

node_id	node_group	node_name	module_level1
1	A	Aglais.urticae	1
2	A	Apis.mellifera	2
		...	
104	P	Trifolium.pratense	2

3.2 Monolayer directed networks with hierarchical structure

In our second example we use the Otago food web described in Mouritsen et al. (2011). In this network, nodes are species or taxonomical groups (e.g., bacteria) and binary links represent consumption. In this analysis, we allow for hierarchical clustering, which can partition the network into modules within modules (Infomap's default). Because this is not a bipartite network, species are numbered consecutively and the from and to columns can contain any species ID. We now use a directed flow model because links do not represent real flow of energy and are directed. The optimal solution has three levels (including the leaves), and the output is described in Table 4.

In Infomap's `tree` output, the `path` column has a tree-like format such as `1:3:4:2` which describes the path from the root of the tree to the leaf node. The first integer is the 'top module'. The last integer after the colon indicates the ID of the leaf in the module, and not the ID of the node. In this example, the node is the second leaf of the fourth submodule of the third submodule of the first top module. This output is automatically parsed in `infomapecology` to levels. In this example, there are three module levels and the fourth level is the leaf. This will create columns: module_level1, module_level2, module_level3 and module_level4. In case a node has fewer levels because its module is not partitioned, then the missing levels will show an `NA value`. A second node in the same network with a path 2:1:5 will be the fifth leaf in module 1, which is nested within top module 2. The values for this node will be: module_level1=2, module_level2=1, module_level3=5, module_level4=NA.

Table 4: Infomap output for a three-level solution. The code reads the tree output of infomap and integrates it with the metadata of nodes provided by the user. This data set is rich and contains node attributes (we include as an example three columns with the metadata provided in the original data set). The algorithm has partitioned the network into three levels but the last level is the ID of the leaves within a module. Note that not all modules have modules nested within them, and this is depicted by NA values.

node_id	node_name	module_level1	module_level2	OrganismalGroup	NodeType	Resolution
1	Phytoplankton	1	1	Plant	Common Name	Assemblage
2	Zooplankton	1	1	Zooplankton	Common Name	Assemblage
			...			
107	Black Swan	3	NA	Bird	Taxon	Species
108	Garfish	4	NA	Fish	Taxon	Species

3.3 Monolayer directed network with metadata

We again use the Otago food web (Mouritsen et al., 2011), but now guide the analysis using metadata (node attributes). One hypothesis, inspired by the group model of Allesina and Pascual (2009), is that the structure of the network is related to the 'guild' of the species because species from the same guild are eaten by the same predators and feed on similar prey. Although the group model identifies both assortative and disassortative groups, the dissortative groups connect to each other and form energy flows in the network (Baskerville et al., 2011) that can be interpreted as groups defined by flows. The organisation of such flow based groups has been suggested to stabilise the system (Rooney et al., 2008). Mouritsen et al. (2011) divided their 140 species to what they term 25 'organismal groups' (analogous to guilds) such as plants, zooplankton, snails, fish or birds. We included these node attributes and tested their effect for different values of η, which result in different partitions (Fig. 5). At $\eta = 0$ node attributes are ignored and the

partition is based solely on network topology. There is a large module that contains 17 of the 25 organismal groups. Increasing η returns partitions with more modules, but that contain species from fewer organismal groups (Fig. S2). This approach can thus reveal a structure that is *not* supported solely by topology, but rather incorporates expert knowledge.

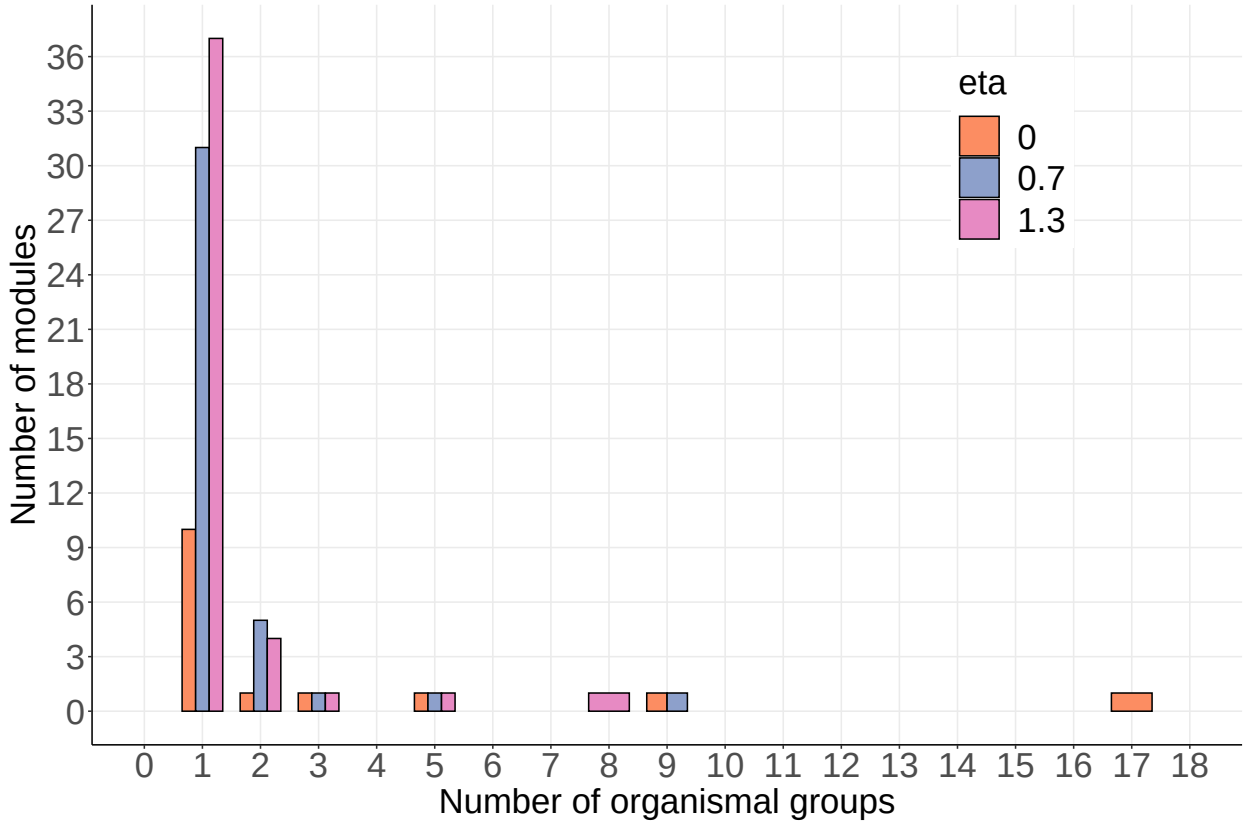


Figure 5: Including metadata changes network partitioning. The histogram shows the number of modules that contain a particular number of organismal groups (out of 25). With increasing values of η—that is, the more weight is given to the metadata—more modules contain fewer combinations of organismal groups. For example, there are 10, 31, and 37 modules that contain a single organismal group for eta values of 0, 0.7 and 1.3, respectively. We can also see that when metadata are not considered $\eta = 0$ there is a single module that contains 17 organismal groups but when $\eta \neq 0$ the maximum number of organismal groups contained in a module is 9.

3.4 Controlling flows

The Otago food web, as most food web data, is unweighted. Therefore, the flow is based solely on node degree and the direction of links. However, there are examples of ecological network data that do include measurements of flows, which are indeed important for the network dynamics. As discussed earlier, there are two ways to refer to the flow. The first is by measuring real flow; the other is to instead let the links represent some constraints on the flow. In Infomap, the first option is achieved by setting a flow model as: `-f rawdir`, and the second by: `-f directed`. A directed flow model will run a PageRank algorithm to set the flows for Infomap, whereas a rawdir flow model will use the flows as specified by the links.

As an instructive example, we here used data from Tur et al. (2016), who measured directed flow of

pollen grains (links) moved from a donor plant's anthers to a recipient plant's stigmas (plants are nodes), in south Andean communities, at three elevations. Because pollen can move between individuals from the same species, the network also contains self loops, representing conspecific pollination. We selected one network and ran Infomap twice. First, we used `-f rawdir` because in this data set real flows (actual number of pollen grains) were measured. Second, for the sake of example alone, we assumed that the links instead represent constraints and used `-f directed`. We found that the module assignments of species differs between these two models (Fig. 6). Specifically, using normalised mutual information (Danon et al., 2005), we find that knowledge on the composition of modules in one of the flow models reveals around 82% of the module composition obtained in the other (Fig. 6c). This discrepancy in module assignments indicates that researchers should choose a model carefully, based on the system and sampling scheme. In this particular example, Tur et al. (2016) state that they "randomly selected five plant individuals per species, whenever possible, and collected five senescent flowers (i.e. post-anthesis) per individual". Even though this is a sample of the true population, these are still measured real flows. Hence, a `-f rawdir` flow model should be preferred.

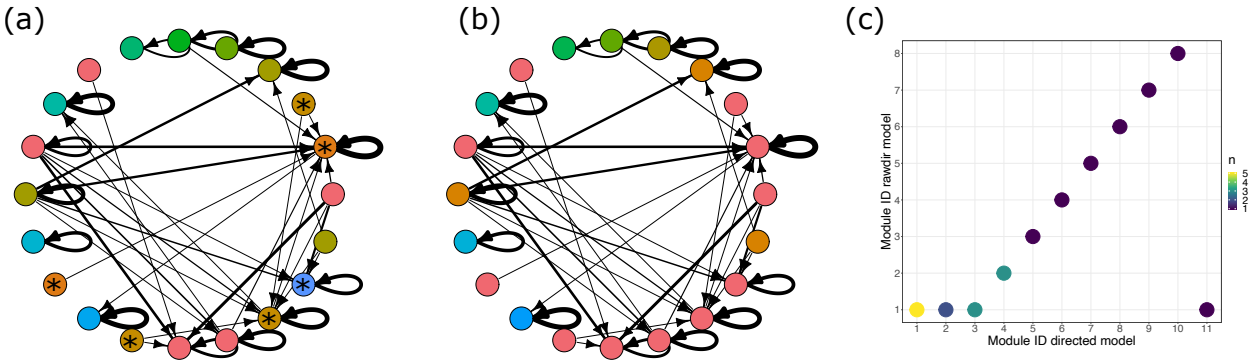


Figure 6: Comparison between flow models. The same network (elevation 2000 from Tur et al. (2016)), with different module assignments are presented for the directed (a) and rawdir (b) flow models. Link weights, depicting number of grains moved from a donor to a recipient plant species are log-transformed. Node colors in each of the panels correspond to different modules. Nodes that differ in module affiliation between the models are marked with an asterisk. (c) A confusion matrix depicting the number of species assigned to the same module. Species that were assigned to module 1 in the rawdir module, are divided between four different modules in the directed model (marked with an asterisk in panel a). The rest of the modules contain the same species in both models. Colors depict the number of species

3.5 Temporal multilayer network

There are many types of multilayer networks in ecological systems and the ability of Infomap to integrate layers of different kinds opens up a gamma of possibilities for their analysis. Per our goal in this paper, we here present an example of a temporal multilayer network, which represents flow in time. We use a host-parasite network recorded over 6 years, in which both interlayer and intralayer links are quantified (Pilosof et al. 2017, see also associated Wiki page), and analysed it in two ways: First, we analysed the network using the interlayer links (see Wiki page for details on input). We find that 47.4% of the modules persisted for the whole 6 layers while 7.89% appeared in only two layers and no module has persisted for a single layer (Fig 7a). This indicates that that the grouping of species has a strong temporal component. A second finding is that species are flexible; 21.8% of the species switched modules at least once during the six-year

period; that is, a species can be at a given module at one time point (layer), and switch to a different module in the next layer (Fig 7b). This happens because each species is represented by different state nodes. These findings are in line with a previous analysis of the same network using an extension of Q for multilayer networks (Pilosof et al., 2017) and with analyses of other temporal multilayer networks in biological (Pilosof et al., 2019) and non-biological (Mucha et al., 2010) systems.

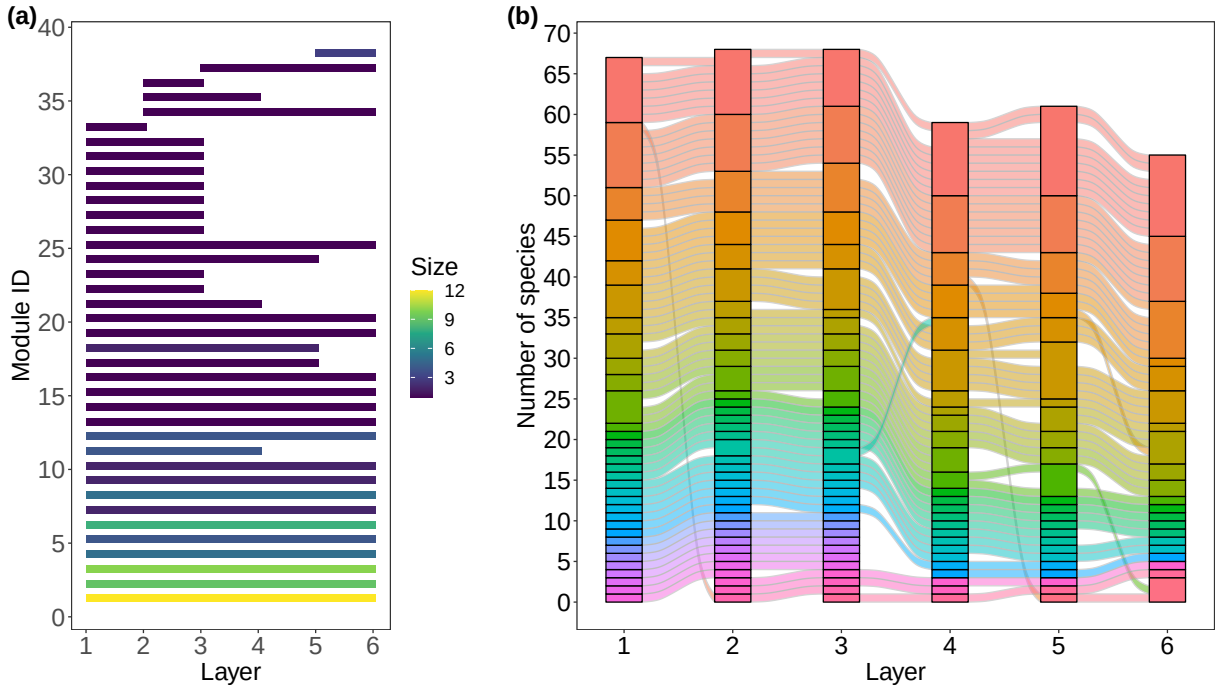


Figure 7: Modular structure of a temporal network. (a) The persistence of each module in time. Colours depict module size: the total number of species in each module. (b) An alluvial diagram for the flow of species from different modules among layers. Species are clustered in modules, presented in coloured blocks. Each line represents a species, and line colours correspond to the module in which the species originates.

To illustrate Infomap's capabilities to model interlayer links in a second analysis we ignored the interlayer links and used global relax rates to mimic the common situation in which interlayer links have not been measured (see Wiki page for details on input). We limited the relaxation of the random walker between layers to one layer forward, with no backwards relaxation because time flows in one direction. To do this, we used `--multilayer-relax-limit-up 1 --multilayer-relax-limit-down 0`. By changing the value of multilayer-relax-rate using `--multilayer-relax-rate`, we effectively examined the effect of increasing interlayer connectivity on structure. The higher the relax-rate r, the more frequent the movement of the random walker between layers, tightening the connection between layers and potentially affecting structure (e.g., creating modules that persist for longer times). While we do detect variation in number of modules, module composition and persistence, this variation is not large (Fig 8). Nonetheless, these results are specific for this network and we recommend this kind of sensitivity analysis to choose the appropriate relax-rate that best expresses the dynamics of the network. Moreover, the precise definition of interlayer links, or the use of relax rates should be one of the main considerations when analysing multilayer networks (Pilosof et al., 2017; Hutchinson et al., 2018).

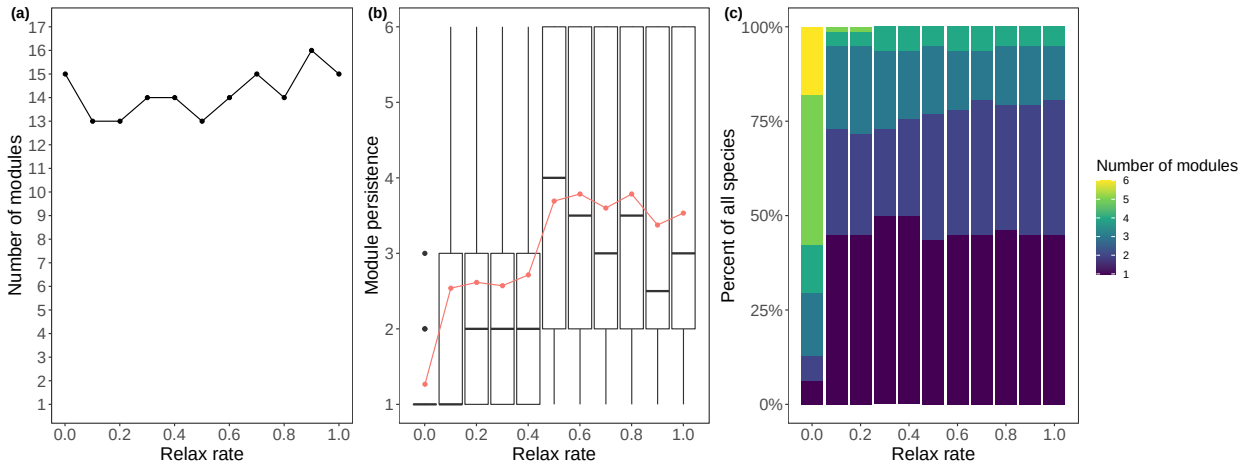


Figure 8: Structure as a function of increasing global relax rate. (a) Variation in the number of modules. (b) Distribution of module persistence (i.e., the number of layers throughout which a module exists). Boxplots represent the range, 95% quantiles and median (black line). The average is marked with red points and line. As expected, module persistence increases with increasing relax rate. (c) Species flexibility: The bars depict the percentage of species appearing in different numbers of modules.

3.6 Hypothesis testing

Many objective functions used for community detection, including the map equation, assume that the network data are complete. In practice, however, network data are rarely complete. Consequently, ecologists commonly ask if an observed partition reflects a true biological process or whether it can also be obtained at random. For example, both modularity Q and the map equation can capitalise on noise in sparse random networks. To overcome this problem, one approach is to incorporate a Bayesian prior with assumptions about network uncertainties into the map equation framework (Smiljanić et al., 2019). Another approach is to compare the obtained value of the objective function with values obtained by optimising it on a set of randomised versions of the observed network (Bascompte et al., 2003; Fortuna et al., 2010). The statistical test is carried out by calculating the probability, P_{value}, of obtaining a network with a better objective function value than that of the observed network. For the map equation, where short code lengths correspond to finding more modular regularities, $P_{\text{value}} = (|L_{\text{sim}} < L_{\text{obs}}|)/n$, where L_{sim} is the code length of a single randomised network, L_{obs} is the code length of the observed network, and n is the number of simulations.

The R package capitalises on previous methods to shuffle networks from the `vegan` package, particularly for bipartite networks to automatically run analysis for hypothesis testing. It also has a dedicated function to plot the results. A code for this example is detailed in the relevant Wiki page.

Here, we demonstrate hypothesis testing using the network of Tur et al. (2016), which we have used in our previous examples, for a single elevation - 1800 m. In randomising the network, we explicitly consider the flow dynamics of pollen. Specifically, we shuffle the network 1,000 times by randomly distributing non-diagonal cell matrices. We did so separately for lower and upper matrix triangles, constraining the total amount of flow that goes into and out of plants as in the observed network but shuffling the plant identities that distribute the flow. This randomisation scheme tests the hypothesis that the way pollen flows in and out *between* plants (we left the diagonal intact) is non-random and less modular in simulated networks than observed in nature. We found that in the observed network flows tend to be more constrained within

modules than in any of the random networks (Fig. 9a) ($P_{value} < 0.001$). In addition, the observed network had statistically significantly more modules than any of the simulated networks ($P_{value} < 0.001$; Fig. 9b). Together, these results point to a non-random structure in the flows of pollen to and from plants.

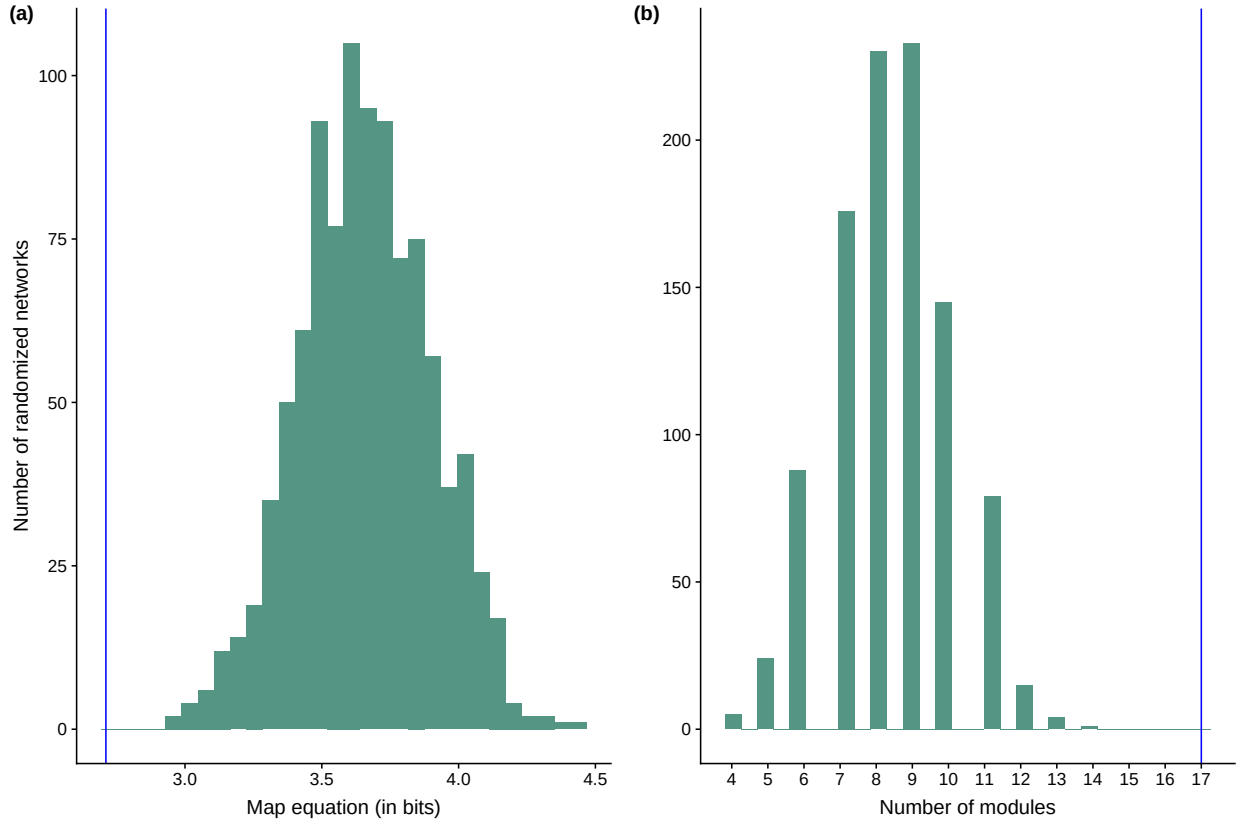


Figure 9: Results of null model hypothesis testing. The plot shows a comparison between shuffled versions and the observed network in (a) the value of the map equation and (b) the number of modules. The observed value is marked with a vertical blue line. Results are for the network from elevation 1800m in (Tur et al., 2016))

4 Discussion

Community detection is a cornerstone in ecological network analysis because it provides a higher-level simplification of complex ecological systems. Modularity for example can result from, and therefore be a signature of, ecological and/or coevolutionary dynamics (Beckett and Williams, 2013). For example, in host-pathogen systems, a modular structure has been shown to be a signature of negative-frequency dependent selection that leads to the formation of niches (hosts as resources for pathogens), with consequences for epidemiology and evolutionary dynamics of both the hosts and parasites (He et al., 2018; Pilosof et al., 2019, 2020). In mutualistic systems, it can indicate coevolutionary associations between plants and their mutualistic patterns resulting from trait matching (Olesen et al., 2007; Dormann et al., 2016). In biogeography, modules can form due to species variation in the ability to disperse to, or colonise, different habitats (Calatayud et al., 2019). A modular structure also has functional consequences. For example, modularity slows the spread of

perturbations such as disease or species extinctions (Fortuna et al., 2009; Salathé and Jones, 2010; Stouffer and Bascompte, 2011) and under certain conditions stabilises ecological networks (Grilli et al., 2016). Other community detection methods have also shown to be highly relevant for ecological networks, such as stochastic block models which can identify species that are performing unique roles in ecological communities (Sander et al., 2015) or represent spatial guilds (Baskerville et al., 2011).

Another core concept in research on ecological networks is analyses of the dynamical processes taking place on the network (e.g., (Otto et al., 2007). Nevertheless, the community detection algorithms commonly used in ecology focus on network topology and do not specifically view communities as dynamical building blocks. Here, we aimed to fill this gap by introducing Infomap to ecological research. Infomap is focused on mapping flow modules, detecting groups of nodes where the flow tends to stay for longer times, thus capturing the modular structure of networks with respect to the dynamics. This method may be particularly adequate for ecological systems in which flow is the dominant processes. Communities revealed by different community detection methods will indeed highlight different aspects of networks (Rosvall et al., 2018). Infomap, which seeks to coarse-grain the system's dynamics will therefore identify flow modules, rather than clusters with high internal density as in modularity analyses. These flow modules will likely better capture structural patterns important for the dynamics of the system.

Like any other method applied to ecological data, Infomap is limited by the availability of well-sampled data. While Infomap performs well in benchmark tests (Aldecoa and Marín, 2013), like other methods it provides no guarantee to identify the optimal solution because that requires checking all possible solutions, which is not feasible even for moderate-sized networks. In addition, it is not clear what is "the true" partitioning of a network (Peel et al., 2017), and as such it is in all types of community detection analyses important to choose a method relevant for the research question (Rosvall et al., 2018). Therefore, we advocate the application of a community detection method appropriate for the question. For example, if the goal is to detect species that consume, and/or are consumed by, similar species, then stochastic block models (e.g., the group model (Allesina and Pascual, 2009)) are more adequate as Infomap cannot capture disassortative modules. When applied to undirected networks Infomap is a powerful clustering algorithm but the classical Newman-Girvan modularity may be more appropriate if the goal is to detect groups by comparing to a random expectation.

For analysing many types of ecological networks, the methodological advantages of Infomap are clear. It is an efficient and fast algorithm, which is particularly useful when analysing a large number of networks (e.g., during hypothesis testing) or large and dense networks. It is also flexible and can be used with any kind of network, including directed multilayer networks. In addition, it can be applied to analyse higher-order interactions in a generalised way by modelling dynamics with state nodes between physical nodes. For example, Infomap can not only capture multilayer dynamics, but also dynamics that depend on varying number of previous steps across the network (Edler et al., 2017). The possibility of using metadata to inform the community detection is another advantage, highly relevant for ecological data, in particular as all interactions rarely are actually captured in the data (Jordano, 2016). Additional information from other systems (e.g. information on the role of species traits Eklöf et al. (2013) or taxonomic classification for interactions Eklöf et al. (2012)) or expert knowledge can then be valuable information for community detection. It is however important to remember that the metadata by no means reflect the true solution on how nodes should be partitioned (Peel et al., 2017). Therefore, the role it is allowed to take should be carefully chosen by adjusting the parameter η.

The usefulness of Infomap in providing novel insights into ecological systems is yet to be discovered, though few studies have already taken a step in this direction (**?**Calatayud et al., 2019; Pilosof et al., 2019). Nevertheless, as an algorithm specifically designed for coarse-graining the dynamics and identifying flow modules, Infomap is likely highly relevant for ecological networks. With this paper we provide the first incentives and guidelines for venturing into that line of research.

Data and code

All the data are included as data sets within the R package *infomapecology*, which is publicly available in the infomap_ecology_package GitHub repository, with detailed instructions for installation. Data were downloaded from the sources described in their respective publications, which are fully referenced in infomapecology, as well as in the Wiki within the infomap_ecology GitHub repository that accompany the paper. All the code is also publicly available in the GitHub repository and described in the Wiki.

Acknowledgements

D.E. and M.R. were supported by the Swedish Research Council, Grant No. 2016-00796. A.E. was supported by the Swedish Research Council, Grant No. 2016-04919. We thank Ana M. Martín González for advice on data sets and comments on drafts.

References

Aldecoa, R. and Marín, I. (2013). Exploring the limits of community detection strategies in complex networks. *Sci. Rep.*, 3:2216.

Allesina, S. and Pascual, M. (2009). Food web models: a plea for groups. *Ecol. Lett.*, 12(7):652–662.

Allesina, S. and Tang, S. (2015). The stability–complexity relationship at age 40: a random matrix perspective. *Popul. Ecol.*, 57(1):63–75.

Aslak, U., Rosvall, M., and Lehmann, S. (2018). Constrained information flows in temporal networks reveal intermittent communities. *Phys. Rev. E*, 97(6):062312.

Barber, M. (2007). Modularity and community detection in bipartite networks. *Physical Review E*, 76(6):1–9.

Bascompte, J., Jordano, P., Melián, C. J., and Olesen, J. M. (2003). The nested assembly of plant–animal mutualistic networks. *Proc. Natl. Acad. Sci. U. S. A.*, 100(16):9383–9387.

Bascompte, J. and Stouffer, D. B. (2009). The assembly and disassembly of ecological networks. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 364(1524):1781–1787.

Baskerville, E. B., Dobson, A. P., Bedford, T., Allesina, S., Anderson, T. M., and Pascual, M. (2011). Spatial guilds in the serengeti food web revealed by a bayesian group model. *PLoS Comput. Biol.*, 7(12):e1002321.

Beckett, S. J. (2016). Improved community detection in weighted bipartite networks. *R Soc Open Sci*, 3(1):140536.

Beckett, S. J. and Williams, H. T. P. (2013). Coevolutionary diversification creates nested-modular structure in phage-bacteria interaction networks. *Interface Focus*, 3(6):20130033.

Bernardo-Madrid, R., Calatayud, J., González-Suárez, M., Rosvall, M., Lucas, P. M., Rueda, M., Antonelli, A., and Revilla, E. (2019). Human activity is altering the world's zoogeographical regions. *Ecol. Lett.*, 22(8):1297–1305.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008(10):P10008.

Bohlin, L., Edler, D., Lancichinetti, A., and Rosvall, M. (2014). Community detection and visualization of networks with the map equation framework. In *Measuring Scholarly Impact*, pages 3–34. Springer International Publishing.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Brose, U., Williams, R. J., and Martinez, N. D. (2006). Allometric scaling enhances stability in complex food webs. *Ecol. Lett.*, 9(11):1228–1236.

Calatayud, J., Bernardo-Madrid, R., Neuman, M., and others (2019). Exploring the solution landscape enables more reliable network community detection. *arXiv preprint arXiv*.

Danon, L., Díaz-Guilera, A., Duch, J., and Arenas, A. (2005). Comparing community structure identification. *J. Stat. Mech.*, 2005(09):P09008.

De Domenico, M., Lancichinetti, A., Arenas, A., and Rosvall, M. (2015). Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Phys. Rev. X*, 5(1):011027.

De Domenico, M., Solé-Ribalta, A., Cozzo, E., Kivelä, M., Moreno, Y., Porter, M. A., Gómez, S., and Arenas, A. (2013). Mathematical formulation of multilayer networks. *Phys. Rev. X*, 3(4):041022.

Dormann, C. F., Fründ, J., Blüthgen, N., and Gruber, B. (2009). Indices, graphs and null models: analyzing bipartite ecological networks. *The Open Ecology Journal*, 2(1):7–24.

Dormann, C. F., Fründ, J., and Schaefer, H. M. (2016). Identifying causes of patterns in ecological networks: Opportunities and limitations. *Annu. Rev. Ecol. Evol. Syst.*

Dormann, C. F. and Strauss, R. (2014). A method for detecting modules in quantitative bipartite networks. *Methods Ecol. Evol.*, 5(1):90–98.

Edler, D., Bohlin, L., and Rosvall, M. (2017). Mapping higher-order network flows in memory and multilayer networks with infomap. *Algorithms*.

Eklöf, A., Helmus, M. R., Moore, M., and Allesina, S. (2012). Relevance of evolutionary history for food web structure. *Proc. Biol. Sci.*, 279(1733):1588–1596.

Eklöf, A., Jacob, U., Kopp, J., Bosch, J., Castro-Urgal, R., Chacoff, N. P., Dalsgaard, B., de Sassi, C., Galetti, M., Guimarães, P. R., Lomáscolo, S. B., Martín González, A. M., Pizo, M. A., Rader, R., Rodrigo, A., Tylianakis, J. M., Vázquez, D. P., and Allesina, S. (2013). The dimensionality of ecological networks. *Ecol. Lett.*, 16(5):577–583.

Emmons, S. and Mucha, P. J. (2019). Map equation with metadata: Varying the role of attributes in community detection. *Phys Rev E*, 100(2-1):022301.

Fletcher, Jr, R. J., Revell, A., Reichert, B. E., Kitchens, W. M., Dixon, J. D., and Austin, J. D. (2013). Network modularity reveals critical scales for connectivity in ecology and evolution. *Nat. Commun.*, 4:2572.

Fortuna, M. A., Popa-Lisseanu, A. G., Ibáñez, C., and Bascompte, J. (2009). The roosting spatial network of a bird-predator bat. *Ecology*, 90(4):934–944.

Fortuna, M. A., Stouffer, D. B., Olesen, J. M., Jordano, P., Mouillot, D., Krasnov, B. R., Poulin, R., and Bascompte, J. (2010). Nestedness versus modularity in ecological networks: two sides of the same coin? *J. Anim. Ecol.*, 79(4):811–817.

Ghasemian, A., Hosseinmardi, H., and Clauset, A. (2019). Evaluating overfit and underfit in models of network community structure. *IEEE Trans. Knowl. Data Eng.*, pages 1–1.

Grilli, J., Rogers, T., and Allesina, S. (2016). Modularity and stability in ecological communities. *Nat. Commun.*, 7:12031.

Guimerà, R., Sales-Pardo, M., and Amaral, L. A. N. (2007). Module identification in bipartite and directed networks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 76(3 Pt 2):036102.

He, Q., Pilosof, S., Tiedje, K. E., Ruybal-Pesántez, S., Artzy-Randrup, Y., Baskerville, E. B., Day, K. P., and Pascual, M. (2018). Networks of genetic similarity reveal non-neutral processes shape strain structure in plasmodium falciparum. *Nat. Commun.*, 9(1):1817.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Soc. Networks*, 5(2):109–137.

Howe, H. F. (2016). Making dispersal syndromes and networks useful in tropical conservation and restoration. *Global Ecology and Conservation*, 6:152–178.

Huffman, D. A. (1952). A method for the construction of Minimum-Redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101.

Hutchinson, M. C., Bramon Mora, B., Pilosof, S., Barner, A. K., Kéfi, S., Thébault, E., Jordano, P., and Stouffer, D. B. (2018). Seeing the forest for the trees: Putting multilayer networks to work for community ecology. *Funct. Ecol.*, 1:55.

Jordano, P. (2016). Sampling networks of ecological interactions. *Funct. Ecol.*, 30(12):1883–1893.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y., and Porter, M. A. (2014). Multilayer networks. *J. Complex Networks*, 2:203–271.

Krause, A. E., Frank, K. A., Mason, D. M., Ulanowicz, R. E., and Taylor, W. W. (2003). Compartments revealed in food-web structure. *Nature*, 426(6964):282–285.

Lambiotte, R. and Rosvall, M. (2012). Ranking and clustering of nodes in networks with smart teleportation. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 85(5 Pt 2):056107.

Lancichinetti, A. and Fortunato, S. (2009). Community detection algorithms: a comparative analysis. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 80(5 Pt 2):056117.

Marquitti, F. M. D., Guimarães, P. R., Pires, M. M., and Bittencourt, L. F. (2014). MODULAR: software for the autonomous computation of modularity in large network sets. *Ecography*, 37(3):221–224.

May, R. M. (1973). *Complexity and Stability in Model Ecosystems*. Princeton University Press, Princetone, New Jersey.

Maynard, D. S., Serván, C. A., and Allesina, S. (2018). Network spandrels reflect ecological assembly. *Ecol. Lett.*, 21(3):324–334.

Memmott, J. (1999). The structure of a plant-pollinator food web.

Mouritsen, K. N., Poulin, R., McLaughlin, J. P., and Thieltges, D. W. (2011). Food web including metazoan parasites for an intertidal ecosystem in new zealand: Ecological archives E092-173. *Ecology*, 92(10):2006–2006.

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878.

Newman, M. E. J. (2004). Detecting community structure in networks. *Eur. Phys. J. B*, 38(2):321–330.

Newman, M. E. J. and Clauset, A. (2016). Structure and inference in annotated networks. *Nat. Commun.*, 7:11863.

Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E Stat. Nonlin. Soft Matter Phys.*, 69(2 Pt 2):026113.

Olesen, J. M., Bascompte, J., Dupont, Y. L., and Jordano, P. (2007). The modularity of pollination networks. *Proc. Natl. Acad. Sci. U. S. A.*, 104(50):19891–198916.

Olesen, J. M., Dupont, Y. L., O'Gorman, E., Ings, T. C., Layer, K., Melián, C. J., Trøjelsgaard, K., Pichler, D. E., Rasmussen, C., and Woodward, G. (2010). From broadstone to zackenberg: space, time and hierarchies in ecological networks. In Woodward, G., editor, *Advances in Ecological Research*, volume 42 of *Ecological Networks*, pages 1–69. Academic Press.

Otto, S. B., Rall, B. C., and Brose, U. (2007). Allometric degree distributions facilitate food-web stability. *Nature*, 450(7173):1226–1229.

Pascual, M. and Dunne, J. A. (2006). *Ecological Networks: Linking Structure to Dynamics in Food Webs*. Oxford University Press.

Peel, L., Larremore, D. B., and Clauset, A. (2017). The ground truth about metadata and community detection in networks. *Sci Adv*, 3(5):e1602548.

Pilosof, S., Alcala-Corona, S. A., Wang, T., Kim, T., Maslov, S., and others (2020). The network structure and eco-evolutionary dynamics of CRISPR-induced immune diversification. *bioRxiv*.

Pilosof, S., Fortuna, M. A., Vinarski, M. V., Korallo-Vinarskaya, N. P., and Krasnov, B. R. (2013). Temporal dynamics of direct reciprocal and indirect effects in a host-parasite network. *J. Anim. Ecol.*, 82(5):987–996.

Pilosof, S., He, Q., Tiedje, K. E., Ruybal-Pesántez, S., Day, K. P., and Pascual, M. (2019). Competition for hosts modulates vast antigenic diversity to generate persistent strain structure in plasmodium falciparum. *PLoS Biol.*, 17(6):e3000336.

Pilosof, S., Porter, M. A., Pascual, M., and Kéfi, S. (2017). The multilayer nature of ecological networks. *Nature Ecology & Evolution*, 1:0101.

Poulin, R., Krasnov, B. R., Pilosof, S., and Thieltges, D. W. (2013). Phylogeny determines the role of helminth parasites in intertidal food webs. *J. Anim. Ecol.*, 82(6):1265–1275.

Rezende, E. L., Albert, E. M., Fortuna, M. A., and Bascompte, J. (2009). Compartments in a marine food web associated with phylogeny, body mass, and habitat structure. *Ecol. Lett.*, 12(8):779–788.

Rooney, N., McCann, K. S., and Moore, J. C. (2008). A landscape theory for food web architecture. *Ecol. Lett.*, 11(8):867–881.

Rosvall, M., Axelsson, D., and Bergstrom, C. T. (2010). The map equation. *Eur. Phys. J. Spec. Top.*, 178(1):13–23.

Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. U. S. A.*, 105(4):1118–1123.

Rosvall, M. and Bergstrom, C. T. (2011). Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PLoS One*, 6(4):e18209.

Rosvall, M., Delvenne, J.-C., Schaub, M. T., and Lambiotte, R. (2018). Different approaches to community detection. In Doreian, P., Batagelj, V., and Ferligoj, A., editors, *Advances in network clustering and blockmodeling*, pages 71–87. Wiley.

Rosvall, M., Esquivel, A. V., Lancichinetti, A., West, J. D., and Lambiotte, R. (2014). Memory in network flows and its effects on spreading dynamics and community detection. *Nat. Commun.*, 5:4630.

Salathé, M. and Jones, J. H. (2010). Dynamics and control of diseases in networks with community structure. *PLoS Comput. Biol.*, 6(4):e1000736.

Sander, E. L., Wootton, J. T., and Allesina, S. (2015). What can interaction webs tell us about species roles? *PLoS Comput. Biol.*, 11(7):e1004330.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Sys. Tech. J.*, 27:379–423.

Smiljanić, J., Edler, D., and Rosvall, M. (2019). Mapping flows on sparse networks with missing links. *arXiv [physics.soc-ph]*.

Stouffer, D. B. and Bascompte, J. (2011). Compartmentalization increases food-web persistence. *Proc. Natl. Acad. Sci. U. S. A.*, 108(9):3648–3652.

Thébault, E. (2013). Identifying compartments in presence-absence matrices and bipartite networks: insights into modularity measures. *J. Biogeogr.*, 40(4):759–768.

Timóteo, S., Correia, M., Rodríguez-Echeverría, S., Freitas, H., and Heleno, R. (2018). Multilayer networks reveal the spatial structure of seed-dispersal interactions across the great rift landscapes. *Nat. Commun.*, 9(1):140.

Traag, V. A., Waltman, L., and van Eck, N. J. (2019). From louvain to leiden: guaranteeing well-connected communities. *Sci. Rep.*, 9(1):5233.

Truitt, L. L., McArt, S. H., Vaughn, A. H., and Ellner, S. P. (2019). Trait-Based modeling of multihost pathogen transmission: Plant-Pollinator networks. *Am. Nat.*, 193(6):E149–E167.

Tur, C., Sáez, A., Traveset, A., and Aizen, M. A. (2016). Evaluating the effects of pollinator-mediated interactions using pollen transfer networks: evidence of widespread facilitation in south andean plant communities. *Ecol. Lett.*, 19(5):576–586.

Urban, D. L., Minor, E. S., Treml, E. A., and Schick, R. S. (2009). Graph models of habitat mosaics. *Ecol. Lett.*, 12(3):260–273.

Supplementary Information

S1 Details on Infomap's search and optimisation algorithm

For all mentioned types of dynamics and networks, the map equation measures the description length for a given modular partition. To identify the modular partition that reveals most modular regularities and has the shortest description length is a difficult optimisation problem due to the vast number of possible partitions. Like any other network clustering method, it is so difficult that the optimal clustering can be guaranteed only by checking all possible network partitions, which is not feasible even for medium-sized networks. Instead, Infomap is designed to efficiently explore the partition space with respect to the map equation in a greedy and stochastic way (Edler et al., 2017).

Inspired by the Louvain method for modularity optimisation (Blondel et al., 2008) and similar to the improved Leiden method, Infomap first joins neighboring nodes into modules, subsequently joins them into supermodules, and so on. First, Infomap assigns each node to its own module. Then, in random sequential order, Infomap moves each node to the neighboring module that results in the largest decrease of the map equation. If no move results in a decrease of the map equation, the node stays in its original module. Infomap repeats this procedure, each time in a new random sequential order, until no move generates a decrease of the map equation. Now Infomap rebuilds the network with the modules of the last level forming the nodes at this new level, and, exactly as at the previous level, Infomap joins the new nodes into modules, replacing the previous level. Infomap repeats this rebuilding of the network until the map equation cannot be reduced further. With this core algorithm, Infomap finds a fairly good clustering of the network in a very short time.

Infomap improves the accuracy by breaking the modules of the final state of the core algorithm with submodule movements and single-node movements. In submodule movements, first Infomap treats each cluster as a network on its own and applies the core algorithm to this network. This procedure generates one or more submodules for each module. Then Infomap moves back all submodules to their respective modules of the previous step. At this stage, with the same partition as in the previous step but with each submodule being freely movable between the modules, Infomap re-applies the core algorithm on the submodules.

In single-node movements, Infomap first re-assigns each node to be the sole member of its own module. It then moves all nodes back to their respective modules of the previous step. At this stage, with the same partition as in the previous step but with each single node being freely movable between the modules, Infomap re-applies the core algorithm on the single nodes. In practice, Infomap repeats the two extensions to the core algorithm in sequence, alternating between fine-tuning and coarse-tuning as long as the description length decreases enough.

Finally, because the full algorithm is stochastic and fast, repeated starts from scratch, 100 times or more if possible, increases the chance that the description length of the final partition is close to the global minimum. Moreover, analysing all generated partitions, the so-called solution landscape, can enable more reliable community detection (Calatayud et al., 2019). For example, by introducing a distance threshold between partitions within which a cluster of solutions are considered to be similar, Infomap can restart and generate new solutions – expanding and completing the solution landscaspe – until the probability that a new solution fits within existing clusters is higher than a given accuracy level, say 95 percent.

S1.1 Hierarchical modules

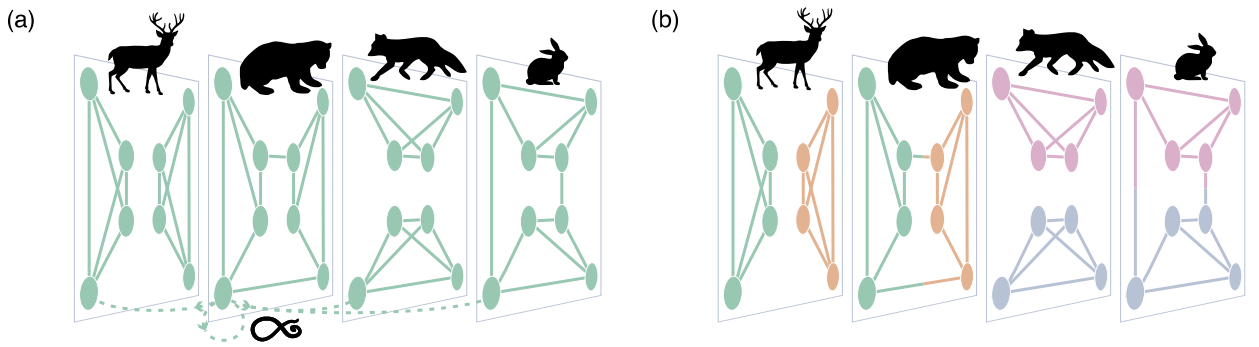
The algorithm described above will give a two-level solution, that is one level of modules on top of the nodes in the network. However, some networks have a hierarchical modular structure where a multilevel solution would give a shorter description length. Infomap can search for that by running two extra steps. First, it repeatedly applies the two-level algorithm to the top modules to add a super module level until no coarser top level description can be found that decreases the hierarchical map equation. Then, for each top module, it recursively tries to find submodules. For a given module, Infomap's recursive search clears any existing submodules and finds the coarsest modular description of the sub-network by applying the two-level algorithm followed by the first step of the multilevel algorithm. If this step doesn't decrease the hierarchical description length, the recursive search does not go further down this branch. Otherwise, it will continue the recursive search for each submodule until no finer modular structure can be found.

S2 Detailed explanation for multilayer figure

In Fig. 4b we detail the calculation of the map equation for this particular multilayer network example. Here we provide an explanation for this calculation. For undirected networks (as is this example), the flow for each node is its degree divided by total degree, here $3/96$ for each state node. As we do not consider two state nodes of the same physical node as two different nodes if they are within the same module, we have for each module four physical nodes with flow $6/96$ (each physical node contains two state nodes with flow $3/96$ each). These are the four $6/96$ entries in each of the module codebooks.

We then add the exit rate for each module: $(2 + 12r)/96$. The 2 comes from two links crossing module boundaries within a layer. The $12r$ is because each module has 24 links and half of relaxed flow at rate r will cross module boundaries. We can also think of $12r$ from the node perspective: each state node relaxes with rate r, and half of that rate exits the module, so $r/2$ of the flow for each node is transitioned to another module. That is, $8 \cdot 3/96 \cdot r/2$ for the eight state nodes within each module, or $4 \cdot 6/96 \cdot r/2$ for the four physical nodes, both ways giving a total of $12r/96$.

The frequency of use of each module codebook is $(26 + 12r)/96$. This is the total rate of use of the codebook: $4 \cdot 6/96 + (2 + 12r)/96$. As for the index codebook, the first term in L, the four $(2 + 12r)/96$ entries are the enter rates for the four modules, which for undirected networks are the same as the exit rates calculated above. The total use of the index codebook is the sum of these four, i.e. $4 \cdot (2 + 12r)/96 = (8 + 48r)/96$.



$$L = H(12/96, 12/96, 12/96, 12/96, 12/96, 12/96, 12/96, 12/96) = 3 \text{ bits}$$

$$\begin{aligned} L = {} & (8+48r)/96 H((2+12r)/96, (2+12r)/96, (2+12r)/96, (2+12r)/96) + \\ & (26+12r)/96 H(6/96, 6/96, 6/96, 6/96, (2+12r)/96) + \\ & (26+12r)/96 H(6/96, 6/96, 6/96, 6/96, (2+12r)/96) + \\ & (26+12r)/96 H(6/96, 6/96, 6/96, 6/96, (2+12r)/96) + \\ & (26+12r)/96 H(6/96, 6/96, 6/96, 6/96, (2+12r)/96) + \\ \approx {} & 2.84 \text{ bits for } r = 0.1 \end{aligned}$$

Figure S1: Fig 4 from the main text of a schematic multilayer network, included here for convenience Within each layer, links represent the dispersal of a species between patches. Each layer is a different species. In the absence of empirical interlayer links, uniform coupling between all layers at relax rate r models flows between layers (depicted with dashed arrows for one layer), which is a proxy for the dispersal of a parasite between patches of different species. Note that the random walker can relax to the current layer, as indicated by a self-loop. (a) A one-module solution. (b) The optimal four-module solution. Modules represent patches tightly connected by the parasite's dispersal on different species. See Supplementary Information for detailed explanations of the codebooks.

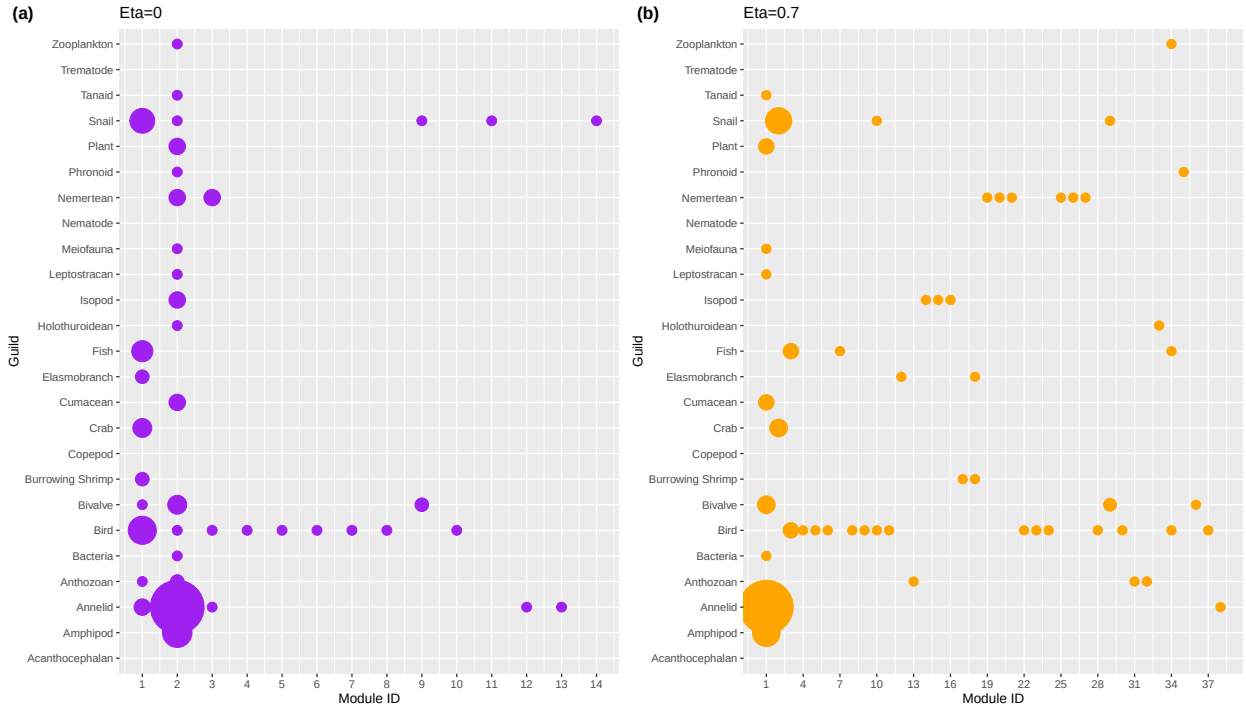
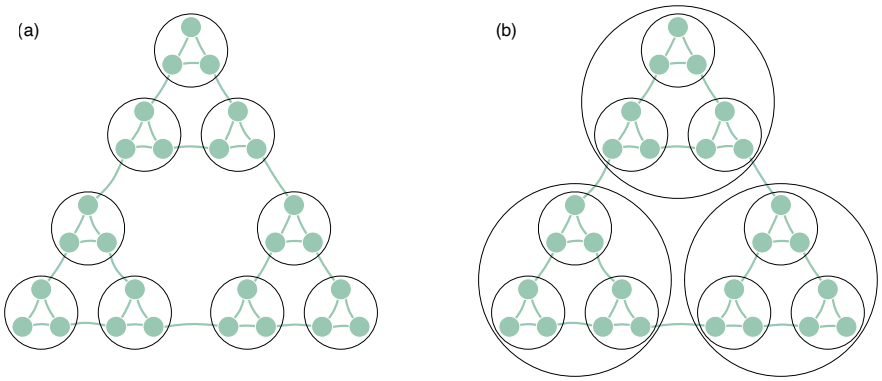


Figure S2: Comparison of structure without and with metadata. (a) The affiliation of different guilds into modules based on topology alone. The size of the circle represents the number of species from a guild in the module. (b) When metadata is included (here, $\eta = 0.7$), there are more modules, with each containing fewer combinations of guilds.



$$
\begin{aligned}
L = {} & 24/78H(2/78, 3/78, 3/78, 3/78, \\
& \quad 2/78, 3/78, 3/78, 3/78, 2/78)+ \\
& 10/78H(2/78, 3/78, 3/78, 2/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 10/78H(3/78, 2/78, 3/78, 2/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 10/78H(3/78, 3/78, 2/78, 2/78) \\
& \approx 3.57 \text{ bits}
\end{aligned}
$$

$$
\begin{aligned}
L = {} & 6/78H(2/78, 2/78, 2/78)+ \\
& 8/78H(2/78, 3/78, 3/78)+ \\
& 8/78H(3/78, 2/78, 3/78)+ \\
& 8/78H(3/78, 3/78, 2/78)+ \\
& 10/78H(2/78, 3/78, 3/78, 2/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 10/78H(3/78, 2/78, 3/78, 2/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 12/78H(3/78, 3/78, 3/78, 3/78)+ \\
& 10/78H(3/78, 3/78, 2/78, 2/78) \\
& \approx 3.48 \text{ bits}
\end{aligned}
$$

Figure S3: The description length according to the map equation for a schematic undirected network partitioned into (a) two levels and (b) three levels. The multilevel code structure compresses the dynamics the most.