Methods in Ecology and Evolution

British Ecological Society

Methods in Ecology and Evolution 2014, **5,** 90–98 doi: 10.1111/2041-210X.12139

APPLICATION

A method for detecting modules in quantitative bipartite networks

Carsten F. Dormann[1*] and Rouven Strauss[2]

[1]*Biometry and Environmental System Analysis, University of Freiburg, Freiburg, Germany; and* [2]*Department of Computer Science, Technion–Israel Institute of Technology, Haifa, Israel*

Summary

1. Ecological networks are often composed of different subcommunities (often referred to as modules). Identifying such modules has the potential to develop a better understanding of the assembly of ecological communities and to investigate functional overlap or specialization.

2. The most informative form of networks are quantitative or weighted networks. Here, we introduce an algorithm to identify modules in quantitative bipartite (or two-mode) networks. It is based on the hierarchical random graphs concept of Clauset *et al.* (2008 Nature 453: 98–101) and is extended to include quantitative information and adapted to work with bipartite graphs. We define the algorithm, which we call QuanBiMo, sketch its performance on simulated data and illustrate its potential usefulness with a case study.

3. Modules are detected with a higher accuracy in simulated quantitative networks than in their binary counterparts. Even at high levels of noise, QuanBiMo still classifies 70% of links correctly as within- or between-modules. Recursively applying the algorithm results in additional information of within-module organization of the network.

4. The algorithm introduced here must be seen as a considerable improvement over the current standard of algorithms for binary networks. Due to its higher sensitivity, it is likely to lead to be useful for detecting modules in the typically noisy data of ecological networks.

Key-words: compartments, groups, modularity, null model, pollination networks, weighted networks

Introduction

The ecological literature is replete with references to interacting groups of species within systems, variously termed compartments (May, 1973; Pimm 1982; Prado & Lewinsohn 2004), modules (Olesen *et al.* 2007; Garcia-Domingo & Saldañ a, 2008; Dupont & Olesen 2009), cohesive groups (Bascompte *et al.* 2003; Danieli-Silva *et al.* 2011; Guimarães, Jordano & Thompson 2011) or simply communities (Fortunato 2010). Their attraction, for ecologists, is that they promise a way to simplify the description and understanding of an ecological system, by representing not each and every species, but aggregating their interactions and energy fluxes into a more manageable set of modules (e.g. Allesina, 2009a). In the following, we will refer to such aggregated sets of interacting species as 'modules'. Their characteristic hallmark is that *within*-module interactions are more prevalent than *between*-module interactions (Newman 2003; Newman & Girvan, 2004; Fortunato 2010).

In the extreme case, modules are completely separated from each other and are then typically called compartments (Pimm 1982). This strict definition has seen some relaxation (Dicks, Corbet & Pywell 2002), but most recent studies converge on the term 'module' for any identifiable substructure in interaction networks (Prado & Lewinsohn 2004; Lewinsohn *et al.* 2006; Olesen *et al.* 2007; Ings *et al.* 2009; Joppa *et al.* 2009; Cagnolo, Salvo & Valladares 2011).

The identification of modules, and the membership of species to modules, has received considerable interest in the physical sciences (as reviewed *in extenso* by Fortunato 2010). Particularly the work of Newman and co-workers (e.g. Newman 2003; Newman & Girvan 2004; Newman 2004a, b, 2006) has practically defined the current paradigm of module definition and identification. Algorithms to identify modules are 'greedy', that is, highly computationally intensive, relying on some way of rearranging module memberships and then quantifying 'modularity' until a maximal degree of sorting has been achieved (Clauset, Newman & Moore 2004; Newman 2004b; Newman 2006; Pons & Latapy 2006; Schuetz & Caflisch 2008). The focus of virtually all these algorithms was on unweighted and one-mode networks (see, e.g., Clauset, Moore & Newman 2008; Kovács *et al.* 2010; Lancichinetti & Fortunato 2011; Jacobi *et al.* 2012, for a recent example). Unweighted (or binary or qualitative) refers to the fact that only the presence of a link between species is known, but not

*Correspondence author. E-mail: carsten.dormann@biom.uni-freiburg.de

its strength (Levins 1975; Pimm 1982). One-mode refers to the structure of the community, in which all species are potentially interacting with each other. The typical ecological example is a species × food web matrix, in which entries of 1 depict an existing interaction.

In recent years, weighted and bipartite interaction networks have become more intensively studied. In a *weighted* network, the link between two species is actually quantified (e.g. by the number of interactions observed or the strength of the interaction inferred from the data: Newman 2004b). In a *bipartite* network, the species fall into two different groups, which interact with members of the other group, but not within their group. A typical example are pollinator–visitation networks (Vázquez *et al*. 2009), where pollinators interact with flowers, but flowers do not interact among themselves (see Fig. 1). Another well-studied examples is host–parasitoid network (e.g. Morris, Lewis & Godfray 2004; Tylianakis, Tscharntke & Lewis 2007).

While popular among ecologists (Blüthgen 2010; Pocock, Evans & Memmott 2012; Poisot *et al*. 2012; Schleuning *et al*. 2012), weighted bipartite graphs are not amenable to any of the existing module detection algorithms for one-mode networks or for unweighted bipartite networks. Existing software uses one-mode networks or, more precisely, one-mode projections of bipartite networks (Guimerà, Sales-Pardo & Amaral 2007; Martín Gonzàlez *et al*., 2012; Thébault 2013), or unweighted (binary) bipartite networks (Guillaume & Latapy 2004; Marquitti *et al*. 2013), while other approaches focus on the identification of crucial leaves through quantifications of their position in the network (e.g. centrality or degree: Ravasz *et al*. 2002; Borgatti 2006; Martín González *et al*., 2010). This lack of an algorithm to identify modules in quantitative, bipartite networks is particularly problematic, as such networks find their way into conservation ecological considerations (Tylianakis *et al*., 2010) and are the focus of a vibrant field of macroecological research (Ings *et al*. 2009). Furthermore, from a statistical point of view, weighted networks offer much more information and are less likely to lead to erroneous conclusions about the system (Li & Chen 2006; Scotti, Podani & Jordan 2007; Blüthgen 2010; Fortunato 2010).

Here, we present an algorithm to identify modules (and modules within modules) in weighted bipartite networks. We build on an algorithm provided by Clauset, Moore & Newman (2008) for unweighted, one-mode networks, the weighted modularity criterion developed by Newman & Girvan (2004), and the bipartite modularity proposed by Barber (2007).

Modularity algorithms

Modules can be interpreted as link-rich clusters of species in a community. An alternative to finding and delimiting such modules is to group species by ordination (Borgatti & Everett 1997; Lewinsohn *et al*. 2006). Correspondence analysis (CA) of the adjacency matrix is a simple and fast way to organize species. Typically, however, correspondence analysis will not be able to identify modules sufficiently well, even if modules are actually compartments (i.e. perfectly separated: Fig. 2 left, centre). The QuanBiMo algorithm we present here can do so, at least in principle (Fig. 2 right). If modules are perfectly separated, with no species interacting with species in another module, they are called compartments and will be visible as clearly separated groups of species. It is relatively straightforward to implement a recursive compartment detection function, but compartments are much coarser than modules and not the topic of this publication.

One algorithm proposed and available for detecting modules in bipartite networks is due to Guimerà, Sales-Pardo & Amaral (2007) called 'bipartmod_w'), which is derived from a one-mode algorithm (Guimerà *et al*. 2005). Their approach differs substantially from single-run bipartite algorithm in that they employ a one-mode modularity algorithm on each level of a bipartite network separately, although they discuss the approach later developed by Barber (2007). The Guimerà *et al*. approach was used in several ecological applications of

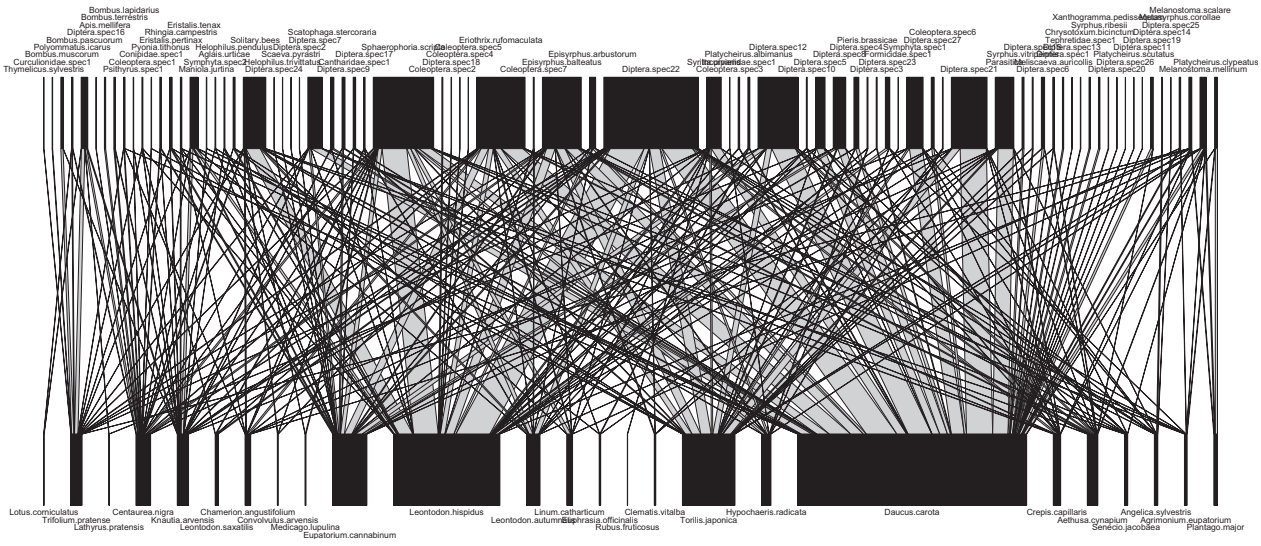

Fig. 1. Bipartite graph of a quantitative pollinator–visitation network (Memmott, 1999).

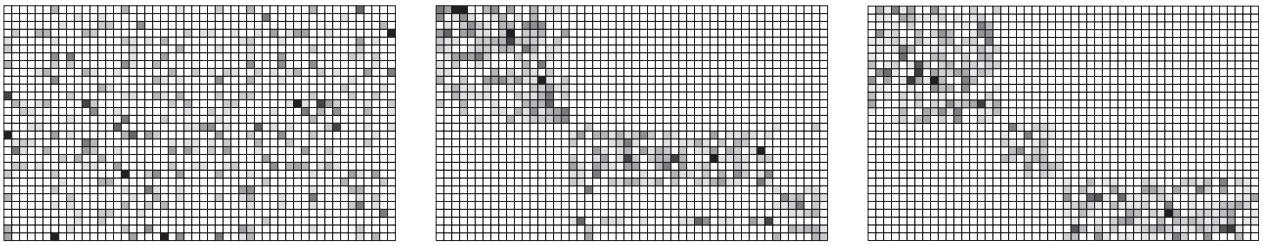

Fig. 2. A simulated three-compartment network in random sequence (left), as sorted by a correspondence analysis (centre) and by the modularity algorithm with default settings (right).

modularity (Olesen *et al*. 2007; Dupont & Olesen 2009; Fortuna *et al*. 2010; Guimerà *et al*. 2010; Carstensen *et al*. 2011; Trøjelsgaard & Olesen 2013), although the algorithm does not explicitly identify combined modules (as stated in Barber 2007; Fortuna *et al*. 2010). Most recently, Thébault (2013) investigated, through simulations, the ability of three modularity measures (those of Newman & Girvan 2004; Barber 2007; Guimerà, Sales-Pardo & Amaral 2007) to identify modules in binary bipartite networks and comes out in support of that of Guimerà *et al*. (2007).

Finally, Allesina & Pascual (2009b) have proposed an approach for one-mode networks. It identifies 'groups', rather than modules, which reveal more about the structure of a food web than modules do, since also their relation towards each other emerges from the analysis. Their approach is based on a binary one-mode matrix, however, even when applied to bipartite networks (as was done by Martín González *et al*., 2012).

QUANBIMO: A QUANTITATIVE BIPARTITE MODULARITY ALGORITHM

Outline

The new algorithm (Dormann & Strauß, 2013) builds on the hierarchical random graph approach of Clauset, Moore & Newman (2008), which builds a graph (i.e. a dendrogram) of interacting species so that nearby species are more likely to interact. It then randomly swaps branches at any level and evaluates whether the new graph is more likely than the previous one, recording and updating the best graph. Fit is computed as modularity of the current graph (detailed in section Goal function). The swapping is a Simulated Annealing-Monte Carlo approach, that is, sometimes a worse graph is chosen as the starting point for the next swap, thereby avoiding being trapped in a local maximum. Each node of the graph contains the information of whether it is part of a module, so that the graph can be transgressed top-down to identify modules.

Our modifications consist of (i) allowing branches between species to be weighted by the number of interactions observed between them, thereby making the algorithm quantitative; and (ii) taking into account that species in one group can only interact with species in the other group, rather than the one-mode network the algorithm was initially developed for. Taken together, our algorithm computes modules in weighted, bipartite networks, based on a hierarchical representation of species link weights and optimal allocation to modules.

Terminology

A graph $G = (V,E)$ denotes a set of vertices $v \in V$ connected by edges $e \in E$. An edge e connects two nodes, thus $e = c(v_i,v_j)$, where $v_i \in V \wedge v_j \in V$. G is a weighted (= quantitative) graph if each edge e has a weight $w \in W$ associated with it ($w \subseteq R^+$). We normalize edge weights so that $\sum_{w \in W} w = 1$. (For binary graphs $w = 1/|E|$ for all existing edges, where $|.|$ symbolizes the number of elements.)

For bipartite graphs, the vertices V are of two non-overlapping subsets, V_H and V_L (higher and lower level), such that $V_H \cap V_L = \emptyset$ and for all edges, the connected vertices are in different subsets: $v_i \in V_H \Leftrightarrow v_j \in V_L$ ($\Leftrightarrow$ symbolises equivalence, i.e. if we know v_i is in V_H, v_j *must* be in V_L, and vice versa).

A graph can be represented as a dendrogram D, that is, a binary tree with the vertices of the graph G being the tips (or leaves) of the dendrogram D (Fig. 3a). Thus, any internal split (or vertex) of D defines a subset of G. The idea of the algorithm is now to find internal vertices of D so that the subset it defines is a module.

Goal function

The algorithm has to divide G into a set of modules C such that

1. each module $c \in C$ is a connected subgraph of G. (This means each species has to have a partner).

2. each vertex v belongs to exactly one module c. (The uniqueness requirement).

3. edge weights within a module are higher than edge weights outside modules. (The modularity definition).

To specify point 3 above, Barber (2007) has defined modularity for bipartite networks as

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - P_{ij}\right)\delta(c_i, c_j) \qquad \text{(eqn 1)}$$

where m is half the total number of observed links in the network and A_{ij} is in fact a binary version of our edge matrix **E**, with values of 1 if a link between i and j exists. The expected value for each link, based on an appropriate null model, is given in the matrix **P**. The module to which a species i or j is

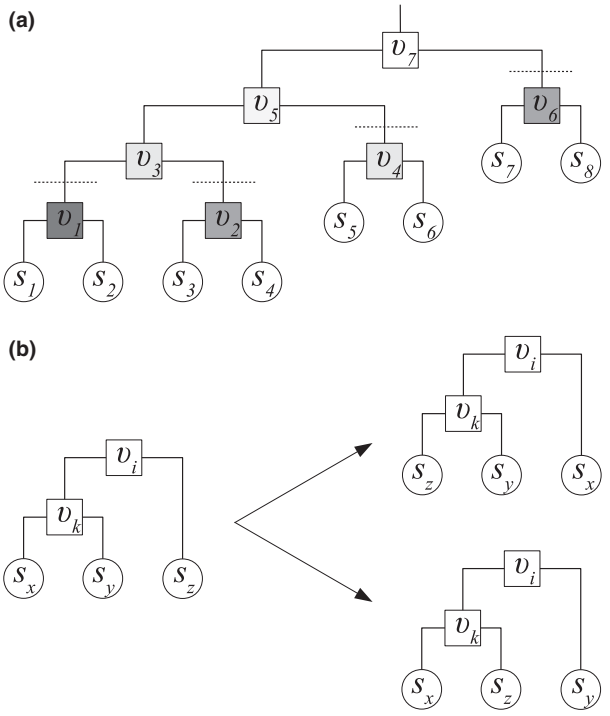

Fig. 3. (a) Representation of bipartite network as dendrogram. *s* refers to subtrees or leaves (= species) of the tree, v to vertices of the dendrogram. Nodes are shaded to illustrate the modularity within the subtree (darker means more strengths in that subtree than expected). The dashed lines indicate where a module is cut out. (b) The two possible moves in the swapping of randomly selected vertices v_i and v_k. The algorithm randomly chooses one of these two possible new configurations.

assigned is c_i, c_j. The indicator function $\delta(c_i,c_j) = 1$ if $c_i = c_j$ and 0 if $c_i \neq c_j$. Q ranges from 0, which means the community has no more links within modules than expected by chance, to a maximum value of 1. The higher Q, the more do the data support the division of a network into modules.

Following Newman (2004b), we can expand eqn. 1 by allowing A_{ij} and P_{ij} to be weighted, rather than binary. To avoid confusion, we then replace P_{ij} by K_{ij}, which is not a normalized probability matrix, but really the matrix of expected weights, based on the null model. Furthermore, for the weighted case, we compute $m = \sum_{i,j} A_{ij}$. We then get:

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - K_{ij}\right)\delta(c_i, c_j), \qquad \text{(eqn 2)}$$

where $\mathbf{K} = \frac{\mathbf{k}_i\mathbf{k}_j^T}{2m}$, with $\mathbf{k}_i = \mathbf{A}_{i.}$ and $\mathbf{k}_j = \mathbf{A}_{.j}$ being the marginal totals for rows and columns of $\mathbf{A}$, respectively (Newman 2004b).

One crucial point of our modifications of the original hierarchical random graph algorithm of Clauset, Moore & Newman (2008) was to assign an indicator value to each dendrogram vertex to label it as being within a module, or not. To do so, we have to compute the expected value for each value of A_{ij} in order to be able to evaluate whether the observed value is lower or higher (the term over which eqn. 2 sums). This step is not required if edges are unweighted, since then the expectation will always be the same. For weighted edges, and under random interactions among individuals, however, we would expect the edge e_{ij} connecting two nodes i and j representing abundant species to have a high value of w_{ij}. Similarly, nodes representing rare species could be expected to have low edge weights.

Thus, at every internal vertex of the tree, the subtree is interpreted as module and the algorithm computes the expectation matrix $\mathbf{K}$ based on the cross-product of marginal totals of all species in the module, divided by the sum of the number of observed interactions in that module $\left(\mathbf{K} = \frac{\mathbf{k}_i\mathbf{k}_j^T}{2m}\right)$. (Since we normalized all edge weight to sum to 1, $\mathbf{K}$ is actually a probability matrix). In other words: the number of interactions we expect in a cell depends on the number of observations we have for both interacting partners; frequently observed species contribute more to the expected number of interactions than rarely observed species. Ecologically speaking, we interpret the marginal totals as being proportional to the probability of observing this species in interactions. This need not reflect abundance, but rather activity in a given community context. In fact, we would expect modules to change when further species are added, as this ecologically affects not only their relative abundance but also their preferences with different resources becoming available. If the vertex gives rise to a module, that is, if $\sum_{ij \in c}(A_{ij} - K_{ij}) > 0$, this vertex is labelled as a module. We can now sum the contributions of all vertices and modules according to eqn. 2 to compute to total modularity of graph G. For a formal description of this part of the algorithm, please see Appendix S1.

Swapping

The algorithm starts with a random dendrogram, where modularity Q is likely to be very low. Through random swapping of branches and their optimization, Q increases during a simulated annealing procedure. The algorithm stops when a pre-defined number of swaps did not further increase the value of Q.

Random swaps are implemented as exchange of two randomly selected vertices in the dendrogram, subject to the following constraint (Fig. 3b). The vertex to be swapped cannot be a leaf. Since terminal vertices always connect leaves from the two bipartitions V_i and V_j, thus representing an interaction, they can be swapped, while their leaves cannot.

After each swap, the modularity of the entire dendrogram is recomputed (for computational efficiency, only those parts affected by the swap). If the new configuration has a higher value of Q, it is stored and becomes the new best dendrogram, otherwise the previous configuration will be used as the starting point for the next swap. A worse configuration is accepted with the probability $P < e^{\frac{\delta Q}{T}}$, where δQ is the change in modularity from the last configuration to the new one and T is the current temperature of the simulated annealing algorithm. We observed that the algorithm converges notably faster if the temperature is not decreased monotonously, but rather set back to the average temperature at which an increase in Q occurs. This is also a better approach in our case, since we do not know, *a priori*, how many steps the algorithm will take or which value of Q can be obtained.

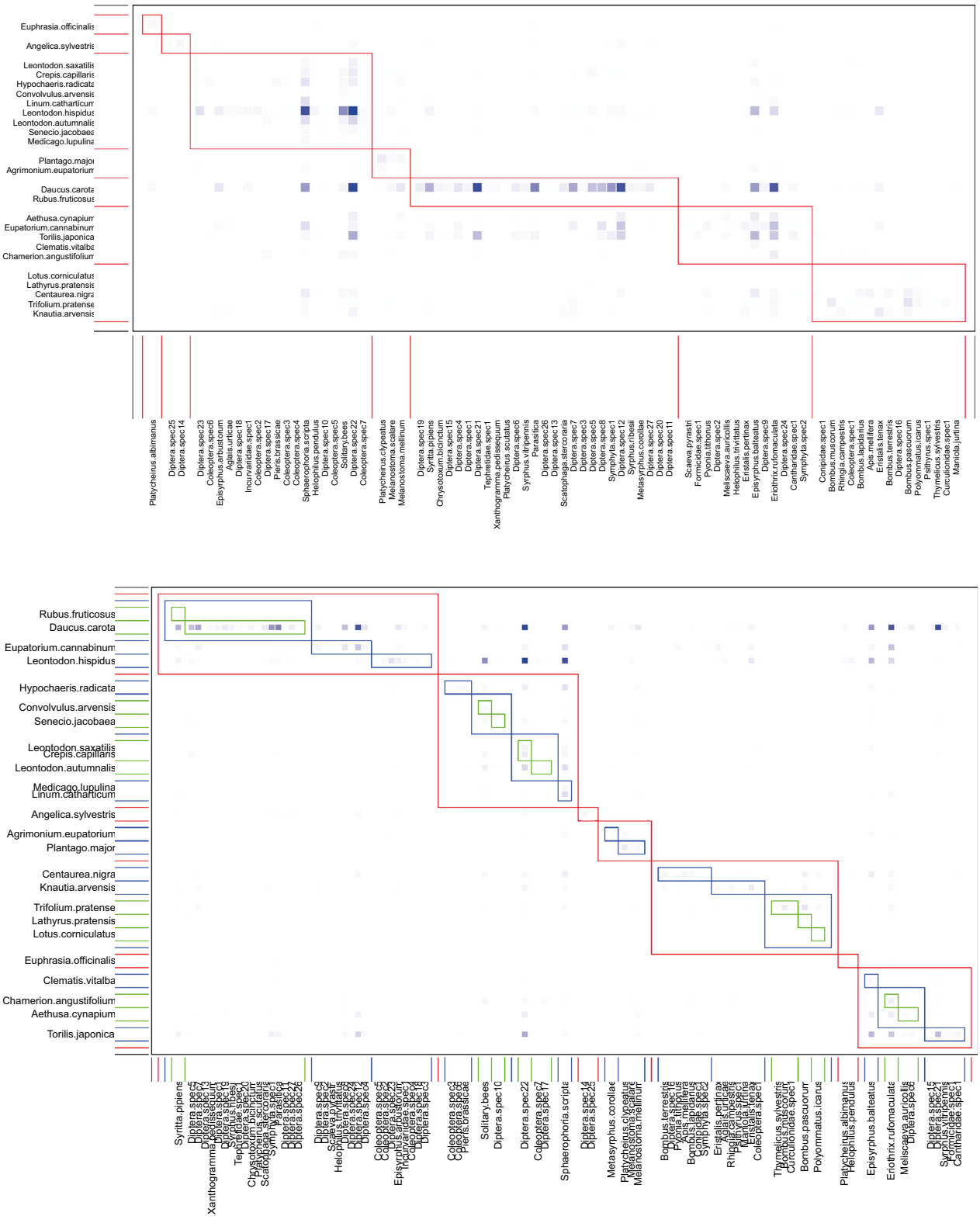

Fig. 4. Interaction matrix featuring modules for the data of Memmott (1999). *Top*: Modules identified by QuanBiMo(with *steps* = *1e10*, running for several hours; $Q = 0{\cdot}30$). Darker squares indicate more observed interactions. Red boxes delineate the seven modules. (Note that results may vary between runs.) In the central module, yellow Asteraceae feature heavily, while a possible ecological cause pattern for the other modules is less apparent. *Bottom*: Nested modules based on a recursive call of QuanBiMo. Module arrangement is slightly different from top, since the algorithm is stochastic.

Since the hierarchical dendrogram is computed through iterative proposing, evaluation and rejecting dendrogram structure in a Markov Chain Monte Carlo approach, Clauset, Moore & Newman (2008)'s, and hence our, algorithm cannot guarantee finding the optimal module configuration. Since the algorithm is coded in C++, even billions of MCMC swaps are

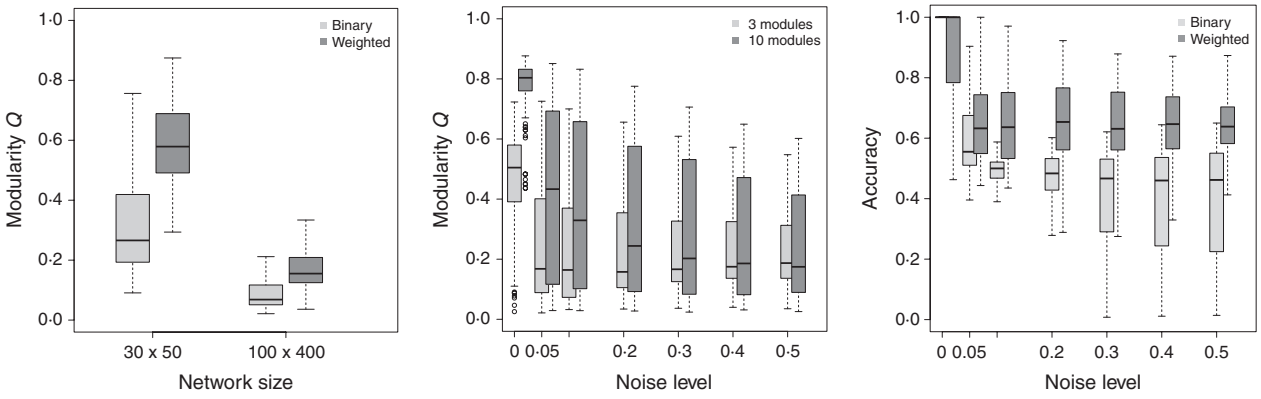

Fig. 5. Quality of modularity detection (left and middle: Q; right: overall accuracy) depends on network size, the amount of noise and type of information (binary or weighted).

feasible in a few minutes, yielding reasonable results for typically sized ecological networks (see below) at acceptable handling time. For large networks, this algorithm can run for hours to days. See appendix for an example session on how to employ the algorithm through R (R Development Core Team 2012).

OUTPUT & NESTED MODULES

The algorithm returns an object identifying modules and sequence vectors for species, as well as a re-order network ready for visualization of modules and the modularity Q. It can be plotted as an interaction matrix (Fig. 4, top).

QuanBiMo can be invoked recursively, searching for modules within modules (Fig. 4, bottom). When doing so, QuanBiMo will ignore links outside the inner module and treat the 'parent' module as all there is. While such nested modules become ever smaller and are thus ever faster to detect, there are plenty of them, and hence, nesting will typically dramatically prolong the search for patterns.

EVALUATION OF QUANBIMO

We analysed over 2000 simulated networks with known module structure, as a factorial combination of two sizes, seven levels of noise, two number of modules and two sampling intensity. The voluminous output is presented in the supplementary material; here, we only briefly summarize our conclusions from these analyses.

Our simulations show that the analysis of quantitative networks is clearly superior to that of binary data of the same network (Fig. 5).

Modularity Q and accuracy are substantially higher in this case. This seems to be the logical consequence of implicitly down-weighting rare species and singletons, whose correct allocation to modules is given a lower importance than that of species with more information (i.e. a higher number of interactions). In a sense, the outcome of such a comparison is trivial: a binary network does not allow any weighting of species by information content. Also, the null model implicit in the modularity criterion of Newman (2004b, eqn. 2) can be computed with more nuances and hence has a higher chance of correctly identifying modules.

The QuanBiMo algorithm is by no means perfect: at high levels of noise, such as one would expect in ecological network data, only two-thirds to three-quarter of all links are correctly classified (Appendix Fig. 9 right). The levelling-off at low noise levels gives us some confidence that a misclassification of 25% in small networks seems realistic, which seems acceptable. For large networks, however, the misclassification of over 40% seems too high for comfort, though. Adapting the settings (specifically the number of steps of no improvement before the algorithm stops) seems a logical option. However, runtimes increase substantially with every order of magnitude of number of steps. The price may be worth paying, as the example of the Memmott network shows, where increasing the values for steps from *1e5* to *1e10* boosted Q from 0·1 to 0·3 (see Fig. 4 for the resulting modules).

Modularity *Q* as a network index

Modularity Q is likely to be correlated with other network metric, as specialization of module members is the prime reason for the existence of modules. Across the 22 quantitative pollination networks of the NCEAS 'interaction webs' database [1], Q was evidently highly positively correlated with complementary specialization H_2' (Fig. 6).

Ecologically, the correlation with specialization makes good sense. Modules only exist because some species do not interact with some others, that is, because they are specialized. An overall low degree of specialization is equivalent to random interactions, which will yield no modules.

Furthermore, the absolute value of Q is, like virtually all network indices (Dormann *et al.* 2009), dependent on network size (i.e. the number of species) as well as the number of links and the total number of interactions observed (see also Thébault 2013). We would thus recommend a null model comparison (e.g. Vázquez & Aizen 2003; Blüthgen *et al.* 2008; Dormann *et al.* 2009), to correct the observed value of Q by

[1]http://www.nceas.ucsb.edu/interactionweb

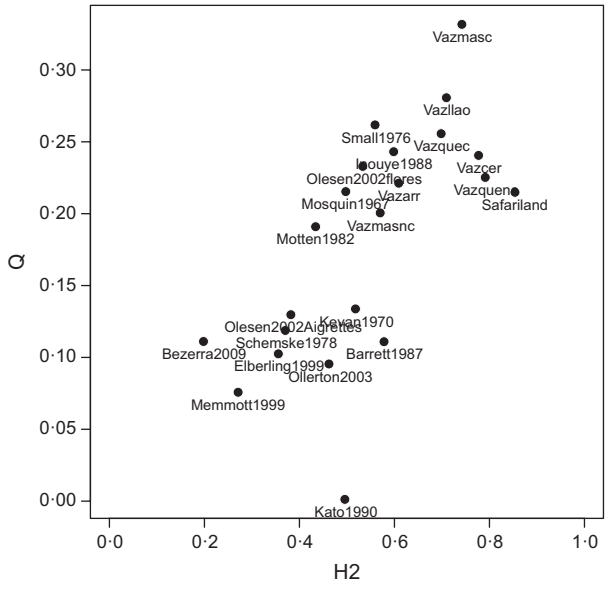

Fig. 6. Modularity (Q) is highly correlated with specialization H_2' (Blüthgen *et al*., 2006) across 22 pollination networks. Names refer to network data sets in *bipartite* which were taken from http://www.nceas.ucsb.edu/interactionweb.

null model expectation (e.g. by standardizing them to z-scores: $z_Q = \frac{Q_{\text{observed}} - \overline{Q}_{\text{null}}}{\sigma_{Q_{\text{null}}}}$; see appendix for an example). The choice of the null model is and will be contentious, since it is not clear whether observed interactions are independent from each other and among species. They may thus differ among network types (e.g. pollination networks may be usefully compared to a null model based on marginal totals, while host–parasitoid networks, with a stronger feedback between the two levels, are likely to be overcorrected by such an approach). Modularity Q is in itself not an index of an ecological feature. It is merely a measure of how well links and interactions can be separated into different modules. Large networks, with many species and links, allow for more combinations of species in modules, leading to higher values of Q, as Allesina (2009a) pointed out for any grouping algorithm.

USING MODULARITY TO IDENTIFYING SPECIES WITH IMPORTANT ROLES IN THE NETWORK

Guimerà *et al*. (2005) and Olesen *et al*. (2007) propose to compute standardized connection and participation values, called c and z, for each species to describe their role in networks, where c refers to the even distribution of links across modules (called 'participation coefficient' P by Guimerà *et al*. 2005) and z refers to within-module degrees. Originally, both are computed based on the number of links, but a weighted version based on species strength (*sensu* Bascompte *et al*. 2006) is implemented, too. Guimerà *et al*. (2005) suggest critical values for a binary and one-mode networks of c and z of 0·625 and 2·5, respectively. Species exceeding both of these values are called 'hubs' because they link different modules, combining high between-module with high within-module connectivity.

In the case of the pollination network of Fig. 1, c-values range between 0 and 0·78 (with 23 of 79 pollinators and 13 of 25 plant species exceeding the threshold of 0·625); z-values range between −1·21 and 5·00 (with two pollinators but no plant species exceeding the value of 2·5: Fig. 7). Put together, only the syrphid *Syritta pipiens* (and hawkbit *Leontodon hispidus* almost) exceeded both thresholds and would thus be called a 'hub species'. As can be seen in Fig. 4 (top), this syrphid is relatively rare but clearly not randomly distributed over the six modules, thus linking modules three, five and six (from the left). In contrast, *Leontodon hispidus* is a common plant species, visited by many different pollinators, and it actually links all modules with the exception of module two.

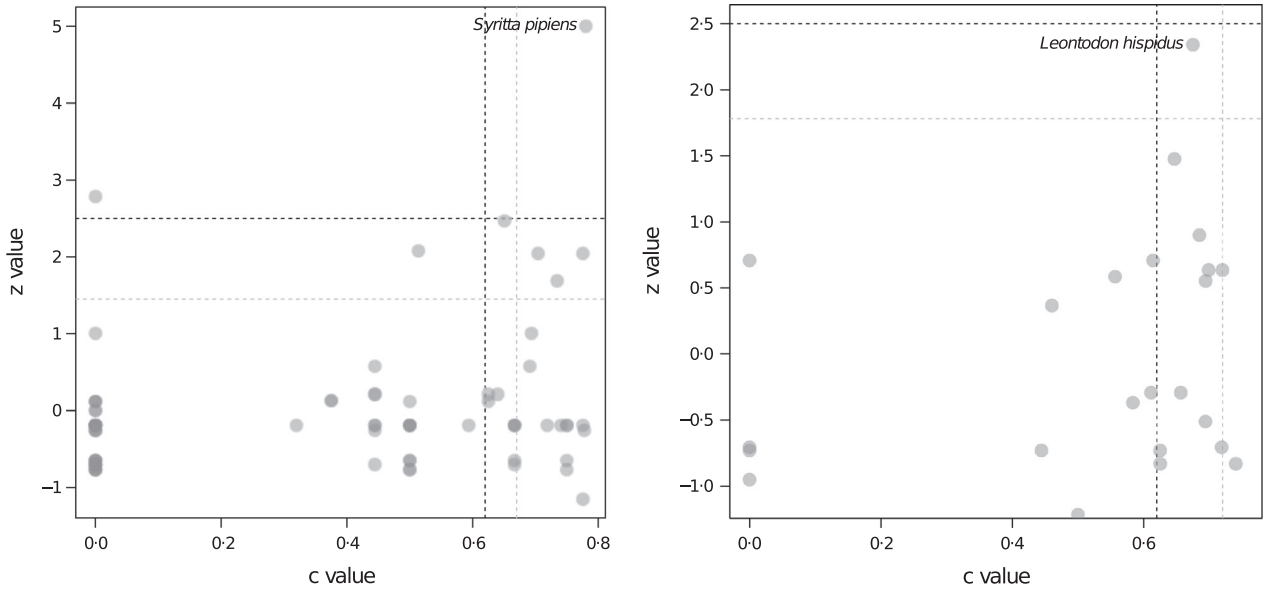

Fig. 7. Connection (c) and participation (z) values for pollinators (left) and plants (right) in the network of Memmott (1999). Dashed black lines indicate critical values according to Olesen *et al*. (2007), those in grey 95% quantiles from 100 null models (see text).

To objectively define this threshold one could run null models of the original network and employ 95% quantiles as critical *c*- and *z*-values. For the pollinators in the network of Fig. 1 these would be 0·67 (±0·039) and 1·45 (±0·220), respectively, based on 100 null models (for plants: $c_{\text{critical}} = 0{\cdot}72 \pm 0{\cdot}036$ and $z_{\text{critical}} = 1{\cdot}78 \pm 0{\cdot}297$; Fig. 7 left). While for plant species this has little effect (except for moving *Leontodon hispidus* across the threshold), three more pollinators would become hub species (the common hoverfly *Episyrphus balteatus*, the tachinid fly *Eriothrix rufomaculata* and undetermined fly '*Diptera* spec.22').

Conclusion

We here presented an algorithm to compute modularity *Q* and detect modules in weighted, bipartite networks. Because it uses the strength of links as quantitative information, this approach is more sensitive and also more specific, than current binary algorithms. By making the algorithm easily available, we hope that network ecology will benefit from new insights into the structure of interaction networks.

Acknowledgements

We like to thank Aaron Clauset for inventing the hierarchical random graph idea and for making his code freely available. CFD acknowledges funding by the Helmholtz Association (VH-NG 247). Many thanks to Lili Ingmann and Matthias Schleuning for extensively testing the robustness and ecological meaningfulness of this algorithm. Additional thanks to Peter Csermely and Pedro Jordano for drawing our attention to some work in progress in this field, to Pedro Peres-Neto and an anonymous reviewer for improving the quality of this publication.

References

Allesina, S. (2009) Cycling and cycling indices. *Ecosystem, Ecology*, (ed. S.E. Jorgensen), pp. 50–57. Elsevier, Amsterdam.

Allesina, S. & Pascual, M. (2009) Food web models: a plea for groups. *Ecology Letters*, **12**, 652–662.

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

Bascompte, J., Jordano, P., Melián, C.J. & Olesen, J.M. (2003) The nested assembly of plant-animal mutualistie networks. *Proceedings of the National Academy of Sciences of the USA*, **100**, 9383–9387.

Bascompte, J., Jordano, P. & Olesen, J.M. (2006) Asymmetric coevolutionary networks facilitate biodiversity maintenance. *Science*, **312**, 431–433.

Blüthgen, N. (2010) Why network analysis is often disconnected from community ecology: A critique and an ecologist's guide. *Basic and Applied Ecology*, **11**, 185–195.

Blüthgen, N., Menzel, F. & Blüthgen, N. (2006) Measuring specialization in species inter-action networks. *BMC Ecology*, **6**, 9.

Blüthgen, N., Fründ, J., Vázquez, D.P. & Menzel, F. (2008) What do interaction network metrics tell us about specialization and biological traits? *Ecology,* **89,** 3387–3399.

Borgatti, S.P. (2006) Identifying sets of key players in a social network. *Computational and Mathematical Organization Theory*, **12**, 21–34.

Borgatti, S.P. & Everett, M.G. (1997) Network analysis of 2-mode data. *Social Networks*, **19**, 243–269.

Cagnolo, L., Salvo, A. & Valladares, G. (2011) Network topology: patterns and mechanisms in plant-herbivore and host-parasitoid food webs. *Journal of Animal Ecology*, **79**, 342–351.

Carstensen, D.W., Dalsgaard, B., Svenning, J.C., Rahbek, C, Fjeldså, J., Sutherland, W.J. & Olesen, J.M. (2011) Biogeographical modules and island roles: a comparison of wallacea and the west indies. *Journal of Biogeography*, **39**, 739–749.

Clauset, A., Newman, M. & Moore, C. (2004) Finding community structure in very large networks. *Physical Review E*, **70**, 066111.

Clauset, A., Moore, C. & Newman, M.E.J. (2008) Hierarchical structure and the prediction of missing links in networks. *Nature*, **453**, 98–101.

Danieli-Silva, A., de Souza, J.M.T., Donatti, A.J., Campos, R.P., Vicente-Silva, J., Freitas,L. & Varassin, I.G. (2011) Do pollination syndromes cause modularity and predict interactions in a pollination network in tropical high-altitude grasslands? *Oikos*, **121**, 35–43.

Dicks, L.V., Corbet, S.A. & Pywell, R.F. (2002) Compartmentalization in plant-insect flower visitor webs. *Journal of Annual Ecology*, **71**, 32–43.

Dormann, C.F. & Strauß, R. (2013) Detecting modules in quantitative bipartite networks: the QuaBiMo algorithm. *arXiv [q-bioQMj*, **1304.3218v.**

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

Dupont, Y.L. & Olesen, J.M. (2009) Ecological modules and roles of species in heath-land plant: insect flower visitor networks. *Journal of Animal Ecology*, **78**, 346–353.

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

Fortunato, S. (2010) Community detection in graphs. *Physics Reports*, **486**, 75–174.

Garcia-Domingo, J.L. & Saldaña, J. (2008) Effects of heterogeneous interaction strengths on food web complexity. *Oikos*, **117**, 333–343.

Guillaume, J.L. & Latapy, M. (2004) Bipartite structure of all complex networks. *Information Processing Letters*, **90**, 215–221.

Guimarães Jr, P.R., Jordano, P. & Thompson, J.N. (2011) Evolution and coevolution in mutualistie networks. *Ecology Letters*, **14**, 877–885.

Guimerà, R., Mossa, S., Turtsehi, A. & Amaral, L.A.N. (2005) The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences of the USA*, **102**, 7794–7799.

Guimerà, R., Sales-Pardo, M. & Amaral, L. (2007) Module identification in bipartite and directed networks. *Physical Review E*, **76**, 036102.

Guimerà, R., Stouffer, D.B., Sales-Pardo, M., Leieht, E.A., Newman, M.E.J. & Amaral, L.A.N. (2010) Origin of compartmentalization in food webs. *Ecology*, **91**, 2941–2951.

Ings, T.C., Montoya, J.M., Bascompte, J., Blüthgen, N., Brown, L., Dormann, C.F. *et al.* (2009) Ecological networks-beyond food webs. *Journal of Animal Ecology*, **78**, 253–269.

Jacobi, M.N., André, C., Döös, K. & Jonsson, P.R. (2012) Identification of subpopulations from connectivity matrices. *Ecography*, **35**, 1004–1016.

Joppa, L.N., Bascompte, J., Montoya, J.M., Solé, R.V., Sanderson, J. & Pimm, S.L. (2009) Reciprocal specialization in ecological networks. *Ecology Letters*, **12**, 961–969.

Kovács, I.A., Palotai, R., Szalay, M.S. & Csermely, P. (2010) Community landscapes: an integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. *PLoS One*, **5**, 1–14.

Lancichinetti, A. & Fortunato, S. (2011) Limits of modularity maximization in community detection. *Physical Review E*, **84**, 066122.

Levins, R. (1975) Evolution in communities near equilibrium. *Ecology and Evolution of Communities*, (eds. M.L. Cody & J.M. Diamond), pp. 16–50. Belknap Press, Cambridge, MA.

Lewinsohn, T.M., Prado, P.I., Jordano, P., Bascompte, J. & Olesen, J.M. (2006) Struetiue in plant-animal interaction assemblages. *Oikos*, **113**, 174–184.

Li, C. & Chen, G. (2006) Modelling of weighted evolving networks with community structures. *Physica A: Statistical Mechanics and its Applications*, **370**, 869–876.

Marquitti, F.M.D., Guimaraes, P.R., Pires, M.M. & Bittencourt, L.F. (2013) Modular: Software for the autonomous computation of modularity in large network sets. *arXiv*, **410**, 1304.2917.

Martín González, A.M., Dalsgaard, B. & Olesen, J.M. (2010) Centrality measures and the importance of generalist species in pollination networks. *Ecological Complexity*, **7**, 36–47.

Martín González, A.M., Allesina, S., Rodrigo, A. & Bosch, J. (2012) Drivers of compartmentalization in a Mediterranean pollination network. *Oikos*, **121**, 2001–2013.

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

Memmott, J. (1999) The structure of a plant-pollinator food web. *Ecology Letters*, **2**, 276–280.

Morris, R.J., Lewis, O.T. & Godfray, H.C.J. (2004) Experimental evidence for apparent competition in a tropical forest food web. *Nature*, **428**, 310–313.

Newman, M.E.J. (2003) The structrue and function of complex networks. *SIAM Review*, **45**, 167–256.

Newman, M. (2004a) Analysis of weighted networks. *Physical Review E*, **70**, 056131.
Newman, M.E.J. (2004b) Fast algorithm for detecting community structure in networks. *Physical Reviews E*, **69**, 066133.
Newman, M.E.J. (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the USA*, **103**, 8577–8582.
Newman, M.E.J. & Girvan, M. (2004) Finding and evaluating community structure in networks. *Physical Review E*, **69**, 1–15.
Olesen, J.M., Bascompte, J., Dupont, Y.L. & Jordano, P. (2007) The modularity of pollination networks. *Proceedings of the National Academy of Sciences of the USA*, **104**, 19891–19896.
Pimm, S.L. (1982) *Food Webs*. Chicago University Press, Chicago.
Pocock, M.J.O., Evans, D.M. & Memmott, J. (2012) The robustness and restoration of a network of ecological networks. *Science*, **335**, 973–977.
Poisot, T., Canard, E., Mouquet, N. & Hochberg, M.E. (2012) A comparative study of ecological specialization estimators. *Methods in Ecology and Evolution*, **3**, 537–544.
Pons, P. & Latapy, M. (2006) Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications*, **10**, 191–218.
Prado, P.I. & Lewinsohn, T.M. (2004) Compartments in insect-plant associations and their consequences for community structure. *Journal of Animal Ecology*, **73**, 1168–1178.
R Development Core Team (2012)*R: A Language and Environment for Statistical Com-pitting*. R Foundation for Statistical Computing, Vienna, Austria.
Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N. & Barabási, A.L. (2002) Hierarchical organization of modularity in metabolic networks. *Science*, **297**, 1551–1555.
Schuetz, P. & Caflisch, A. (2008) Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Reviews E*, **77**, 046112.
Scotti, M., Podani, J. & Jordan, F. (2007) Weighting, scale dependence and indirect effects in ecological networks: A comparative study. *Ecological Complexity*, **4**, 148–159.
Schleuning, M., Fründ, J., Klein, A.M., Abrahamczyk, S., Alarcón, R., Albrecht, M. *et al.* (2012) Specialization of mutualistic interaction networks decreases toward tropical latitudes. *Current Biology*, **22**, 1–7.
Thébault, E. (2013) Identifying compartments in presence-absence matrices and bipartite networks: insights into modularity measures. *Journal of Biogeography*, **40**, 759–768.
Trøjelsgaard, K. & Olesen, J.M. (2013) Macroecology of pollination networks. *Global Ecology and Biogeography*, **22**, 149–162.
Tylianakis, J.M., Tscharntke, T. & Lewis, O.T. (2007) Habitat modification alters the structure of tropical host-parasitoid food webs. *Nature*, **445**, 202–205.
Tylianakis, J.M., Laliberté, E., Nielsen, A. & Bascompte, J. (2010) Conservation of species interaction networks. *Biological Conservation*, **143**, 2270–2279.
Vázquez, D.P. & Aizen, M.A. (2003) Null model analyses of specialization in plant-pollinator interactions. *Ecology*, **84**, 2493–2501.
Vázquez, D.P., Blüthgen, N., Cagnolo, L. & Chacoff, N.P. (2009) Uniting pattern and process in plant-animal mutualistic networks: a review. *Annals of Botany*, **103**, 1445-1457.

Received 21 June 2013; accepted 31 October 2013
Handling Editor: Pedro Peres-Neto

Supporting Information

Additional Supporting Information may be found in the online version of this article.

Appendix S1. (A) Formal definition of the algorithm; (B) description and results of extensive evaluation of the algorithm on simulated data, and (C) R sample session of its use.

Appendix

A Formal definition of the identification of module vertices

Consider an edge $(i, j) \in E$ with weight w_{ij} representing the strength of interaction between vertices i and j. In a bipartite graph $\overline{G}$ maintaining for each vertex its original sum of edge weights, but disregarding the modular structure of G, the weight $\overline{w}_{ij}$ of the edge between vertices i and j is given by

$$\overline{w}_{ij} = \begin{cases} \sum w_{i\cdot} \times \sum w_{\cdot j} \ , & \text{if } i \in V_A \Leftrightarrow j \in V_B \\ 0, & \text{else.} \end{cases} \tag{3}$$

Therefore, the difference of edge weight and expected edge weight

$$w'_{ij} = w_{ij} - \overline{w}_{ij} \tag{4}$$

is positive, if within module, and negative, if outside module.

Therefore, the algorithm attempts to find the best trade-off between a maximum sum of w' within modules and a minimum sum of w' outside.

Given a division of V into a set of non-overlapping subgraphs C, we define

$$g(C) = \begin{cases} \sum\limits_{i \in V_A} \sum\limits_{j \in V_B} \delta_C(i,j) \times w'_{ij} - (1 - \delta_C(i,j)) \times w'_{ij} \ , & \text{if } \forall c \in C\text{: c is connected graph} \\ -\infty, & \text{else,} \end{cases} \tag{5}$$

where

$$\delta_C(i,j) = \begin{cases} 1, & \text{if } i \in c \wedge j \in c \wedge c \in C \\ 0, & \text{else.} \end{cases} \tag{6}$$

Obviously, $g(C)$ has to be maximized in order to find the best division of V into modules C. For achieving this goal, we modify the algorithm of Clauset *et al.* (2008).

Let D be a binary tree with arbitrarily connected internal vertices $v \in V_{\text{intern}}$ and with n leaves representing the vertices of G and initially arranged in an arbitrary order. A module c within D is defined as the set of leaves of the sub-tree rooted at an internal vertex v meeting following requirements:

I v has at least one child being a leaf.

II No ancestor of v has a child being a leaf.

III $c \cap V_A \neq \emptyset \wedge c \cap V_B \neq \emptyset$, i.e. there is at least one vertex $v_A \in V_A$ and at least one vertex $v_B \in V_B$ within c.

Due to requirement I it is obvious that there are at most $min(|V_A|, |V_B|)$ modules. Note that due to requirement II on each path from the root of D to a leaf there is exactly one internal vertex shaping a module. For convenience, we will use the term 'module vertex' for this kind of vertex.

Each internal vertex v is assigned the information r_v whether it is the root of a sub-tree of D representing a module or whether it is below or above such an internal vertex. Let $r_v = 1$ if v is above a module vertex, $r_v = 0$ if v is a module vertex itself and let $r_v = -1$ if v is below a module vertex.

Additionally, each internal vertex v is assigned its contribution g_v to $g(C)$

$$g_v = \begin{cases} + \sum_{i \in \mathcal{L}_v} \sum_{j \in \mathcal{R}_v} w'_{ij} \ , & \text{if } r_v \leq 0 \ \wedge \sum_{i \in \mathcal{L}_v} \sum_{j \in \mathcal{R}_v} w_{ij} > 0 \\ - \sum_{i \in \mathcal{L}_v} \sum_{j \in \mathcal{R}_v} w'_{ij} \ , & \text{if } r_v = 1 \\ -\infty, & \text{else,} \end{cases} \tag{7}$$

where $\mathcal{L}_v$ is the set of leaves of the sub-tree rooted at the left child of v and, analogously, $\mathcal{R}_v$ is the set of leaves of the sub-tree rooted at the right child of v.

For C given by the current state of D, $g(C)$ can now be rewritten as

$$g(C) = \sum_{v \in V_{intern}} g_v \qquad . \tag{8}$$

In order to compute $\max(g(C))$, the subtrees of D have to be re-arranged. The algorithm therefore randomly selects an edge e of D connecting two internal vertices v_i and v_j. Let w.l.o.g. e be the left edge of v_j connecting it to its child v_i. Then there are three subtrees $\mathcal{L}_{v_i}$, $\mathcal{R}_{v_i}$ and $\mathcal{R}_{v_j}$ originating from v_i and v_j, respectively, and two possible rearrangements α and β (Fig. 3) of which one is chosen randomly and simulated. In re-arrangement α, sub-trees $\mathcal{R}_{v_i}$ and $\mathcal{R}_{v_j}$ are permuted, in rearrangement β sub-trees $\mathcal{L}_{v_i}$ and $\mathcal{R}_{v_j}$. The change dg in $g(C)$ resulting from the rearrangement is computed according to r_{v_i} and r_{v_j}.

B Evaluation of the algorithm

The detection of modules has theoretical limits related to the number of between-module links present, the sparceness of the network matrix and the size of the network (e.g. Fortunato & Barthélemy, 2007; Lancichinetti & Fortunato, 2011; Lancichinetti *et al.*, 2010). In the following paragraphs we evaluate the QuaBiMo-algorithm for different simulated networks typical in size and noise for pollination networks. There is no technical reason why the algorithm should not work for much larger networks, too, given enough time for computing a large number of dendrogram configurations. Such an evaluation is outside the scope of this study.

B.1 Simulations to investigate algorithm sensitivity and specificity for noisy network data

We analysed simulated networks of different noisiness to evaluate the performance of the modularity algorithm. We would expect that modules become unidentifiable when the proportion of links within modules becomes as low as between modules. We hence simulate networks with increasing degree of noise by moving, randomly, interactions from within a module to a random position in the adjacency matrix not included in any module (Fig. 8). We simulated two sizes of networks (30×50 and 100×400), two levels of filling (achieved through setting the parameter "size" of the negative binomial distribution to 0.1,"low", or 1, "high"), and two levels of modularisation (3 and 10 modules). Each combination was evaluated for seven noise levels (0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5) and replicated 15 times, yielding 840 different networks. Replicates differ in the size, position of modules and number of interactions per link. Sizes were maintained at the same two levels.

Networks were simulated in three steps. First, we defined the size of the matrix and position and size of the modules. This initial network is a matrix of 0s except for all interactions in a module, which is thus identified by a block of 1s. Then, second, we drew actual interactions for each link of a module from a strongly skewed negative binomial distribution (with size $= 0.05$ and $\mu = 2$), removed 80% (high filling) or 40% (low filling) of 0-values, and then replaced the initial

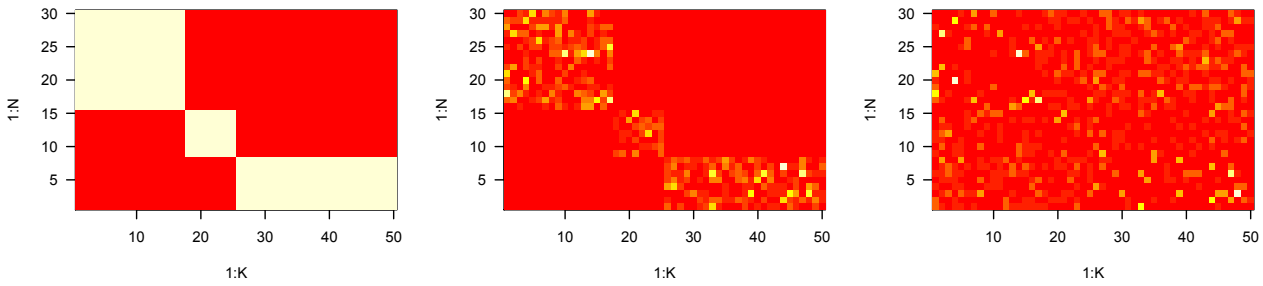

Figure 8: Network simulation starts by defining the modules (left), then allocating to all links a number of interactions drawn from a negative binomial distribution (centre) and finally removing interactions in a module and placing them outside (right). High levels of noise, as shown here, yield poorly defined modules. Cells with a value of 0 are shown in red.

1s of the module blocks by these random values. Accordingly, the modules had a connectance (= filling) of less than 100%. Higher filling of modules generally increases performance. Third, we randomly drew a proportion of interactions from the module and moved it to randomly selected columns and rows of these species outside the module. Thereby we effectively added noise to the network data. There is an upper limit to the third step, where modules become ill-defined. That is the case when the number of interactions outside modules is as high as inside.

We ran the QuaBiMo algorithm five times on each network, saving the result with the highest modularity. This was more efficient in finding a good module configuration than running the algorithm for much longer. For comparison, we also ran the algorithm on a binarised version of the data. The code for simulations and analysis is available in appendix B; runtime for the simulations was approximately two months on a standard desktop computer with 32 GB RAM.

Congruence between the original assignment to modules and the one identified by the algorithm was assessed by means of a confusion matrix. Each link existing in the simulated data was classified as correctly belonging to a module, falsely assigned to a module, falsely not assigned to a module, or correctly not assigned to a module. The confusion matrix was then summarised as sensitivity, specificity and accuracy.

B.2 Simulation results: modularity Q in binary and weighted networks

Modularity Q was strongly dependent on network size, the amount of noise added and the number of modules (Table 1). Most importantly, however, our quantitative approach strongly improved on modularity based on binary data, particularly for large networks (Fig. 9). Deterioration of the module detection with increasing network size could possibly be compensated for by increasing the number of swaps before terminating the search (see example session below). The loss of skill with increasing noise (Fig. 9, right) cannot be alleviated. Here the ability of QuaBiMo to use not only the binary but the weighted link information is already a dramatic improvement.

In the following paragraphs, we shall only be looking at the results for the weighted networks, since that is the explicit focus of the QuaBiMo algorithm.

B.3 Simulation results: modularity Q

Modularity Q and overall accuracy were affected very similarly by network size, noise and the number of modules (Table 2). The most prominent effects were those of size, noise and their interaction, depicted for Q and overall accuracy in Fig. 10. Evidently, larger networks are more difficult to modularise, as are those with a higher level of noise.

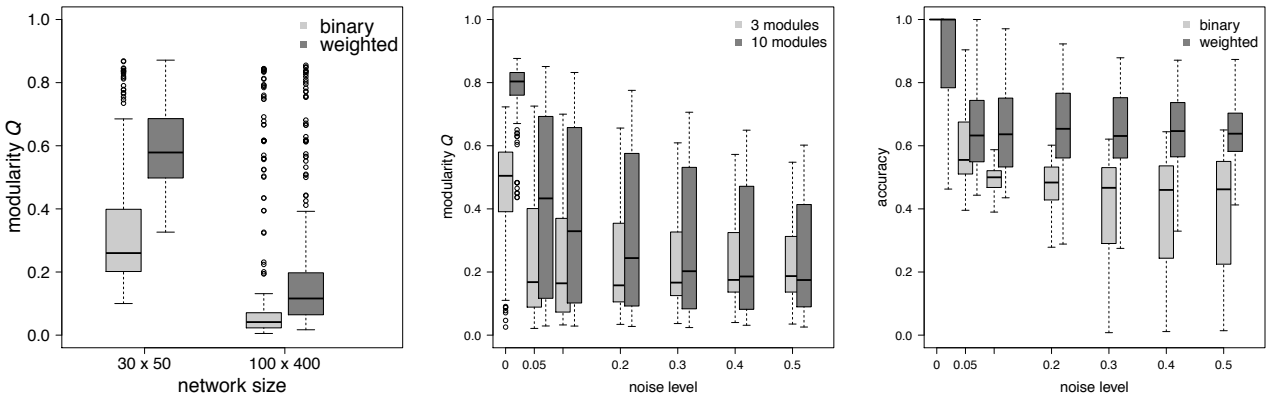

Figure 9: Quality of modularity detection (left and middle: Q; right: overall accuracy) depends on network size, the amount of noise and type of information (binary or weighted). (This figure is the reprint of Fig. 5 in the main body of the publication.)

Table 1: Effect of different simulation parameters on modularity Q and overall accuracy. Sum of squares and F-value can be taken as a measure of how strongly these parameters effect modularity. No significances are given since a test of an effect is nonsensical for simulations. Information refers to binary vs. weighted networks. 'Noise' has seven levels and was analysed as continuous variable.

Modularity Q	df	sum of squares	F value
noise	1	35.16	2038
size	1	76.34	4424
fill	1	1.97	114
no.of.modules	1	12.45	722
information	1	28.02	1624
noise:no.of.modules	1	7.61	441
noise:information	1	0.21	12
size:fill	1	1.81	105
size:no.of.modules	1	3.59	208
size:information	1	9.20	533
Residuals	4106	70.86	
Overall accuracy	df	sum of squares	F value
noise	1	38.31	1568
size	1	1.93	79
fill	1	1.49	61
no.of.modules	1	4.20	172
information	1	22.84	935
noise:no.of.modules	1	0.46	19
noise:information	1	11.63	476
size:fill	1	0.66	27
size:no.of.modules	1	0.57	23
size:information	1	3.73	153
fill:no.of.modules	1	0.34	14
fill:information	1	0.43	18
no.of.modules:information	1	3.58	147
Residuals	4103	100.21	

Table 2: Effect of different simulation parameters on modularity Q for weighted networks. Sum of squares and F-value can be taken as a measure of how strongly these parameters effect modularity.

	df	sum of squares	F value
noise	1	14.90	1164
size	1	69.19	5402
fill	1	0.88	69
no.of.modules	1	7.39	577
noise:size	1	0.11	9
noise:no.of.modules	1	2.16	169
size:fill	1	0.74	58
size:no.of.modules	1	1.15	90
Residuals	2048	26.23	

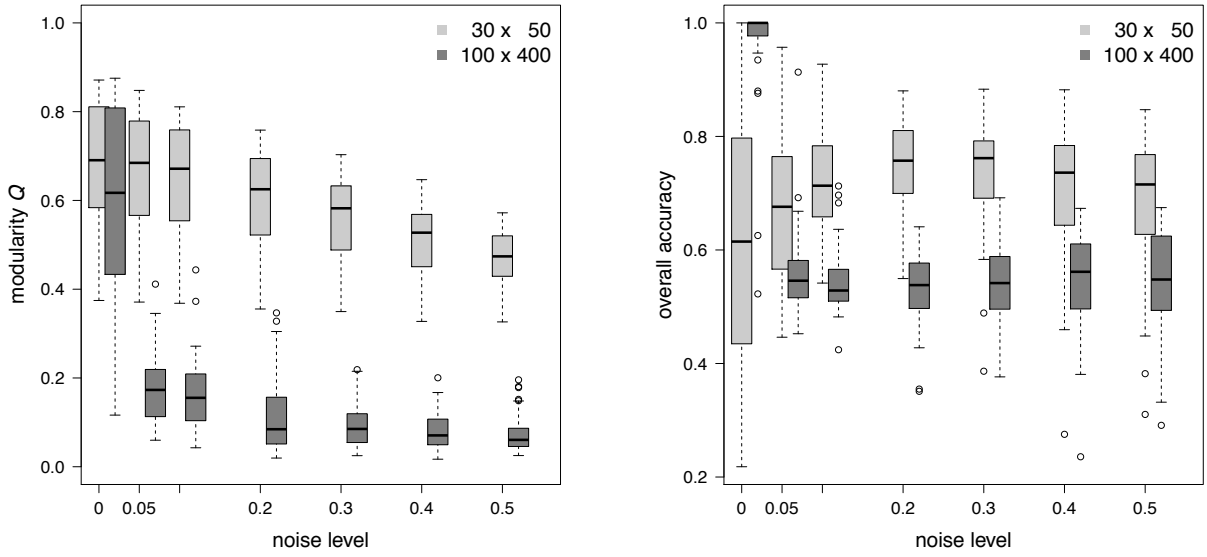

Figure 10: Effect of noise and network size on modularity Q (left) and overall accuracy (left).

Table 3: Effect of different simulation parameters on module identification accuracy (weighted networks only).

	df	sum of squares	F value
noise	1	3.84	237
size	1	5.49	339
fill	1	1.75	108
no.of.modules	1	0.01	1
noise:size	1	1.03	64
noise:no.of.modules	1	1.96	121
size:fill	1	0.33	20
size:no.of.modules	1	0.67	42
fill:no.of.modules	1	0.08	5
Residuals	2047	33.18	

B.4 Simulation results: classification accuracy

While modularity Q gives an indication of how well observed links could be grouped into modules (with a value of 1 indicating that all links are within and none between modules), we can also quantify the algorithm's accuracy based on a confusion table. Overall accuracy (= correct classification rate) is the proportion of all links correctly placed, i.e. (number of links correctly placed into modules + number of links correctly placed between modules)/total number of links. Since the purpose of the algorithm is the use of weighted network data, we here only present results for the weighted and not for the binary networks.

The overall accuracy of module detection decreased with increasing noise levels (Table 3), an effect more pronounced for large networks than for small ones (Fig. 10 right). Again, this interaction probably could have been reduced if more steps until termination were allowed for the larger networks.

B.5 Simulation results: sensitivity and specificity

Classification accuracy has two elements: the correct classification of all module links as belonging to modules (sensitivity) and the correct identification of between-module links as *not* belonging into modules (specificity). For the detection of patterns in networks high sensitivity is desirable, although this may inflate type II errors (i.e. we may identify modules that do not really exist). High specificity indicates that links allocated into modules are indeed correct, but possibly at the expense of not allocating many links to modules overall (leading to inflated type I errors).

Sensitivity and specificity of the QuaBiMo-algorithm were driven by the same factors as overall accuracy (Table 4). Increasing noise levels reduced both sensitivity and specificity, as did larger networks (Fig. 11). Specificity, i.e. the correct identification of non-module links, was strongest driven by the number of modules (0.55 ± 0.17 (1 sd) for networks with three modules *vs* 0.75 ± 0.16 in networks with ten modules). Apparently the incomplete filling of the network led to an identification of several small modules, which reduced sensitivity for networks with few, but not with many modules.

C Identifying modules - a sample session

The QuaBiMo-algorithm is implemented in C++ and is made available through the open source R-package `bipartite` (Dormann *et al.*, 2009). The most important function is `computeModules`, which takes three arguments: the matrix representing the bipartite network data ("web"), a specification of how many MCMC moves should yield no improvement before the algorithm

Table 4: Effect of different simulation parameters on sensitivity and specificity of module identification (weighted networks only).

Sensitivity	df	sum of squares	F-value
noise	1	6.63	382
size	1	4.81	278
fill	1	0.84	48
no.of.modules	1	20.44	1178
noise:size	1	4.32	249
noise:fill	1	0.67	39
size:no.of.modules	1	2.72	157
Residuals	2049	35.54	
Specificitiy	df	sum of squares	F-value
noise	1	9.43	465
size	1	12.91	636
fill	1	4.55	224
no.of.modules	1	5.52	273
noise:size	1	0.17	8
size:fill	1	0.55	27
fill:no.of.modules	1	13	
Residuals	2049	41.61	

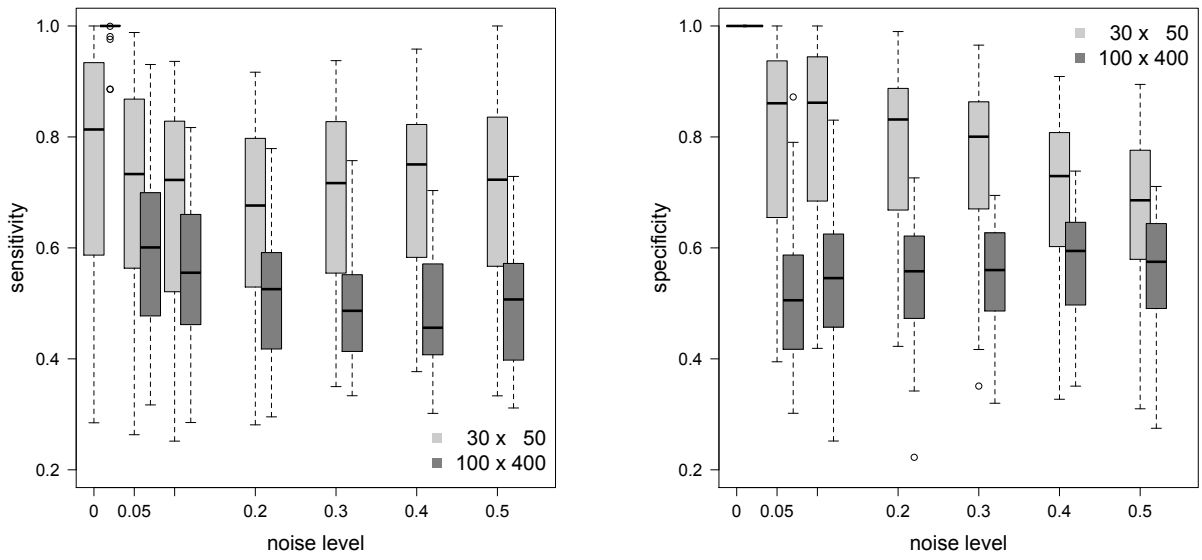

Figure 11: Effect of noise and network size on sensitivity (right) and specificity (left) of the classification of links into modules.

stops ("steps", with default `1e6`) and a logical switch for computing nested modules ("deep", defaulting to `FALSE`). The number of steps should be adapted to the size of the network (see previous sections). We found that Q levels off very soon, once the default of one million is exceeded. However, we have not extensively trialled this setting for networks larger than that used below.

As a typical analysis we shall use the relatively large (25×79) and well-sampled pollination network of Memmott (1999), which is provided along with the bipartite package:

```
> library(bipartite)
> mod <- computeModules(web=memmott1999, steps=1E8)
```

The evaluation of these two lines will usually take about one minute and perform around 20 million MCMC moves. The resulting object stores the module composition and the likelihood of the solution found. The modularity value Q of this solution is simply the likelihood value (0.18, this value may vary between runs; random seeding is not supported):

```
> mod@likelihood
[1] 0.18
```

We can now plot the resulting modules to visualise the compartments (Fig. 4 top).

```
> plotModuleWeb(mod)
```

To identify nested modules, we choose a lower value for steps (to reduce computation time), thus also yielding a different module structure at the highest level. Modularity value Q will still be based on the non-recursive algorithm.

```
> modn <- computeModules(memmott1999, steps=1E6, deep=T)
```

To be able to ecologically interpret these modules (Fig. 4 bottom), expert knowledge on the system is required. The computation of modularity is primarily an explorative tool helping the user to objectively detect pattern in typically noisy network data.

C.1 Using Q as an index

To account for Q's dependence on network size and sampling intensity, we compute null model-expectations and turn the observed value of Q into a z-score. In R, this could be achieved by the following code (which will take more than one hour since we are computing modules in 100 null model networks):

```
> nulls <- nullmodel(memmott1999, N=100, method="r2d")
> modules.nulls <- sapply(nulls, computeModules)
> like.nulls <- sapply(modules.nulls, function(x) x@likelihood)
> (z <- (mod@likelihood - mean(like.nulls))/sd(like.nulls))

[1] 7.088665
```

This means that the observed modularity is 7 standard deviations higher than would be expected from random networks with the same marginal totals (representing abundance distributions of plants and pollinators). Since z-scores are assumed to be normally distributed, values above ≈ 2 are considered significantly modular.

C.2 Identifying species with importance for modularity

To compute c- and z-values as proposed by Guimerà *et al.* (2005) and Olesen *et al.* (2007), which, for each species, describe their role in networks, we simply call the function `czvalues`. Originally, and by default, both are computed based on the number of links, but a weighted version based on species strength (*sensu* Bascompte *et al.*, 2006) is implemented, too.

```
> czvalues(mod) # for all species
> czvalues(mod, level="lower") # for  lower trophic level
> czvalues(mod, level="lower", weighted=TRUE) #based on strength
```

The output of the last command is:

```
$c
 Agrimonium.eupatorium    Leontodon.autumnalis      Lotus.corniculatus       Medicago.lupulina
             0.64092549              0.21672839              0.00000000              0.00000000
        Rubus.fruticosus    Hypochaeris.radicata         Centaurea.nigra  Euphrasia.officinalis
             0.00000000              0.08889145              0.30144047              0.41741018
      Linum.catharticum   Convolvulus.arvensis         Knautia.arvensis        Aethusa.cynapium
             0.65209520              0.31525184              0.44070394              0.58554442
 Eupatorium.cannabinum          Plantago.major     Leontodon.saxatilis      Lathyrus.pratensis
             0.32225369              0.00000000              0.25853701              0.48347107
        Clematis.vitalba        Senecio.jacobaea     Trifolium.pratense Chamerion.angustifolium
             0.55612161              0.35602360              0.11623699              0.62822506
     Leontodon.hispidus       Crepis.capillaris         Torilis.japonica    Angelica.sylvestris
             0.33256384              0.66497432              0.66366078              0.15796269
         Daucus.carota
            0.36472241

$z
 Agrimonium.eupatorium    Leontodon.autumnalis      Lotus.corniculatus       Medicago.lupulina
             -0.7071068              0.0383543              -1.0496042              -0.4840071
        Rubus.fruticosus    Hypochaeris.radicata         Centaurea.nigra   Euphrasia.officinalis
             -0.7071068              0.1170909               0.9731430                      NA
      Linum.catharticum    Convolvulus.arvensis         Knautia.arvensis        Aethusa.cynapium
             -0.4629702             -0.4838904               0.5310580              -0.5293687
 Eupatorium.cannabinum          Plantago.major     Leontodon.saxatilis      Lathyrus.pratensis
              1.5704676              0.7071068              -0.4780049              -1.1121634
        Clematis.vitalba        Senecio.jacobaea     Trifolium.pratense Chamerion.angustifolium
             -0.7399423             -0.3949799               0.6575666              -0.7252687
     Leontodon.hispidus       Crepis.capillaris         Torilis.japonica     Angelica.sylvestris
              2.5923834             -0.4439763               0.4241122                      NA
         Daucus.carota
             0.7071068
```

Supplementary material references

Bascompte, J., Jordano, P. & Olesen, J.M. (2006) Asymmetric coevolutionary networks facilitate biodiversity maintenance. *Science*, **312**, 431–433.

Clauset, A., Moore, C. & Newman, M.E.J. (2008) Hierarchical structure and the prediction of missing links in networks. *Nature*, **453**, 98–101.

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

Fortunato, S. & Barthélemy, M. (2007) Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the USA*, **104**, 36–41.

Guimerà, R., Mossa, S., Turtschi, A. & Amaral, L.A.N. (2005) The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences of the USA*, **102**, 7794–9.

Lancichinetti, A. & Fortunato, S. (2011) Limits of modularity maximization in community detection. *Physical Review E*, **84**, 066122.

Lancichinetti, A., Kivelä, M., Saramäki, J. & Fortunato, S. (2010) Characterizing the community structure of complex networks. *PLoS ONE*, **5**, e11976.

Memmott, J. (1999) The structure of a plant-pollinator food web. *Ecology Letters*, **2**, 276–280.

Olesen, J.M., Bascompte, J., Dupont, Y.L. & Jordano, P. (2007) The modularity of pollination networks. *Proceedings of the National Academy of Sciences of the USA*, **104**, 19891–19896.