Do presence-absence bipartite networks share the same communities as their quantitative counterparts? Methods for detecting modularity in weighted networks

Stephen Beckett

December 15, 2014

Biosciences, College of Life and Environmental Sciences, University of Exeter, EX4 4QE

Key words: matrices, plant-pollinator, modular structure, network ecology, community detection, two-mode networks, compartments

Note this document is currently a draft. It is not yet finished!

Introduction

Bipartite (two-mode) networks are composed of interactions between two distinct sets of nodes. Identifying structure within these networks is a first step to understanding their formation, function and behaviour. Modularity is an evaluation of the partitioning of nodes into separate subsets, forming modules (also known as groups, compartments, communities, subgraphs) which in bipartite networks are composed of both types of nodes. This property is important in networks across a diverse set of fields including sociology, ecology and the physical sciences.

Modularity is highest when modules are isolated from the rest of the network i.e. nodes in each module highly interact amongst themselves, but between module interactions are limited. A negative modularity score indicates there are fewer edges between nodes within the same module than expected; whilst a positive score indicates that within module connectivity is higher than expected (there is a strict upper limit of 1, though this often cannot be reached) (Newman, 2010).

QuanBiMo (Dormann and Strauss, 2014) is the first algorithm to maximise weighted modularity in bipartite networks with quantitative data. Previously (with the exception of (Newman, 2004)) modularity algorithms (and community detection methods in general) have only been designed and tested on networks of binary data (Fortunato, 2010) i.e. focussing only on whether two nodes have associations regardless of the strength of those associations. It may be possible to adapt some of the methods available for binary data to deal with quantitative information, rather than having to discard this important data dimension.

The LPAb+ algorithm (Liu and Murata, 2010) for maximising modularity in binary bipartite networks has been shown to outperform seven other available methods for binary networks (Liu and Murata, 2010; Costa and Hansen, 2014) whilst retaining a fast time complexity. These qualities made it a good candidate choice to see if this algorithm could be extended for dealing with weighted networks.

I show how the LPAb+ algorithm can be modified so it can detect weighted modularity and denote this algorithm LPAwb+. A further modification to allow a more thorough search of modularity space called Exhaustive LPAwb+ is also presented. I compare the three weighted modularity algorithms using a dataset of 23 plant-pollinator networks. These experiments show that Exhaustive LPAwb+ and QuanBiMo perform well on smaller networks and that the speed of LPAwb+ makes it suitable for usage on larger datasets. We also reveal how the inclusion of quantitative information in the network can alter identified modular structure and discuss the implications of this.

Methods

Finding Modules

We first outline the LPAb+ algorithm(Liu and Murata, 2010) for binary networks then show how it can be extended to weighted networks.

Binary bipartite modularity

For an undirected binary network (showing presence or absence of a connection) of n nodes with m edges, described by the adjacency matrix A; modularity can be defined as (Newman, 2006):

$$Q = \frac{1}{2m} \sum_{u=1}^{n} \sum_{v=1}^{n} (A_{uv} - P_{uv}) \, \delta (L_u, L_v) \tag{1}$$

where P is the probability matrix associated with the network describing the probability of a connection occurring between two nodes when the total number of edges between nodes is constrained. The connection probability between two nodes u and v is calculated as $P_{uv} = \frac{k_u k_v}{2m}$ where k_u is the number of nodes that connect to node u and k_v is the number of nodes that connect to node v. Each node in the network is assigned a module label L - nodes with the same label are in the same module. The Kronecker delta function $\delta (L_u, L_v)$ is equal to one when nodes u and v are classified as being in the same module (i.e. they have the same label value) or zero otherwise.

Bipartite or two-mode networks are made of two types of distinctive nodes such that interactions only occur between nodes of the opposite type. For example in plant-pollinator networks pollinating species cannot pollinate other pollinating species and plants cannot visit each one another; the only interactions are between different plants and pollinators as shown in figure 1. To generalise we can say that we have two node types: red and blue - and that interactions are only allowed between red and blue nodes. If we say there are r nodes of the red type and c nodes of the blue type such that $n = r + c$, we can rewrite A and P in block diagonal form as:

$$\begin{cases} A = \begin{pmatrix} 0_{r \times r} & \tilde{A}_{r \times c} \\ \tilde{A}^T_{c \times r} & 0_{c \times c} \end{pmatrix} \\ P = \begin{pmatrix} 0_{r \times r} & \tilde{P}_{r \times c} \\ \tilde{P}^T_{c \times r} & 0_{c \times c} \end{pmatrix} \end{cases} \tag{2}$$

where $\tilde{A}$ and $\tilde{P}$ are the incidence(biadjacency) matrices describing the connections between the different types of nodes (here T indicates the matrix transpose). This formulation allows bipartite modularity to be written as(Barber, 2007):

$$Q_B = \frac{1}{m} \sum_{u=1}^{r} \sum_{v=1}^{c} \left(\tilde{A}_{uv} - \tilde{P}_{uv} \right) \delta (g_u, h_v) \tag{3}$$

where g are the labels for red nodes and h are the labels for blue nodes. I will leave this equation in the form:

$$Q_B = \frac{1}{m} \sum_{u=1}^{r} \sum_{v=1}^{c} \left(\tilde{A}_{uv} - \frac{k_u d_v}{m} \right) \delta (g_u, h_v) \tag{4}$$

where k describes the node degree for red nodes (the number of blue nodes each red node interacts with) and d describes the node degree for blue nodes (the number of red nodes each blue node associates with).

LPAb+

We first use the dimensions of the network to decide how to run the algorithm; this is because a bipartite community can have at the most $F = min(r, c)$ communities. The LPAb+ algorithm is then initialised by giving a unique label to each of the nodes in the smallest of the two sets. The LPAb+ algorithm is sensitive to the initial labelling of nodes - this can lead to different values of modularity

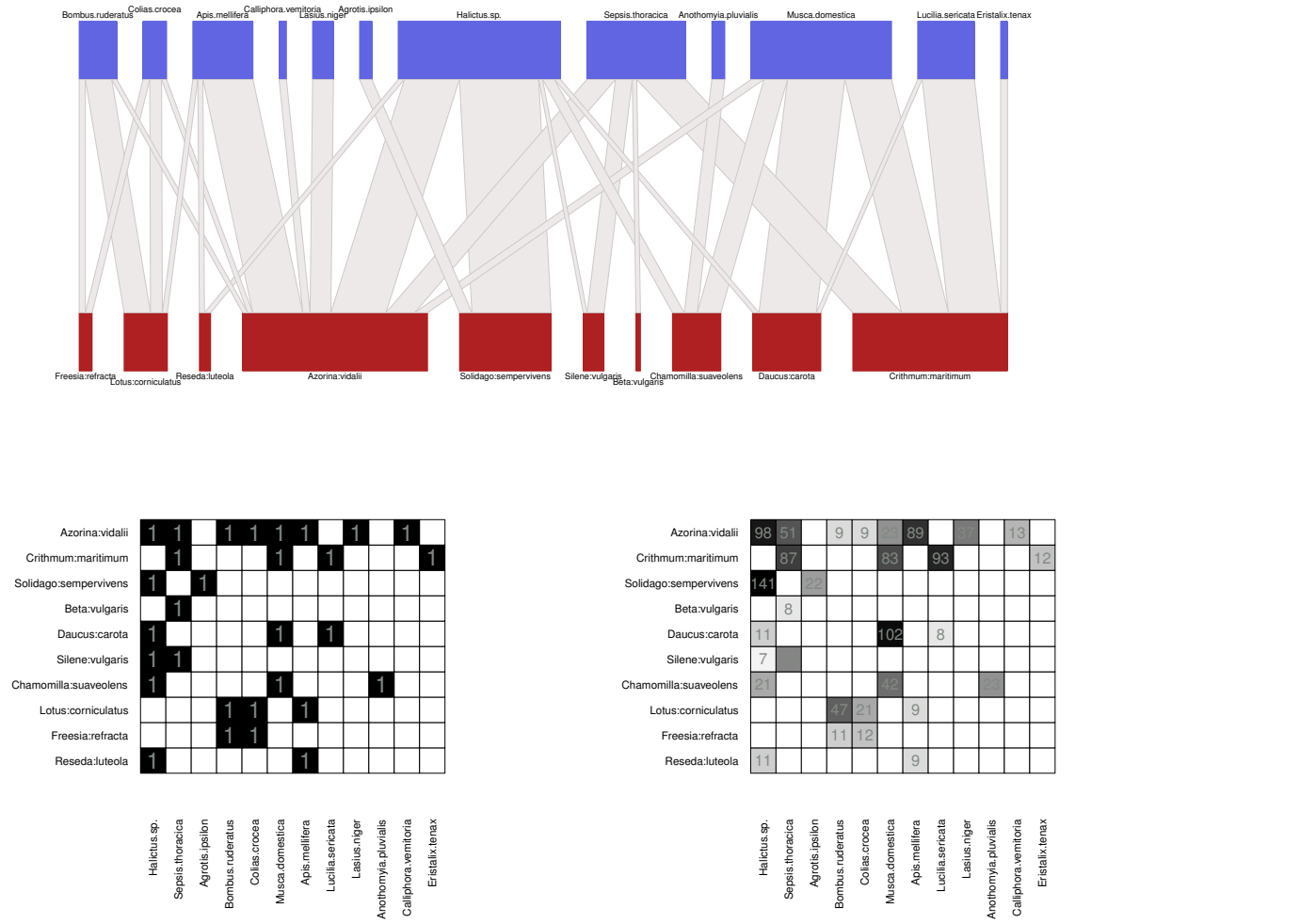

Figure 1: (A) The olesen2002flores bipartite network of 12 species of pollinators (blue nodes) visiting 10 plant species (red nodes). The width of the edges represents the number of pollinator-plant visitations. (B) The same network represented by the interaction matrix denoted $\tilde{A}$ in the text, where the plant species are represented as rows and the pollinator species as columns and the presence of visitations between a pollinator and plant species is represented by a 1. (C) The incidence matrix $\tilde{A}$ is the binary equivalent of $\tilde{W}$, the weighted interaction matrix shown here. The cell numbers correspond to the number of visitations that occurred (where there is no number - there were 0 visitations)

being reported. To combat this issue we randomly assign these initial node labels and suggest running the LPAb+ algorithm multiple times on a given network to find the greatest modularity score.

Algorithm 1 LPAwb+ pseudo-code (Liu and Murata, 2010)

```
Inputs :  an interaction matrix
Output :  row module labels, column module labels, modularity score

1    start
2
3        Find the smallest of the matrix dimensions and make these the red nodes
4        Initialise and randomly assign a unique label to each red node
5        Initialise the blue labels
6        run Stage1:  Repeatedly update labels to locally maximise modularity
7
8        find the number of communities
9
10        while joining communities will result in increased modularity:  {
11            run Stage2:  Merge two communities that will increase modularity most
12            run Stage1:  Repeatedly update labels to locally maximise modularity
13            find the number of communities
14        }
15
16        Assign red and blue labels to row and column labels (see line 3)
17
18    return row labels, column labels and modularity
```

Stage 1 - label propagation stage - bottom up

Asynchronous updating of red, then blue labels on the network is performed to locally maximise modularity. For each node we try to maximise modularity (equation (4)). For a particular red node x this can be written as choosing a new label g_x by trying to maximise the condition:

$$\begin{aligned} g_x &= \left(\sum_{v=1}^{c}\left(\tilde{A}_{xv} - \tfrac{k_x d_v}{m}\right)\right)\delta\left(g, h_v\right) \\ &= \left(\sum_{v=1}^{c}\tilde{A}_{xv}\delta\left(g, h_v\right) - \sum_{v=1}^{c}\left(\tfrac{k_x d_v}{m}\right)\delta\left(g, h_v\right)\right) \end{aligned} \tag{5}$$

Red nodes only use information about the blue nodes to update their labels (g) and similarly blue node labels (h) are updated only using information about the red nodes. Simplifying equation (5) and creating an analogue for the updating rules for blue node labels leads to the following set of conditions:

$$\begin{cases} g_x^{new} = \arg\max\limits_{g}\left(N_{xg} - \tfrac{k_x D_g}{m}\right) \\ h_x^{new} = \arg\max\limits_{h}\left(N_{xh} - \tfrac{K_h d_x}{m}\right) \end{cases} \tag{6}$$

where the new label assigned to node x of type g (red) or h (blue) is that which maximises g or h on the right-hand side (if more than one solution exists, one is chosen at random). Here N_{xg} is the number of nodes that connect to x labelled g, while D_g is the sum of blue node degrees labelled g and K_h is the sum of red node degrees labelled h. As these 'bottom-up' updating rules (equation (6)) are mutually exclusive of one another they can be applied asynchronously such that blue labels are updated, then red nodes are updated, then blue nodes are updated and so on until modularity (equation (4)) can no longer be increased.

Stage 2 - agglomeration stage - top down

When modularity can no longer be increased via stage 1's 'bottom-up' steps, a localised maximum of modularity for the network is reached, however this may not be the global maximum. The second stage seeks to prevent the algorithm getting stuck at local maxima by merging groups of communities

together. Each identified community t is composed of blue and red nodes that share the same label i.e. when $g_u = h_v$. If there are F communities in total, then the merging of two different communities t_i and t_j can only occur if this would result in an increase in network modularity and if there is no third community t_k ($1 \leq k \leq F$, $i \neq j \neq k$) whose merger with t_i or t_j would result in a larger increase to modularity.

Once this merger of communities is completed, stages 1 and then stage 2 are repetitively repeated until the networks communities can no longer be combined - this occurs when it is no longer possible to increase network modularity by merging any of the possible communities together. These modules (communities) and the modularity of this partition are the solution provided by the LPAb+ algorithm.

Weighted bipartite modularity

Weighted modularity can be defined as (Newman, 2004; Dormann and Strauss, 2014):

$$Q = \frac{1}{2M} \sum_{u=1}^{n} \sum_{v=1}^{n} \left(W_{uv} - E_{uv}\right) \delta\left(L_u, L_v\right) \tag{7}$$

where W is the weighted adjacency matrix, M is the sum of all the edge weights in W and E is a probability matrix defined as $E_{uv} = \frac{T_u T_v}{2M}$ with T_u being the marginal total of nodes connected to node u. As before this can be rewritten in biadjacency matrix form, simplifying to (Dormann and Strauss, 2014):

$$\begin{aligned} Q_W &= \frac{1}{M} \sum_{u=1}^{r} \sum_{v=1}^{c} \left(\tilde{W}_{uv} - \tilde{E}_{uv}\right) \delta\left(g_u, h_v\right) \\ &= \frac{1}{M} \sum_{u=1}^{r} \sum_{v=1}^{c} \left(\tilde{W}_{uv} - \frac{y_u z_v}{M}\right) \delta\left(g_u, h_v\right) \end{aligned} \tag{8}$$

where y is the row marginal totals and z is the column marginal totals of $\tilde{W}$. Q_W is the quantity that we wish to maximise with this algorithm for our observed networks. For a binary network W will be equivalent to the adjacency matrix, the marginal totals will equal the node degrees and M will equal the fill - thus equation (8) will reduce to equation (4) for a binary network. Furthermore we can reformulate equation (8) into its matrical form (Barber, 2007; Flores et al., 2014) to allow for vectorised computation as:

$$Q_W = \frac{1}{M} tr\left(R\left(\tilde{W} - \tilde{E}\right) C\right) \tag{9}$$

where for a network with F communities, R is the $F \times r$ red label matrix and C is the $c \times F$ blue label matrix. R and C are binary matrices with a single 1 in each row (column) indicating which community each row (column) node belongs to (this information is held by the red and blue labels). This definition of weighted bipartite modularity can now be used in the modified framework of (Liu and Murata, 2010)'s LPAb+ algorithm.

LPAwb+

To maximise weighted modularity alterations are needed so that quantitative information rather than the binary features of the network are utilised. The node label updating rules in stage 1 are redefined as:

$$\begin{cases} g_x^{new} = \arg\max\limits_{g} \left(\Omega_{xg} - \frac{y_x Z_g}{M}\right) \\ h_x^{new} = \arg\max\limits_{h} \left(\Omega_{xh} - \frac{Y_h z_x}{M}\right) \end{cases} \tag{10}$$

where Z_g is the sum of the values in z (column marginal totals) whose red nodes are labelled by g, whilst Y_h is the sum of the values in y (row marginal totals) for the blue nodes labelled h. Ω_{xg} is the sum of the weighted interactions between node x and all of its neighbours with the label g. Note that equation (10) will reduce to equation (6) for a binary network.

The second stage of the LPAwb+ algorithm is as before, but hinges on the new definition of weighted modularity, such that communities are merged together only when this results in the greatest of the possible changes to the networks weighted modularity score.

QuanBiMo

The quantitative bipartite modularity algorithm (QuanBiMo) of (Dormann and Strauss, 2014) uses a simulated annealing method to attempt to maximise weighted bipartite modularity. It is a C++ routine that is available in the R package bipartite(Dormann et al., 2008) through the function `computeModules`. We used the default settings in bipartite version 2.04. QuanBiMo is stochastic and can also find different modular configurations for the same network - we suggest running it multiple times to find the partition with the greatest modularity score.

Exhaustive LPAwb+

Exploratory research with QuanBiMo and LPAwb+ revealed LPAwb+ often got stuck in a suboptimal solution with a larger number of modules, when compared with QuanBiMo, as LPAwb+ starts by identifying the largest possible number of modules, then iteratively merges them until modularity cannot be increased.

Knowing that LPAwb+ is sensitive to node label initialisation (Liu and Murata, 2010) and that it performs faster than QuanBiMo I designed a new algorithm, Exhaustive LPAwb+ (see algorithm 2). Exhaustive LPAwb+ essentially computes LPAwb+ multiple times with different random initialisations of node labels chosen from μ unique possible labels; and returns the solution which finds the greatest modularity score.

Exhaustive LPAwb+ takes three inputs; the incidence matrix for the network of interest, the number of times that LPAwb+ should be run for each value of μ, and the minimum number of unique labels (modules) to start running LPAwb+ with. Therefore μ ranges between this minimum value and the number of modules returned by a single execution of the LPAwb+ algorithm (when each node is initialised with a unique label) which is used as an upper limit.

Algorithm 2 Pseudo-code for Exhaustive LPAwb+

```
Inputs :  an interaction matrix, minimum number of modules, repititions
Output :  row module labels, column module labels, modularity score

start

    Sol1 = run LPAwb+
    M = number of modules found in Sol1

    for A from minimum number of modules up to M: {
        for every repetition:  {
            Sol2 = run LPAwb+ with A initial modules
            if Sol2 has greater modularity than Sol1:
                 Sol1 = Sol2
         }
     }

 return row labels, column labels and modularity from Sol1
```

Setting the minimum number of modules to search for small, and the number of repetitions high will increase the chance of detecting the global modularity optimum for a network; but is likely to be computationally costly. We choose to give Exhaustive LPAwb+ default settings of ten repetitions for each value of μ, starting from a minimum of four modules (note this does not preclude solutions with fewer modules being identified due to the merging process in LPAwb+) as the speed taken to perform these calculations appeared favourable to QuanBiMo for the test datasets.

Comparing Modularity

Normalised Modularity

The modularity values of Q_B and Q_W found above are network specific - properties such as the size and number of links in a network effect the magnitude of the modularity found (Newman, 2010; Thébault, 2013; Dormann and Strauss, 2014). In order to assess the strength of assortative mixing across different networks it is neccessary to account for the possibility of these effects. (Dormann and Strauss, 2014) recommend using a null model to generate an ensemble of networks from wich the standardised effect size of modularity can be assessed as a z-score. However, it is unclear what would make an appropriate null model for weighted networks. An alternative method is to normalise the modularity values by the maximum value that modularity can take, found in the perfectly mixed network, where all edges are assigned to a module and there are no between module links (Newman, 2010). For bipartite networks this is given as:

$$Q_{max} = \frac{1}{M}\left(M - \sum_{u=1}^{r}\sum_{v=1}^{c}\frac{y_u z_v}{M}\right)\delta\left(g_u, h_v\right) \tag{11}$$

, where as before M is the sum of the edges in the incidence matrix with y, the marginal row totals and z, the marginal column totals. Then normalised modularity is found as:

$$Q_{norm} = \frac{Q}{Q_{max}} \tag{12}$$

Normalised Mutual Information

The normalised mutual information criterion(Ana and Jain, 2003) is used as a way to compare the similarity of network structures found by different community detection methods (Danon et al., 2005; Thébault, 2013). For two different partitions A and B of the same network with a total of n nodes (red and blue), with C_A and C_B modules respectively, the normalised mutual information is:

$$NMI(A;B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B}N_{ij}\log\left(\frac{N_{ij}n}{N_i N_j}\right)}{\sum_{i=1}^{C_A}N_i\log\left(\frac{N_i}{n}\right) + \sum_{j=1}^{C_B}N_j\log\left(\frac{N_j}{n}\right)} \tag{13}$$

where N is the confusion matrix with elements N_{ij} which indicate the number of nodes that appear in the ith module of partition A and the jth module of partition B; N_i is the number of nodes in module i of partition A and N_j is the number of nodes in module j of partition B. If $NMI(A;B) = 0$ there is no shared information between partitions A and B - they each have identified very different community structures; whilst if $NMI(A;B) = 1$ the information given by partitions A and B is identical - the same community structure has been found by A and B.

Realised Modularity

Realised modularity has been suggested as a posterior measure of modularity(Poisot, 2013) to classify the proportion of links in a network that are within rather than between modules. Here we extend this measure so it can be applied to weighted as well as binary networks. If M is the sum of all edge weights in a network and H is the sum of all within-module edge weights realised weighted modularity is expressed as:

$$Q_R^{'} = 2\left(\frac{H}{M}\right) - 1 \tag{14}$$

$Q_R^{'}$ takes values between -1, indicating that no edges exist between nodes in the same module, and 1 in which case all edges in the network are contained within-modules and no interactions exist between different modules. If $Q_R^{'} = 0$ half of the edge weights in the network are found connecting nodes within the same module and the remaining edge weights are node connections between different modules. Note that in a weighted network $Q_R^{'}$ says nothing about the actual number of edges between/within modules, only the strength of the connecting edges.

Data

We used the 23 plant-pollinator networks available in the bipartite R package (22 of which were used in (Dormann and Strauss, 2014) and the junker2013 network) taken from the NCEAS dataset (https://www.nceas.ucsb.edu/interactionweb/resources.html). These networks show the number of observed visitations by each recorded pollinator species to each recorded plant species at different field sites across the world. Network properties are shown in table 3.

Computing Modularity

We computed the binary and quantitative networks for each of the datasets, removing rows and columns that contained no interaction data from the analysis. QuanBiMo, LPAwb+ and Exhaustive LPAwb+ were run 100 times for each binary and each weighted network in order to assess the modular structures found and the fidelity of the algorithms. We then quantified the differences between the modular structures found by the binary and weighted algorithms using the normalised mutual information criterion.

Code implementations for the LPAwb+ algorithm are currently available online for the Julia, MATLAB/Octave and R programming languages. This and the R code used to create the figures and perform the analysis presented in this paper is available online (https://github.com/sjbeckett/weighted-modularity-LPAwbPLUS). For fair comparison all computations were performed in R version 3.1.1 using version 2.04 of the bipartite package on an Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz desktop computer.

Results

Figure 2 shows the maximum modularity scores detected by each algorithm (from 100 replicates) for each of the networks. Full details are shown in table 1 for binary networks and table 2 for weighted networks. As expected by definition Exhaustive LPAwb+ scores were always equal or greater than those detected by LPAwb+. Each algorithm detected similar maximum modularity scores for each network, with the exception of the datasets of kato1990, junker2013, barrett1987 and elberling 1999 in binary networks (figure 2*a*) and kato1990, junker2013, elberling1999, kevan1970 and barrett1987 for weighted networks (figure 2*b*) in which LPAwb+ and Exhaustive LPAwb+ detected much better than QuanBiMo.

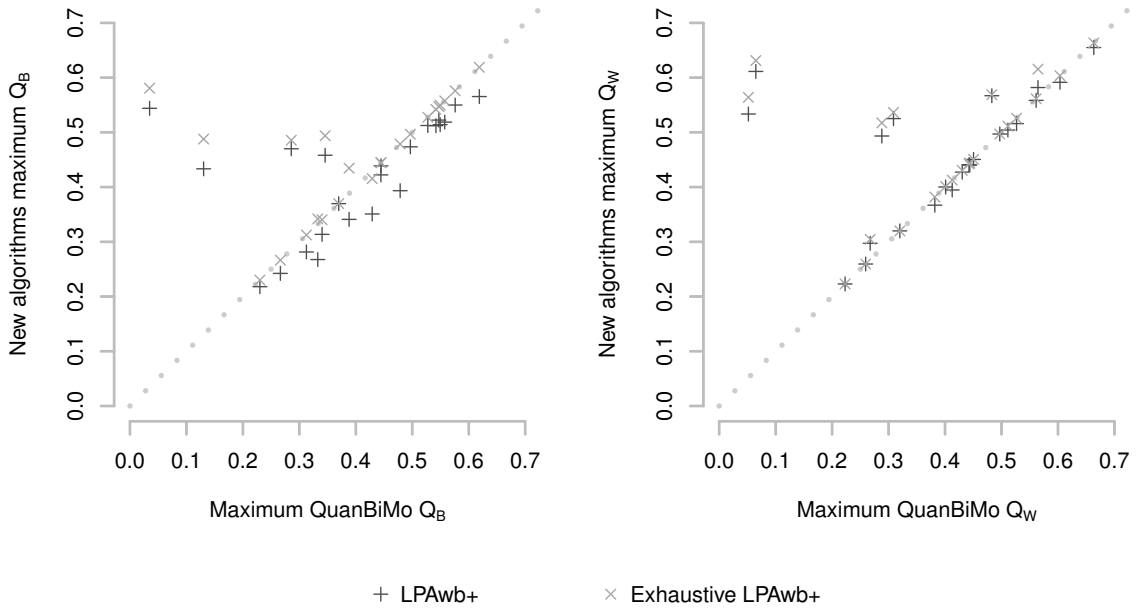

Figure 2: Comparing the maximum detected modularity scores by each algorithm (from 100 repetitions on each of the 23 plant-pollinator networks). The dotted line indicates unity i.e. QuanBiMo and the other algorithms are in perfect correspondence. Points below the dotted line indicate QuanBiMo maximises modularity more effectively; whilst points above the dotted line show that LPAwb+ (+) or Exhaustive LPAwb+ (×) detected partitions with greater modularity than QuanBiMo. (*a*) shows a comparison of Q_B, binary modularity scores, whilst (*b*) shows Q_W, weighted modularity scores.

Table 1 shows the greatest modularity scores detected by each algorithm, the number of modules in this particular partition and the average execution time for each algorithm in the analysis of binary networks. The same partition was found by all three algorithms in only the schemske1978 network; both QuanBiMo and Exhaustive LPAwb+ found the same partitions for another 15 networks; whilst Exhaustive LPAwb+ found the greatest modularity score for 6 networks and QuanBiMo found the best modularity score in the inouye1988 network. LPAwb+ was by far the algorithm with the quickest execution time. Exhaustive LPAwb+ performs faster on small networks than QuanBiMo and more slowly on larger networks, however it generally found a much greater modularity score than QuanBiMo for these networks. The partitions found by LPAwb+ had more modules than those found by the solution with the greatest modularity.

For weighted networks table 2 shows there were 5 networks for which the same maximum modularity was detected by all 3 algorithms, 10 networks in which QuanBiMo and Exhaustive LPAwb+ found the greatest modularity, 7 networks for which Exhaustive LPAwb+ found the greatest modularity and a single network, small1976, that was maximised by QuanBiMo. QuanBiMo had a similar average performance time to the binary networks, with LPAwb+ finding modularity more quickly in weighted than in binary networks. Exhaustive LPAwb+ has a similar performance time for smaller networks as under binary conditions and performs faster for the larger networks - which can be ascribed to the lower number of modules detected by LPAwb+ for the weighted networks. LPAwb+ detects partitions which generally have more modules than that with the greatest modularity, while QuanBiMo generally finds partitions with fewer modules than the solution found with greatest modularity.

Figure 3 shows the median detected modularity scores for each algorithm against the overall maximum modularity score for each network. Figure 3*a* shows that Exhaustive LPAwb+ consistently finds modularity scores closest to the maximal value, that LPAwb+ scores were close, but not so close and that whilst QuanBiMo could achieve consistency as good as the Exhaustive LPAwb+, for several networks QuanBiMo had a median value much lower than the maximum modularity detected. Similarly in figure 3*b* Exhaustive LPAwb+ shows high consistency as does LPAwb+ (more so than for binary

networks), whilst QuanBiMo in general performs less consistently for weighted networks than binary networks.

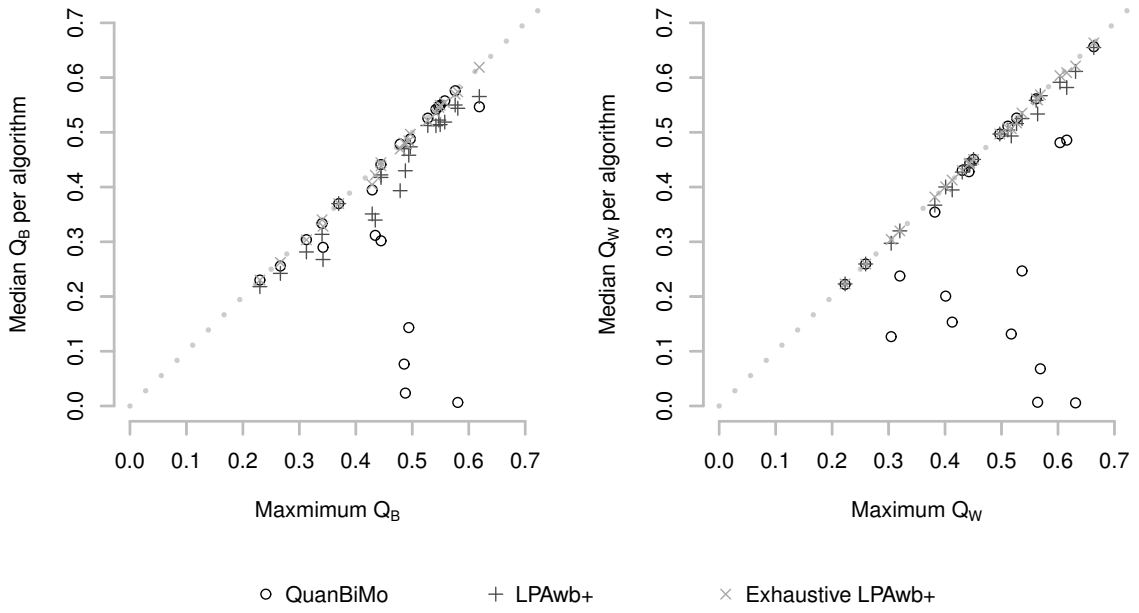

Figure 3: Comparison of median modularity scores found by each algorithm (from 100 repetitions on each of the 23 plant-pollinator networks) to the maximum of the modularity scores found across the algorithms. (a) shows results for binary networks, whilst (b) shows the results for weighted networks. The dotted line represents unity, where median modularity score is equal to the maximum modularity score that was detected.

Network	QuanBiMo Q_B	M	t	LPAwb+ Q_B	M	t	Exhaustive LPAwb+ Q_B	M	t
Safariland	**0.558**	6	1.067	0.519	9	0.014	**0.558**	6	0.641
barrett1987	0.286	4	11.811	0.470	11	0.070	**0.486**	8	3.667
bezerra2009	**0.230**	3	1.106	0.218	5	0.008	**0.230**	3	0.734
elberling1999	0.346	6	24.470	0.458	22	0.382	**0.494**	8	23.353
inouye1988	**0.429**	9	18.532	0.351	31	0.710	0.415	11	74.624
junker2013	0.130	5	46.076	0.433	55	2.762	**0.488**	19	405.596
kato1990	0.035	5	1551.827	0.544	74	14.196	**0.581**	20	3441.840
kevan1970	0.388	6	29.303	0.341	23	0.279	**0.434**	5	43.059
memmott1999	0.333	5	10.598	0.268	19	0.151	**0.342**	5	19.302
mosquin1967	**0.479**	6	0.916	0.393	11	0.014	**0.479**	6	0.819
motten1982	**0.313**	6	2.763	0.281	10	0.032	**0.313**	6	2.512
olesen2002aigrettes	**0.340**	4	1.149	0.314	7	0.011	**0.340**	4	1.254
olesen2002flores	**0.444**	4	0.949	0.422	7	0.008	**0.444**	4	0.533
ollerton2003	**0.445**	6	5.334	0.439	8	0.026	**0.445**	6	1.179
schemske1978	**0.370**	6	1.869	**0.370**	6	0.009	**0.370**	6	0.359
small1976	**0.266**	5	1.803	0.242	8	0.021	**0.266**	5	2.103
vazarr	**0.542**	7	1.431	0.512	9	0.016	**0.542**	7	0.865
vazcer	**0.619**	6	2.000	0.565	9	0.015	**0.619**	6	0.744
vazllao	**0.576**	6	1.129	0.550	8	0.016	**0.576**	6	0.915
vazmasc	**0.547**	6	1.340	0.522	8	0.011	**0.547**	6	0.486
vazmasnc	**0.527**	6	1.969	0.512	8	0.014	**0.527**	6	0.533
vazquec	**0.497**	4	1.529	0.474	7	0.013	**0.497**	4	0.479
vazquenc	**0.549**	5	0.834	0.514	7	0.009	**0.549**	5	0.300

Table 1: Comparison of QuanBiMo, LPAwb+ and Exhaustive LPAwb+ algorithms on binary ecological interaction networks. Q_B is the greatest value of binary modularity from 100 replicates on the network, M is the corresponding number of modules found in this partition and t is the mean time taken to compute each algorithm once. Numbers have been rounded to 3 d.p. Numbers shown in bold are those with the highest Q_B score.

Network	QuanBiMo Q_W	M	t	LPAwb+ Q_W	M	t	Exhaustive LPAwb+ Q_W	M	t
Safariland	**0.430**	5	1.258	0.427	7	0.014	**0.430**	5	0.721
barrett1987	0.483	5	10.107	0.567	9	0.057	**0.569**	7	3.577
bezerra2009	**0.223**	5	1.178	**0.223**	5	0.008	**0.223**	5	0.645
elberling1999	0.288	6	25.416	0.493	18	0.190	**0.517**	10	21.288
inouye1988	0.565	11	25.368	0.582	22	0.413	**0.615**	9	54.377
junker2013	0.052	5	83.548	0.533	33	1.133	**0.564**	17	287.774
kato1990	0.065	5	2355.046	0.611	48	6.000	**0.631**	23	2425.382
kevan1970	0.309	2	35.164	0.525	10	0.096	**0.536**	5	26.133
memmott1999	0.267	4	12.333	0.297	10	0.065	**0.305**	7	11.660
mosquin1967	**0.444**	6	0.970	0.440	7	0.009	**0.444**	6	0.669
motten1982	**0.382**	4	3.292	0.367	6	0.020	**0.382**	4	1.902
olesen2002aigrettes	**0.259**	5	1.181	**0.259**	5	0.008	**0.259**	5	0.905
olesen2002flores	**0.497**	5	0.989	**0.497**	5	0.006	**0.497**	5	0.415
ollerton2003	**0.413**	6	6.023	0.395	7	0.024	**0.413**	6	1.243
schemske1978	**0.320**	4	1.792	**0.320**	4	0.009	**0.320**	4	0.392
small1976	**0.527**	8	1.984	0.516	11	0.026	0.526	9	1.909
vazarr	**0.442**	6	1.733	0.441	7	0.014	**0.442**	6	0.883
vazcer	**0.604**	6	2.317	0.591	7	0.015	**0.604**	6	0.725
vazllao	**0.561**	6	1.386	0.558	8	0.013	**0.561**	6	0.839
vazmasc	**0.663**	6	1.436	0.655	7	0.010	**0.663**	6	0.456
vazmasnc	**0.401**	6	2.291	0.400	7	0.012	**0.401**	6	0.565
vazquec	**0.511**	6	1.835	0.504	7	0.013	**0.511**	6	0.474
vazquenc	**0.450**	4	0.815	**0.450**	4	0.007	**0.450**	4	0.265

Table 2: Comparison of QuanBiMo, LPAwb+ and Exhaustive LPAwb+ algorithms on weighted ecological interaction networks. Q_W is the greatest value of weighted modularity from 100 replicates on the network, M is the corresponding number of modules found in this partition and t is the mean time taken to compute each algorithm once. Numbers have been rounded to 3 d.p. Numbers shown in bold are those with the highest Q_W score.

Contrasting Binary and Quantitative Modular Structure

Different modular network representations were found for binary and weighted networks. Figure 4*a* shows the partition with the greatest bianry modularity for the olesen2002flores network, whilst figure 4*b* shows the partition with the greatest weighted modularity. They show that the same dataset is qualitatively and quantitatively differently (the shared normalised mutual information for these two partitions is $I = 0.619$) structured using weighted and binary representations.

Figure 5*a* shows the differences in modularity and normalised mutual information between the binary and weighted network representations. 11 networks have greater weighted modularity; and 12 networks have greater binary modularity. In general it appears that networks that have greater binary modularity than weighted modularity also have more similar partitions; though it would be expected that normalised mutual information would equal 1 when binary and weighted modularity is equal - which would occur when every edge in the network has the same interaction strength.

Not only were the detected modularity scores different between the binary and weighted networks - but the modular configurations of these networks also differed. Only 8 of the networks had the same number of modules under binary and weighted conditions; whilst 8 had more modules in the weighted networks and 7 had more modules in the binary network representation.

There appears to be a positive relationship between realised modularity and modularity (figure 5*b*), however there does not appear to be a relationship between the binary and quantitative versions of these measures.

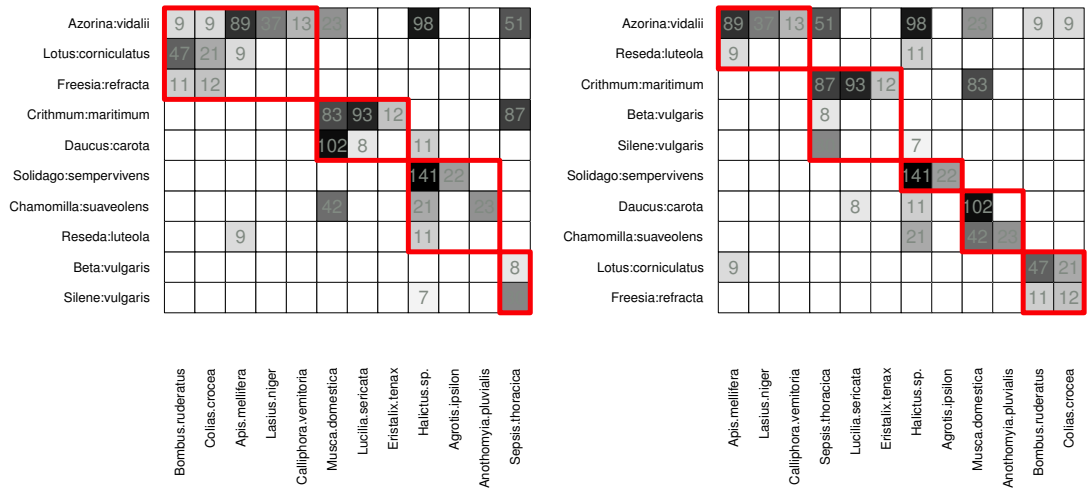

Figure 4: A visual comparison of the modular structures identified for the olesen2002flores dataset as a (a) binary ($Q_B = 0.444$, 4 modules, $Q^B_{norm} = 0.625$) and (b) quantitative ($Q_w = 0.497$, 5 modules, $Q^W_{norm} = 0.625$) network. The normalised mutual information shared between these two modular compositions is $NMI = 0.619$ indicating a qualitative difference in the revealed modular structure.

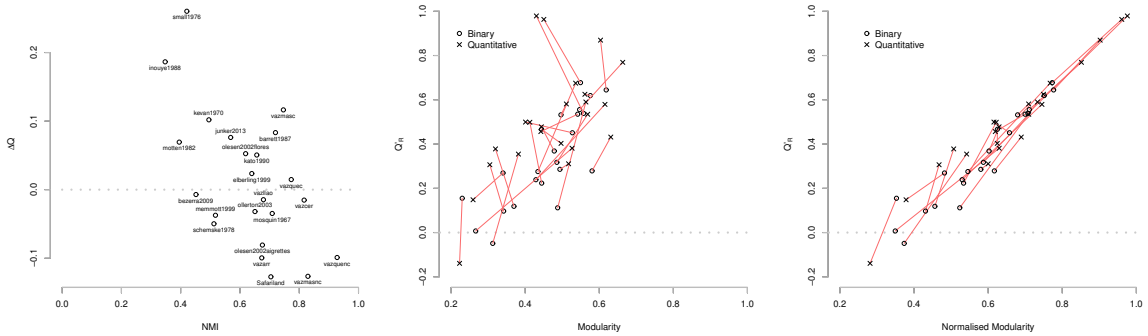

Figure 5: (a) The change in greatest modularity scores found between the weighted and binary networks ($\Delta Q = Q_W - Q_B$) against NMI scores found by comparing the corresponding weighted and binary partitions for each network. (b) The greatest modularity scores (Q_B and Q_W) for each network and their corresponding realised modularity scores (Q'_R). (c) The normalised modulariy scores of each of the greatest modularity scores plotted against their corresponding realised modularity scores. Red lines join together the binary and quantitative scores of the same network.

Discussion

Weighted modularity challenges

Maximising bipartite modularity is an NP-hard problem (Miyauchi and Sukegawa, 2014) and it may be difficult to find an algorithm which performs well on this problem for any given network. There are four challenges to address when attempting to maximise modularity (Good et al., 2010) which are also relevant to weighted modularity. Any modularity maximisation algorithm only uses information within the incidence matrix and is thus agnostic to heirarchies within the dataset - the algorithm will find communities at the resolution that has the greatest modularity it can compute; which may be different to the resolution which correpsonds best with any additional information we know about the network. This is further complicated as several heirarchical levels may exist within an individual network. Some

work has started to address this problem in terms of visualising the network as a multiscale structure (Flores et al., 2012, 2014; Dormann and Strauss, 2014), but this requires finding a suitable starting resolution. As we found with QuanBiMo, the ability of algorithms to maximise modularity is highly dependent on network properties such as size. Finally it is recognised that the modularity landscape is "glassy" - there are many local modularity maxima; but detecting the global peak is extremely difficult and finding an algorithm that can capably traverse this "glassy" landscape is a challenge.

We focussed on specific definitions of modularity in this paper - but note that others do exist (Murata, 2010; Thébault, 2013). (Thébault, 2013) compared two binary bipartite modularity based measures that have been applied in ecology and concluded that different forms of modularity may be useful in different contexts; but that the form of modularity used here (Barber, 2007) corresponded well with that for unipartite networks (Newman and Girvan, 2004) - and is well suited for identifying densely connected modules. (Murata, 2010) proposes a different modularity measure that allows overlapping community structures to exist. Other modularity definitions and their weighted extensions are worth further investigation.

A further challenge will be to find appropriate null models to test weighted modularity against in order to standardise the effect size of modularity in different networks (Dormann and Strauss, 2014). (Staniczenko et al., 2013) introduce a null model in which the interaction structure between nodes and the edge weights are kept fixed, but that edge weights are randomly assigned across the nodes. However, in principle it would be good to test against a null ensemble in which both structure and connections are variable.

Algorithmic comments

In this paper we introduced two new algorithms LPAwb+ and Exhaustive LPAwb+, which we compared to the existing QuanBiMo algorithm. We found all three algorithms were able to detect greater modularity than previously reported (figure 6 (Dormann and Strauss, 2014)) and that they were generally able to perform well on binary and quantitative networks (a binary network is a special case of a quantitative network).

QuanBiMo found fewer modules and LPAwb+ found too many modules when compared to Exhaustive LPAwb+, which can be regarded as the goldilocks case of the three algorithms examined here.

The input values for the minimum number of modules and the number of repetitions to use in Exhaustive LPAwb+ were set to default values for comparison with QuanBiMo's default values here. However - both algorithms settings should be tuned for the input network. Unlike QuanBiMo where it is not known how input values for tolerance and number of steps to perform will effect modularity, the time it takes to find it or if degenerate solutions will be returned; suitable input values for Exhaustive LPAwb+ can be chosen based on the performance and outcome of a single LPAwb+ run; and any prior knowledge about the network in terms of how many modules it is likely to have.

We performed experiments for all algorithms in R to make a fair comparison in terms of timings - however as QuanBiMo calls an executable written natively in C++, whilst the LPAwb+ algorithms are written natively in R - it may not be as fair as first appears - and it may be possible to improve the speed of the LPAwb+ algorithms. For especially large networks employing parallelised computation of node label updates in LPAwb+ may improve computation (Liu and Murata, 2010).

It can be argued that small disagreements in modularity scores are not worth quibbling over. However, we found major deficiencies with the QuanBiMo algorithm in its current form. There is no diagnostic available for QuanBiMo to suggest whether detected modularity is high; it is unclear for a given network what input parameters should be chosen to find a high modularity value; the algorithm can produce highly variable evaluations of modularity and degenerate solutions are possible. Some of these problems can be mitigated by running the QuanBiMo algorithm multiple times and using the partition with the optimal value of modularity as done in this study, or by increasing the number of MCMC steps performed. However, both options are computationally expensive and may not guarantee the best solution.

We found that QuanBiMo performed particularly poorly on the larger networks of junker2013 and kato1990 in comparison to the approaches introduced here. This indicates that the default settings of QuanBiMo are not appropriate for these networks. (Dormann and Strauss, 2014) address this issue by saying that using larger step sizes will allow a more thorough search of the network which may lead to better partitions. However increasing the number of steps will (a) increase the time needed to perform

QuanBiMo and (b) it is unclear how many steps are needed in general to find 'good' modularity scores for a particular network - other than by running QuanBiMo with multiple step numbers. We also note that the QuanBiMo algorithm sometimes fell into solutions that were so suboptimal their modularity score was less than the tolerance level of the QuanBiMo algorithm - which led it to fail (no results returned to the user).

The stochasticity present in all three of the modularity maximising algorithms discussed here necessitates that they should by run multiple times. This is an underdiscussed issue and potential source of bias - especially when it comes to significance testing - in which the chosen modularity algorithm should be applied multiple times to each individual null network in the null ensemble. This source of bias may be particularly problematic with regards to the QuanBiMo algorithm which has a much lower median modularity score than the LPAwb+ algorithms for several of the networks (see figure 3, table 4 and table 5).

Plant-pollinator networks and quantitative information

Binary and weighted modularity appear positively correlated to complementary specialisation(Dormann and Strauss, 2014) despite the different modularity values reported here (results not shown); which makes intuitive sense as those that well modulate must be specialised amongst there fellow modulees. There is no guarantee of finding biologically relevant information from the modularity partitions found. Just as different modularity measures have different contextual meanings - so does the usage of binary or weighted measurements. Both measurements contain different information - though we expect that the weighted measurements will in general contain more relevance for the analysis of real world networks as the strength of existing interactions is undoubtably an important feature of a network is structured.

Conclusions

Real world networks are not formed of binary interactions. We encourage researchers to apply weighted modularity measures to their datasets and evaluate the community partitions that are identified.

LPAwb+ is an algorithm that would be well suited for exploratory analysis and useage on large networks - as it is fast and whilst it did not return the best modularity values of the methods tested here, the solutions it did find were consistently high. Care has to be taken with both QuanBiMo and Exhaustive LPAwb+ in setting appropriate input parameter settings such that the analysis is not computationally infeasible. We would recommend using Exhaustive LPAwb+ over QuanBiMo; as Exhaustive LPAwb+ has more meaningful input parameters, can perform no worse than LPAwb+ and it's performance was less variable than QuanBiMo on the networks tested in this study.

We have made the code for our algorithms and analysis available online (https://github.com/sjbeckett/weighted-modularity-LPAwbPLUS) to allow researchers to replicate our findings and encourage those with access to potentially interesting weighted bipartite datasets to analyse them using these methods.

Acknowledgements

I thank Xin Liu for the hints and nudges that helped me setup the LPAb+ algorithm - eventually leading to the weighted counterpart presented here. I thank Timothée Poisot and Hywel Williams for comments that improved the manuscript.

References

Newman, M. *Networks: an introduction*; Oxford University Press, 2010.

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

Newman, M. E. (2004) Analysis of weighted networks. *Physical Review E 70*, 056131.

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

Liu, X., and Murata, T. (2010) An Efficient Algorithm for Optimizing Bipartite Modularity in Bipartite Networks. *JACIII 14*, 408–415.

Costa, A., and Hansen, P. (2014) A locally optimal hierarchical divisive heuristic for bipartite modularity maximization. *Optimization Letters 8*, 903–917.

Newman, M. E. (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences 103*, 8577–8582.

Barber, M. J. (2007) Modularity and community detection in bipartite networks. *Physical Review E 76*, 066102.

Flores, C. O., Poisot, T., and Weitz, J. S. (2014) BiMAT: a MATLAB (R) package to facilitate the analysis and visualization of bipartite networks. *arXiv preprint arXiv:1406.6732*

Dormann, C., B., G., and Fruend, J. (2008) Introducing the bipartite Package: Analysing Ecological Networks. *R news*

Thébault, E. (2013) Identifying compartments in presence–absence matrices and bipartite networks: insights into modularity measures. *Journal of Biogeography 40*, 759–768.

Ana, L., and Jain, A. Robust data clustering. Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. 2003; pp II–128–II–133 vol.2.

Danon, L., Diaz-Guilera, A., Duch, J., and Arenas, A. (2005) Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment 2005*, P09008.

Poisot, T. (2013) An a posteriori measure of network modularity. *F1000Research 2*.

Miyauchi, A., and Sukegawa, N. (2014) Maximizing Barber's bipartite modularity is also hard. *Optimization Letters* 1–17.

Good, B. H., de Montjoye, Y.-A., and Clauset, A. (2010) Performance of modularity maximization in practical contexts. *Physical Review E 81*, 046106.

Flores, C. O., Valverde, S., and Weitz, J. S. (2012) Multi-scale structure and geographic drivers of cross-infection within marine bacteria and phages. *The ISME journal 7*, 520–532.

Murata, T. In *Data Mining for Social Network Data*; Memon, N., Xu, J. J., Hicks, D. L., and Chen, H., Eds.; Annals of Information Systems; Springer US, 2010; Vol. 12; pp 109–123.

Newman, M. E., and Girvan, M. (2004) Finding and evaluating community structure in networks. *Physical review E 69*, 026113.

Staniczenko, P. P., Kopp, J. C., and Allesina, S. (2013) The ghost of nestedness in ecological networks. *Nature communications 4*, 1391.

Vázquez, D. P., and Simberloff, D. (2002) Ecological specialization and susceptibility to disturbance: conjectures and refutations. *The American Naturalist 159*, 606–623.

Vázquez, D. P. (2002) Interactions among introduced ungulates, plants, and pollinators: a field study in the temperate forest of the southern Andes.

Vázquez, D. P., and Simberloff, D. (2003) Changes in interaction biodiversity induced by an introduced ungulate. *Ecology Letters 6*, 1077–1083.

Barrett, S. C., and Helenurm, K. (1987) The reproductive biology of boreal forest herbs. I. Breeding systems and pollination. *Canadian Journal of Botany 65*, 2036–2046.

Bezerra, E. L., Machado, I. C., and Mello, M. A. (2009) Pollination networks of oil-flowers: a tiny world within the smallest of all worlds. *Journal of Animal Ecology 78*, 1096–1101.

Elberling, H., and Olesen, J. M. (1999) The structure of a high latitude plant-flower visitor system: the dominance of flies. *Ecography 22*, 314–323.

Inouye, D. W., and Pyke, G. H. (1988) Pollination biology in the Snowy Mountains of Australia: comparisons with montane Colorado, USA. *Australian Journal of Ecology 13*, 191–205.

Junker, R. R., Blüthgen, N., Brehm, T., Binkenstein, J., Paulus, J., Martin Schaefer, H., and Stang, M. (2013) Specialization on traits as basis for the niche-breadth of flower visitors and as structuring mechanism of ecological networks. *Functional Ecology 27*, 329–341.

others„ et al. (1990) Insect-flower relationship in the primary beech forest of Ashu, Kyoto: an overview of the flowering phenology and the seasonal pattern of insect visits.

Kevan, P. (1970) High arctic insect-flower visitor relations: the inter-relationships of arthropods and flowers at Lake Hazen, Ellesmere Island, Northwest Territories, Canada. *PhD thesis*

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

Mosquin, T., and Martin, J. (1967) Observations on the pollination biology of plants on Melville Island, NWT, Canada. *Canadian Field Naturalist 81*, 201–205.

Motten, A. F. (1986) Pollination ecology of the spring wildflower community of a temperate deciduous forest. *Ecological Monographs* 21–42.

Olesen, J. M., Eskildsen, L. I., and Venkatasamy, S. (2002) Invasion of pollination networks on oceanic islands: importance of invader complexes and endemic super generalists. *Diversity and Distributions 8*, 181–192.

Ollerton, J., Johnson, S. D., Cranmer, L., and Kellie, S. (2003) The pollination ecology of an assemblage of grassland asclepiads in South Africa. *Annals of Botany 92*, 807–834.

Schemske, D. W., Willson, M. F., Melampy, M. N., Miller, L. J., Verner, L., Schemske, K. M., and Best, L. B. (1978) Flowering ecology of some spring woodland herbs. *Ecology* 351–366.

Small, E. (1976) Insect pollinators of the Mer Bleue peat bog of Ottawa. *Canadian field-naturalist*

Appendix

Network	Rows	Columns	Edges	Fill	Reference
Safariland	9	27	39	1130	(Vázquez and Simberloff, 2002; Vázquez, 2002; Vázquez and Simberloff, 2003)
barrett1987	12	102	167	550	(Barrett and Helenurm, 1987)
bezerra2009	13	13	71	28224	(Bezerra et al., 2009)
elberling1999	23	118	238	383	(Elberling and Olesen, 1999)
inouye1988	41	83	268	1459	(Inouye and Pyke, 1988)
junker2013	56	257	572	3053	(Junker et al., 2013)
kato1990	91	679	1206	2392	(Kato et al., 1990)
kevan1970	30	114	312	2523	(Kevan, 1970)
memmott1999	25	79	299	2183	(Memmott, 1999)
mosquin1967	11	18	38	134	(Mosquin and Martin, 1967)
motten1982	13	44	143	2225	(Motten, 1986)
olesen2002aigrettes	14	13	52	1512	(Olesen et al., 2002)
olesen2002flores	10	12	30	1139	(Olesen et al., 2002)
ollerton2003	9	56	103	594	(Ollerton et al., 2003)
schemske1978	7	32	59	299	(Schemske et al., 1978)
small1976	13	34	141	992	(Small, 1976)
vazarr	10	29	43	515	(Vázquez and Simberloff, 2002; Vázquez, 2002; Vázquez and Simberloff, 2003)
vazcer	9	33	45	613	(Vázquez and Simberloff, 2002; Vázquez, 2002; Vázquez and Simberloff, 2003)
vazllao	10	29	42	677	(Vázquez and Simberloff, 2002; Vázquez, 2002; Vázquez and Simberloff, 2003)
vazmasc	8	26	36	286	(Vázquez and Simberloff, 2002; Vázquez, 2002; Vázquez and Simberloff, 2003)
vazmasnc	8	35	51	719	(Vázquez and Simberloff, 2002; Vázquez, 2002; Vázquez and Simberloff, 2003)
vazquec	8	27	47	592	(Vázquez and Simberloff, 2002; Vázquez, 2002; Vázquez and Simberloff, 2003)
vazquenc	7	24	31	761	(Vázquez and Simberloff, 2002; Vázquez, 2002; Vázquez and Simberloff, 2003)

Table 3: Network properties of the datasets used in this study.

	QuanBiMo					LPAwb+					Exhaustive LPAwb+				
Network	R	$\tilde{x}$	U	F	Q'_R	R	$\tilde{x}$	U	F	Q'_R	R	$\tilde{x}$	U	F	Q'_R
Safariland	89	0.558	1	0	0.538	100	0.519	2	0	0.519	40	0.554	1	0	0.538
barrett1987	1	0.077	1	0	0.593	98	0.470	1	0	0.470	1	0.481	1	0	0.317
bezerra2009	73	0.230	1	0	0.155	100	0.218	1	0	0.218	52	0.230	1	0	0.155
elberling1999	1	0.143	1	5	0.311	100	0.458	2	0	0.458	1	0.484	1	0	0.286
inouye1988	1	0.395	1	0	0.239	51	0.351	1	0	0.351	1	0.404	1	0	0.082
junker2013	1	0.024	1	0	0.619	44	0.430	27	0	0.433	1	0.479	1	0	0.112
kato1990	1	0.006	1	0	0.945	92	0.544	85	0	0.544	1	0.574	1	0	0.279
kevan1970	1	0.312	1	1	0.276	6	0.340	4	0	0.341	1	0.422	1	0	0.276
memmott1999	1	0.290	1	0	0.124	57	0.268	8	0	0.268	1	0.328	1	0	0.097
mosquin1967	64	0.479	1	0	0.368	100	0.393	1	0	0.393	25	0.470	1	0	0.368
motten1982	6	0.304	1	0	-0.049	100	0.281	1	0	0.281	8	0.304	1	0	-0.049
olesen2002aigrettes	19	0.334	1	0	0.269	98	0.314	1	0	0.314	80	0.340	1	0	0.269
olesen2002flores	24	0.441	1	0	0.467	98	0.422	2	0	0.422	61	0.444	1	0	0.467
ollerton2003	1	0.302	1	3	0.223	43	0.418	1	0	0.439	9	0.439	1	0	0.223
schemske1978	53	0.370	1	0	0.119	100	0.370	1	0	0.370	100	0.370	1	0	0.119
small1976	9	0.256	1	0	0.007	100	0.242	1	0	0.242	13	0.262	1	0	0.007
vazarr	100	0.542	1	0	0.535	100	0.512	1	0	0.512	17	0.535	1	0	0.535
vazcer	28	0.547	1	0	0.644	100	0.565	1	0	0.565	73	0.619	1	0	0.644
vazllao	100	0.576	1	0	0.619	82	0.550	2	0	0.550	39	0.570	1	0	0.619
vazmasc	100	0.547	2	0	0.556	100	0.522	1	0	0.522	48	0.546	2	0	0.556
vazmasnc	14	0.526	1	0	0.451	100	0.512	2	0	0.512	8	0.521	1	0	0.451
vazquec	26	0.488	1	0	0.532	100	0.474	1	0	0.474	73	0.497	1	0	0.532
vazquenc	100	0.549	1	0	0.677	100	0.514	1	0	0.514	74	0.549	1	0	0.677

Table 4: Extra results from the evaluations of the binary version of these networks. R is the number of times that the best partitions (with highest Q_B) were found from the 100 tests, $\tilde{x}$ is the median Q_B score, U is the number of unique configurations found with the maxmium Q_B score (for each method), F is number of times that the algorithms reported a failure (from the 100 runs) and Q'_R is the realised modularity of the partition with highest Q_B score (for each method). Numbers have been rounded to 3 d.p.

	QuanBiMo					LPAwb+					Exhaustive LPAwb+				
Network	R	$\tilde{x}$	U	F	Q'_R	R	$\tilde{x}$	U	F	Q'_R	R	$\tilde{x}$	U	F	Q'_R
Safariland	91	0.430	1	0	0.979	100	0.427	1	0	0.963	42	0.430	1	0	0.979
barrett1987	1	0.068	1	0	0.836	100	0.567	1	0	0.560	11	0.568	1	0	0.535
bezerra2009	21	0.222	1	0	-0.139	100	0.223	1	0	-0.139	100	0.223	1	0	-0.139
elberling1999	1	0.131	1	3	0.530	100	0.493	4	0	0.180	1	0.507	1	0	0.311
inouye1988	1	0.486	1	0	0.565	100	0.582	1	0	0.406	1	0.609	1	0	0.579
junker2013	1	0.007	1	0	0.743	100	0.533	1	0	0.452	1	0.559	1	0	0.590
kato1990	1	0.006	1	0	0.903	100	0.611	1	0	0.355	1	0.621	1	0	0.431
kevan1970	1	0.247	1	0	0.739	100	0.525	1	0	0.583	7	0.535	1	0	0.675
memmott1999	1	0.127	1	0	0.532	100	0.297	1	0	0.132	2	0.304	1	0	0.306
mosquin1967	78	0.444	1	0	0.478	100	0.440	1	0	0.403	89	0.444	1	0	0.478
motten1982	16	0.354	1	0	0.355	100	0.367	1	0	0.212	100	0.382	1	0	0.355
olesen2002aigrettes	96	0.259	1	0	0.148	100	0.259	1	0	0.148	100	0.259	1	0	0.148
olesen2002flores	67	0.497	1	0	0.403	100	0.497	1	0	0.403	100	0.497	1	0	0.403
ollerton2003	1	0.153	1	2	0.498	100	0.395	1	0	0.431	98	0.413	1	0	0.498
schemske1978	5	0.238	1	0	0.378	100	0.320	1	0	0.378	100	0.320	1	0	0.378
small1976	33	0.526	1	0	0.381	100	0.516	1	0	0.260	1	0.517	1	0	0.337
vazarr	21	0.428	1	0	0.456	100	0.441	1	0	0.449	93	0.442	1	0	0.456
vazcer	30	0.481	1	0	0.869	100	0.591	1	0	0.830	80	0.604	1	0	0.869
vazllao	100	0.561	1	0	0.625	100	0.558	1	0	0.586	61	0.561	1	0	0.635
vazmasc	31	0.656	1	0	0.769	100	0.655	1	0	0.727	80	0.663	1	0	0.769
vazmasnc	26	0.201	1	0	0.499	100	0.400	1	0	0.497	31	0.401	1	0	0.499
vazquec	56	0.511	1	0	0.581	100	0.504	1	0	0.544	22	0.508	1	0	0.581
vazquenc	100	0.450	1	0	0.963	100	0.450	1	0	0.963	100	0.450	1	0	0.963

Table 5: Extra results from the evaluations of the weighted version of these networks. R is the number of times that the best partitions (with highest Q_W) were found from the 100 tests, $\tilde{x}$ is the median Q_W score, U is the number of unique configurations found with the maxmium Q_W score (for each method), F is number of times that the algorithms reported a failure (from the 100 runs) and Q'_R is the realised modularity of the partition with highest Q_W score (for each method). Numbers have been rounded to 3 d.p.

Network	Q^B_{norm}	Q^W_{norm}	NMI
Safariland	0.707	0.976	0.704
barrett1987	0.587	0.710	0.720
bezerra2009	0.353	0.281	0.452
elberling1999	0.580	0.600	0.640
inouye1988	0.530	0.745	0.348
junker2013	0.523	0.733	0.569
kato1990	0.617	0.689	0.657
kevan1970	0.545	0.767	0.495
memmott1999	0.431	0.468	0.517
mosquin1967	0.602	0.630	0.709
motten1982	0.373	0.542	0.395
olesen2002aigrettes	0.482	0.379	0.676
olesen2002flores	0.625	0.625	0.619
ollerton2003	0.534	0.622	0.650
schemske1978	0.456	0.507	0.512
small1976	0.349	0.630	0.421
vazarr	0.700	0.619	0.674
vazcer	0.777	0.902	0.816
vazllao	0.751	0.750	0.679
vazmasc	0.711	0.852	0.746
vazmasnc	0.658	0.616	0.829
vazquec	0.680	0.709	0.773
vazquenc	0.773	0.961	0.928

Table 6: Normalised modularity and normalised mutual information scores for the partitions with greatest modularity. Numbers have been rounded to 3 d.p.