Improved community detection in weighted bipartite networks

Stephen Beckett

December 21, 2014

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

Key words: modular structure, network ecology, two-mode networks, compartments

Abstract

Real-world complex networks are composed of non-random quantitative interactions, yet many community detection algorithms only use the presence or absence of interactions between nodes. Weighted modularity is a potential method for evaluating the quality of communities in quantitative networks. Modularity optimisation is a method for finding communities in a network. QuanBiMo has been proposed to maximise weighted modularity in bipartite networks. This paper introduces two new algorithms, LPAwb+ and Exhaustive LPAwb+, for maximising weighted modularity in bipartite networks. These algorithms robustly identify partitions with high modularity scores. Exhaustive LPAwb+ consistently matched or outperformed QuanBiMo, whilst the speed of LPAwb+ makes it an attractive choice for detecting the modularity of larger networks. Searching for modules using weighted data (rather than binary data) provides a different and potentially insightful method for evaluating network partitions.

Introduction

Bipartite networks are the representation of interactions between two distinct classes of nodes. Identifying structure within these networks is useful in explaining their formation, function and behaviour. Modularity is an evaluation of the partitioning of nodes into separate subsets, forming modules. These are also known as groups, compartments, communities or subgraphs. Determining functional groups of networks is an important challenge for a diverse set of fields including sociology, ecology and the physical sciences.

Maximising the modularity of a network is one method for detecting communities originally developed for unipartite (in which all nodes are allowed to interact with one another) networks (Newman and Girvan, 2004). Modularity is highest when each module appears isolated from the rest of the network. This occurs when nodes interact often with nodes in the same module, but there are few between module interactions. Negative modularity scores imply fewer interactions occur within modules than expected in a random network. But, positive modularity indicates that within module connectivity is higher than expected. The smallest and largest possible modularity scores that can be found are network dependent (Newman, 2010).

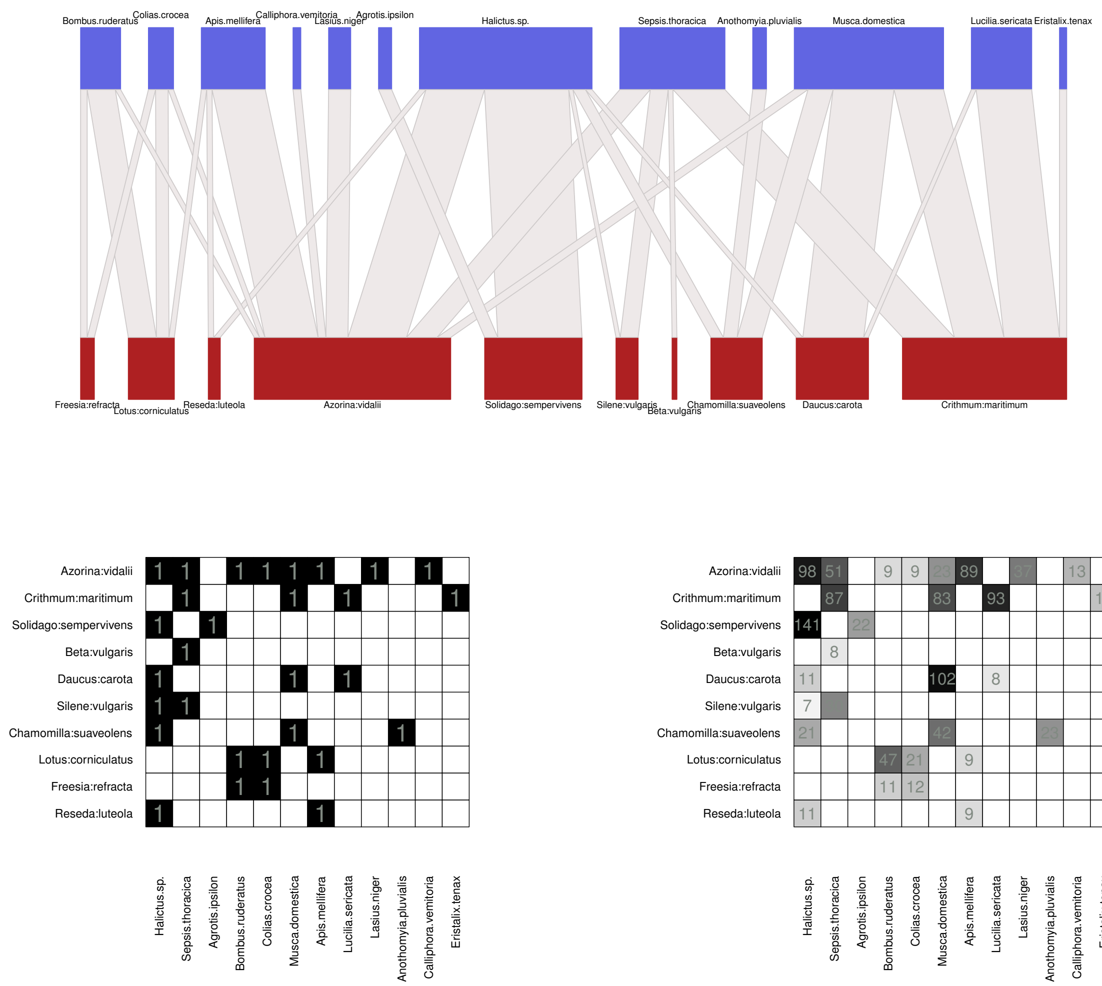

Figure 1: (a) The olesen2002flores bipartite network of 12 species of pollinators (blue nodes (top)) visiting 10 plant species (red nodes (bottom)). The width of the edges linking the nodes represents the number of pollinator-plant visitations, whilst the width of the nodes represents the marginal total of visits made by a pollinator species or received by a plant species. (b) The same network represented by the incidence 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 observed pollinator-plant visitations that occurred (where there is no number in a square there were 0 visitations)

There are several definitions of modularity used in bipartite networks. Guimerà's modularity (Guimerà et al., 2007) and Barber's modularity (Barber, 2007) were recently reviewed (Thébault, 2013) in the context of ecological networks. Guimerà's modularity uses weighted projections to identify separate communities within each node type. In contrast, Barber's modularity identifies joint communities composed of both types of node. In this paper I concentrate on the modularity definition proposed by Barber and its extension to weighted networks (Dormann and Strauss, 2014) to search for communities composed of both node types, which in the context of this study are communities of plants and their respective pollinators.

Modularity is a major feature of plant-pollinator networks (Olesen et al., 2007) and may contribute to network stability in these systems. They can be represented as bipartite networks with interactions between pollinators and plants. Pollinating species cannot pollinate other pollinating species, while plants cannot visit each one another – the only allowed interactions are between different plants and pollinators (an example network is shown in figure 1).

The majority of approaches to community detection only focus on whether two nodes have an association, regardless of the strength of those associations (Fortunato, 2010). However, there are some exceptions in unipartite networks (Newman, 2004). QuanBiMo (Dormann and Strauss, 2014) is the first algorithm to maximise weighted modularity in bipartite networks with quantitative data. 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 fast time complexity. These qualities make it a good candidate for extension to the case of weighted networks.

The definitions of binary and weighted modularity are presented. I show how to alter the LPAb+ algorithm so it can detect weighted modularity and denote this algorithm LPAwb+. A further modification allowing a more thorough search of modularity space is also presented. I call this Exhaustive LPAwb+. The three algorithms for maximising weighted modularity are compared on a dataset of 23 plant-pollinator networks. I find that QuanBiMo is highly sensitive to its input parameters, which may lead to reporting of modularity far below the optimal value in a given network. QuanBiMo reported less consistent modularity scores than either LPAwb+ or Exhautive LPAwb+. These experiments show that Exhaustive LPAwb+ and QuanBiMo performed well on smaller networks, whilst the speed of LPAwb+ makes it particularly suitable for use on larger datasets. The inclusion of quantitative information in networks alters the structure of detected modules.

Methods

Modularity

Barber's modularity

Bipartite or two-mode networks are made of two disjoint sets of nodes such that interactions only occur between nodes of opposite types. To generalise we say there are two node types: red and blue - and that interactions are only allowed between red and blue nodes. If there are r nodes of the red type and c nodes of the blue type, the adjacency matrix A is given in block diagonal form as:

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

where $\tilde{A}$ is the incidence matrix 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):

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

where m is matrix fill - the number of edges in $\tilde{A}$, 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). Red node labels are denoted g , whilst h are the labels for blue nodes and the Kronecker delta function $\delta\left(g_u, h_v\right)$ 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.

Weighted bipartite modularity

Weighted bipartite modularity, Q_W , can be defined as (Dormann and Strauss, 2014):

$$Q_W = \frac{1}{M}\sum_{u=1}^{r}\sum_{v=1}^{c}\left(\tilde{W}_{uv} - \tilde{E}_{uv}\right)\delta\left(g_u, h_v\right)$$

$$= \frac{1}{M}\sum_{u=1}^{r}\sum_{v=1}^{c}\left(\tilde{W}_{uv} - \frac{y_u z_v}{M}\right)\delta\left(g_u, h_v\right) \tag{2}$$

$\tilde{E}$ is the matrix of the null expectations of interaction between two nodes, where y is the row marginal totals and z is the column marginal totals of $\tilde{W}$, the weighted incidence matrix. In a binary network $\tilde{W}$ is equivalent to the binary incidence matrix $\tilde{A}$, the marginal totals will equal the node degrees ($y = k$ and $z = d$) and M, the sum of edge weights will equal m, the fill. Thus equation (2) will reduce to equation (1) for a binary network. Furthermore equation (2) can be reformulated into its matrical form (Barber, 2007; Flores et al., 2014) to allow for vectorised computation as:

$$Q_W = \frac{1}{M}tr\left(R\left(\tilde{W} - \tilde{E}\right)C\right) \tag{3}$$

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 red (blue) node belongs to (this information is held by the red and blue labels). These definitions of weighted bipartite modularity can now be used in the modified framework of the LPAb+ algorithm.

Weighted modularity maximising algorithms

QuanBiMo

The quantitative bipartite modularity algorithm (QuanBiMo) of (Dormann and Strauss, 2014), based on the heirarchical random graph algorithm (Clauset et al., 2008), 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`. The default settings available in bipartite version 2.04 were used (steps= 10^6, tolerance= 1^{-10}).

LPAwb+

A key feature of the LPAwb+ algorithm is that it simplifies to the LPAb+ algorithm when a binary network is used as input. The algorithm is made from two stages - a ‘bottom up’ step that maximises modularity on a node-by-node basis using label propagation; and a ‘top down’ step that joins modules together when it results in increased network modularity. First the dimensions of the network are used to decide how to run the algorithm (whose pseudocode is given in Algorithm 1); this is because a bipartite network can have at the most $F = min(r, c)$ communities with our chosen definition of modularity. The LPAwb+ algorithm is initialised by giving a unique label to each of the nodes in the smallest of the two sets. The LPAwb+ algorithm is sensitive to the initial labelling of nodes - this can lead to different values of modularity being reported. To combat this issue the initial node labels are randomly assigned and it is suggested that the LPAb+ algorithm is run multiple times on a given network to find the greatest modularity score (Liu and Murata, 2010).

Algorithm 1 LPAwb+ pseudo-code

```
Inputs :  an incidence 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 (equation (2)). For a particular red node x this can be written as choosing a new label g_x by trying to maximise the condition:

$$\begin{aligned} g_x &= \left(\sum_{v=1}^{c}\left(\tilde{W}_{xv} - \frac{y_x z_v}{M}\right)\right)\delta\left(g, h_v\right) \\ &= \left(\sum_{v=1}^{c}\tilde{W}_{xv}\delta\left(g, h_v\right) - \sum_{v=1}^{c}\left(\frac{y_x z_v}{M}\right)\delta\left(g, h_v\right)\right) \end{aligned} \tag{4}$$

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 (4) and creating an analogue for the updating rules for blue node labels leads to the following set of conditions:

$$\begin{cases} g_x^{new} = \arg\max\limits_{g}\left(N_{xg} - \frac{y_x Z_g}{M}\right) \\ h_x^{new} = \arg\max\limits_{h}\left(N_{xh} - \frac{Y_h z_x}{M}\right) \end{cases} \tag{5}$$

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 connecting to x labelled g, while Z_g is the sum of blue node degrees labelled g and Y_h is the sum of red node degrees labelled h. As these 'bottom-up' updating rules (equation (5)) are mutually exclusive of one another they are applied asynchronously such that blue labels are updated, then red nodes are updated, then blue nodes are updated and so on until modularity (equation (2)) 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 performed until 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 LPAwb+ algorithm.

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+ 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 incidence 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 each value 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. I chose 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 affect the magnitude of modularity that can be found (Newman, 2010; Thébault, 2013; Dormann and Strauss, 2014). In order to compare the strength of assortative mixing across different network studies 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 which 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, in which all edges are assigned to a module and there are no links between different modules (Newman, 2010). Extending this for weighted bipartite networks gives:

$$Q^{max} = \frac{1}{M}\left(M - \sum_{u=1}^{r}\sum_{v=1}^{c}\frac{y_u z_v}{M}\right)\delta\left(g_u, h_v\right) \tag{6}$$

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

$$Q^{norm} = \frac{Q}{Q^{max}} \tag{7}$$

Realised Modularity

Realised modularity (Poisot, 2013) has been suggested as a posterior measure of modularity that classifies the proportion of links in a network that are within, rather than between modules. Here I 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, then realised weighted modularity is expressed as:

$$Q_R^{'} = 2\left(\frac{H}{M}\right) - 1 \tag{8}$$

$Q_R^{'}$ takes values between -1, indicating that no edges exist between nodes in the same module, and 1, when all edges are interactions within-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 or within modules, only the strength of the connecting edges.

Normalised Mutual Information

The normalised mutual information criterion 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:

$$NMI(A;B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij}\log\left(\frac{N_{ij}n}{N_iN_j}\right)}{\sum_{i=1}^{C_A} N_i\log\left(\frac{N_i}{n}\right) + \sum_{j=1}^{C_B} N_j\log\left(\frac{N_j}{n}\right)} \tag{9}$$

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.

Data

I used the 23 plant-pollinator networks available in the bipartite R package (22 of which were used in (Dormann and Strauss, 2014) and the additional 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. Some network properties are shown in table 3.

Computing Modularity

I 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. I then quantified the differences between the modular structures found by the binary and weighted algorithms using the normalised mutual information criterion and investigated the differences in normalised and realised modularity.

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 in timing the algorithms 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 2a) and kato1990, junker2013, elberling1999, kevan1970 and barrett1987 for weighted networks (figure 2b) in which LPAwb+ and Exhaustive LPAwb+ detected much greater modularity scores than QuanBiMo.

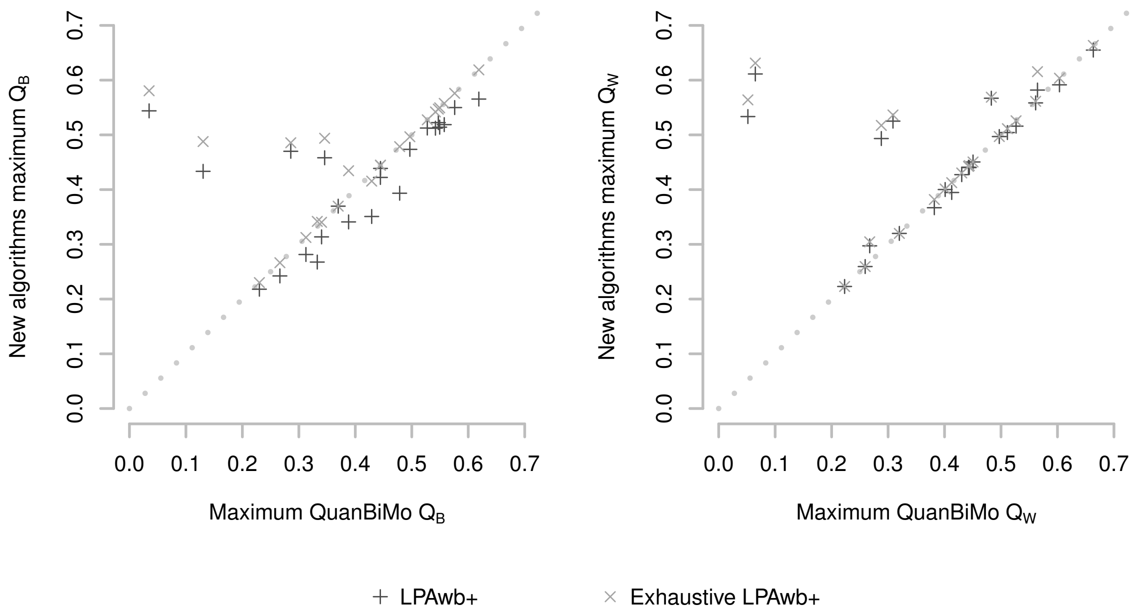

Figure 2: Comparing the maximum detected modularity scores by each algorithm (from 100 repetitions on each of the 23 plant-pollinator networks). The dotted line indicates a consensus i.e. QuanBiMo and the other algorithms are in perfect correspondence. Points below the dotted line indicate QuanBiMo maximises 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 binary modularity scores,Q_B, whilst (b) shows the weighted modularity scores, Q_W.

Table 1 shows the greatest modularity scores detected by each algorithm, the number of modules in these partitions 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 three 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.

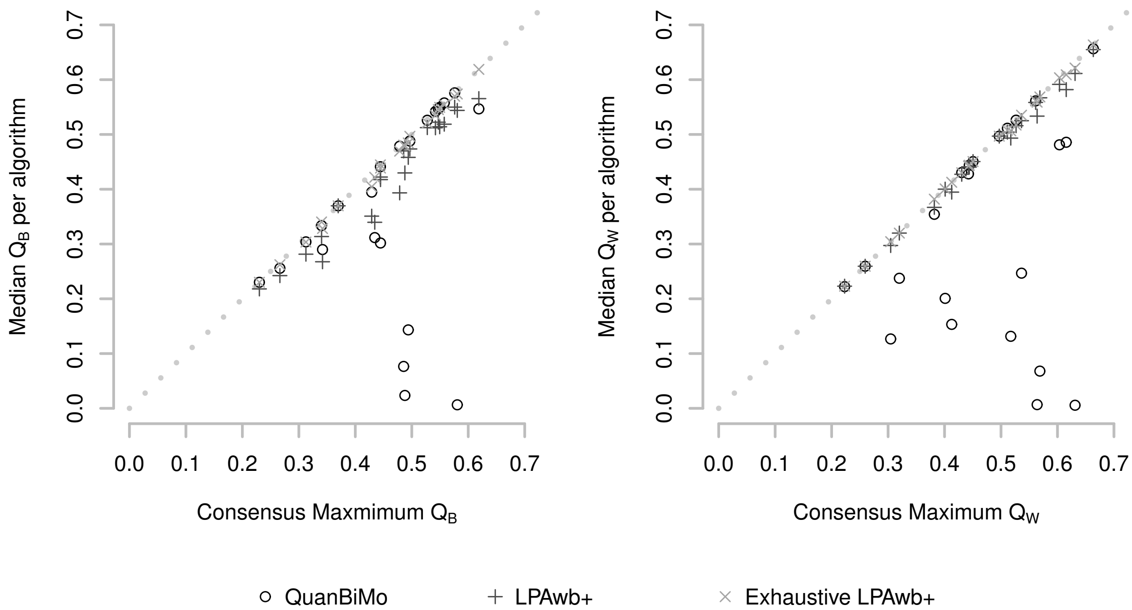

Figure 3: Comparison of median modularity scores found by each algorithm (from 100 repetitions on each of the 23 plant-pollinator networks) to the maximum of the modularity scores found across the algorithms - the consensus maximum modularity. (a) shows results for binary networks, whilst (b) shows the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

Figure 3 shows the median detected modularity scores for each algorithm against the overall maximum modularity score for each network. Figure $3a$ 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 $3b$ 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.

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.

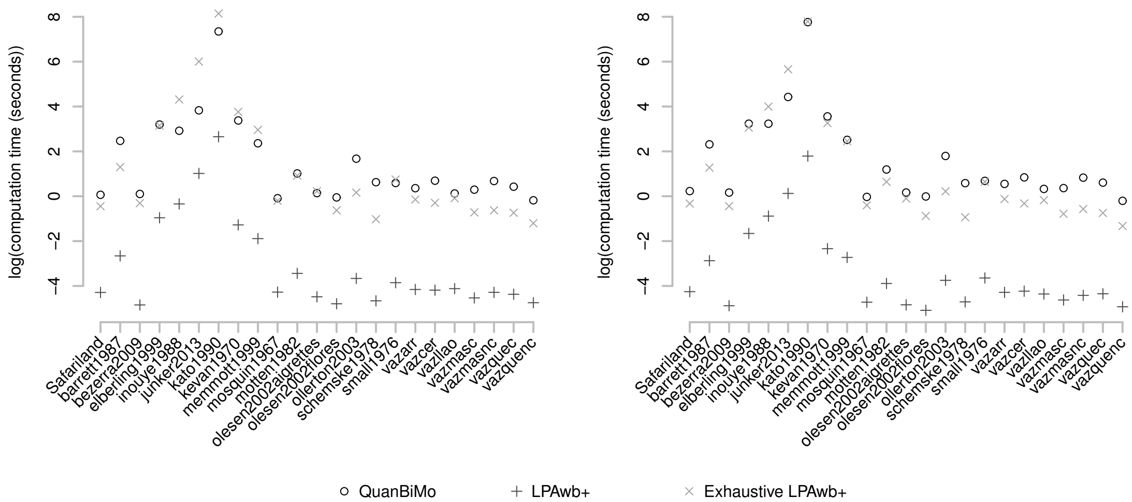

Figure 4: Average computational time for each algorithm (measured over 100 replicates) on the (a) binary and (b) quantitative representations of each plant-pollinator network.

	QuanBiMo			LPAwb+			Exhaustive LPAwb+		
Network	Q_W	M	t	Q_W	M	t	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.

The average time to run each algorithm is shown in figure 4. Performance time is network dependent; where it takes longer to report modularity for larger networks. LPAwb+ performed quickest on all networks by roughly 2 orders of magnitude. Performance on the binary (figure 4a) and quantitative (figure 4b) network representations was similar. However, QuanBiMo performed faster for binary (rather than quantitative) inputs on 20 of the 23 networks. On the other hand, LPAwb+ ran quicker with quantitative network representations (20 out of 23), as did Exhaustive LPAwb+ (18 out of 23). For the ten cases where Exhaustive LPAwb+ took longer than QuanBiMo, Exhaustive LPAwb+ found a partition with greater modularity seven times, both QuanBiMo and Exhaustive LPAwb+ found greatest modularity twice and QuanBiMo found the greatest modualrity score once (the binary representation of inouye1988).

Contrasting Binary and Quantitative Modular Structure

Maximising binary modularity and maximising weighted modularity results in different identified modular structures. Figure 5a shows the partition with the greatest binary modularity for the olesen2002flores network, whilst figure 5b shows the partition with the greatest weighted modularity. The same dataset has qualitatively different structure between its weighted and binary representations. The shared normalised mutual information for these two partitions is $NMI = 0.619$, quantifying this difference.

Figure 6a shows the differences in normalised modularity and normalised mutual information between the binary and weighted network representations. Only 3 of the networks (vazquenc, vazmasnc and vazcer) have a normalised mutual information greater than 0.8 - indicating major differences in identified binary and quantitative modular structures. The strength of assortative mixing, measured by normalised modularity, was generally greater in weighted than binary networks. However, 4 networks (olesen2002aigrettes, vazarr, bezerra2009, vazmasnc) showed greater assortative mixing in their binary representations and for 2 networks (olesen2002flores, vazllao) the assortative mixing strength was nearly the same in both binary and weighted networks - though the community partitions are very different.

Not only were the detected modularity scores different between the binary and weighted networks - but the number of modules found in each partition 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 (table 1,table 2).

There appears to be a weak positive relationship between realised modularity and modularity (figure 7a), however normalised and realised modularity appear to be much more strongly correlated (figure 7b). There does not appear to be a relationship between the binary and quantitative measures for each network.

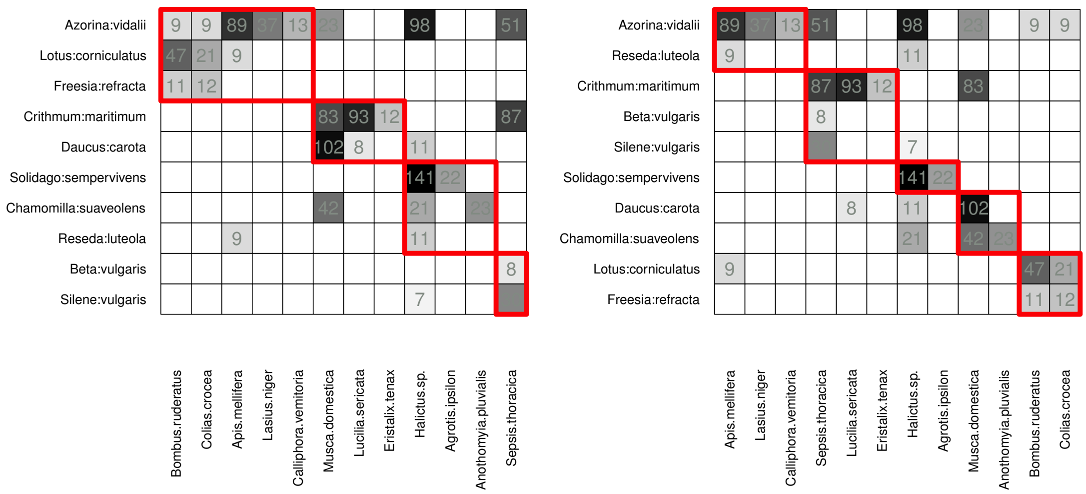

Figure 5: A visual comparison of the modular structures identified for the olesen2002flores dataset of plant-pollinator visitations as a (a) binary ($Q_B = 0.444$, 4 modules, $Q_B^{norm} = 0.625$) and (b) quantitative ($Q_w = 0.497$, 5 modules, $Q_W^{norm} = 0.625$) network. Modules are identified in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

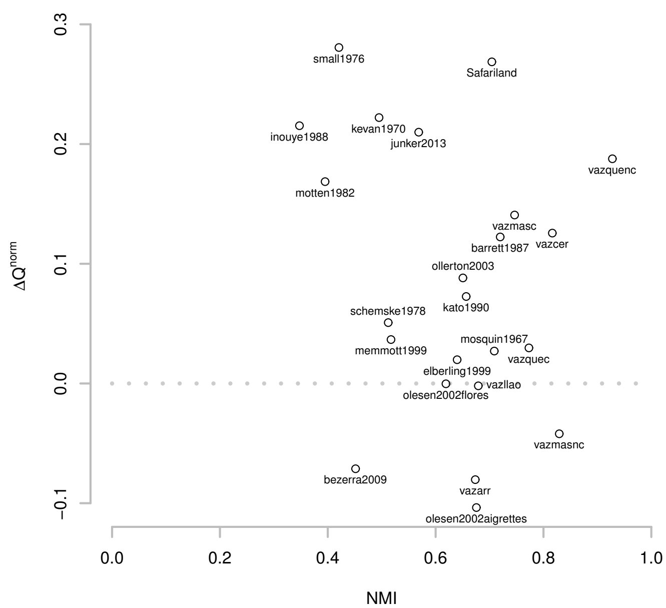

Figure 6: The change in normalised modularity scores found between the weighted and binary networks ($\Delta Q^{norm} = Q_W^{norm} - Q_B^{norm}$) against the normalised mutual information between the weighted and binary partitions for each network.

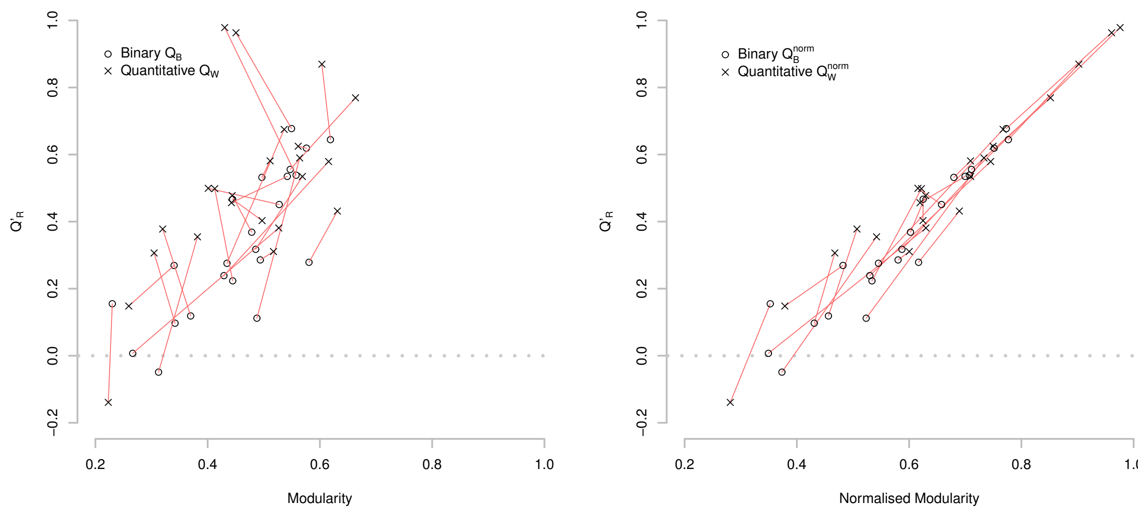

Figure 7: (a) The greatest modularity scores (Q_B and Q_W) for each network and their corresponding realised modularity scores (Q'_R). (b) The normalised modularity scores (Q_B^{norm} and Q_w^{norm}) calculated using the partitions with greatest modularity scores plotted against their corresponding realised modularity scores. Each red line joins together the binary and quantitative scores of the same network.

Discussion

I tested the efficacy of three algorithms maximising Dormann's bipartite modularity in plant-pollinator networks. LPAwb+ and Exhaustive LPAwb+ gave more consistent modularity scores than those found by QuanBiMo across the test networks. The robustness of modularity maximisation algorithms is important when considering the reproducibility of results. QuanBiMo struggled to report "good" modularity scores in the larger datasets. All three algorithms were able to detect greater modularity than previously reported (figure 6 in (Dormann and Strauss, 2014)) and were generally performed well on the binary and quantitative test networks (a binary network can be seen as a special case of a quantitative network). But, QuanBiMo has the potential to fall into below par solutions and there is no diagnostic to show when this occurs.

Different modular structures were found for each of the binary and weighted representations of plant-pollinator networks. In binary networks modules are formed by attempting the maximise the density of edges; whilst in quantitative networks modules are formed that maximise the density of edge weights. In the former, strongly interacting nodes are just as important as nodes that only rarely interact; whilst in the latter modules are likely to form around the strongest node-node interactions.

Normalised modularity measures the strength of assortative mixing and is a useful network index that can be used as a comparison indicator across different network studies. Modularity by itself is often used as a network indicator - but this is not appropriate when comparing different networks whose theoretical modularity maximums may differ. I find normalised modularity is strongly correlated with the proportion of within module interactions (realised modularity) which is an intuitive way for understanding modularity.

LPAwb+ was not able to maximise modularity so well as Exhaustive LPAwb+ or QuanBiMo on the majority of datasets (though the modularity found was near the maximal value found here), but its fast performance makes it an ideal algorithm for exploratory research and for investigating modularity in larger networks, where parallelisation of the algorithm (Liu and Murata, 2010) may become useful.

There is no guarantee that the greatest possible modularity was found in any of the test networks here; indeed 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 possible network.

The QuanBiMo algorithm takes two input values; the number of algorithmic steps that should be performed to attempt to find greater modularity than the current partitions modularity; and the tolerance threshold for greater modularity scores. Clearly the default values were not appropriate for some of the networks assessed here; where much greater modularity was detected by the new algorithms. However, there is no diagnostic to tell that QuanBiMo has returned a sub-par modularity value without comparisons (which may be a lengthy process); or what suitable input parameters may be for a particular network. There is a strong tradeoff between computational effort and the accuracy of the returned modularity. On the other hand LPAwb+ takes no input parameters and was able to quickly find modularity scores near to the consensus maximum modularity. Exhaustive LPAwb+ has two input parameters; the minimum number of modules to search for and the number of times that LPAwb+ should be initialised for each module number. Unlike QuanBiMo, these parameters have physical meaning in the context of the network – and the time complexity of this algorithm can be estimated from the number of calls that will be made to the LPAwb+ algorithm (as LPAb+'s time complexity is known (Liu and Murata, 2010)).

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 hierarchies 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 corresponds best with any additional information known about the network. This is further complicated as several hierarchical 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 found with QuanBiMo, the ability of algorithms to maximise modularity can be 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.

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). In principle it would be good to test against a null ensemble in which both the allowed interactions and the strength of these interactions are allowed to vary. However, in this paper I have only focussed on the optimisation of weighted modularity.

Another limitation of the weighted modularity definition explored here is that it is only valid on networks where all connections are positive. However, methods have been created to search for modules in weighted networks with positive and negative link strengths in unipartite networks that could easily be extended for bipartite networks (Gómez et al., 2009).

I focussed on a specific definition of modularity in this paper - but note that others do exist (Guimerà et al., 2007; Murata, 2010). Thébault (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; Dormann and Strauss, 2014) corresponded well with that for unipartite networks (Newman and Girvan, 2004; Newman, 2004) - and is well suited for identifying densely connected modules. Other modularity measures (Guimerà et al., 2007; Murata, 2010) do not identify joint communities made of both types of nodes – but rather identify communities within each type of node, though neither of these approaches has yet been extended to weighted networks to my knowledge.

The major advantage in a definition of weighted modularity is that it allows for much more information about a network to be used to detect communities. Both binary and weighted measurements contain different information about a network and may be useful - though I expect weighted measurements may in general contain more relevance for the analysis of real world networks – the strength of interactions is undoubtedly an important component of network structure. Other modularity definitions and their weighted extensions are also in need of further investigation to consider communities within each type of node and how these may overlap with the joint communities considered here.

Conclusions

Real world networks are not formed of binary interactions. I 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 use 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. I would recommend using Exhaustive LPAwb+ over QuanBiMo; as Exhaustive LPAwb+ has more meaningful input parameters, can perform no worse than LPAwb+ and its performance was less variable than QuanBiMo on the networks tested in this study.

I have made the code for the LPAwb+ and Exhaustive LPAwb+ algorithms; as well as the analysis performed in this paper available online (https://github.com/sjbeckett/weighted-modularity-LPAwbPLUS) to allow researchers to replicate my 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 - leading to the weighted counterpart presented here. I thank Timothée Poisot and Hywel Williams for comments that improved the manuscript.

References

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

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

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

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

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

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

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

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

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

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.

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*

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

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

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

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.

Gómez, S., Jensen, P., and Arenas, A. (2009) Analysis of community structure in networks of correlated data. *Physical Review E 80*, 016114.

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.

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.

Kato, M., Kakutani, T., Inoue, T., and Itino, T. (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.

Network	QuanBiMo R	$\tilde{x}$	U	F	$Q_R^{'}$	LPAwb+ R	$\tilde{x}$	U	F	$Q_R^{'}$	Exhaustive LPAwb+ 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) judged by comparing the normalised mutual information of partitions sharing this value, 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.

Network	QuanBiMo R	$\tilde{x}$	U	F	$Q_R^{'}$	LPAwb+ R	$\tilde{x}$	U	F	$Q_R^{'}$	Exhaustive LPAwb+ 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) judged by comparing the normalised mutual information of partitions sharing this value, 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.