# EXEUTIVE SUMMRY

The project is about evaluating selective algorithms primarily developed for finding "*communities"* in real world systems; one of them is "*food web*". Communities are set of highly connected entities in a network. They exist inherently in innumerable real networks. Their presence has been observed, for example, from small networks like a joint-family that involves more than one sub families as its community to massive networks like virtual groups created on social networking websites for people with common interest. There are number of examples of communities in various fields, such as, Sociology (e.g. social network), Neurology (neural networks), Information Technology (e.g. Internet, World Wide Web) etc. In the field of Ecology, food web is such a network where a group of various species living in common habitat or having some common characteristics forms a community. There are hundreds of algorithms for detecting communities in network but few, if any that can work properly on food web, since food webs exhibits special kind of network characteristic. This project shows experimentation done with carefully selected algorithm on few existing datasets of food web and its analysis.

The work was non-trivial since food webs are mostly bipartite or directed–weighted networks. The first challenge was to research and gather all the algorithms (as well as its corresponding binaries) which can work on these type of networks. Secondly bipartite, directed and weighted graphs (network) are in general harder to evaluate than uni-partite or un-weighted graphs. Previous work done in this area (e.g. the algorithm developed by Girvan and Newman [4]) didn't consider weights while evaluating. Other study [22] involved using only single algorithm [21-30] for various food webs around the world.

This thesis reports the research carried out for comparison and evaluation of seven algorithms:

- The algorithms are divided into two categories based on type of network they can handle. In the first category, four algorithms namely Simulated Annealing, BRIM, MAGA and COPRA were applied on three different bipartite plant-pollinator food webs. Three algorithms in the second category namely Infomap, Louvain and OSLOM were applied on three directed and weighted predator-prey food webs. "Modularity" (defined later in the thesis) is used to measure the quality of algorithms.
- Further for bipartite plant-pollinator, food webs were split into pair of uni-partite network and check again for community structure just to see the change in previous results.
- For predator - prey food webs, the algorithms were evaluated with and without weights to check whether weights are misleading.
- The plant-pollinator network showed significant modular structure i.e. high chances of existence of community where as predator-prey network displayed modular structure which is not good enough for it to be called a community.

# Table of Contents

# 1. INTRODUCTION

Any real word networked system in today's world can be represented with a graph which is neither regular nor random in nature. Here entities of the system corresponds to vertices in graph and any type of connection between them corresponds to edges. Examples of such network system are present in many fields. For example in technology, networks like*World Wide Web*[1] where different web pages can be represented by vertices and the links through which users are navigated from one page to other page as edges. In sociology , *social-network* [2] of people where naturally people are the nodes and their relationships to each other are edges. In ecology, we have *food webs*[3] which can be viewed as graph by representing different spieces as nodes and interactions (i.e predators feeding on preys) as edges.There are numerous other kinds of networks in real world as well. All of these networks have quite a few underlying properties which allows us to understand the functioning of the system in a precise way. Some special statistical properties of such real networks such as *small-world phenomenon, power-law degree distribution* and *network transitivity* have turned the focus of many researchers recently. Another such property of these networks is *community structure* or *clustering*[4] which exhibits the fact that there exists abundance of interactions within the communities of the network and relatively few interactions between them.

Finding a community in a large network has found number of applications. For example in e-commerce, finding a group of online customers with similar likes and dislikes has an advantage of setting up efficient recommendation system, that not only guides the customer through other relevant items of the retailer but also adds value by increasing the sale[5]. Data clusters in massive disk-size graphs are used to devise new data stucture that enables searching within the database fast and efficient without the expense of additional storage space[6]. In the field of Mobile Computing, Ad hoc networks are configured in order to stabilize the signal streangth of moving devices (i.e. cellular phone, laptop etc), such networks cannot maintain a routing table that generally keeps routing information (i.e which node has to communicate with which other node in the network). Forming a cluster of these devices enables ad hoc networks to generate and maintain efficient routing tables, making the communication more reliable [7]. If the clusters and their boundaries are correctly identified, one can determine the approximate location of the vertices which may reveal certain properties of the network. For example, one can infer that any vertex positioned at the center of a cluster and having many links to the surrounding vertices plays a major role in the stability of the network. Also any vertex protecting the boundaries of a cluster does the work of gluing different modules together. Removing such vertices may result into two clusters, completely isolating from eachother which further leads to a completely isolated network.

## 1.1 Aims & Objectives

The aim of this project is to find inherent and natural communities in the food webs using an algorithm that best suits the kind of networks they represent. Food web exhibits special types of network structure i.e. bipartite or directed that poses extra challenge on community detecting power of an algorithm.

To achieve this aim:

- First of all, thorough research should be carried out to select algorithms that deal with this special kind of network.
- Evaluate each of them using various options available on existing food web datasets.
- Compare and analyse the results obtained with one another and have meaningful conclusion.

## 1.2 Scope

This thesis consists of 9 key sections. As you have observed, the project name involves two different terminologies "*Community Detection*" and "*Food Web*" from the field of *Computer Science* and *Ecology* respectively. In order to get the insight of the actual topic, it is very important to understand these two terms individually. Therefore, section 2 provides detailed understading of community detection. Here we shall assume that the reader has little knowledge of *Graph Theory*. Section 3 acquaints the reader with food web and some related terminologies. Section 4 discusses the previous work done in this area (i.e. different techniques to find communities in food web). Section 5 discusses each algorithm used in the study briefly. In section 6 method used for evaluation are discussed. Section 7 shows the results obtained and comaprison & critical analysis for that is given in section **Error! Reference source not found.**. In section 8 concluding remarks are given. Finally future directions are given in section 9.

# 2. COMMUNITY DETECTION

## 2.1 History

In 1927, Stuart Rice analysed a group of people voting for the election of a leader. He divided them into a number of clusters, based on their votes. This activity brought community detection in the picture. Few decades later in 1955, Weiss and Jacobson [9] formally analysed community structure. In that, they analysed all the people working in a government agency and their relationship with others by arranging personal interviews. They studied how the interactions were taking place within and between group of people and the consequences of removing members who were working with people of different groups. Results showed different groups being isolated from one another. This very idea of removing the connectors of different groups serves the basis for a few modern community detection techniques.

In 2002, Girvan and Newman [4] came up with an algorithm which was based on the same concept derived by Weiss and Jacobson, that is, identify those vertices acting as connecting links between different modules. By iteratively removing these links, entire network transforms into isolation of modules. The interconnecting links were identified by searching for those edges which are "least" central to or most "between" communities with the help of edge/vertex betweenness centrality measure.

## 2.2 Basics

In this section we present several fundamental concepts related to community detection. First, we shall attempt to give you a general overview of the community. We then try to formally define community in graph and finally we shall review the notion of partition.

### 2.2.1 Overview

The goal of community detection algorithms is to identify the inherent modules, fix their bounderies and sometimes derive the hierarchical organization by making use of information encoded in the graph topology [10]. Fig. 1 shows the example of small network with a community structure.

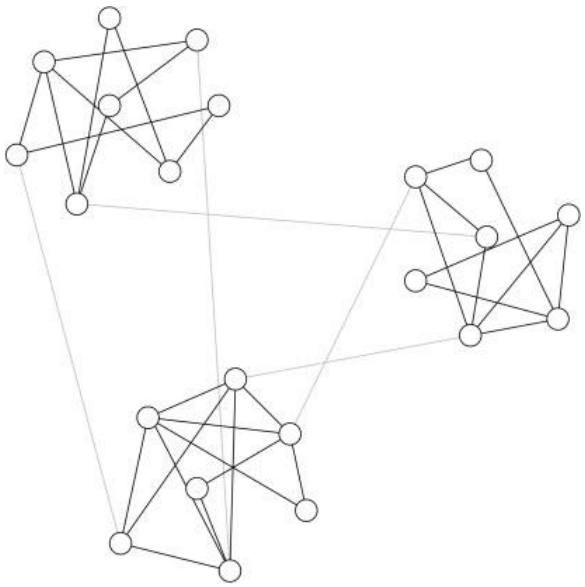

**Figure 1: Representation of small network with community's structure. Taken from [4]. Edges within the communities are relatively dense than they are between them.**

The interaction between the members of any real system is not necessarily reciprocal in nature. There are many networks in which they are one-way or unidirectional. For example, network of authors and article or paper they have written, where interaction can only happen between different classes of vertices and not amongst them. (i.e. auther-to -article and not

auther-to-auther or article-to-article). Food web is other example where interaction can only occur between predator and prey.

### 2.2.2 Definition

The most unfortunate thing about community detection is that till date no definition of community has been universally accepted. The definitions vary with the specific system under consideration. By seeing at Figure 1 one can clearly observe, there are more edges inside any of the three communities than they are outside. This characteristic provides the basic reference for formal definition of community. The definition given below is taken from [10].

Assume that we have graph $\mathcal{G}$ with $n$ vertices, which includes sub-graph $g$ with vertices $n_c$. Internal and external degrees (i.e. number of edge connecting vertex to any other vertex of the graph) of any vertex $v \in g$ are defined as $\mathcal{K}_v^{int}$ and $\mathcal{K}_v^{ext}$ respectively. If $\mathcal{K}_v^{ext} = 0$ then vertex $v$ has visibility only within sub-graph $g$ which is one of the characteristics of a good cluster. On the other hand, if $\mathcal{K}_v^{int} = 0$ applies that vertex $v$ should be part of sub-graph under evaluation. Similarly, internal degree of graph $g$, $\mathcal{K}_{int}^{g}$, is the sum of internal degrees of all its member vertices and external degree $\mathcal{K}_{ext}^{g}$ is sum of the external degrees of all its member vertices. This implies the total degree of sub-graph $g$ :$\mathcal{K}^{g} = \mathcal{K}_{int}^{g} + \mathcal{K}_{ext}^{g}$.

Finally we define intra-cluster density $\delta_{int}(g)$ for sub-graph $g$ as the ratio between the number of internal edges of $g$ andthe number of all possible internal edges:

$$\delta_{int}(g) = \frac{K_{int}^{g}}{\left[n_g(n_g - 1)/2\right]} \tag{1}$$

And inter-cluster density $\delta_{int}(g)$ as the ratio between the number of external edges of and the number of possible external edges:

$$\delta_{ext}(g) = \frac{K_{ext}^{g}}{\left[n_g(n - n_g)\right]} \tag{2}$$

If $\delta_{int}(g)$ is substantially bigger than the average $\delta(g)$ and $\delta_{ext}(g)$ is substantially smaller than $\delta(g)$ we can call sub-graph $g$ a *community* of the original graph $\mathcal{G}$, where $\delta(\mathcal{G})$ is defined as ratio between total number of edges and number of all possible edges $n(n-1)/2$.

Please note providing formal definition of community like one given above does not make it universal. It is defined with the aim to help the user understand community better in general. Literature offers many other definitions of community. These are often based on degrees of

internal cohesion among vertices. Presenting those definitions is beyond the scope of this document.

## 2.3 Graph Partitions

A graph can be partitioned by dividing it into disjoint or (sometimes) overlapping modules. In real systems like social network, elements are often parts of more than one community (e.g. Communities in Facebook). The terminology used in place of *partition* in such overlapping communities is *cover* [10]. Sometimes there exists a hierarchical structure in partition where original graph exhibits the characteristics of different level of organization. For example, in business firm employees along with their managers form a team, a number of different teams altogether form a department and these departments in turn form an organization. What we observe is the group of departments displaying a hierarchical community structure as large communities containing small communities, all of which in turn have smaller communities inside them. Since these clusters are properly nested, they can be represented with help of a tree diagram shown in Fig.2. In that, the nodes connected at the lowest level, represents the strength of the link that resulted in those two nodes, first becoming member of the same community. Tree structure of this kind is called *dendrogram*. Dendrograms are often used in tradition community detection algorithms like *hierarchical clustering*. As shown in the Figure 2, if we cut the dendrogram through a horizontal line it results in two different partitions each displaying a hierarchical structure. Nodes representing community at specific level are included in a community at the higher level.

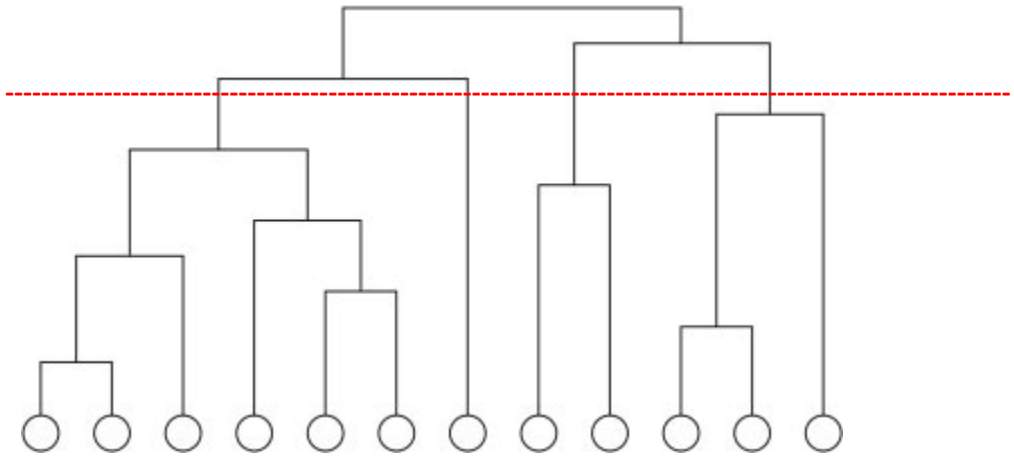

**Figure 2: A Dendrogram of hierarchical organization (i.e. business firm). The cut through the graph represents the partition of a graph into two clusters. Adapted from [4].**

## 2.4 Modularity : Quality function for partitioning

### 2.4.1 Null Model

Since the concept of *modularity* relies heavily on the term *null model*, it becomes necessary to define the latter before defining the former one. As with the community detection,

definition of null model is often considered arbitrary and varies according to the system at hand. Our definition is based on Girvan and Newman [11] which is as follows:

Null model is copy of the original graph (displaying some community structure) which preserves some of its structural properties such as the expected degree distribution of all the vertices or edge distribution between the vertices and despite of that, it is a random graph (i.e. graph without any community structure). In short, a null model is a randomized version of the original graph. The definition gives the intuition that null models can be used for the purpose of comparison with the original graph to check whether or not it exhibits any characteristics of community structure.

Barber [34] and Guimera *et al* [30] addressed that null model for bipartite network can't be produced in the same fashion as uni-partite network. They mentioned that, in the null model of uni-partite network, edges can exist between any pair of vertices and this approach can't be extended for bipartite where edges only exist between two classes of vertices (See section 3.2.1). Barber [34] proposed new definition of null model for bipartite network in order to define bipartite modularity. Now we have sufficient knowledge to define modularity.

### 2.4.2 Modularity

Modularity ,proposed by Girvan and Newman [11], is essentially a quality function that measures the quality of the partition of graph into modules resulting from any partitioning algorithms[10]. It is the key concept behind most of the modern graph clustering algorithms and is often considered a global critearian for defining a community and its goodness. As an example, any subgraph is called community if the total number of edges residing in sub-graph exceeds its expected value that was derived from the same sub-graph in null model i.e.

$$Q = [Fractions\ of\ edges\ within\ communites] - [Expected\ fraction\ of\ edges\ in\ null\ model]$$

This expected value is calculated by taking average over all possible realization of null model (i.e. not from an arbitrary null model). Then modularity of graph is formally expressed as follows:

$$Q = \left(\frac{1}{2m}\right)\sum\left(A_{ij} - P_{ij}\right)\delta\left(C_i - C_j\right) \tag{3}$$

where sum is taken over all pair of vertices, $A_{ij}$ is the value for the vertices pair *i* and *j* of the adjacency matrix $A$, m is the size of the graph (i.e. total number of edges), $P_{ij}$ is the expected value for the number of edges between vertices $i$ and $j$ in a null model and $\delta$ is a function that outputs 1 if $C_i = C_j$ and 0 otherwise. Some null model requires that expected degree sequence of the graph calculated by taking average over all possible configuration of the model, matches the actual degree sequence of the graph. In this kind of arrangement, probability that a vertex $i$ with degree $k_i$ is attached to any other vertex $j$ with degree $k_j$ of the graph can be

calculated without problems. In order to create an edge between $i$ and $j$ one has to join two half-edges that incident with $i$ and $j$. The probability $p_i$ for picking up random half edge incident with $i$ is ($k_i/2m$), since there are $k_i$ half edges that incident with $i$ out of a total of 2m. The probability that different vertices $i$ and $j$ is connected is given by the product $p_i p_j$, as edges are placed independently of each other. Therefore, $P_{ij}$ becomes $2m * p_i p_j$ and the above expression thus becomes

$$Q = (1/2m)\sum\left(A_{ij} - \left(2m * p_i p_j\right)\right)\delta\left(C_i, C_j\right)$$

$$= (1/2m)\sum\left(A_{ij} - \left(k_i k_j/4m^2\right)\right)\delta\left(C_i, C_j\right) \quad (4)$$

In this expression vertex pairs lying in the same module are the one contributing to the sum. We can aggregate this sum over pair of vertices to the sum over entire module, which gives following expression.

$$Q = \sum_{c=1}^{n_c}[(l_c/m) - (d_c/2m)^2] \quad (5)$$

From to Eq. (5) one can infer that a subgraph is called a community if its contribution to modularity in sum is positive. In other words, the more the number of intra-edges of the module exceeds the expected value, the better it displays the community. Now we understand that why modularity is the measure of quality of the partition, as the larger the values of the modularity, the better the quality of partitions.

Apart from modularity literature provides many other quality functions. Example incluldes *performance* that counts number of correctly interpreted pair of vertices. By correctly interpreted means, whether pair of vertices belongs to the same community and connected by an edge or belongs to different community and not connected by an edge. One more example of quality function is *Coverage*, which is the ratio of intra-module edges and total number of edges.

The maximum possible value of modularity generally grows with the size and/or the number of clusters [12] within the graph. As a matter of fact, modularity should never be used to compare the quality of the community structure of graphs with large difference in their size. The modularity of the graph taken as single entity, is zero, as in above equation the two terms of the only summand becomes equal and opposite. Modularity is always smaller than one, and may also be negative. Assume the partition in which each vertex represents community. In this case the sum runs over all n vertices, which are all negative since the first term of each

summand is zero. This implies that, if there are no partitions with positive modularity, the graph doesn't contain a community structure. On the other hand, if there exists partition with large negative modularity values, subgroups in them have very few internal edges and more inter cluster edges.

## 2.5 Example of existance of community in real life networks

Here we present two well-known examples of community in real life. This is just to get oneself acquainted with the manifestation of community (i.e. how do they look like) and to understand the reason why they are important.

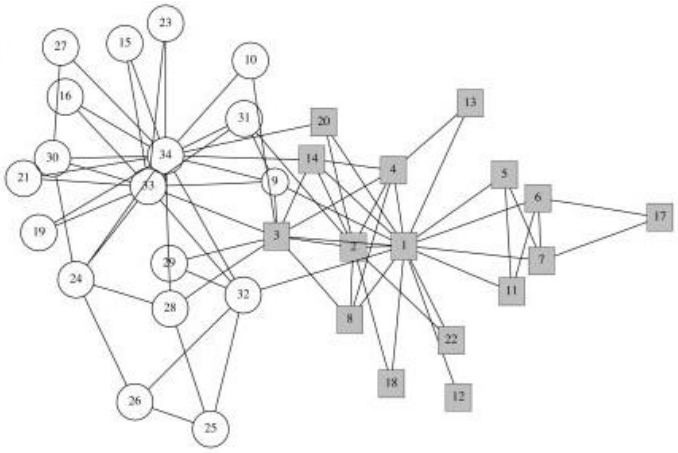


**Figure 3: The graph representation of Zachary's study of karate club. Nodes (members) supporting president are drawn as circle and those supporting instructors are drawn as squares.**

The first example is drawn from social networks. That is, karate club study of Zachary [12]. In that Zachary observed one karate club in the United States. The club consisted of 34 members who were observed over the span of two years. During the study, some disagreement between the president and the instructor resulted into division of the club in two parts. Zachary demonstrated the network of friendship between members of the club, using different measures to estimate the strength of ties between them. In Figure 3 nodes represents all the members and edges shows their interaction outside the club. By carefully observing the figure one can spot that most of the vertices are centered on either vertex 34 (the president) or vertex 1 (the instructor) forming two overlapping communities. Vertices connecting both primary vertices (1 and 34) are those whose membership is often hard to classify by community detection algorithms.

Next example of existence of community structure is taken from food webs. Food webs are discussed in depth in section 3. Figure 4 shows the graphical representation of community structure of the Chesapeake Bay food web. It has 45 nodes representing different taxa, divided mainly in two groups, (A) pelagic (living on the surface of water) and (B) benthic (living in sediments). The edge between two taxa represents their predator-prey relationship (predator feeding on prey), where arrow originates from predator and ends towards prey. The thickness

of the edge shows rank of their interaction strength. A consist of 28 pelagic taxa and B consist of 17 benthic taxa and both are linked with each other through some weak interactions. Position of a taxon shows its importance within the group. For group B, taxa 45, 22 and 2 are centrally located, indicating their high importance in comparison with those protecting the boundaries, such as 33, 41 and 21.

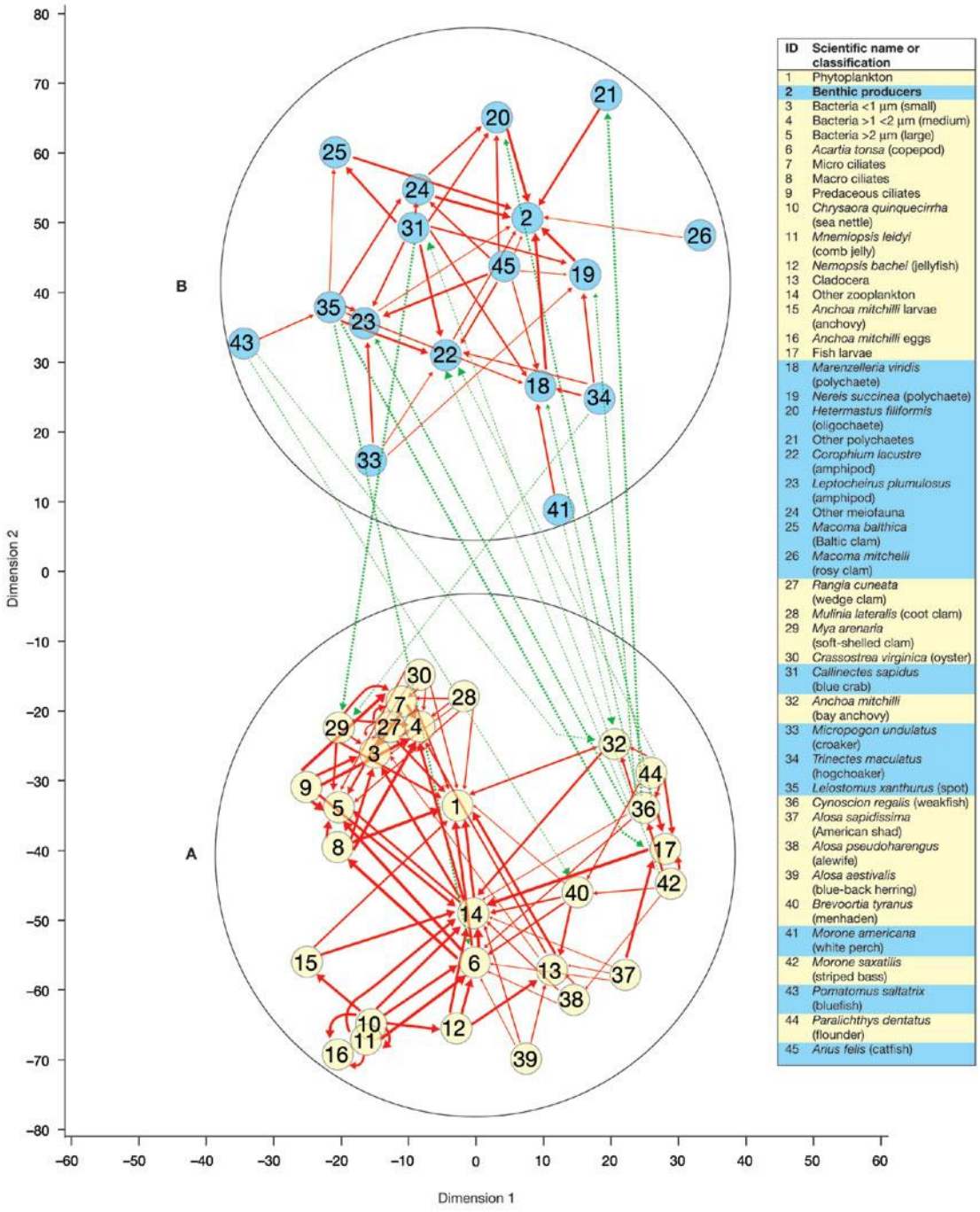


**Figure 4: Graphical display of community structure revealed in Chesapeake Bay food web. Units displayed on the graph shows relative distances based on the inverse of the density of interaction. Image is taken from [13].**

## 2.6 Community Detection Algorithms

In this section, we shall look into few of the existing methods for detecting community in real systems. These methods (i.e. algorithms) can be classified into different categories. There are *traditional methods*, primarily based on *graph partitioning*. Examples of traditional methods include *hierarchical clustering*, *partitional clustering* and *spectral clustering*. Then there are *divisive methods* based on idea of removing the edges that connects inter-module vertices in order to find communities in given network. The best example of this is the algorithm given by Girvan and Newman (described later in this section). *Modularity-based methods,* as the name suggests makes use of modularity as quality function. Examples include *simulated annealing*, *extremal optimization*, *spectral optimization* etc. Methods are large in numbers and therefore we pick up three out of all, and describe them briefly. Next three sub sections explain *hierarchical clustering*, *Girvan and Newman's algorithm* and *simulated annealing method* respectively.

### 2.6.1 Hierarchical Clustering

Hierarchical clustering is used when the system under analysis exhibits some sort of hierarchical structure, where small clusters are included within large ones, which are in turn part of the larger ones. The method described here is adapted from [10].

The method essentially requires defining a concept called *similarity measure* that tells how closely a pair of vertices is related. After having chosen the similarity measure, the same is computed for every pair of vertices in the graph (even unconnected pair of vertices) and new similarity matrix (called X) n × n is formed. Next, the clusters of vertices with high similarity are identified and merged by iteratively connecting those pair of vertices with edges. In this process, we start from vertices as singleton clusters and end up with the graph as a unique cluster.

Here we define a measure that estimates cluster's similarity from entries of the matrix X. There exist several techniques for that. In *single linkage* clustering, the similarity between two clusters is given by minimum element $x_{ij}$, with *i* in one cluster and *j* in the other. In *complete linkage similarity,* the same is given by maximum element $x_{ij}$ for vertices of different groups. In average linkage clustering, the average of the $x_{ij}$ is taken. Since the clusters are properly nested they all can be represented by means of dendrograms like the one in Fig. 2 (See Section 2.3).

Although this method looks simple and accurate at first glance, it has number of drawbacks. *First*, it does not recognize the differences between all the partitions obtained by the procedure. *Second,* the results of the method highly depend on the specific similarity measure chosen at the beginning. *Third*, the method also outputs a hierarchical structure even for those networks that may not have it at all. *Fourth*, the method often incorrectly classifies many vertices, sometimes even those, on which other vertices are aggregated [14]. In a case, where a vertex is connected to the rest of a network by only a single edge, it should obviously be considered to belong to the community at the other end of that edge. This method often

classifies such vertices as isolated clusters. *Fifth* and the last, the hierarchical clustering method doesn't scale well. Take an example where nodes are embedded in space and by using distance as (dis) similarity measure, the computational complexity turns out to be $O(n^2)$ for single linkage and $O(n^2 \log n)$ for the complete and average linkage schemes. The only plus point hierarchical clustering has is that it does not require a preliminary knowledge on the number and size of the clusters. These five drawbacks against one and the only advantage make the hierarchical clustering method, although useful, far from being perfect.

### 2.6.2 Girvan and Newman's Algorithm

Girvan and Newman's algorithm [4] is based on the concept of centrality measure called "*edge betweenness*" (which is in turn based on *vertex/site betweenness* introduced by Freeman [17]) and its iterative removal. Edge betweenness is defined as the number of path between different pair of vertices that run through the edge [4]. From the definition, we get an intuition that edge betweenness for all the inter-community edges will be greater than that of the intra-community edges. This is due to fact that all the shortest paths connecting vertices of two different communities run through one of these inter-community edges. This illustration is depicted in Figure 5. If we remove these inter-community edges, the modules will become isolated smaller networks and the underlying community structure will be exposed.

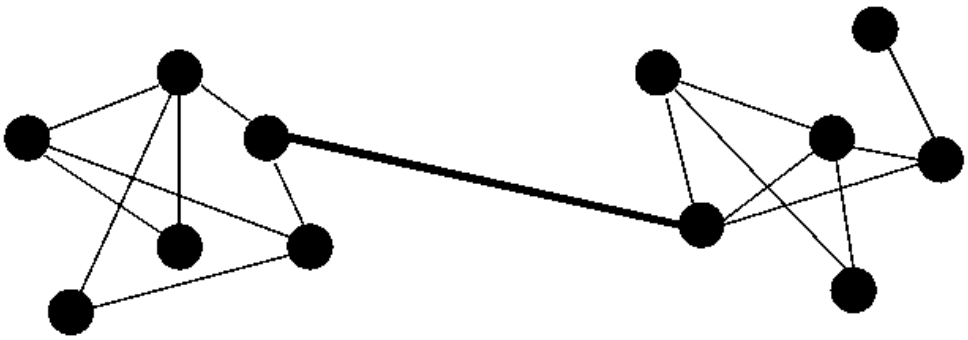

**Figure 5: Edge Betweenness. The thickness of the edge shows the strength of its betweenness centrality. The edge connecting two communities has greater strength than the edge inside any community.**

Girvan and Newman proposed four step simple algorithm based on the idea described above.

1. Compute the edge betweenness for all edges in the graph.
2. Remove the edge with the highest edge betweenness. If there are more than one with same value, pick one at random.
3. Re-compute betweennesses for all edges affected by the removal.
4. Repeat from step 2 until there are no edges in the graph.

Edge betweenness of the graph can be computed using algorithm given by Newman [18]. The algorithm computes the betweenness for graph with total $m$ edges and $n$ vertices in $O(mn)$ time. Since in step 3 we re-compute the edge betweenness the worst case running time for the algorithm becomes $O(m^2n)$. But the step says we only have to re-compute this for all edges affected by the removal. This enables us to run the algorithm much faster than the worst case scenario. This happens usually in the network with high community structure since, in such networks, most of the edges are within the community and only edges exist between communities, making calculations in step 3 faster.

In this algorithm, since each vertex is assigned to exactly one cluster, it doesn't give accurate results for the network with the overlapping communities. Many modifications have been made to this algorithm to recover this very weakness of the algorithm. In the algorithm proposed by Pinney and Westhead [19] membership of vertex can be divided between communities. In that we also have to compute the betweenness of all vertices of the graph (i.e. vertex betweenness introduced by Freeman [19]). Pinney and Westhead claimed that the two end-vertices of an inter-cluster edge have similar betweenness values, since the shortest paths crossing one of them are likely to reach the other one as well through the edge. Therefore, we take the edge with highest betweenness value and remove it only if the ratio of the betweenness values of its end-vertices is between α and 1/α, with α= 0.8. Otherwise, the vertex with highest betweenness along with all its adjacent edges is temporarily removed. Finally, when a subgraph is broken by vertex or edge removal, all temporarily removed vertices and associated edges belonging to that subgraph are simply copied back in each subcomponent. In another modification proposed by Gregory [20] vertices are split among clusters if their betweenness exceeds the highest edge betweenness value. Vertex splitting is done by making a copy of the same vertex in the other community of which the vertex is assumed to be part of too. The edges connecting the vertex to others in network are split between the copies of vertex such that they sum up the actual degree in the original graph. To do that, Gregory introduced a new centrality measure called *split betweenness* defined as number of shortest paths that would run between two parts of a vertex if the vertex were split. The method has a worst-case running time of $O(m^3)$, or $O(n^3)$ on a sparse graph.

### 2.6.3 Simulated Annealing

## 3. FOOD WEB

Here we attempt to acquaint the reader with food-webs, their basics, types and some terminologies. This is just to help imagine, what kind of network, a food-web constitutes and how any community detection algorithm exploits some of its properties. In section 3.2 we explain the two types of graphs (networks) that most of the food-webs usually projects to. We have considered only these two types of networks for evaluating different algorithms.

## 3.1 Basics

Plants and animals play a major role in maintaining the right ecological balance and further support the human life to a great extent. Every living being on earth needs food for survival. Food web shows transfer of food and energy between the species and therefore can be defined as a diagrammatic representation of 'who eats whom' among species in an ecological community. Community ecology has been termed a `mess' [15], but a recent synthesis has cleaned up this mess by realising: "*At the most general level, patterns in the composition and diversity of species - the subject matter of community ecology - are influenced by only four classes of process: selection, drift, speciation, and dispersal. Selection represents deterministic fitness differences among species, drift represents stochastic changes in species abundance, speciation creates new species, and dispersal is the movement of organisms across space.*" [16].

The concept of the food web (referred to as food cycle then) was introduced by the pioneering animal ecologist Charles Elton in 1927. A food web is referred to all the food chains in a community. Food chains always start with plant life and ends up with an animal. The process of photosynthesis enables the plants to produce food which when consumed by humans provides them with energy.

✓ Plants are referred to as *producers* and they use light from the sun to produce food.

✓ Animals are called *consumers***.** They can eat plants and/or other animals.

- Animals only surviving on plants (primary consumer) are referred to as *Herbivores***.**
- Animals that eat other animals in order to survive are *Carnivores***.**

Further, Carnivores that eat herbivores are *secondary* consumers and carnivores that eat other carnivores are *tertiary* consumers.

✓ *Omnivores* represent the group of animals that eat both plants and animals.

An example of the above mentioned is, squirrels and mice that eat both plants as well as insects.

It can also be rightfully observed that animals that eat variety of animal and plants have better chances of survival than the ones which have limited food resources.

✓ *Decomposers* – This is bacteria & fungi and feed on decaying matter.

Each species holds a certain position in the chain which is called the trophic level. For example, owls eat mice, so if a food chain contains an owl and a mouse, the owls will be at a higher level. The question concerning the number of trophic levels in a food chain is determined by number of species present. The only difference between a food web & food chain is that the latter represents only a part of food web connected by feeding links whereas food web shows a more complete diagram of the feeding interlinks.

### 3.1.1 Kinds of food-webs[2]

- *Source web* - one or more node(s). It represents all predators and all food that is eaten by them.
- *Sink web* - one or more node(s), it shows all prey and their food.
- *Community (or connectedness) web* - a group of nodes and all the connections of who eats whom. The one we are primarily interested in.
- *Energy flow web* - quantified stream of energy between nodes and also depicts links between a resource and a consumer.
- *Paleoecological web* - a web that reconstructs ecosystems from the fossil record.
- *Functional web* - emphasizes the functional significance of certain connections having strong interaction strength and greater bearing on community organization, more so than energy flow pathways.

### 3.1.2 Keystone species

Keystone species is said to have strong indirect effects. Nature is said to comprise of both, the big players (usually predators) and the little players (usually prey). Here the biggest players are referred to as the keystone species. The big players are the ones whose addition or subtraction notably affects the existence of other species and hence the diversity of the entire food web. The name has been derived from the supporting center stone which if removed results in collapsing the arch.

Starfish *Pisaster ochraceus* is illustrated as a keystone species in the rocky intertidal communities of North America. It's a predatory starfish that feeds on mussel. When the starfish was removed on experimental basis, the mussel population flourished speedily and ended up covering the rocky shores in such a manner that it was impossible for other species to base themselves. By this example, we can correctly observe that a species can be a keystone species in some communities but not in every community.

## 3.2 Types of food-web used for evaluation

In this study, we have analysed two types of food webs. Plant – pollinator mutualistic food web and Predator – prey food web. When projected to a graph, they form *bipartite* and *directed-weighted* network respectively. Here we describe each of these in detail.

### 3.2.1 Plant-pollinator (bipartite) network

A bipartite network also called *two- mode* network is a graph having exactly two types of vertices that form two disjoint sets such that, edges exist only between the vertices of two sets and not within them. An example of this kind of network is a plant-pollinator network. Pollinators (usually insects) visit plants in search of food. In return they help the plants in carrying pollens from one flower to another [32]. They help each-other in a mutual way and therefore these networks are often called *mutualistic* networks. Figure 6 shows the graphical representation of a plant-pollinator network found in Wetland, United Kingdom. Some of the

vertices and edges are not shown and the names are shortened for the sake of clarity in visualisation.

In this study, we have used three datasets of plant–pollinator network. When projected to graph, it displays similar bipartite structure.

### 3.2.2 Predator – prey (directed and weighted) network

Directed weighted network, as the name suggests, has edges with direction as well as weights. An example of this kind of network is a predator – prey network. It basically shows who eats whom in the entire food chain. The edges are directed with an arrow going from predator to prey. The thickness of the edge is used to display weights. It is used to show the exchange of carbon from prey to predator when the latter consumes the former. This differs from a bipartite network, since, firstly predators and preys do not form two disjoint sets (i.e. a predator can be a prey for another predator) and the edges also exist within (as opposed to between) two types of animal. Figure 7 shows a predator –prey network found in Chesapeake Bay food web [25]. Some of the vertices and edges are not shown for the sake of clarity in visualisation.

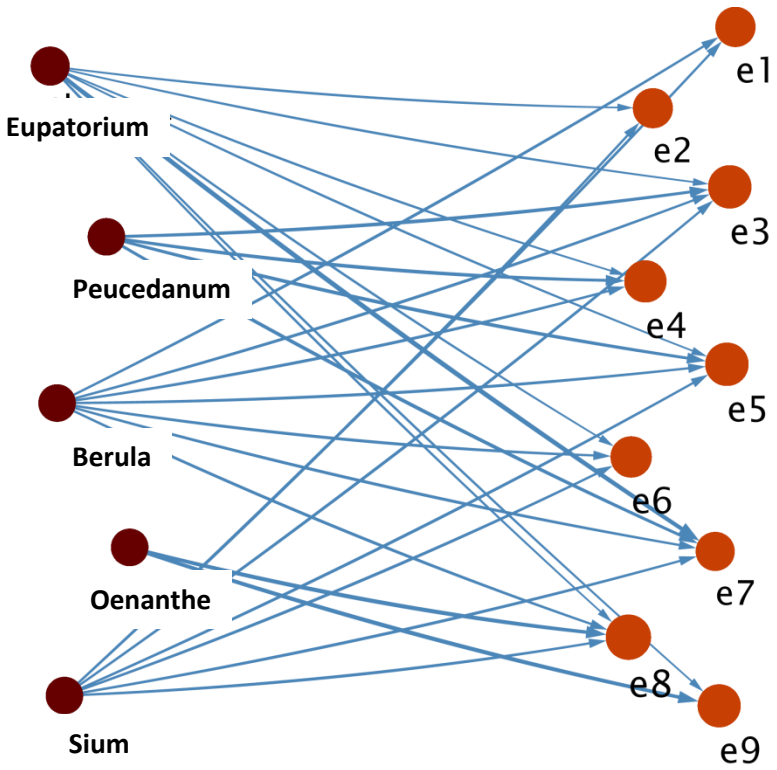


Figure 6 : Wetland's plant-pollinator network displaying bipartite structure. Names are shortened for the sake of clarity

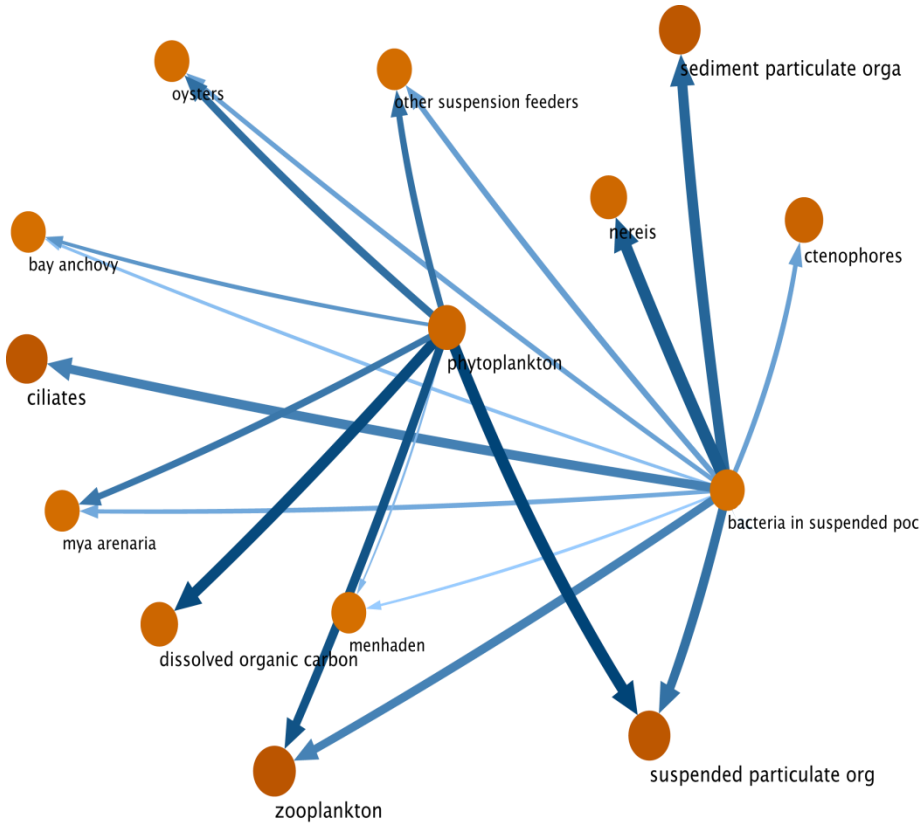


Figure 7 : Chesapeake food web displaying directed-weighted structure

We have used three predator-prey network's datasets in this work. When projected to graph, it displays similar bipartite structure.

# 4. PREVIOUS WORK

"Community Structure in Food Web" has been in sight of researchers since last decade. In this section, we shall look into some previous work done in this area. In particular, we will see the application of two of the three techniques for community detection discussed in section 2.6 on food web.

## 4.1 Girvan & Newman Algorithm on Chesapeake Bay food web

Girvan and Newman, in [4], applied their algorithm described in section 2.6.2 to a food web of marine organisms living in the Chesapeake Bay situated on east coast of the United States. Taxa are represented either at the species level or genus level where some vertices represent larger groups of related species. Edges from one taxa to another are generally directed and display trophic relationships (i.e predator feeding on prey) between them.The algorithm, however, did not consider these directions. Graphical representation of community structure of the Chesapeake Bay food web is shown in Figure 4.

Girvan and Newman applied their algorithm on this database of taxa and extracted two well defined communities of almost equal size. The algorithm couldn't classify few taxa to either community. The results of the test are represented using dendrogram in Figure 8. The division of species shown in Figure 4 almost corresponds division shown in Figure 8. Their results seem to derive the fact that pelagic and benthic organisms in the Chesapeake Bay can be separated into *reasonably* self-contained ecological subsystems [4]. Word reasonably is used since the separation is not perfect in that a small number of benthic organisms find their virtual residence in the pelagia. For example, clams reprensented with vertices 27, 28 and 29 and oyster with vertex 30 in Figure 4 physically reside in the benthos but because of their strong interaction with bacteria (vertices 3, 4 and 5) and ciliates (vertices 7, 8 and 9) in group A they have become a part of the same group. These species are often found to be "**Error! Reference source not found.**" discussed in section 3 and play important role in maintaining diversity of the food web. From the result one can deduce that simple traditional division of taxa as shown in Figure 4 may not be an ideal classification in this case. The algorithm was also applied to number of other food webs with mediocre success.

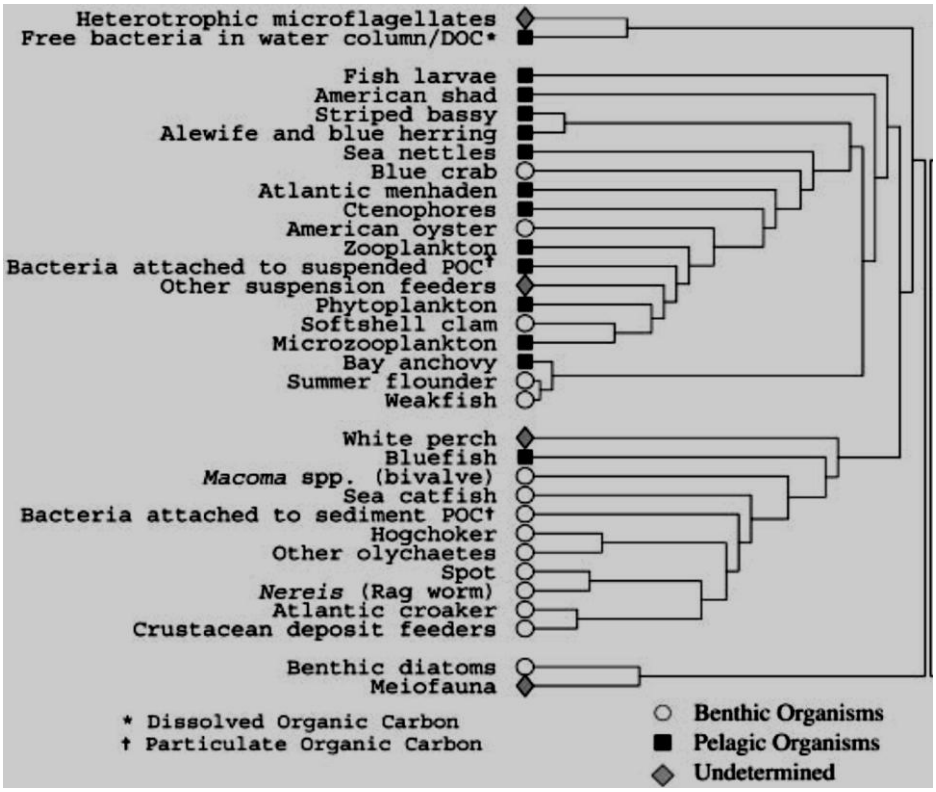


Figure 8: Hierarchical Structure for Chesapeake Bay food web shown using dendrogram. Taken from [4]

## 4.2 Communities in Pollination Networks

J.M. Olesen *et al* [22] applied Simulated Annealing algorithm described in section 2.6.3 to 51 pollination networks in order to find underlying community structure and to measure their *modularity*. Networks consisted of almost 10000 species with 20000 links altogether. Algorithm was used, also to classify species into different roles with respect to their position within and

among modules. The outcomes were (1) modularity index M, which was a measure of the degree to which the network was organized into distinct modules. (2) The significance level of M of the real network by comparing its value to *null model* (see section 2.4.1), (3) the number of modules in a network, and (4) the content of species of each module. The results showed 57% networks (29 out of 51) were significantly modular. Few things they observed for these modular networks were (1) the lower bound on number species was 50 and that on number modules was 5 (2) Modularity index M was independent of network size (3) On an average 60% links were intra module.

Networks were further analysed to identify the role of each species. This was based on vertex's (i.e. specie's) within-module degree $z$ (i.e., number of links to other species within module) and its among-module connectivity $c$, i.e., the level to which the species was linked to modules other than its own). Calculation showed that Species with both a low $z$ and a low $c$ were *specialists* (also called *peripheral species*), i.e., the ones with only a few links that too with species in their module. Species with either a high value of $z$ or $c$ were *generalists*. These included module hubs, i.e., highly connected species linked to many species within their own module (high z, low c), and connectors linking several modules (low z, high c). Species with both a high z and a high c were network hubs or super generalists, acting as both connectors and module hubs. Primarily *module hubs*, i.e., densely connected species within their own module (high $z$, low $c$), and *connectors* linking several modules (low $z$, high $c$). Species with both a high $z$ and a high $c$ were *network hubs*, acting as both connectors as well as module hubs. Results showed (See Figure 9) that 85% of all species were specialists and only 15% were generalists. Out of 15%, 3% were module hubs, 11% were connectors and 1% was devoted to network hubs.

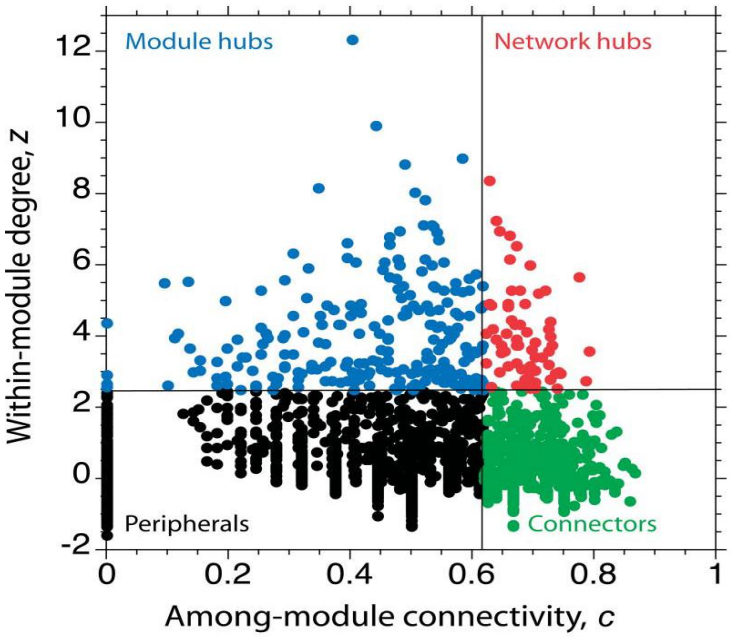


Figure 9: Distribution of pollinator and plant species according to their role in a network.

They also found that networks with less than 50 species were non modular and claimed that this may be due to lack of module detecting power of Simulated Annealing algorithm. This contrasts with the conclusion derived in [21].

# 5. CHOICE OF ALGORITHMS & MEASURES OF TESTING

One has to consider a number of factors while choosing an algorithm for finding communities in any real system. It requires in-depth knowledge and great amount of analysis. In addition, the long lasting trade-off between speed (running time) and accuracy is no exception here. Danon *et at* [27] has presented good analysis on this. While some of the methods invented were aimed at specific field (Girvan and Newman [4] for sociology & biology), most of them were developed to operate on a particular class (es) of graph. Keeping this in mind, we have chosen seven algorithms that can be applied on one of the two types of food webs that are being analysed here. Few of them [28-29] are proven to be best overall in the field [31]. Danon's study [27] showed that the modularity optimization via simulated annealing [21] yields the best results. The rest is chosen because they are able to deal with the special kind of network that a typical food web represents.

Here, while choosing, we have not considered the running time of an algorithm since food-webs, in general, are small in size (Approximately $10^4$ species at max.) and slower algorithms usually perform better [27]. The seven algorithms that are chosen here are broadly divided into two categories i.e. those which work on bipartite (or bipartite-weighted) networks i.e. plant-pollinator network and the ones dealing with directed-weighted network i.e. predator-prey network. Both of these networks are explained in section 3.2. The given description of all the algorithms is not exhaustive. For complete technical details, please refer to the corresponding cited article.

## 5.1 For bipartite graphs

### 5.1.1 Simulated Annealing

Guimera in [21, 30] proposed an algorithm which was based on simulated annealing [35] to find communities in real systems. The method that he proposed in ref. [30] was the first ever proposal to be able to deal with bipartite graph without converting it into pair of uni-partite graphs. Guimera emphasized that the widely used modularity expression (Eq. (5)) was not ideal to detect modules in bipartite network and suggested a new expression for modularity. In that, the vertices are divided into two classes, which they called *actors* and *teams*. Each vertex i in the actor class has degree $t_i$ and vertex a in the team class has degree $m_a$. Modularity is defined by following expression:

$$Q_B = \sum_{c=1}^{n_c} \left[ \frac{\sum_{i \neq j \in C} C_{ij}}{\sum_a m_a (m_a - 1)} - \frac{\sum_{i \neq j \in C} t_i t_j}{(\sum_a m_a)^2} \right]$$

(6)

Simulated Annealing [30] is an optimization technique based on random variables that enables one to find "low cost" configuration without getting trapped in "high cost" local minima. This is based on *computational temperature T* which is defined as follows: At high values of *T*, the system can explore configurations of high cost; while T is low the system only explores low cost regions. Starting at high T and gradually decreasing it, the system stabilizes gradually toward deep minima.

While searching for the communities in the network our primary objective is to maximize its modularity and, therefore, the cost is C = −M, where M is the modularity as defined in equation (5) (See section **2.4.2**). For each values of temperature, we perform a number of random updates and accept them with probability.

$$P = \begin{cases} 1 & if\ C_f \leq C_i \\ exp\left(-\frac{(C_f - C_i)}{T}\right) & if\ C_f > C_i \end{cases}$$

(7)

Where $C_i$ is the cost prior to the update, $C_f$ is the cost after the update.

We define two types of movements. 1) Node Movement: For each value of temperature total of $n_i = f\mathcal{N}$[2] vertices will be moved from one module to another, where $\mathcal{N}$ is the total number of vertices in the network. 2) Collective Movement: $n_c = f\mathcal{N}$ which involve mergers of two communities or splitting a community. Value of $f$ is typically taken as 1. After having evaluated the movements at a certain *T*, temperature is cooled down until it reaches the running value for the global optimization. Splitting of community is carried out in several ways. Collective movements reduce the risk of getting trapped in "high cost" local minima.

This algorithm when tested on the small random network[11], performed better over Girvan and Newman's algorithm [4].In practice, it can be used for small graphs consisting of up to about$10^4$ vertices [10].

[1]The Simulated Annealing algorithm [21] was tested on network comprised on 128 nodes, divided into 4 modules, each consisting of 32 nodes. The obtained results outperformed Girvan and Newman's algorithm significantly and also showed that it is able to identify community in a network with nodes having as many as 50% of their connection outside their own module.

### 5.1.2 BRIM

As pointed out by Guimera, M.J. Barber [34] also thought that the original expression of modularity is not accurate for finding communities in bipartite network. He proposed a different expression to calculate modularity suitable for bipartite network.

Let *p* be number of red and *q* be number of blue vertices that forms overall network. The degree of red vertex *i* is given by $K_i$ and of blue vertex *j* is given by $D_j$. The adjacency matrix **A** of the graph has all zero elements for same pair of vertices. From this fact, Barber concluded that its equivalent null model (section 2.4.1) would have same structure i.e. zero likelihood of edges existing between vertices with same color. So if **P** is null model matrix of **A,** every element $p_{ij}$ where *i* and *j* shares the same color will be zero. The null model matrix **P** is defined as follows:

$$P = \begin{bmatrix} O_{p\times p} & \tilde{P}_{p\times q} \\ \tilde{P}^T{}_{p\times q} & O_{q\times q} \end{bmatrix} \tag{8}$$

where $O_{i\times j}$ are all zero elements. $\tilde{P}_{i\times j} = K_i D_j / m$ as in the null model of modularity. The modularity matrix **B** has the same block off-diagonal diagram as **P.** It can be computed with following expression. **B** = **A – P.**

Barber then proposed a community finding algorithm using this definition of bipartite modularity. It is called Bipartite Recursively Induced Modules (BRIM). It is based on the idea that, once partition of one of the two classes of vertices is known, it is easy to compute the partition with maximum modularity involving the other class of vertices. To say, if it starts from any partition consisting of few clusters, let's assume, red vertices, it recovers the partition for blue vertices. That partition will be used as an input to find partition again in red vertices. Process continues until its modularity value starts declining.

### 5.1.3 MAGA

Zhan *et al* [40] came up with an idea that any uni-partite or directed network can be represented as a bipartite network and further claimed that, finding communities in uni-partite and directed network could be transformed into finding communities in bipartite network. The proposed transformation [40] of uni-partite into bipartite is as follows:

Each node $i$ of uni-partite network is represented by two nodes $A_i$ and $B_i$. Also each edge $i - j$ is converted into two edges $A_i$ - $B_j$ and $A_j$ -$B_i$. The new transformed bipartite network will have $2N$ vertices with $2M$ edges. Figure 10 shows the illustration. They went on to define, for the transformed network, a new expression for modularity based on Barber's modularity expression (Eq. (8)). Barber's matrix form expression can be represented as follows:

$$Q_B = \frac{1}{M}\sum_{i=1}^{p}\sum_{j=p+1}^{N}\left[\tilde{A}_{i,j} - \frac{k_i k_j}{M}\right]\delta(C_i - C_j)$$

(9)

Where vertices $1$ to $p$ constitute class-1 set of vertices and $p+1$ to $N$ constitute class-2. Adjacency matrix $A_b$ is

$$A_b = \begin{bmatrix} O_{p\times p} & \tilde{A}_{p\times q} \\ A^T{}_{p\times q} & O_{q\times q} \end{bmatrix}$$

(10)

Rest of the term are same as defined in section 2.4.2. Based on this equation they define new bipartite modularity in next three equations:

$$Q_B = \frac{1}{M}\sum_{i=1}^{p}\sum_{j=N+1}^{2N}\left[\tilde{A}_{i,j} - \frac{k_i k_j}{2M}\right]\delta(C_i - C_j)$$

$$= \frac{1}{M}\sum_{i=1}^{p}\sum_{j'=1}^{N}\left[\tilde{A}_{i,N+j'} - \frac{k_i k_{N+j'}}{2M}\right]\delta(C_i - C_{N+j'})$$

$$= \frac{1}{M}\sum_{i=1}^{p}\sum_{j=1}^{N}\left[A_{i,j} - \frac{k_i k_j}{2M}\right]\delta(C_i - C_j) = Q$$

The first expression is derived from equation 10 considering the fact that bipartite transformation has $2N$ nodes and $2M$ edges. Second equation is standard change of indices and the last one is obtain by converting $A_b$ to $A$ (by considering only right upper matrix of $A_b$). In doing this they assumed that node $A_i$ and $B_i$ should have the same degree and lie in the same community.

They also demonstrated transformation of directed network into bipartite one and defined identical expression for its bipartite. Further, they proposed an algorithm using this definition of bipartite modularity to detect communities in bipartite network and called it **M**odified **A**daptive **G**enetic **A**lgorithm (MAGA) which was based on the **M**utation-**O**nly **G**enetic **A**lgorithm (MOGA).

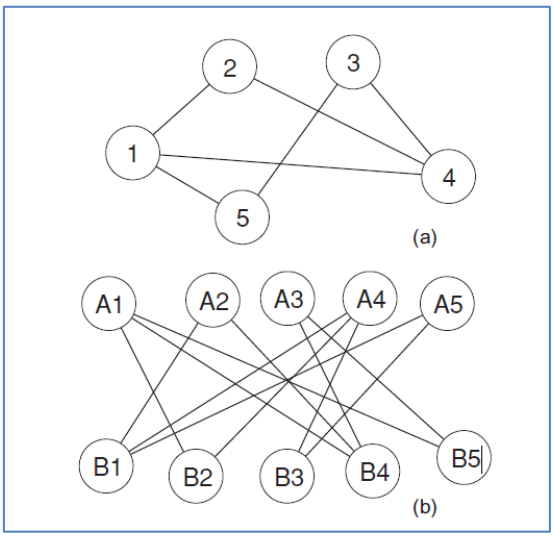


Figure 10: Transformation of uni-partite graph into bipartite. Adapted from [40].

### 5.1.4 COPRA

Gregory [33] extended an algorithm proposed by Raghavan *et al* [36] to find communities in network through network propagation and called it **C**ommunity **O**verlap **PR**opagation **A**lgorithm (COPRA). He modified Raghavan's algorithm called RAK so that it can be applied to network with overlapping communities (also called *covers*). The basic RAK algorithm [36] is as follows:

1. Initialize every vertex with separate label.
2. Updates label for each vertex, replacing it with the label used by highest number of neighbour vertices. In case of tie, choose the vertex randomly and use its label. Repeat this step until each vertex's label become associated with other vertices of the same community.
3. Group of vertices with the same label are added to a community.

After several iterations, the algorithm tends to terminate, when all the vertices are gradually assigned a label used by a maximum number of neighbour vertices. Gregory thought if communities are of overlapping nature each vertex label should be allowed to contain more than one community identifier. He labelled each vertex with pair $(c, b)$ where $c$ and $b$ are *community identifier* and *belonging co-efficient* (indicates the strength of vertex $x$ membership of community in way that it sums to 1) respectively. During each propagation step label for each vertex $x$ is updated to – union of its neighbour's labels i.e. $C$, the sum of their belonging coefficient. This can be expressed as follows:

$$b_t(c,x) = \frac{\sum_{y \in N(x)} b_{t-1}(c,y)}{|N(x)|}$$

(9)

The term on the left hand side maps vertex $x$ and $c$ to its belonging coefficient in iteration $t$. At the end of this step, all the pairs whose belonging co-efficient is smaller than some threshold are removed. The threshold is set as $1/v$ where $v$ is the parameter to the algorithm representing the maximum number of communities that a vertex can be part of. In this process, it could be possible that more than one pair of vertices has the same belonging co-efficient less than the set threshold. In that case a pair is randomly chosen making the algorithm non-deterministic. With value of $v = 2$ the algorithm gives result as depicted in Figure 11.

Gregory also extended this algorithm to work on bipartite network. In that, for each iteration, vertex label is propagated between two classes of vertices i.e. First labels are propagated from class-1 set of vertices to class-2 and in the same way resultant vertices of class-2 are propagated back to class-1. We have used the bipartite version of COPRA in our evaluation. Compared to other algorithms [38-39] COPRA operates very fast on a large and dense network.

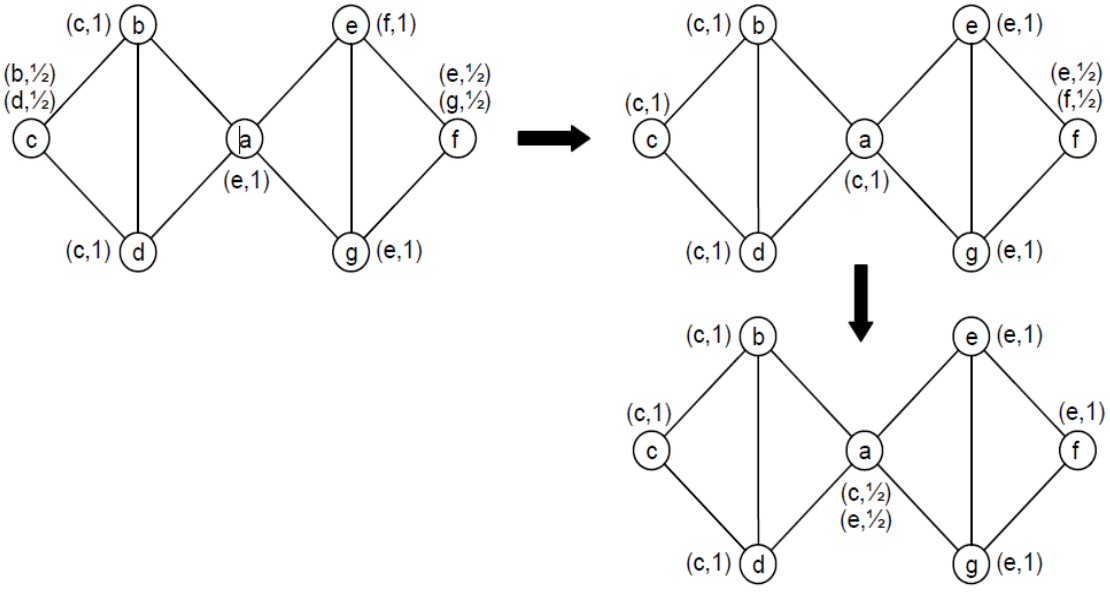


Figure 11: Label propagation with $v = 2$

## 5.2 For directed & weighted graphs

### 5.2.1 Infomap[2]

Infomap [28] is based on the idea of describing a graph by using information essentially less than that encoded in the full adjacency matrix. It aims to compress the required information to describe the process of information diffusion across the graph. It uses random walk as an alternative of information diffusion. The modular structure is represented by two-level description based on Huffman coding. It assigns unique names to structures marked as important and the vertices within it and makes sure that vertex names are reused between different structures. An example of this could be street names in different cities. City represents a structure and within that, street names are reused, as long as the pair of city–street name is unique. The random walker is expected to spend more time within this structure. Therefore, these structures will form communities. One would say that the problem of finding community structure becomes a problem in finding the partition that produces smallest description length of potentially infinite random walk. The description would contain the Shannon entropy of the random walk within and also between clusters. When random walker jumps from one cluster to another, one needs to inform the decoder of the jump, by mentioning the code word of the new cluster in the description. It is intuitive for the graph with almost disjoint clusters that these jumps will be rare. Code words of the clusters are not going to be repeated more often in the description of random walk since we are using the map with cluster as regions; the fact that leads to saving in the description. On contrary, while the graph doesn't reflect clear or partial modular structure, the jumps of the random walker becomes more frequent and the use of two level descriptions become nearly futile.

Infomap can be applied to weighted & un-weighted, normal (i.e. undirected) as well directed graphs. We have used only directed version both with and without weights in our analysis.

### 5.2.2 Louvain

Blondel *et al* [29] proposed a fast algorithm called *Louvain* which is able to deal with weighted graph. The algorithm also gives qualitative results in terms of modularity[3]. Blondel *et al* extended the definition of modularity to handle weights which is as follows:

$$Q_w = \left(\frac{1}{2w}\right)\sum_{i,j}\left(W_{i,j} - \frac{s_i s_j}{2W}\right)\delta(C_i, C_j) \tag{10}$$

The algorithm starts by placing all the vertices into separate clusters. In the first phase, all vertices of the graph are put in different communities. The first step consists of a sequential sweep over all vertices and for each node *i,* it considers the neighbours *j* of *i* and calculates the gain in weighted modularity (Eq. (6)) that would take place by removing *i* from its community

[2] The description is adopted from [10]
[3] When tested on Belgian mobile phone network involving 2.6 million users [29]

and by placing it into *j*'s community. The gain has to be positive. After repeating this procedure for all the nodes until there are no improvements, it brings out first level partition. In the second phase, communities found in the first phase become new nodes (call it super nodes). The weights are aggregated and these super nodes are connected if there exists at least an edge between their vertices. The algorithm is repeated, on the intermediate value received at the end of second phase until there are no more changes and a maximum value of modularity is achieved. Figure 10 illustrates the two phase algorithm.

This method suffers from storage crisis if the network is huge in size but on a small network like food web, storage is not an issue.

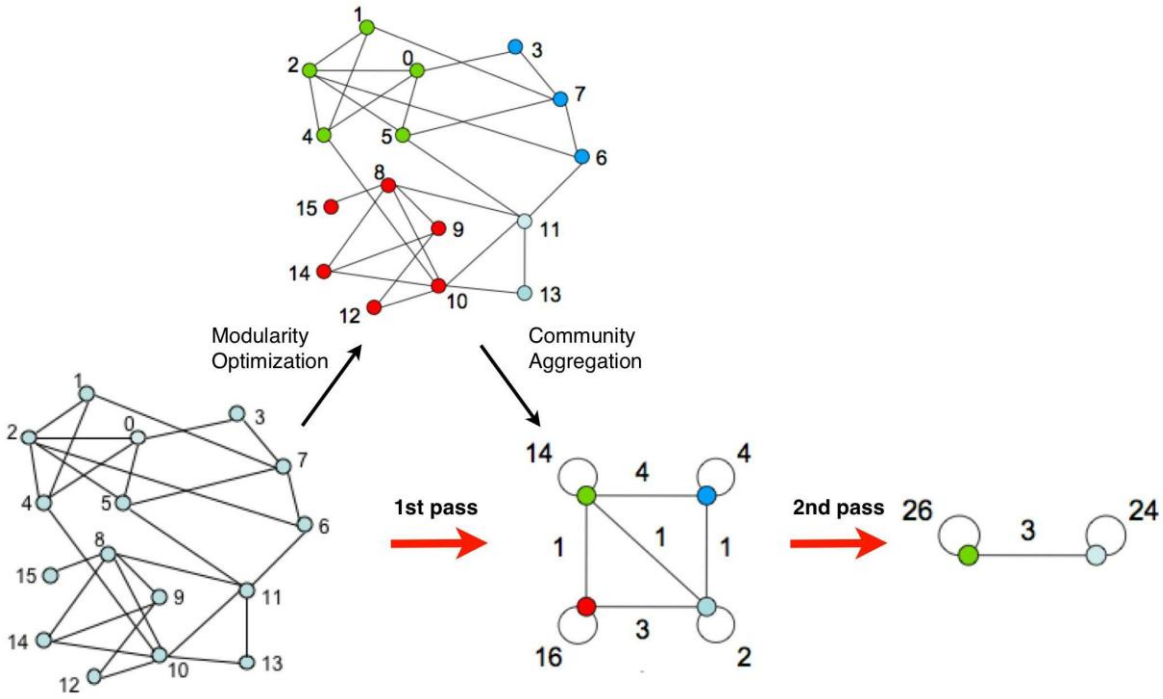


**Figure 12: Modularity maximization method proposed by Bloden *et al*. The figure shows two phase approach towards finding modules in graph. Refer to the description for explanation. Image taken from [29].**

### 5.2.3 OSLOM

OSLOM [37] is based on local optimization of a fitness function that expresses statistical significance of discovered communities with respect to random fluctuations approximated by tools of Extreme and Order Statistics. The fact that makes OSLOM special is that, it can be applied to a range of networks accounting for directed and weighted edge, overlapping community as well as hierarchical community dynamics. We have described the algorithm in abstract fashion without going into in-depth technicalities, since OSLOM accommodates three sub algorithms and we feel providing details of each component would be of very little importance. The algorithm is as follows:

1. Search for significant cluster until convergence.
2. Analyse the cluster received from step 1 then try to detect their internal structure or possible union thereof.
3. Find out hierarchical structure of the clusters.

Once the set of minimal clusters are found in step 1, the network is reformed wherein the new nodes represent the clusters found. These new nodes are called super-vertices and an edge exists between them, if the vertices inside them are connected. After having built the super network such as this, the method can be applied repetitively to each super vertex in order to find further hierarchical structure until it produces no clusters. We are only interested in the clusters having highest modular structure.

## 5.3 Testing Algorithms

How do we know an algorithm is better than rest of the others? How do we know which one is the best or an ideal one? In the field of community detection, indeed, the issue of testing has received relatively less attention. As a result, it is still impossible to decide the most reliable algorithm for any specific system. In this study, we aim to find an algorithm which would give although not completely reliable, but the most accurate results on any real food web.

While testing an algorithm's performance on any artificial or real system (i.e. food web), one has to ask whether the underlying community structure is known a priory? The answer to this question in yes or no, gives us two mostly used methods for testing an algorithm.

When we know the community structure, the best way to test the performance is to compare the real communities with the communities identified by the algorithm. Again there are multiple criterions for measuring similarity of two communities or partition. One of them is *Normalized Mutual information* which was first discovered by Fred *et al* [26] and later adopted by Danon *et al* [26] in his study. The measure is derived from a confusion matrix $M$ which has as many rows as number of real communities and as many columns as number of identified communities by an algorithm. Each element $m_{i,j}$ of the matrix represents the number of nodes incorrectly classified the algorithm i.e. nodes classified in community $i$ by the given algorithm, when they belong to real community $j$. When the algorithm doesn't deliver correct number of communities, the matrix takes the rectangular shape.

$$NMI = \frac{2\sum i\sum j \, \mathrm{m}_{ij} \log(\mathrm{nm}_{ij} \,/\, (\sum_j \mathrm{m}_i \times (\sum_i \mathrm{m}_j))}{\sum i \, \mathrm{m}_i \log(\left(\sum j \frac{\mathrm{mi}}{\mathrm{n}}\right) + \sum j \, \mathrm{m}_j \log(\left(\sum i \frac{\mathrm{m}_j}{\mathrm{n}}\right)} \tag{11}$$

The value of NMI ranges from 0 to 1. It is 1, if the communities are identical, 0 if they are independent or fractional, if there is some similarity.

When we don't know the underlying community structure of the system, the only way to predict it, is through quality function, as it is the case in food web networks. Modularity is the most popular and widely accepted quality function so far. Therefore, in this study we will have to rely on it to measure the strength module detecting power of all the algorithms. As pointed out in section 2.4.2, modularity optimization suffers from a resolution limit that often hampers its module detecting capacity; especially this drawback of modularity gives misleading results in system where communities are substantially smaller with respect to the network as a whole. This may happen even if system has well defined communities like cliques. The reason behind this shortcoming of modularity lies within its definition. If you recollect, the value of modularity depends largely on underlying *null model* (see section 2.4.1) which assumes that every vertex of graph can directly interact with every other vertex, since it randomly distributes all edges between pair of vertices. In reality, this is very unlikely or not possible. For example, how likely it is for person living in London, United Kingdom to know or interact with person living in Antarctica! Ideally each vertex should be able to interact with all the vertices in its vicinity. However, the food webs we have used in this study are considerably smaller in size as compared to other real world system and thus measured value of modularity in those shall deliver expected results.

# 6. METHOD

The algorithms described in section 5.1 and 5.2 were divided into two categories as mentioned earlier. Algorithms falling in first category (i.e. for bipartite networks) were initially applied on three datasets namely Wetland, Heathland and Ashton Court. All three datasets were collected by J Memmott [32] in 1999. Places are located in south-west England.

In the field of community detection, the common approach taken for detecting community in bipartite network is to split the network into pair of uni-partite networks i.e. construct a uni-partite projection one part (class containing majority of vertices) and detect community in them. For example, in the scientist – publication network, where scientists are originally connected to their corresponding publication, the uni-partite projection can be constructed by joining two scientists who have co-authored the publication. Then we can apply any algorithm applicable to uni-partite network to detect communities in it. This gives more options for algorithms because most of them are designed to work on uni-partite network.

The algorithms chosen in the first category, however, are capable of dealing with original bipartite networks and finding modules in them. Therefore, firstly these algorithms are applied onto bipartite plant-pollinator networks. The various equations used for calculating modularity are provided in the corresponding algorithm description. Then we constructed uni-partite projections of each of them using the concept mentioned above for scientist-publication network i.e. two pollinators are connected if they pollinates same plants. And this time, we applied the same algorithms onto the network of pollinators. Each algorithm except COPRA was run 10 times on the same dataset and the highest value of modularity was noted. Due to the large fluctuation in results delivered by COPRA, it was considered for running it 100 times. In addition to modularity, we also took note of number of communities found in each network for

each algorithm. This extra constraint was undertaken to check the consistency of algorithm's performance.

Algorithms falling in second category (i.e. for directed- weighted) were initially applied on three datasets namely Chesapeake Bay Masohaline Net [25], Florida Bay [41] and Mangrove Estuary, dry season[4]. Weights in these networks displays exchange flow of carbon from donor to recipient[4] i.e. from prey to predator. There are other ways of describing these weights as well. For example, Krause *et al* [13] in their study considered interaction strength as weights. Weight could also represent interaction frequency of animals.

These three algorithms were also applied on the un-weighted version of the same datasets. This was done to check the changes in modularity values. As with bipartite algorithms, each algorithm was run 10 times and the highest modularity values along with number of communities found were taken. For each algorithms, we have used same modularity equation i.e. Eq. 5.

# 7. RESULTS & COMPARISON

As mentioned in the previous section, we have applied algorithms in each category in two different ways. The following two sections summarise the results we got for each algorithms in two modes of execution.

## 7.1 Bipartite v/s Uni-partite

Each algorithm in this category produces good quality solution overall for all three datasets undertaken. Out of four algorithms evaluated in this phase, COPRA and MAGA gives highly modular results. MAGA showed consistency in the each run i.e. modularity value didn't fall down or went up by substantial margin. Also the number of communities found were unfluctuating in nature on all three networks.

In comparison with MAGA, COPRA delivered better values in terms of modularity. However, we observed that number of communities calculated by this method ranges from two to six. This happens primarily because of non-deterministic nature of algorithm as pointed out in section 5.1.4. In this worth noting that COPRA relies on the original expression of modularity. So if the clusters found are bipartite in nature, COPRA gives two values of modularity corresponding to each set (type) of vertices in found communities. We have only noted modularity of pollinators' communities.

The most consistent results were delivered by Simulated Annealing. Due to the stochastic nature of this algorithm, the algorithm is expected to yield different results in each run. However, since we used same seed for random number generation in all runs, the resultant

[4] Formal reference not found. Dataset were downloaded from URL < http://vlado.fmf.uni-lj.si/pub/networks/data/bio/foodweb/foodweb.htm >

communities were always constant in size and contents. BRIM delivered comparatively poor results[5].

These values are summarised in Table 1 and the column chart above it presents the graphical view of it.

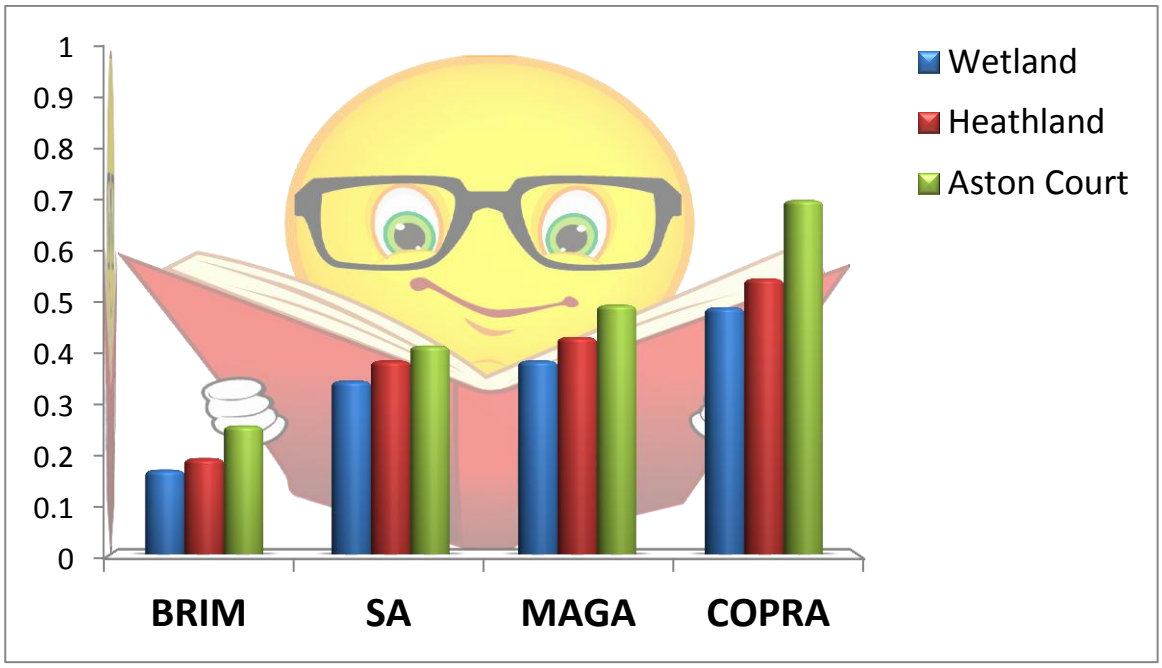


**Table 1: Comparisons of 4 algorithms in terms of Modularity (Q) and Number of Communities found (M). Algorithms were applied on original bipartite network.**

| | SA | | BRIM | | MAGA | | COPRA | |
|---|---|---|---|---|---|---|---|---|
| | M | Q | M | Q | M | Q | M | Q |
| *WETLAND* | 8 | 0.345 | 4 | 0.168 | 6 | 0.385 | 3 | 0.491 |
| *HEATHLAND* | 7 | 0.385 | 5 | 0.191 | 4 | 0.432 | 2 | 0.548 |
| *ASHTON COURT* | 5 | 0.415 | 5 | 0.256 | 4 | 0.496 | 2 | 0.704 |

When the bipartite networks are projected to 1-mode uni-partite network, results showed some changes. Once again BRIM delivered figures, that poor in modularity. Rest of the algorithms also showed significant decline in modularity values, except COPRA, which gives essentially the same results. Even number of communities were different in this case.

These values are summarised in Table 2 and the column chart above it, presents the graphical view of it.

[5] We have derived these values using the executable code for MAGA which also incorporates BRIM algorithms as addition functionality.

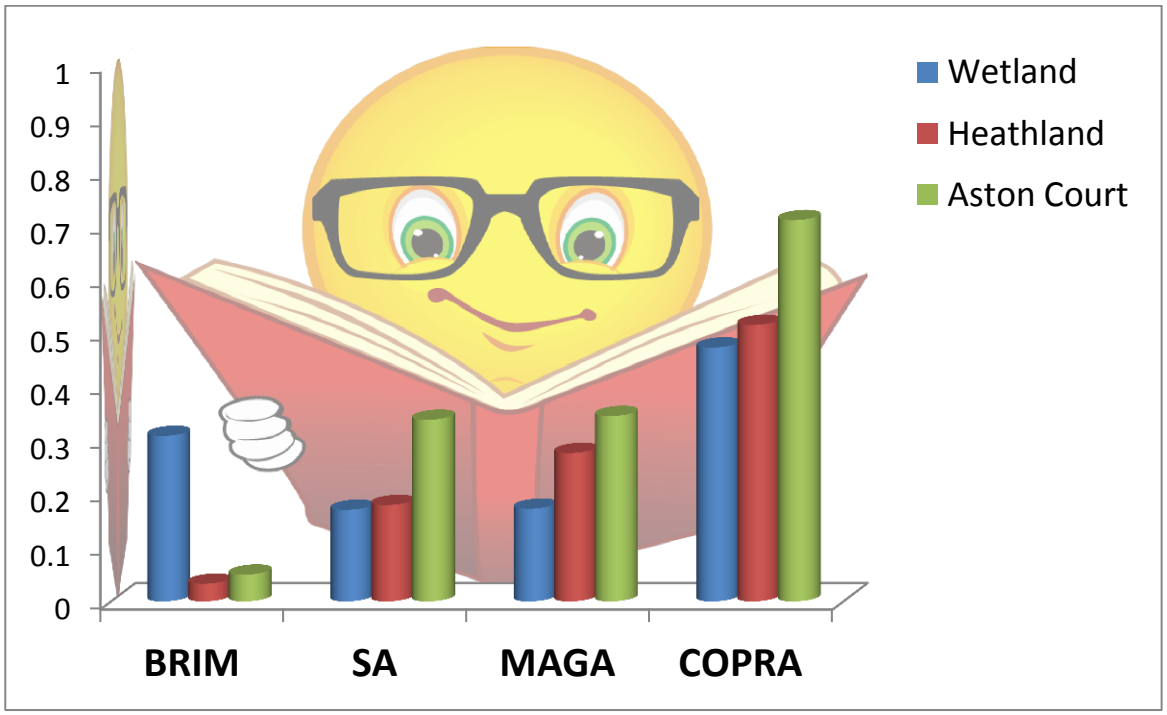


Table 2: Comparisons of 4 algorithms in terms of Modularity (Q) and Number of Communities found (M). Algorithms were applied on one mode projection of original bipartite networks.

| | SA | | BRIM | | MAGA | | COPRA | |
|---|---|---|---|---|---|---|---|---|
| | M | Q | M | Q | M | Q | M | Q |
| *WETLAND* | 3 | 0.171 | 6 | 0.309 | 3 | 0.173 | 2 | 0.474 |
| *HEATHLAND* | 2 | 0.18 | 5 | 0.033 | 3 | 0.277 | 2 | 0.516 |
| *ASHTON COURT* | 3 | 0.339 | 6 | 0.05 | 3 | 0.346 | 2 | 0.711 |

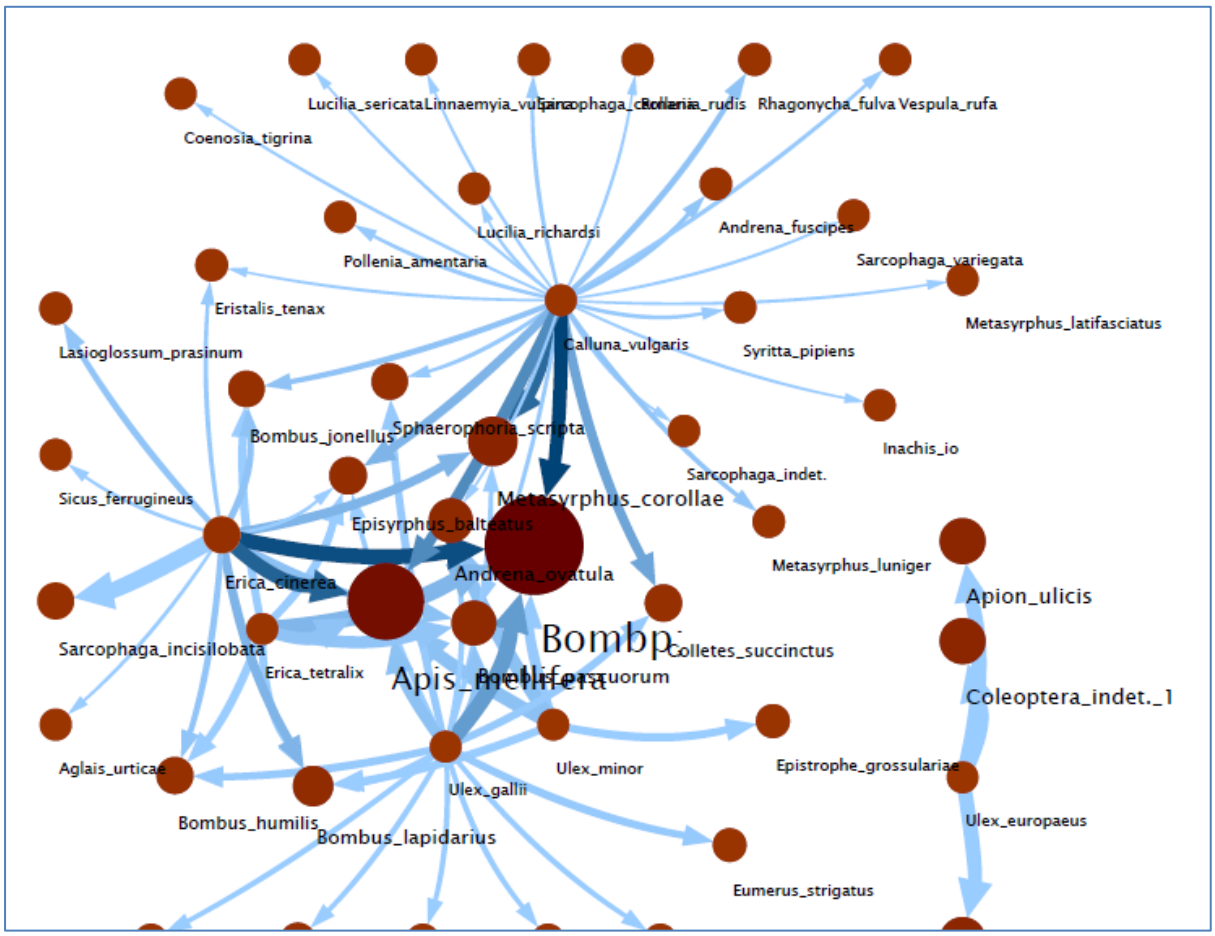


Figure 13: Communities structure found in HEATHLAND using MAGA.

## 7.2 Weighted v/s Un-weighted

Results found in this category were bit shocking. None out of three algorithms that were employed gives modular results. The highest modularity values obtain in 0.339 by OSLOM in Mangrove Bay food web.

In the previous work done by Krause et al [13] on Chesapeake Bay network, two distinct communities of benthic and pelagic species were found (See section 2.5) using KliqueFinder. But in that the dataset was consist of 45 species. Also they didn't consider direction. In addition the interaction strength were used as weights between edges as opposed to carbon flows. That's why it is hard to say the chances of existence of these well-defined communities in Chesapeake bay. Modularity values are summarised in Table 3 and the column chart above it, presents the graphical view of it.

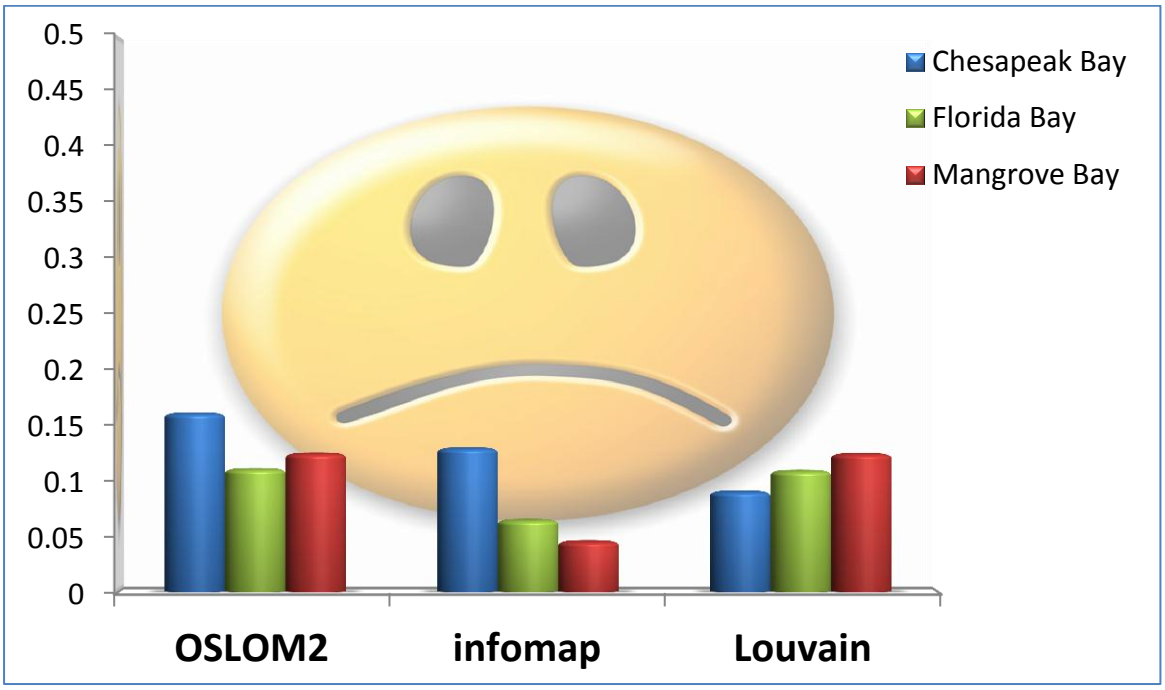


Table 3: Comparisons of 3 algorithms in terms of Modularity (Q) and Number of Communities found (M). Algorithms were applied on one mode projection of original directed & weighted networks

| | OSLOM | | Infomap | | Louvain | |
|---|---|---|---|---|---|---|
| | M | Q | M | Q | M | Q |
| *Chesapeak Bay* | 7 | 0.171 | 4 | 0.129 | 10 | 0.090 |
| *Florida Bay* | 6 | 0.180 | 2 | 0.065 | 8 | 0.109 |
| *Mangrove Bay* | 4 | 0.339 | 6 | 0.046 | 5 | 0.124 |

It was thought that the weights placed on the edges could be misleading for algorithms, because weights are used to refer to different characteristics of interaction between species and moreover it could be very hard for any community detection algorithm to conceptualize the meaning behind these weights. (Algorithms cannot think!!).

Hence we removed the weights and considered only directions while applying the same algorithms for the second time. Unfortunately, no improvements were observed even in the un-weighted version of all three algorithms. Modularity values are summarised in Table 3 and the column chart above it, presents the graphical view of it.

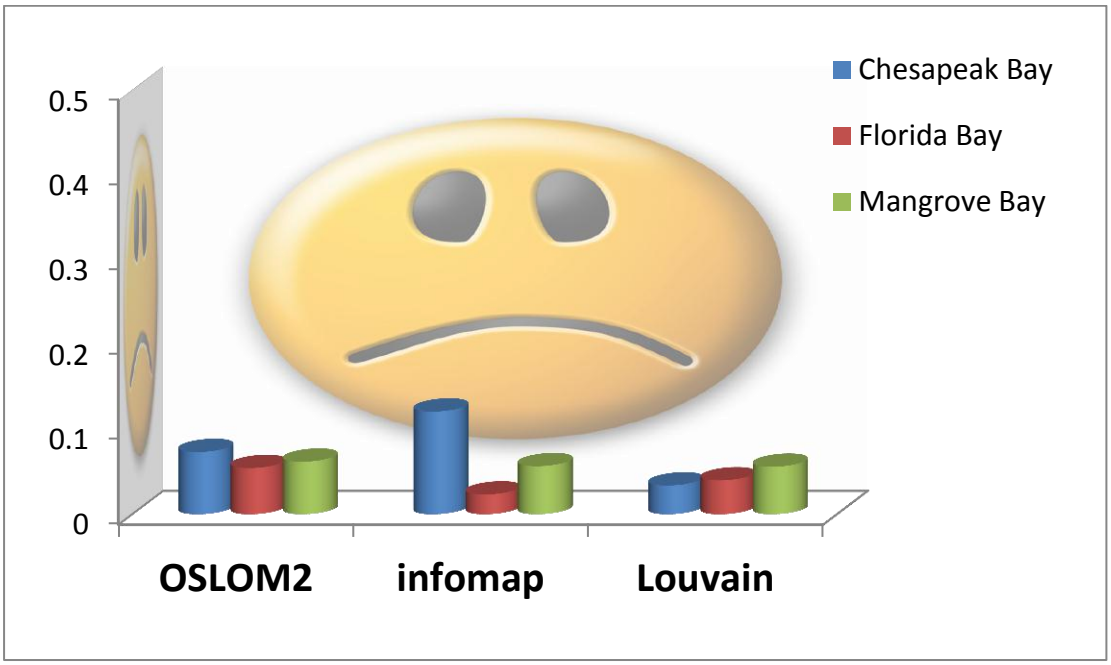


Table 4: Comparisons of 3 algorithms in terms of Modularity (Q) and Number of Communities found (M). Algorithms were applied on one mode projection of original directed & un-weighted networks

| | OSLOM | | Infomap | | Louvain | |
|---|---|---|---|---|---|---|
| | M | Q | M | Q | M | Q |
| *Chesapeak Bay* | 4 | 0.073 | 10 | 0.121 | 7 | 0.034 |
| *Florida Bay* | 2 | 0.055 | 8 | 0.023 | 6 | 0.040 |
| *Mangrove Bay* | 6 | 0.062 | 5 | 0.057 | 4 | 0.056 |

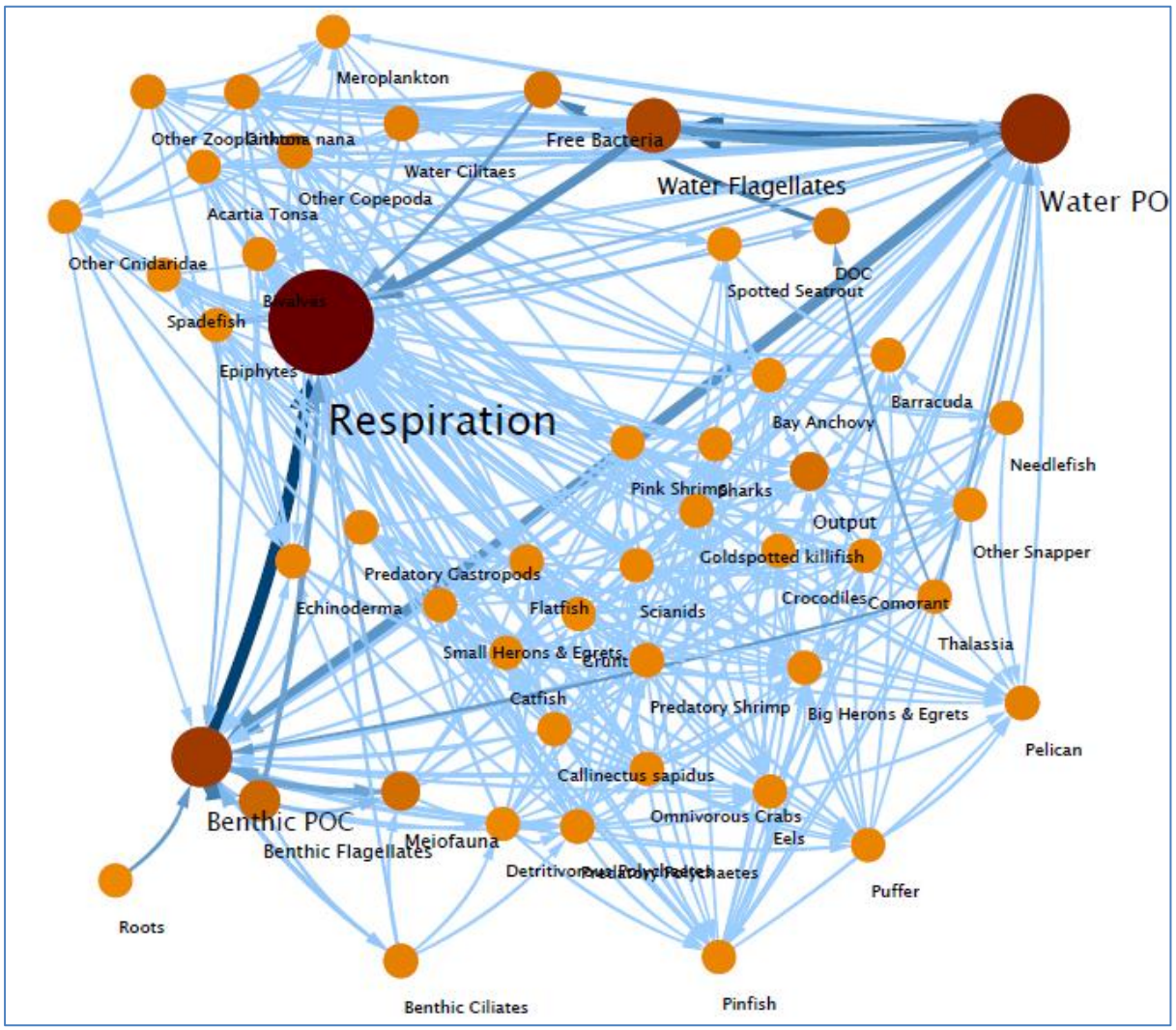


Figure 14: Community Stucture found in Florida Bay food web using infomap

## 8. SUMMARY & CONCLUSION

In this review, we primarily focused on the technique of community detection with a specific interest shown towards food webs. We understood the basics of community detection followed by their real life examples. We also talked about food web and some of its basic terminology. Community detection algorithms are the techniques to find inherent communities in a graph representing a real or artificial network. Out of large number of existing algorithms for community detection in real-world network, we carefully selected seven algorithms. We applied four out seven on different datasets of plant-pollinator networks and rest three on predator prey food webs.

In plant-pollinator networks, COPRA delived highest results in terms of modularity, however, the results obtained in different runs were fluctuating. MAGA and Simulated Annealing also delived good results. These results strongly justified the existence of community in plant-pollinator networks, althought it's still not possible to draw exact boundaries between identified community.

In predator-prey network, the values obtained for modularity is very low for all three algorithms. However, these values are not sufficient enough in order to show that communities exists in this network. This could happen because the weights put on each edges indicate of something which is misleading for these algorithms. But even when weights were taken off, none of the algorithms showed improvement in the results. This could be because these networks are genuinely not composed of communities or because of lack of module detecting power of algorithms. It might also be possible that some of the algorithms not considered in this study,performs better on the unweighted network. These algorithms were chosen for their capability of handling direction as well as weights..

## 9. FUTURE WORK

Out of all quality functions available, in this study, we have chosen modularity as a sole criterion to assess the quality of community detection algorithm. Since food webs are network with unknown community structure one cannot use *Normalize Mutual Information* as a criterion. A natural extension to this work could be to evaluate these networks using other quality functions. Apart from modularity, literature provides many other quality functions. Example incluldes *performance* that counts number of correctly interpreted pair of vertices. By correctly interpreted means, whether pair of vertices belongs to the same community and connected by an edge or belongs to different community and not connected by an edge. One more example of quality function is *Coverage*, which is the ratio of intra-module edges and total number of edges. The interest of use of such measure can shed additional light on the present evaluation.

## 10. REFERENCES

[1] R. Albert, H. Jeong, A.-L. Barabási, Internet: Diameter of the world-wide web, *Nature* 401 (1999) 130 - 131.

[2] S. Wasserman, K. Faust, Social Network Analysis, Cambridge University Press, Cambridge, UK, 1994.

[3] R.J. Williams, N.D. Martinez, Simple rules yield complex food webs, *Nature* 404 (2000) 180 – 183.

[4] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, *Proc. Natl. Acad. Sci*. USA 99 (12) (2002) 7821 – 7826.

[5] K.P. Reddy, M. Kitsuregawa, P. Sreekanth, S.S. Rao, A graph based approach to extract a neighborhood customer community for collaborative filtering, in: DNIS '02: Proceedings of the Second International Workshop on Databases in Networked Information Systems, *Springer-Verlag*, London, UK, (2002) 188 – 200.

[6] R. Agrawal, H.V. Jagadish, Algorithms for searching massive graphs, *Knowl. Data Eng*. 6 (2) (1994) 225 – 238.

[7] M. Steenstrup, Cluster-Based Networks, Addison Wesley, Reading, USA, 2001, (Chapter 4).

[8] S.A. Rice, The identification of blocs in small political bodies, *Am. Polit. Sci. Rev*. 21 (1927) 619 – 627.

[9] R.S. Weiss, E. Jacobson, A method for the analysis of the structure of complex organizations, *Am. Sociol. Rev.* 20 (1955) 661 – 668.

[10] S. Fortunato,Community structure in graphs,*Physics Reports*.486(3-5) (2010) 75 – 174.

[11] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, *Phys. Rev. E* 69 (2) (2004) 026113.

[12] W.W. Zachary, An information flow model for conflict and fission in small groups, *J. Anthropol. Res.* 33 (1977) 452 – 473.

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

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

[15] J. H. Lawton, Are there general laws in ecology? *Oikos* (1999)177 – 192.

[16] M. Vellend, Conceptual synthesis in community ecology. *Q. Rev. Biol.,* 85(2) (2010)183 – 206.

[17] L.C. Freeman, A set of measures of centrality based on betweenness, *Sociometry* 40 (1977) 35 – 41.

[18] M.E.J. Newman, Fast algorithm for detecting community structure in networks, *Phys. Rev. E* 69 (6) (2004) 066133.

[19] J.W. Pinney, D.R. Westhead, Betweenness-based decomposition methods for social and biological networks, in: Interdisciplinary Statistics and Bioinformatics, Leeds University Press, Leeds, UK, (2006) 87 – 90.

[20] S. Gregory, An algorithm to find overlapping community structure in networks, in: Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD 2007, Springer-Verlag, Berlin, Germany, (2007) 91 – 102.

[21] R. Guimerà, L.A.N. Amaral, Cartography of complex networks: Modules and universal roles, *J. Stat. Mech.* P02001 (2005).

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

[23] P. Jordano. Patterns of mutualistic interactions in pollination and seed dispersal: connectance, dependence asymmetries and co-evolution. *American Naturalist*, 129 (5) (1987) 657–677

[24] Freeman L. C. Finding social groups: a meta-analysis of the Southern Women data, *Dynamic Social Network Modeling and Analysis* ed R Breiger, K Carley and P Pattison (Washington, DC: The National Academies Press) (2003) 39-77

[25] D. Baird & R.E. Ulanowicz . The seasonal dynamics of the Chesapeake Bay ecosystem. *Ecological Monographs* 59 (1989) 329-364.

[26] A. L. N. Fred and A. K.Jain. Robust Data Clustering. *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2 (2003) 128-133.

[27] L. Danon, A. Díaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, *J. Stat. Mech.* P09008 (2005).

[28] M. Rosvall, C. Bergstrom, "Maps of random walks on complex networks reveal community structure", *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, p. 1118, 2008

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

[30] R. Guimerà, M. Sales-Pardo, L.A.N. Amaral, Module identification in bipartite and directed networks, *Phys. Rev.* E 76 (3) (2007) 036102.

[31] G.K. Orman, V. Labatut, H. Cherifi. On Accuracy of Community Structure Discovery Algorithms, *Journal of Convergence Information Technology* 6(11)(2011) 283-292.

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

[33] S. Gregory. Finding overlapping communities in networks by label propagation. *New J. Phys.* 12 (2010) 103018

[34] M. J. Barber. Modularity and community detection in bipartite networks. *Phys. Rev. E* 76, (2007) 066102.

[35] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi. Optimization by simulated annealing, *Science* 220 (1983) 671 – 680.

[36] U.N. Raghavan, R Albert, S Kumara. Near linear time algorithm to detect community structures in large-scale networks, *Phys. Rev. E* 76 (2007) 036106

[37] A. Lancichinetti, F. Radicchi, J. J. Ramasco JJ, S. Fortunato. Finding Statistically Significant Communities in Networks. *PLoS ONE* 6(4) (2011) e18961.

[38] G. Palla, I. Derényi, I. Farkas, T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society *Nature* 435 (2005) 814.

[39] A. Lancichinetti, S. Fortunato, J. Kertész. Detecting the overlapping and hierarchical community structure of complex networks *New J. Phys.* 11 (2009)033015

[40] W. Zhan, Z. Zhang, J. Guan, and S. Zhou. Evolutionary method for finding communities in bipartite networks *Phys. Rev. E* 83 (2011) 066120

[41] R.E. Ulanowicz, C. Bondavalli and M.S. Egnotovich. Network Analysis of Trophic Dynamics in South Florida Ecosystem, FY 97: *The Florida Bay Ecosystem*. Ref. No. [UMCES]CBL 98-123. Chesapeake Biological Laboratory, Solomons, MD 20688-0038 USA (1998).