OvSbChain: An Enhanced Snowball Chain Approach for Detecting Overlapping Communities in Social Graphs

Jayati Gulati, Muhammad Abulaish
Department of Computer Science
South Asian University
New Delhi, India

Sajid Yousuf Bhat
Department of Computer Sciences
University of Kashmir
J&K, India

***Abstract*—Overlapping Snowball Chain is an extension to Snowball Chain, which is based on the concept of community formation in line to the snowball chaining process. The inspiration behind this approach is from the snowball sampling process, wherein a snowball grows to form chain of nodes, leading to the formation of mutually exclusive communities in Snowball Chain. In the current work, the nodes are allowed to be shared among different snowball chains in a graph, leading to the formation of overlapping communities. Unlike its predecessor Snowball Chain, the proposed technique does not require the use of any hyper-parameter which is often difficult to tune for most of the existing methods. The proposed algorithm works in two phases, where overlapping chains are formed in the first phase, and then they are combined using a similarity-based criteria in the second phase. The communities identified at the end of the second phase are evaluated using different measures, including *modularity*, *overlapping NMI* and *running time* over both real-world and synthetic benchmark datasets. The proposed Overlapping Snowball Chain method is also compared with eleven state-of-the-art community detection methods.**

***Keywords*—Clustering coefficient; community detection; overlapping communities; snowball sampling; social graph**

I. Introduction

In recent years, there has been a tremendous growth in the study of linked data in the form of networks, such as Internet, World Wide Web, and social networks. The relationships among the entities existing in these networks provide rich insights pertaining to various dynamic interactions and might prove to be beneficial in various applications [1]. To analyse and study these networks, graph is used as a data structure, which consists of a set of nodes joined by links or edges that can be labelled/unlabelled, directed/undirected, or signed/unsigned. The representation of an online social network is termed as *social graph*, which provides a good visualization and eases the interpretation of the network.

One of the emerging research areas in social network analysis is community detection, which digs deep into the social graph and mines the most dense subgraphs that are highly cohesive in nature. A community in a network is represented by a set of nodes with high density links among themselves, but low-density links among inter-community connections [2]. These subgraphs are called communities or modules. Community detection in a social graph mainly involves splitting it into its constituent functional groups. The task has largely been addressed in a distinct community context wherein the communities are considered to be mutually exclusive. However, in case of real-world networks, community structures can be overlapping wherein a node belongs to multiple communities. A density-based approach called CMiner in [3], aims to find similarity among nodes and defines a distance function. Overlapping communities are identified based on this distance function. Another work in [4], detects overlapping communities along with their evolution, called as OCTracker. A similar work in [5], identifies hierarchical communities called HOCTracker which works for dynamic social networks.

The work in this paper aims to address this issue by proposing a novel overlapping community detection algorithm which extends the existing *SbChain* algorithm. The proposed method, named *OvSbChain*, starts with identification of the seed or core nodes in a social graph based on a node parameter, called *normalized degree*. The nodes in the entire social graph are ranked on this parameter and processed in a non-increasing order of their ranks. The method works in two phases. In the first phase, every node is paired with its best suited neighbor in accordance to a score value in each iteration. After several iterations, chains of nodes are formed that may share nodes with each other, i.e., there could be overlapping nodes among different chains. Therefore, the proposed technique is called overlapping snowball chains. The second phase tries to combine chains based on a similarity criteria as discussed in Section III, which finally leads to the formation of overlapping communities. Therefore, the technique focuses on resolving the problem in hand, i.e., community detection using an uncomplicated and elementary strategy. The major enhancements in this work can be summarized as follows:

1) *OvSbChain* introduces overlapping communities unlike *SbChain*, which produces only crisp communities.
2) There is no hyper-parameter tuning required in *OvSbChain*, hence, it always produces the same set of communities every time it is run.
3) *SbChain* uses a maximum common neighbor criteria for finding its best neighbor. Whereas, *OvSbChain* uses normalized degree function to find its best neighbor. Also, both the techniques differ in the way they find the seed nodes. This is discussed in detail in Section III.
4) The results are evaluated and compared based on

Nicosia modularity measure [6], two types of *ONMI* [7], [8] and their *running time*, as discussed in Section IV.

The proposed *OvSbChain* method is compared with eleven state-of-the-art community detection methods, including CFinder [9], LAIS [10], CONGA [11], PEACOCK [11], COPRA [12], SLPA [13], Demon [14], BIGCLAM [15], MULTICOM [16], Lemon [17] and ANGEL [18]. The results from all these methods are evaluated on different parameters including *modularity*, *ONMI* and *running time*, as discussed in Section IV.

The rest of the paper is organized as follows. Section II presents a brief review of the existing literatures on overlapping community detection. Section III presents the preliminary concepts, along with the proposed approach. This section also presents the functional details of the *OvSbChain* method. Section IV describes the details about the datasets, evaluation parameters, experimental settings, and analysis of the results. Section V concludes the paper and finally, Section VI provides future directions of research.

II. Related Work

This section presents a brief description of the state-of-the-art in the area of overlapping community detection. A review of the current community detection methods is described in [19]. It segregates the detection methods into probability-based and deep learning-based. The classical methods use probability-based models for community identification. Whereas, complex networks are generally converted to lower dimensional data using deep learning methods so as to ease the process. A few other works like [20], [21], [22], discuss various community detection algorithms based on their weakness and strengths, performance of algorithm and other domains. We mainly discusses all the traditional approaches for overlapping community detection and compare them with *OvSbChain* in Section IV.

CFinder is an overlapping community detection technique that makes use of the Clique Percolation Method (CPM) [23] to identify the k-cliques in a network. A k-clique is a complete subgraph consisting of k nodes. This method finds dense groups of overlapping nodes in a network [9].

LAIS [10] is an algorithm that combines two functions List Aggregate (LA) and Improved Iterative Scan (IS^2). The LA procedure initializes the clusters, and the IS^2 procedure improves upon these set of clusters in an iterative manner. The IS procedure starts with a seed node and processes clusters by expanding or shrinking them according to a metric value, and IS^2 improves upon this by focussing on nodes within a cluster and its neighboring nodes, instead of considering the entire graph. The overall algorithm detects overlapping community in a network.

CONGA (Cluster-Overlap Newman Girvan Algorithm) [11] is an overlapping community detection algorithm that uses the concept of split-betweenness, i.e., it counts the shortest paths that exist between all pairs of nodes in the network. It keeps removing edges with high betweenness, and thus, keeps splitting the network into singleton clusters. The partition with the desired number of clusters is picked up. However, it requires number of communities as an input for the algorithm.

PEACOCK algorithm [11] consists of two phases; the first phase is similar to CONGA, where the network is split using split betweenness. The altered network is processed by a disjoint community detection algorithm, called centrality of detecting communities based on node centrality or CNM.

COPRA [12] technique extends the previous work on the label propagation by Raghavan, Albert, and Kumara [24], and it is able to detect overlapping communities in a social network. The main extension is to make the label and propagation step to include information about more than one community. Therefore, it allows each node to belong to up to v communities, where v is a hyper-parameter.

In SLPA (Speaker-listener Label Propagation algorithm) [13], the nodes store multiple labels, and act either as the provider or consumer of information. A node keeps gathering information about the observed labels without removing the previously stored label. The frequency of observation of a label by a node is directly related to the spreading of the label among other nodes. It requires a threshold input parameter that gives the minimum probability of occurrence of a label, before it is deleted from the memory of the node.

Demon (Democratic Estimate of the Modular Organization of a Network) [14] is a simple approach for community detection which works on the modular structure of networks. Firstly, each node finds and votes the communities present in its local neighborhood, using a label propagation algorithm. These local communities are merged to form a global collection by combining all the votes, leading to the formation of overlapping modules. However, this algorithm requires a minimum threshold parameter.

BIGCLAM (Cluster Affiliation Model for Big Networks) [15] is a model-based community detection algorithm that allows for identification of dense overlapping, hierarchical communities in massive networks. Each node-community pair is assigned a non-negative latent factor that decides the degree of membership between them. The probability of a connection between a pair of nodes in the network is modeled as a function of the shared community affiliations. Further, the communities are identified using non-negative matrix factorization methods and block stochastic gradient descent.

MULTICOM is another community detection technique that produces overlapping communities starting with an initial seed set. Local community is detected around the seed nodes using a transformation function. After this step, each node belongs to a single community. Thereafter, the transformation function is used to transform a node into its respective vector, that is clustered using a local clustering technique. For each cluster produced in the previous step, a ratio value is calculated using a function mentioned in [16]. The clusters having ratio value less than a pre-defined threshold are considered for further exploration. The process keeps repeating until the number of communities is greater than the set value or if there is no new seed.

In [17], the technique called Lemon (Local Expansion via Minimum One Norm) detects overlapping communities by finding a sparse vector in the local spectra span, such that all the seeds are in its support. The span of vector dimensions produced by random walk is used as an approximate invariant subspace, called the local spectra. However, this local spectral

approach is used for community detection from a small seed set.

ANGEL [18] is a faster successor of Demon that uses a bottom-up approach to find overlapping communities. It works in two phases, where the first phase produces local communities using ego network of the nodes. The second phase merges communities until convergence or a threshold value is met.

The work in [25], develops a PageRank algorithm with constraints so as to obtain tightly packed overlapping communities. Using probability-based methods, a walker avoids irrelevant communities. Therefore, it results in communities with good fitness score. In [26], a method called Adjacency Propagation Algorithm (APA) is developed using adjacent nodes as seed nodes. It uses a threshold parameter to identify subgraphs based on their intraconnectivity. Another work in [27], can produce disjoint as well as overlapping communities in a two-step process that uses genetic algorithm. In the first step, mean path length of a community is calculated in relation with its respective ER random graph. And the second step shrinks the search space by selecting a subset of nodes. Another work in [28], influential nodes are identified to form local communities. These communities expand as nodes join these local communities. Overlapping communities are merged and evaluated on a model.

An application-based work in [29], exploits community detection to protect the privacy of individuals on social platforms It discusses community detection attacks and rewiring of connections for development of effective attack approach.

OvSbChain approach focuses on local community detection using graph parameters such as degree and global clustering coefficient. If these local communities are identical they are merged. The motivation behind this work is that it exploits simple topological features of the graph to detect communities without any expensive overhead in two simple levels, (i) formation of local communities, and (ii) combining local communities based on two criteria.

III. Proposed Approach

The *OvSbChain* approach discussed in this section is an extension to the previously developed *SbChain* [30] method. It detects overlapping communities, i.e., nodes are allowed to be shared among more than one community. The approach works on two levels. In the first level, it starts with finding the best suited pairs of nodes according to an initial criterion. This level ends up with formation of overlapping snowball chains. In the second level, these chains are merged to form the larger chains, and eventually form communities based on global clustering coefficient or majority overlapping criteria.

A. Preliminaries

For a graph $G(V, E)$, V represents the set of vertices or nodes in the graph, i.e $\{v \in V\}$, where n is the number of nodes. And E is the set of edges, i.e., $\{e_{uv} = (u, v) : u, v \in V\}$. This section presents the details about frequently used terms and their meanings, as mentioned in table I.

OvSbChain works at two levels that are described in the following paragraphs:

TABLE I. Notations and their Descriptions

Notation	Description
$\mathcal{N}(v)$	Set of immediate neighbors of a node v
$k(v) = \lvert\mathcal{N}(v)\rvert$	Degree of a node v
k_{max}	Maximum degree value in the graph
$\mathcal{N}_{best}(v)$	Best scoring neighbor of node v
$s^{(n)}$	n^{th} snowball chain
$GCC(s^{(n)})$	Global clustering coefficient of a snowball chain $s^{(n)}$

1) Level-I: It starts by finding the seed nodes and sorting them in non-increasing order, based on the following criteria so as to begin the processing.

1) Seed function - A seed v can by identified by sorting nodes according to their normalized degree value function, given by equation 1. This also represents the score $score(v)$ of a node v.

$$score(v) = k(v)/k_{max} \tag{1}$$

These sorted nodes are processed in non-increasing order of this function value. It should be noted that *SbChain* used a combination of normalized degree and normalized local clustering coefficient for sorting of nodes.

2) $\mathcal{N}_{best}(v)$ function - The best suited neighbor for a seed v is identified using the same score value, i.e., the normalized degree. This neighbor further combines with the seed v to form a snowball chain. Whereas, *SbChain* used maximum number of overlapping neighbors for finding its best neighbor.

It should be noted that these functions have been chosen and designed empirically.

2) Level-II: The second level starts with the chains formed in the first level. These chains are merged to form communities, so as to eliminate almost similar chains. The snowball pairs/chains formed in first level are combined based on global clustering coefficient (GCC) or majority overlapping criteria to form a community. GCC signifies the number of closed triangles to the number of triplets in a graph. Therefore, the technique focuses on finding higher values of GCC for a community, so as to find coherent communities. The first criteria involves calculation of GCC of the formed community, along with GCC of each individual snowball chain. If the combined GCC is higher than the GCC of each chain, then their combination is permitted, otherwise it is discarded, i.e., the chains remain undisturbed. Communities can also be combined as per the second criteria of majority overlapping. This allows communities to get merged if they have atleast 70% overlapping nodes. This percentage is decided empirically, as the value of communities do not change after this point. Also, the minimum percentage overlap was decided to be above 50% so as to form coherent communities. The majority overlapping test prevents the existence of two or more similar communities.

It should be noted that *OvSbChain* creates overlapping communities because it does not follow *non-redundant node strategy*, previously used by *SbChain*. According to this

Algorithm 1: bestNeighbor($v, \mathcal{N}(v)$)

Input : Node v, neighbor list $\mathcal{N}(v)$
Output: Best neighbor of v i.e, $\mathcal{N}_{best}(v)$

$maxWeight \leftarrow 0$
foreach $v \in \mathcal{N}(v)$ **do**
  if *score(v) ¿ maxWeight* **then**
    $maxWeight \leftarrow score(v)$
    $\mathcal{N}_{best}(v) \leftarrow v$
  end
end
return $\mathcal{N}_{best}(v)$

strategy, a node could join with a single node per iteration which creates mutually exclusive communities. The focus of *OvSbChain* is to develop communities that share nodes among themselves. Therefore, it discards this strategy and allows a node to be a part of multiple chains within a single iteration itself.

TABLE II. DIFFERENT TYPES OF REAL-WORLD DATASETS

Dataset	Nodes	Edges
Zachary [31]	34	78
Dolphin [32]	62	159
Football [33]	115	613
Books[1]	105	441
Netscience [34]	379	914
Jazz [35]	198	5484
Email [36]	1133	5451
Power [37]	4941	6594
Blogs [38]	3982	6803
Protein [39]	2445	6265

TABLE III. PARAMETERS USED TO GENERATE LFR-1K NETWORK

Parameter	Value
Nodes (N)	1000
Average degree ($\langle k \rangle$)	15
Minimum community size (c_{min})	20
Maximum community size (c_{max})	50
Maximum degree (k_{max})	50
Number of overlapping nodes (o_n)	100
Number of memberships of the overlapping nodes (o_m)	30
Mixing parameter (μ)	[0.1, 0.5]

B. Algorithm

As discussed in the algorithm 2, *OvSbChain* starts with the pre-processing, i.e., it calculates neighbor list $\mathcal{N}(v)$, degree list $k(v)$ and $score(v)$ (equation 1) for each node v in the social graph. These nodes are then sorted in non-increasing order of their respective $score$ and processed one at a time. Snowball chains are formed by finding the best neighbor (algorithm 1) for each node on the basis of this $score$ value itself, i.e., for a given node v, the best neighboring node $\mathcal{N}_{best}(v)$ with highest value of $score$ parameter is chosen.

In the first iteration, best suited node pairs are combined. The snowball chains $s^{(n)}$ so formed grow internally and new chains are also formed in each iteration, as the nodes find their matches. This sums up the level-I of the proposed technique.

The level-II starts with calculation of global clustering coefficient $GCC(s^{(n)})$ for each snowball chain $s^{(n)}$ formed in level-I. These chains are combined and added to community list C if the GCC of the union of two chains $GCC(s^{(j)} \cup s^{(k)})$

Algorithm 2: OvSbChain(G)

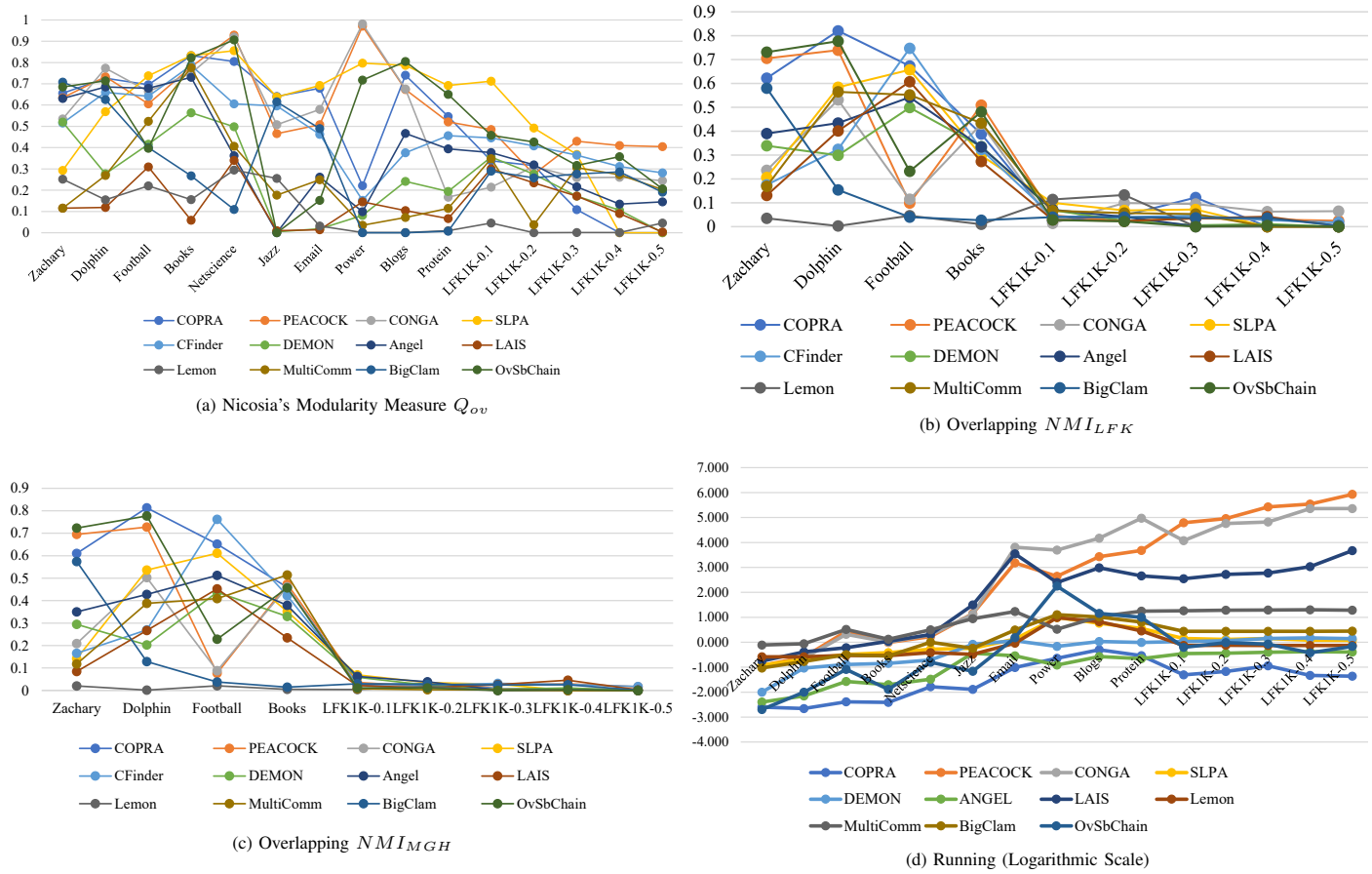
Input : A graph $G(V, E)$
Output: Community list C

1 Pre-processing calculates $\mathcal{N}(v)$, $k(v)$, and $score(v)$ for each node v
2 Arrange $score(v)$ in non-increasing order
3 Initialize new lists snowball s, community C
4 $i \leftarrow 0$
`// Level-I`
5 **foreach** $v \in score.keys$ **do**
6 $\mathcal{N}_{best}(v) \leftarrow bestNeighbor(v, \mathcal{N}(v))$
  `// Algorithm 1`
7 **if** $s = \emptyset$ **then**
8 $i \leftarrow i + 1$
9 Append $\langle v, \mathcal{N}_{best}(v) \rangle$ into $s^{(i)}$
10 Goto 5
11 **end**
12 $counter \leftarrow 0$
13 **for** $j \leftarrow 1$ *to* $len(s)$ **do**
14 **if** $\mathcal{N}_{best}(v) \in s^{(j)}$ *and* $v \notin s^{(j)}$ **then**
15 Append $\langle v \rangle$ into $s^{(j)}$
16 **else**
17 **if** $\mathcal{N}_{best}(v) \notin s^{(j)}$ *and* $v \in s^{(j)}$ **then**
18 Append $\langle N_{best}(v) \rangle$ into $s^{(j)}$
19 **else**
20 $counter \leftarrow counter + 1$
21 **end**
22 **end**
23 **end**
24 **if** *counter = i* **then**
25 $i \leftarrow i + 1$
26 Append $\langle v, \mathcal{N}_{best}(v) \rangle$ into $s^{(i)}$
27 **end**
28 **end**
  `// Level-II`
29 **while** $C \neq s$ **do**
30 **if** $C \neq \emptyset$ **then**
31 $s \leftarrow C$
32 $C \leftarrow \emptyset$
33 **end**
34 **foreach** $s^{(j)} \in s$ **do**
35 **foreach** $s^{(k)} \in s$ **do**
36 **if** $(GCC(s^{(j)} \cup s^{(k)}) > GCC(s^{(j)})$ *and* $GCC(s^{(j)} \cup s^{(k)}) > GCC(s^{(k)}))$ *or* $\frac{\left\|(s^{(j)} \cap s^{(k)})\right\|}{min(\left\|s^{(j)}\right\|, \left\|s^{(k)}\right\|)} > 0.7$ **then**
37 Append $\langle s^{(j)}, s^{(k)} \rangle$ into C
38 **else**
39 Append $\langle s^{(j)} \rangle$, $\langle s^{(k)} \rangle$ into C
40 **end**
41 **end**
42 **end**
43 **end**
44 return C

is greater than either of their individual GCC or if majority of their nodes overlap (i.e., $\geq 70\%$) as mentioned in step 36 of the algorithm. The chains keep combining until both of the criteria fail. If the chains do not combine with other chains, they are directly added to C. The end result is the final set of communities C.

IV. EXPERIMENTAL SETUP AND RESULTS

In this section, the performance of the *OvSbChain* algorithm is evaluated over different datasets using various parameters. The *OvSbChain* is compared with several other overlapping community detection techniques. The following



(a) Nicosia's Modularity Measure Q_{ov}

(b) Overlapping NMI_{LFK}

(c) Overlapping NMI_{MGH}

(d) Running (Logarithmic Scale)

Fig. 1. Comparison of *OvSbChain* with Various Techniques on different Evaluation Metrics

subsections discuss the various datasets used in our experimental evaluations and all the parameters used for assessment of the identified communities.

A. Dataset

The efficacy of *OvSbChain* and other techniques is evaluated over ten real-world datasets and five computer-generated Lancichinetti Fortunato Radicchi (LFR) benchmark datasets [40], as discussed in Tables II and III. The LFR benchmark datasets consists of 1000 nodes with value of the mixing parameter (μ) varying from 0.1 to 0.5. Hence, the datasets are named as LFR1K-0.1, LFR1K-0.2, ..., LFR1K-0.5, respectively.

B. Evaluation Metrics

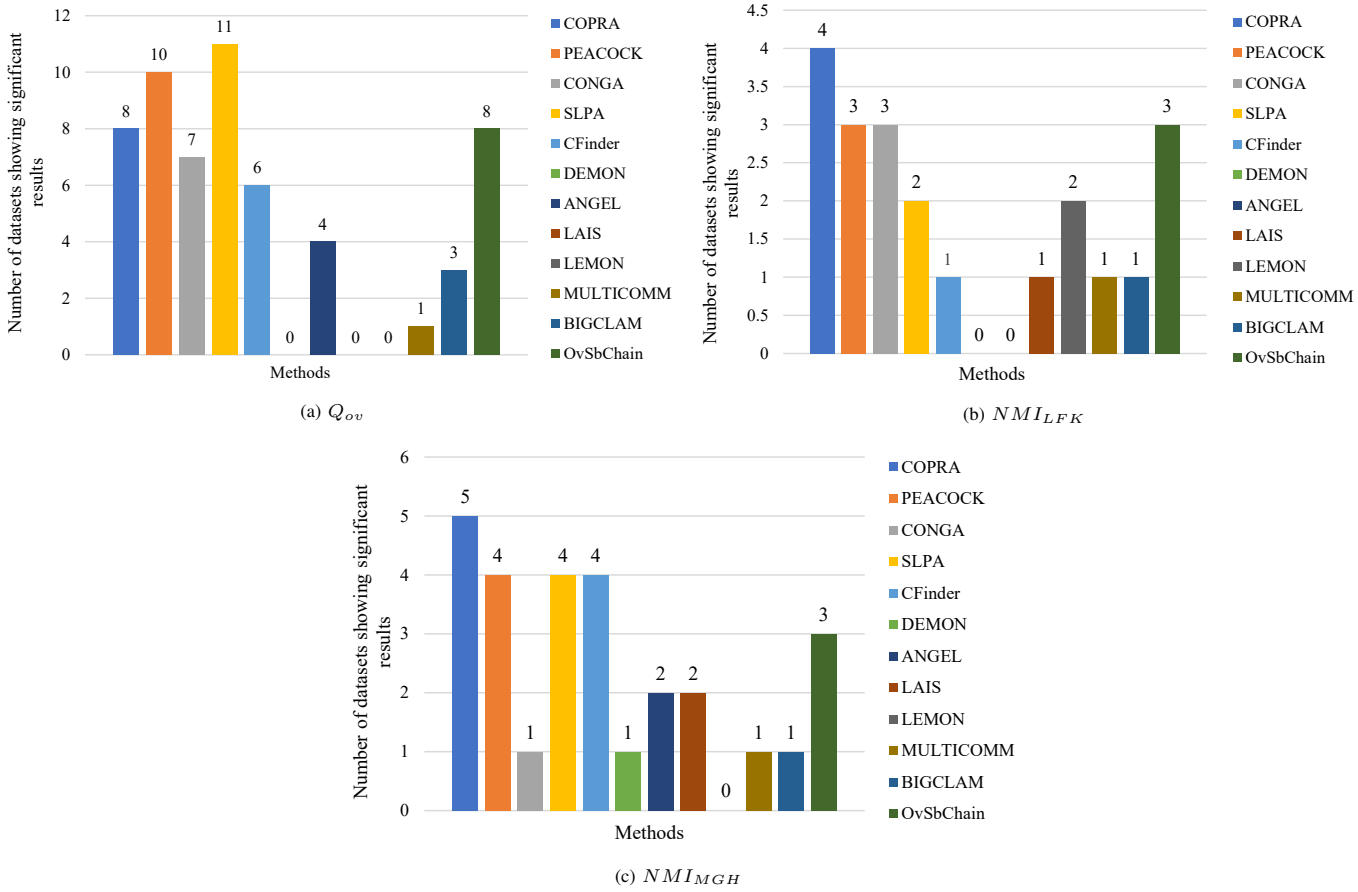
The communities identified as a result of algorithm 2 are analyzed by an overlapping *modularity* measure given by [6], two types of *ONMI* (Overlapping Normalized Mutual Information), and their *running time*.

It should be noted that the modularity measure given by [6] is represented as Q_{ov}. By definition, $Q_{ov} = 0$, for singleton communities or if all nodes belong to a community. Q_{ov} uses a belonging coefficient for each node which defines the percentage contribution of a node in a community. The sum of this coefficient in 1, for each node.

ONMI is an extension of the NMI score that accommodates overlapping partitions within a network. There are two types of ONMI used in this section; one is LFK (Lancichinetti Fortunato Kertesz) [7], which is referred as NMI_{LFK}, but it overestimates the similarity of two clusters in some cases. To fix this, another ONMI called MGH (McDaid Greene Hurley) is used. This version uses a different normalization than the original LFK based ONMI [8], and it is represented as NMI_{MGH}.

C. Results

Techniques like COPRA, PEACOCK, CONGA, SLPA, CFinder, Demon, and ANGEL use a parameter for tuning. Hence, the values represented in this paper are the best values for Q_{ov}. Fig. 1a shows the results of various overlapping techniques compared with *OvSbChain* on Q_{ov}, respectively. The same is also represented via Table IV. Also, Fig. 2a represents the number of datasets for each technique that have their respective value greater than or equal to 80% of the maximum Q_{ov} that exists for all the techniques. It can be observed that *OvSbChain* has an above average performance in terms of Q_{ov}. Though other techniques are seen to show a better value in terms of Q_{ov}, it is seen that the respective ONMI values drops. Hence, high modularity does not necessarily guarantee good partitions.



(a) Q_{ov}

(b) NMI_{LFK}

(c) NMI_{MGH}

Fig. 2. Comparison of Various Techniques on the Number of Datasets having Values Greater than or Equal to 80% of the Maximum Existing Value of Different Evaluation Metrics

It can be seen that although modularity values are comparable or average in comparison to existing techniques, the ONMI values are promising. As an example, SLPA has the highest modularity among all techniques, it does not produce high ONMI values. *OvSbChain* is faster for smaller datasets and produces comparable or better results for certain cases in terms of both NMI_{LFK} and NMI_{MGH}.

Both NMI_{LFK} and NMI_{MGH} are calculated and compared on both real-world and LFR datasets, as shown in Fig. 1b and 1c for *OvSbChain* and other techniques. *OvSbChain* is seen to perform well in most of the cases. Fig. 2b and 2c also show the comparison of the number of datasets that have NMI values greater than or equal to 80% of the maximum existing value of NMI (among all the given datasets). Tables VI and VII show both the ONMI values. It can be observed that the performance of *OvSbChain* is above average for both NMI_{LFK} and NMI_{MGH} measures.

A comparison of the running time of all the techniques is presented in Fig. 1d. Logarithmic scale is used for this comparison because it provides a better visualization. CFinder technique is excluded from this comparison because it does not mention the time it takes to evaluate the communities so formed. It can be observed that *OvSbChain* is works fast on smaller datasets, and it is comparable to other techniques on larger datasets. The same can be seen through table V. As mentioned before, a few techniques use a parameter which needs to be defined every time they are executed. Therefore, in our experimental evaluation, these techniques were run for different parameter values and the best value for Q_{ov} was chosen and the corresponding ONMI and run time values are represented. On the other hand, our proposed *OvSbChain* approach does not need any parameter value to be set, hence, produces the same result every time it is run.

V. Conclusion

It can be seen that the technique *OvSbChain* discussed in the current article works well on real-world datasets with good results in terms of NMI_{LFK} and NMI_{MGH}. It gives comparable results on a few benchmark datasets as well. It should be noted that the running speed of the algorithm was at par with other techniques, or even better in a few cases. The experiments show average results on modularity measure as well. *OvSbChain* does not use any external parameter like most of its counterparts. Also, it produces the same results every time it is run, unlike the other techniques, e.g., COPRA. It gives different results each time it is run (for same parameter value). Hence, it is run for ten times, and the results are averaged. Therefore, it can be established that our technique works well without any parameter tuning, unlike the other

approaches. The overhead of calculations involved in the technique slows it down, but that can be resolved using better hardware options.

VI. Future Work

The future scope of improvement includes extension of the technique to directed graphs as real-world networks are generally directed in nature. OvSbChain can be improvised to find faster and high coverage seed nodes for snowball chains formation and eventually communities.

TABLE IV. COMPARISON AMONG DIFFERENT OVERLAPPING COMMUNITY DETECTION TECHNIQUES ON VARIOUS REAL-WORLD AND BENCHMARK DATASETS ON NICOSIA'S OVERLAPPING MODULARITY MEASURE

Q_{ov}	Techniques											
Dataset	COPRA	Peacock	CONGA	SLPA	CFinder	Demon	ANGEL	LAIS	Lemon	MULTICOM	BIGCLAM	OvSbChain
Zachary	0.655 (5)	0.634 (2)	0.534 (3)	0.292 (0.2)	0.515 (3)	0.519 (0.5)	0.631 (0.4-0.6)	0.115	0.252	0.115	**0.708**	0.685
Dolphin	0.726 (4)	0.732 (2)	**0.773** (4)	0.569 (0.4-0.5)	0.659 (3)	0.276 (0.4)	0.685(0.3)	0.119	0.154	0.269	0.625	0.714
Football	0.695 (2)	0.605 (2)	0.664 (2)	**0.737** (0.3-0.5)	0.641 (4)	0.415 (0.6)	0.678 (0.3)	0.309	0.220	0.523	0.400	0.397
Books	0.832(3)	0.779 (2)	0.749 (3)	**0.832 (0.5)**	0.786 (3)	0.564 (0.6)	0.730 (0.6)	0.058	0.155	0.776	0.267	0.822
Netscience	0.804 (4)	**0.929 (5)**	0.919 (5)	0.855 (0.4-0.5)	0.605 (3)	0.497 (0.3)	0.362 (0.4)	0.339	0.294	0.406	0.109	0.905
Jazz	**0.640 (3)**	0.465 (3)	0.508 (5)	0.638 (0.3-0.5)	0.595 (10)	0.004 (0.8)	0.0004 (1)	0.009	0.255	0.176	0.615	0.000^{1}
Email	0.678 (2)	0.507 (2)	0.579 (5)	**0.692 (0.5)**	0.462 (3)	0.018 (0.5)	0.261 (0.6)	0.013	0.031	0.249	0.489	0.159
Power	0.221 (5)	0.970 (5)	**0.981 (5)**	0.797 (0.5)	0.152 (3)	0.081 (0.1)	0.099 (0.1-0.2)	0.145	0.0007	0.035	0.000^{1}	0.717
Blogs	0.740 (9)	0.672 (9)	0.675 (9)	**0.786 (0.5)**	0.376 (3)	0.241 (0.2)	0.466 (0.3)	0.104	0.001	0.071	0.000^{1}	0.804
Protein	0.545 (5)	0.520 (5)	0.167 (5)	**0.692 (0.5)**	0.456 (3)	0.194 (0.5)	0.393 (0.4)	0.066	0.009	0.114	0.007	0.650
LFR1K-0.1	0.337 (2)	0.484 (2)	0.214 (5)	**0.712 (0.3)**	0.446 (4)	0.355 (0.5)	0.377 (0.6)	0.300	0.045	0.347	0.289	0.458
LFR1K-0.2	0.315 (3)	0.275 (3)	0.311 (5)	**0.492 (0.5)**	0.408 (4)	0.275 (0.5)	0.319 (0.4)	0.234	0.0001	0.037	0.258	0.426
LFR1K-0.3	0.108 (2)	**0.429 (2)**	0.259 (5)	0.368 (0.5)	0.364 (4)	0.172 (0.5)	0.215 (0.5)	0.172	0.0006	0.305	0.275	0.316
LFR1K-0.4	0.000^{1} (5)	**0.409 (2)**	0.260 (4)	0.000^{1} (0.1)	0.310 (3)	0.105 (0.4)	0.134 (0.5)	0.089	0.001	0.272	0.285	0.357
LFR1K-0.5	0.000^{1} (2)	**0.404 (2)**	0.244 (2)	0.000^{1}(0.1)	0.281 (3)	0.001 (0.2)	0.145 (0.6)	0.003	0.045	0.202	0.190	0.206

TABLE V. COMPARISON AMONG DIFFERENT OVERLAPPING COMMUNITY DETECTION TECHNIQUES ON VARIOUS REAL-WORLD AND BENCHMARK DATASETS BASED ON (IN SECONDS)

Time	Techniques										
Dataset	COPRA	Peacock	CONGA	SLPA	Demon	ANGEL	LAIS	Lemon	MULTICOM	BIGCLAM	OvSbChain
Zachary	0.003	0.116	0.104	0.120	0.010	0.004	0.175	0.259	0.774	0.093	0.002
Dolphin	0.002	0.254	0.229	0.208	0.092	0.007	0.413	0.262	0.873	0.171	0.010
Football	0.004	2.668	2.081	0.321	0.129	0.027	0.607	0.293	3.2	0.315	0.082
Books	0.004	0.979	1.007	0.385	0.143	0.020	1.092	0.288	1.316	0.281	0.013
Netscience	0.017	1.684	1.957	0.525	0.194	0.033	2.049	0.381	3.1	0.998	0.158
Jazz	0.013	14.065	13.1	0.567	0.830	0.362	31.2	0.328	8.8	0.575	0.069
Email	0.096	1516.04	6428.1	1.352	1.148	0.285	3526.8	0.921	17.02	3.067	1.502
Power	0.226	442.4	4948.8	9.807	0.680	0.122	246.9	9.7	3.3	12.7	177.1
Blogs	0.498	2675.7	14746.6	5.756	1.066	0.266	971.2	6.6	10.691	10.3	14.3
Protein	0.289	4781.4	93087.7	3.535	0.977	0.221	454.5	2.815	17.5	6.3	9.969
LFR1K-0.1	0.049	61766.5	11830.8	1.408	1.068	0.353	353.5	0.741	18.1	2.7	0.627
LFR1K-0.2	0.068	90091.7	57434.7	1.294	1.148	0.362	522.1	0.760	19.3	2.7	0.968
LFR1K-0.3	0.114	265414.9	65968.1	1.409	1.415	0.399	590.1	0.748	19.6	2.7	0.833
LFR1K-0.4	0.047	344527.9	228258.1	1.203	1.493	0.419	1069.1	0.747	20.1	2.731	0.375
LFR1K-0.5	0.043	851668.6	229433.5	1.178	1.378	0.404	4712.3	0.737	19.3	2.7	0.682

TABLE VI. COMPARISON AMONG DIFFERENT OVERLAPPING COMMUNITY DETECTION TECHNIQUES ON VARIOUS REAL-WORLD AND BENCHMARK DATASETS ON NMI_{LFK} MEASURES

NMI_{LFK}	Techniques											
Dataset	Copra	Peacock	CONGA	SLPA	CFinder	Demon	ANGEL	LAIS	Lemon	MULTICOM	BIGCLAM	OvSbChain
Zachary	0.622	0.705	0.236	0.205	0.174	0.338	0.390	0.131	0.034	0.168	0.580	**0.730**
Dolphin	**0.820**	0.739	0.531	0.583	0.323	0.298	0.433	0.401	0.002	0.564	0.153	0.776
Football	0.672	0.098	0.115	0.657	**0.747**	0.498	0.541	0.607	0.046	0.552	0.039	0.231
Books	0.387	**0.588**	0.400	0.292	0.323	0.336	0.333	0.273	0.004	0.433	0.027	0.484
LFR1K-0.1	0.024	0.047	0.014	0.100	0.043	0.068	0.071	0.028	**0.114**	0.067	0.039	0.029
LFR1K-0.2	0.046	0.022	0.099	0.067	0.039	0.024	0.040	0.024	**0.132**	0.057	0.039	0.022
LFR1K-0.3	0.024	0.034	**0.094**	0.072	0.046	0.005	0.002	0.035	0.000	0.053	0.037	0.000
LFR1K-0.4	0.000	0.029	**0.062**	0.000	0.027	0.0104	0.002	0.041	0.000[1]	0.000	0.035	0.004
LFR1K-0.5	0.000	0.024	**0.065**	0.000	0.0175	0.000	0.000	0.002	0.000	0.000	0.004	0.000

TABLE VII. COMPARISON AMONG DIFFERENT OVERLAPPING COMMUNITY DETECTION TECHNIQUES VARIOUS REAL-WORLD AND BENCHMARK DATASETS ON NMI_{MGH} MEASURES

NMI_{MGH}	Techniques											
Dataset	COPRA	Peacock	CONGA	SLPA	CFinder	Demon	ANGEL	LAIS	Lemon	MULTICOM	BIGCLAM	OvSbChain
Zachary	0.610	**0.694**	0.208	0.142	0.165	0.294	0.349	0.084	0.020	0.117	0.574	0.723
Dolphin	**0.813**	0.727	0.502	0.536	0.269	0.202	0.427	0.267	0.001	0.387	0.129	0.776
Football	0.651	0.076	0.088	0.610	**0.762**	0.437	0.512	0.452	0.020	0.409	0.0381	0.223
Books	0.459	0.473	0.460	0.351	0.421	0.330	0.379	0.234	0.004	**0.514**	0.014	0.456
LFR1K-0.1	0.022	0.035	0.008	**0.069**	0.029	0.0586	0.0628	0.022	0.0042	0.008	0.029	0.011
LFR1K-0.2	0.033	0.017	0.021	0.034	0.024	0.023	**0.0389**	0.0122	0.0188	0.003	0.027	0.010
LFR1K-0.3	0.009	0.023	0.008	0.028	**0.031**	0.0054	0.0022	0.026	0.000	0.002	0.024	0.000
LFR1K-0.4	0.000	0.029	0.008	0.000	0.0277	0.010	0.0021	**0.046**	0.000	0.000	0.025	0.003
LFR1K-0.5	0.000	0.016	0.005	0.000	**0.017**	0.000	0.000	0.003	0.000	0.000	0.001	0.000

REFERENCES

[1] M. E. J. Newman, "The structure and function of complex networks," *Society for Industrial and Applied Mathematics (SIAM) Review*, vol. 45, no. 2, pp. 167–256, March 2003.

[2] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Physical Review E*, vol. 80, no. 5, pp. 056 117–(1–11), November 2009.

[3] S. Y. Bhat and M. Abulaish, "A density-based approach for mining overlapping communities from social network interactions," in *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics (WIMS), Craiova, Romania*, June 2012, pp. 1–7.

[4] ——, "OCTracker: A density-based framework for tracking the evolution of overlapping communities in OSNs," in *Proceedings of IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Istanbul, Turkey*, August 2012, pp. 501–505.

[5] ——, "HOCTracker: Tracking the evolution of hierarchical and overlapping communities in dynamic social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1019–1031, April 2015.

[6] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending the definition of modularity to directed graphs with overlapping communities," *Journal of Statistical Mechanics Theory and Experiment*, vol. 2009, pp. P03 024–P03 046, February 2008.

[7] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New Journal of Physics*, vol. 11, no. 3, pp. 033 015–033 032, March 2009.

[8] A. McDaid, D. Greene, and N. Hurley, "Normalized mutual information to evaluate overlapping community finding algorithms," *Computing Research Repository*, October 2011.

[9] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, "CFinder: Locating cliques and overlapping modules in biological networks," *Bioinformatics*, vol. 22, no. 8, pp. 1021–1023, April 2006.

[10] J. Baumes, M. Goldberg, and M. MagdonIsmail, "Efficient identification of overlapping communities," in *Proceedings of International Conference on Intelligence and Security Informatics (ISI), Berlin, Heidelberg*, May 2005, pp. 27–36.

[11] S. Gregory, "An algorithm to find overlapping community structure in networks," in *Proceedings of Knowledge Discovery in Databases (PKDD), Berlin, Heidelberg*, September 2007, pp. 91–102.

[12] ——, "Finding overlapping communities in networks by label propagation," *New Journal of Physics*, vol. 12, no. 10, pp. 103 018–103 043, October 2010.

[13] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM) Workshops, Vancouver, Canada*, September 2011, pp. 344–349.

[14] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi, "DEMON: A local-first discovery method for overlapping communities," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Beijing, China*, August 2012, pp. 615–623.

[15] J. Yang and J. Leskovec, "Overlapping community detection at scale: A nonnegative matrix factorization approach," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining (WSDM), Rome, Italy*, February 2013, pp. 587–596.

[16] A. Hollocou, T. Bonald, and M. Lelarge, "Multiple local community detection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 45, pp. 76–83, March 2018.

[17] Y. Li, K. He, K. Kloster, D. Bindel, and J. E. Hopcroft, "Local spectral clustering for overlapping community detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 2, pp. 17–(1–27), March 2018.

[18] G. Rossetti, "Exorcising the demon: Angel, efficient node-centric community discovery," in *Proceedings of International Conference on Complex Networks and Their Applications VIII, Lisbon, Portugal*, November 2019, pp. 152–163.

[19] D. Jin, Z. Jin, P. Jiao, S. Pan, D. He, J. Wu, P. Yu, and W. Zhang, "A survey of community detection approaches: From statistical modeling to deep learning," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–22, August 2021.

[20] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics Reports*, vol. 659, pp. 1–44, 2016.

[21] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *Journal of Network and Computer Applications*, vol. 108, pp. 87–111, 2018.

[22] P. Bedi and C. Sharma, "Community detection in social networks," *WIREs Data Mining and Knowledge Discovery*, vol. 6, no. 3, pp. 115–135, 2016.

[23] I. Derényi, G. Palla, and T. Vicsek, "Clique percolation in random networks," *Physical Review Letter*, vol. 94, pp. 160 202–(1–4), April 2005.

[24] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, pp. 036 106–(1–11), October 2007.

[25] Y. Gao, X. Yu, and H. Zhang, "Overlapping community detection by constrained personalized pagerank," *Expert Systems with Applications*, vol. 173, pp. 114 682–(1–12), February 2021.

[26] O. Doluca and K. Oğuz, "APAL: Adjacency propagation algorithm for overlapping community detection in biological networks," *Information Sciences*, vol. 579, pp. 574–590, August 2021.

[27] A. K. Ghoshal, N. Das, and S. Das, "Disjoint and overlapping community detection in small-world networks leveraging mean path length," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 2, pp. 406–418, July 2021.

[28] T. Ma, Q. Liu, J. Cao, Y. Tian, A. Al-Dhelaan, and M. Al-Rodhaan, "LGIEM: Global and local node influence based community detection," *Future Generation Computer Systems*, vol. 105, pp. 533–546, 2020.

[29] J. Chen, L. Chen, Y. Chen, M. Zhao, S. Yu, Q. Xuan, and X. Yang, "GA-based Q-attack on community detection," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 491–503, 2019.

[30] J. Gulati and M. Abulaish, "A novel snowball-chain approach for detecting community structures in social graphs," in *Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China*, December 2019, pp. 2462–2469.

[31] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, November 1976.

[32] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations - can geographic isolation explain this unique trait?" *Behavioral Ecology and Sociobiology*, vol. 54, pp. 396–405, January 2003.

[33] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 99, no. 12, pp. 7821–7826, January 2002.

[34] M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, pp. 036 104–(1–19), October 2006.

[35] P. M. Gleiser and L. Danon, "Community structure in jazz," *Advances in Complex Systems*, vol. 6, no. 04, pp. 565–573, December 2003.

[36] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical Review E*, vol. 68, no. 6, pp. 065 103–(1–4), January 2003.

[37] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, July 1998.

[38] S. Gregory, "An algorithm to find overlapping community structure in networks," in *Proceedings of European Conference on Principles of Data Mining and Knowledge Discovery (PKDD), Warsaw, Poland*, September 2007, pp. 91–102.

[39] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 7043, pp. 814–818, July 2005.

[40] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, pp. 046 110–(1–5), November 2008.