Cooperative Co-evolutionary Module Identification with Application to Cancer Disease Module Discovery

Shan He*, Guanbo Jia, Zexuan Zhu, Daniel A. Tennant, Qiang Huang, Ke Tang, Jing Liu, Mirco Musolesi, John K. Heath, and Xin Yao *Fellow, IEEE*

***Abstract*—Module identification or community detection in complex networks has become increasingly important in many scientific fields because it provides insight into the relationship and interaction between network function and topology. In recent years, module identification algorithms based on stochastic optimization algorithms such as Evolutionary Algorithms have been demonstrated to be superior to other algorithms on small to medium scale networks. However, the scalability and resolution limit problems of these module identification algorithms have not been fully addressed, which impeded their application to real-world networks. This paper proposes a novel module identification algorithm called Cooperative Co-evolutionary Module Identification to address these two problems. The proposed algorithm employs a cooperative co-evolutionary framework to handle large scale networks. We also incorporate a recursive partitioning scheme into the algorithm to effectively address the resolution limit problem. The performance of our algorithm is evaluated on twelve benchmark complex networks. As a medical application, we apply our algorithm to identify disease modules that differentiate low and high grade glioma tumours to gain insights into the molecular mechanisms that underpin the progression of glioma. Experimental results show that the proposed algorithm has a very competitive performance compared with other state-of-the-art module identification algorithms.**

***Index Terms*—Modularity identification, community detection, cooperation co-evolutionary, complex networks.**

I. Introduction

Many complex systems, such as social [1], [2] and biological networks [3], can be naturally represented as complex networks. A complex network consists of nodes (or vertices) and edges (or links) which respectively represent the individual members and their relationships in systems. Based on complex network representation, many theories and methods in graph theory can be applied to enable us to gain insights into complex systems. In recent years, the study of complex networks has attracted more and more attention. One of the most studied complex network properties is community structure [4], which is usually considered as the division of networks into subsets of vertices within which intra-connections are dense, while between which inter-connections are sparse. The identification of the community structure provides important information about the relationship and interaction between network function and topology.

In the past few years, many methods have been proposed to detect the underlying community structures in complex networks [4], [5], [6]. Among them, the most popular methods are modularity maximization based on the definition of modularity [4]. The popularity of the methods is mainly due to the superior performance on real-world complex networks to other methods. Many deterministic optimization algorithms such as greedy algorithms have been employed to maximize the modularity in order to find the optimal division of complex networks [4]. However, in [7] the authors found that as a complex network becomes more modular, the global optimal partition becomes harder to find among the exponentially growing number of suboptimal, but competitive, alternatives. This so-called extreme degeneracy problem indicates that we should treat the results from those deterministic optimization algorithms, which only return unique usually suboptimal solution, with "particular caution" because it might "obscure the magnitude of the degeneracy problem and the wide range alternative solutions" [7].

Several stochastic modularity maximization methods have been proposed to optimize modularity [8], [9], [10], [11], [12], [13]. Among them, the most successful ones are those based on evolutionary algorithms, e.g., the genetic algorithm [9], [10], [11] and differential evolution [12], [13], [14], [15]. However, although these algorithms have achieved satisfactory results on small to medium scale networks, their performance on large-scale networks is not even competitive compared with that of the greedy search algorithms.

In order to address the scalability problem, this paper proposes **Co**operative **Co**-evolutionary **M**odularity **I**dentification (CoCoMi) algorithm. CoCoMi incorporates a Cooperative Co-evolution (CC) framework which employs a divide-and-conquer strategy to divide a large-scale network into several small-scale subnetworks and evolves those subnetworks independently and co-adaptively [16], [17], [18]. Compared with traditional EAs [19], [20], [21], [22], the advantages of

Shan He, Guanbo Jia, Mirco Musolesi and Xin Yao are with CERCIA, School of Computer Science, University of Birmingham.

Shan He and John K. Heath are with the Centre for Systems Biology, School of Biosciences, University of Birmingham.

Daniel A. Tennant is with School of Cancer Sciences, University of Birmingham.

Jing Liu is with Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education. Xidian University, China.

Ke Tang is with Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China, China

Qiang Huang is with School of Software, Sun Yat-sen University, China.

Zexuan Zhu is with College of Computer Science and Software Engineering, Shenzhen University, China.

*Corresponding author

the CC framework are: 1) it is capable of handling large-scale optimization problems; and 2) it can better deal with problems with complex structures. Such framework is very natural and attractive to modularity identification because of two distinctive properties of complex networks: 1) large scale, e.g., consisting of thousands or even millions of nodes; and 2) highly structured, e.g., community structure. We propose a novel node grouping scheme to dynamically decompose a complex network into several smaller scale subnetworks. A subpopulation with a local moving (LM) scheme is then employed to detect communities in each subnetwork. Subsequently, for the subpopulation corresponding to a subnetwork, a representative individual from each of the other subpopulations is selected and concatenated to every individual in this subpopulation to construct a combined population of complete n-dimensional candidate solutions to evolve this subnetwork. In addition, an adapted Kernighan-Lin (KL) [23] moving scheme is adopted for the whole network to improve the results.

Using the aforementioned CC framework, our algorithm can obtain partitions with better modularity values than other EA-based algorithms. However, due to the resolution limit (RL) of the modularity [24], it is known that some communities in large real-world complex networks are deem to be undetectable by modularity maximization methods because the maximum modularity does not necessarily indicate the optimal or natural partition of networks. This resolution limit (RL) of the modularity is serious since it will distort the true community structure of networks. We tackle this problem by incorporating recursive partitioning scheme [25] into CoCoMi, that is, we apply CoCoMi iteratively to the communities found by the previous run of CoCoMi to detect small communities hidden in a large community.

We first thoroughly evaluate the performance of CoCoMi on a large set of 12 benchmark complex networks including 5 medium to large scale networks and three resolution limit benchmark networks in comparison to other state-of-the-art MI algorithms. We then apply to identify disease modules that differentiate low and high grade glioma tumours to help us to understand the molecular mechanisms that underpin the progression of gliomas, a major kind of the brain cancer.

II. Related work

Various kinds of algorithms have been proposed to detect the community structures in networks during the past decade. The most well-known algorithm perhaps is the Girvan-Newman (GN) algorithm [26] which is a divisive method that iteratively removes the edges with the greatness betweenness value based on betweenness centrality. Later, Newman [27] presented an agglomerative hierarchical clustering method based on the greedy optimization of the network modularity. This method iteratively joins communities of nodes in pairs and chooses the join with the greatest increase in the modularity at each step. Moreover, based on the original strategies, its faster version [4] was proposed by using some shortcuts and some sophisticated data structures.

Duch and Arenas [8] proposed a divisive algorithm referred as to EO to detect communities in complex networks. This algorithm uses a heuristic search based on the extremal optimization to optimize the network modularity to detect communities in networks, and basically operates optimizing a global variable by improving extremal local variables that involve coevolutionary avalanches.

Pujol et al. [9] presented an agglomerative hierarchical clustering algorithm named PBD to identify communities in networks. PBD is based on the spectral analysis and modularity optimization to achieve a good compromise between efficiency and accuracy in clustering networks.

Pizzuti [11] proposed a multiobjective genetic algorithm named MOGA-Net to uncover community structure in networks. This MOGA-Net is not to optimize the network modularity but to optimize two objective functions in which one objective is to maximize the number of connections inside each community while the other one is to minimize the number of interconnections between communities in a partition of networks.

Gong et al. [10] presented a memetic community detection algorithm named Meme-Net to detect communities in networks. This Meme-Net optimizes the quality function named modularity density including a tunable parameter that allows one to explore the network at different resolutions. Moreover, Meme-Net combines a genetic algorithm with a hill-climbing strategy as the local search procedure.

We have also proposed a Cooperative Co-evolutionary Differential Evolution based Community Detection (CCDECD) algorithm [13] recently. Based on the Differential Evolution framework, a randomized grouping scheme is designed to decompose a complex network into smaller subnetworks in CCDECD. However, it is worth mentioning that, different from CoCoMi, the grouping scheme in CCDECD does not take the connectivity information into account, which might generate lots of isolated nodes. Moreover, a new mutation operator is also designed in CCDECD to make use of connectivity information of networks to improve the search ability. Unlike CoCoMi, no local search operator is used in CCDECD. To validate its performance, CCDECD is applied to several benchmark social networks and a protein-protein interaction networks.

However, it is worth mentioning that, all the algorithms mentioned above, especially those based on evolutionary algorithms, suffered from the scalability problem. For example, the largest network (Erdös co-author network) tackled in [11], [10], [12], [13] consists only 6927 nodes. The results, e.g., the modularity values obtained by those algorithms, are even worse than those of deterministic algorithms.

III. The proposed algorithm

We give the details of the proposed CoCoMi algorithm. The key steps of CoCoMi are summarized in Algorithm 1 and its flowchart is shown in Fig. 1. The representation, fitness function, initialization, node grouping scheme, two moving operators and recursive partitioning scheme of the algorithm are detailed in the following sections.

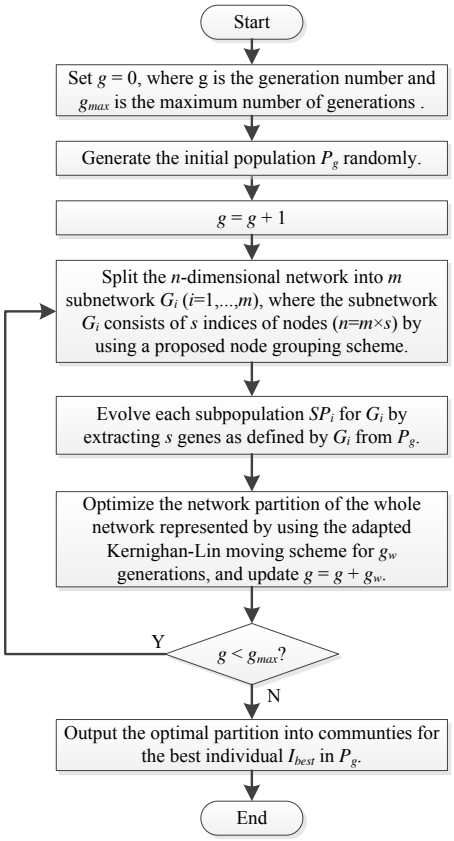

Fig. 1. The flowchart of CoCoMi.

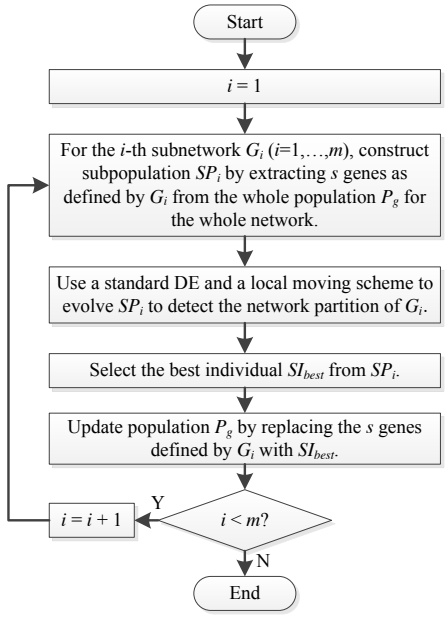

Fig. 2. The evolving of subnetworks.

Algorithm 1 CoCoMi algorithm

1: Set g as the generation number and $g_{\max}$ as the maximum number of generations.
2: $g = 0$.
3: Randomly initialize the population P_g.
4: **while** $g < g_{\max}$ **do**
5: $g = g + 1$.
6: Split the n-dimensional complex network into m subnetworks G_i $(i = 1, \ldots, m)$, where G_i consists of s indices of nodes $(n = m \times s)$ using a proposed node grouping scheme (See Section III-C and Section III-E for details).
7: **for** $i = 1$ to m **do**
8: Construct subpopulation SP_i for G_i by extracting s genes as defined by G_i from P_g.
9: Optimize the network partition by using a standard DE and a local moving scheme to maximize the modularity of G_i with g_s generations for subpopulation SP_i (See Section III-F for details).
10: $g = g + g_s$.
11: Select the best individual SI_{best} from SP_i.
12: Update population P_g by replacing the s genes defined by G_i with SI_{best}.
13: **end for**
14: Optimize the network partition of the whole network represented by using the adapted Kernighan-Lin moving scheme with g_w generations (See Section III-G for details).
15: $g = g + g_w$.
16: **end while**
17: Output the optimal partition of communities for the best individual I_{best} in P_g.

A. Standard Differential Evolution

Differential evolution (DE) proposed by Storn and Price [28], [29], [30] is a very simple yet efficient evolutionary algorithm. It starts the search with an initial population containing NP individuals randomly sampled from the search space, and then one individual called the target vector in the population is used to generate a mutant vector by the mutation operation. One popular mutation strategy "rand/1" [31], [32], [33], [34] employed in CoCoMi is as follows:

$$\vec{v}_i = \vec{x}_{r_1} + F \times (\vec{x}_{r_2} - \vec{x}_{r_3}), \tag{1}$$

where $i \in \{1, 2, \cdots, NP\}$, r_1, r_2 and r_3 are integers randomly selected from $\{1, 2, \cdots, NP\}$ and satisfy $r_1 \neq r_2 \neq r_3 \neq i$, the scaling factor F is usually a real number between 0 and 1, the decision vector $\vec{x}_i = (x_{i1}, x_{i2}, ..., x_{in})$ with n decision variables is the individual in the population and also called the target vector, and $\vec{v}_i = (v_{i1}, v_{i2}, ..., v_{in})$ is the mutant vector.

After mutation, all the components of the mutant vector are checked whether they violate the boundary constraints. In the mutant vector $\vec{v}_i$, if the jth component v_{ij} violates the boundary constraint, then v_{ij} is reflected back from the violated boundary constraint as follows [33]:

$$v_{ij} = \begin{cases} 2LB_j - v_{ij}, & if\ v_{ij} < LB_j \\ 2UB_j - v_{ij}, & if\ v_{ij} > UB_j \\ v_{ij}, & otherwise \end{cases} \quad (2)$$

where LB_i and UB_i are the lower and upper bounds of the ith decision variable x_i, respectively.

Subsequently, the crossover operation is implemented on the mutant vector $\vec{v}_i$ and the target vector $\vec{x}_i$ to generate a trial vector $\vec{u}_i$. And a commonly used crossover operation is the binomial crossover which is executed as follows:

$$u_{ij} = \begin{cases} v_{ij}, & if\ rand \leq CR\ or\ j = j_{rand} \\ x_{ij}, & otherwise \end{cases} \quad (3)$$

where $i \in \{1,2,\cdots,NP\}$, $j \in \{1,2,\cdots,n\}$, $rand$ is a uniformly distributed random number between 0 and 1, j_{rand} is a randomly selected integer from 1 to n, CR is the crossover control parameter, and u_{ij} is the jth component of the trial vector $\vec{u}_i$.

Finally, the target vector $\vec{x}_i$ is compared with the trial vector $\vec{u}_i$ in terms of their objective function values (i.e., $f(\vec{x}_i)$ and $f(\vec{u}_i)$) considering the maximum optimization and the better one survives into the next generation:

$$\vec{x}_i = \begin{cases} \vec{u}_i, & if\ f(\vec{u}_i) \geq f(\vec{x}_i) \\ \vec{x}_i, & otherwise \end{cases} \quad (4)$$

B. Individual representation

CoCoMi uses the community identifier-based representation [35] to represent individuals in the population for modularity identification in networks. For a graph $G = (V, E)$ with n nodes, the kth individual in the population is a vector that consists of n genes $\vec{x}_k = \{x_1, x_2, \ldots, x_n\}$ in which each gene x_i can be assigned an allele value j in the range $\{1, 2, \ldots, n\}$. The gene and allele represent the node and the community identifier (commID) of communities in G respectively. Thus, $x_i = j$ denotes that the node i belongs to the community whose commID is j, and nodes i and d belong to the same community if $x_i = x_d$. Since at most n communities exist in G the maximum value of commID is n. Fig. 3 provides an illustration example.

In the above representation, all the communities in G and all the nodes belonging to each community can be identified straightforward from individuals in the population. For instance, in Fig. 3 the individual {1,1,1,1,2,2,2} represents the partition {{1,2,3,4},{5,6,7}} of the graph G. The community identifier-based representation is very simple and effective. Moreover, the number of communities is automatically determined by the individuals and no decoding process is required in this representation.

C. Fitness function

Newman and Girvan [26] proposed the modularity to measure the strength of the community structure found by algorithms. The modularity is a very efficient quality metric for estimating the partitioning of a network into communities and has been used by many modularity identification algorithms recently ([4], [27], [35]).

CoCoMi also employs the modularity which is maximized as the fitness function to evaluate individuals in the population. The modularity is defined as follows [35].

$$Q = \sum_{j=1}^{n_c} \left[\frac{l_j}{L} - \left(\frac{d_j}{2L} \right)^2 \right], \quad (5)$$

where j is the commID, n_c is the total number of communities, l_j is the number of edges in community j, L is the total number of edges in the network and d_j is the sum of the degrees of all nodes in community j.

Using the network in Fig. 3 as an example, there are 7 nodes and 10 edges in total (i.e., $L = 10$). Considering the community structure found by using the genotype as shown in Fig. 3, there are two communities with commID from 1 to 2 (i.e., $n_c = 2$ and $j = 1, 2$). In the first community ($j = 1$), there are $l_1 = 6$ edges in this community, four nodes with the degree of 3, 3, 3 and 4 respectively, which gives $d_1 = 3 + 3 + 3 + 4 = 13$, and 6 edges in total, while, in the second community ($j = 2$), there are $l_2 = 3$ edges in this community, three nodes each of which has a degree of 2, 2 and 3 respectively, which gives $d_2 = 2 + 2 + 3 = 7$, and 3 edges in total. By plugging these numbers into Equation (5), we can calculate the modularity of this community structure of this network $Q = 0.355$.

D. Initialization

At the beginning of the initialization process, CoCoMi places each node into a random community by assigning a random commID and generates individuals in the initial population. However, such random generation of individuals is likely to generate individuals that consist of isolated nodes having no connectivity with each other in the original network. Considering that nodes in the same community should connect with each other and in the simple case are neighbors, the initialization process proposed in [35] is used to overcome the above drawbacks. The process works as follows: once an individual is generated, some nodes in an individual are randomly selected and their commIDs are assigned to all of their neighbors.

E. Node grouping scheme

Similar to the random grouping framework in [16], the main idea behind CoCoMi is also to split a large network into m subnetworks, and then identify communities in each of them with a standard EA. Previously, we have already applied the random grouping framework in [13] to community detection. However, we found that the random grouping scheme is not suitable for modularity identification in complex networks because it does not employ connectivity information of networks when splitting them into subnetworks. As a result, the generated subnetworks may have many isolated nodes, which will deteriorate the search performance of EA for modularity identification in networks. Therefore, we introduce a novel node grouping scheme which utilizes the neighborhood information of nodes when splitting networks into subnetworks. This node grouping scheme works as Algorithm 2 and its flowchart is shown in Fig. 4.

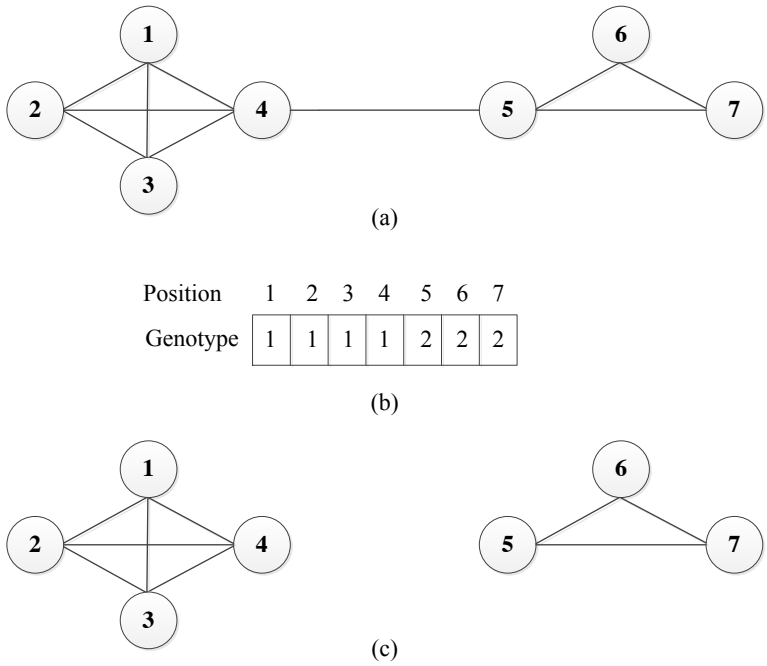

Fig. 3. (a) A network with seven nodes. (b) A genotype of the network. (c) The community structure identified by the genotype.

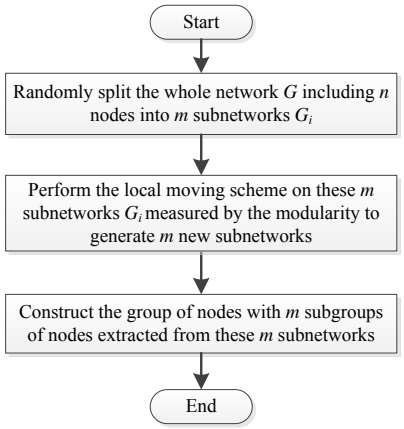

Fig. 4. The flowchart of the node grouping scheme.

Algorithm 2 Node grouping scheme

1: Split the whole network G with n nodes randomly into m subnetworks $\{G_1,\ldots,G_m\}$ with each subnetwork G_i containing s nodes and edges connecting them.
2: Perform the local moving scheme on these m subnetworks $\{G_1,\ldots,G_m\}$ measured by the modularity and generate m new subnetworks $\{G'_1,\ldots,G'_m\}$.
3: Construct the group of nodes containing m subgroups in which each subgroup consists of nodes extracted from each subnetwork G'_i $(i = 1,\ldots,m)$.

In Algorithm 2, firstly the whole network G including n nodes is split into m subnetworks $\{G_1,\ldots,G_m\}$ in which each subnetwork G_i contains s nodes (i.e., $n = m \times s$) and edges connecting these nodes. Then, the local moving scheme described in Section III-F is performed on these m subnetworks, and outputs m new subnetworks $\{G'_1,\ldots,G'_m\}$ with high quality measured by the modularity (see Section III-C for details). According to these m new subnetworks, we construct the group of nodes containing m subgroups in which each subgroup consists of nodes extracted from each subnetwork G'_i $(i = 1,\ldots,m)$. By the above node grouping scheme, CoCoMi can divide all nodes of a network into several subgroups of nodes with high quality, that is, those tightly interacting nodes will be grouped together, which will ultimately generate a better grouping of nodes than putting nodes with no any connections together in networks.

F. Local moving scheme

The local moving (LM) scheme [36] employed in CoCoMi works as follows. First, we find all its neighbors of each node i in the network. Then, for its each neighbor j, we generate a new partition by moving node i from its community into the community of its neighbor j. Subsequently, the increase of modularity of the new partition is computed. After having computed all the increase of modularity for all neighbors of node i, we select the best move for node i, that is, moving node i into the community for which the increase of modularity of the new partition is positive and maximum. However, if all the above generated partitions for node i have no positive

Algorithm 3 Adapted Kernighan-Lin moving scheme

Set PG as the best found partition of a network and Q_{best} as its corresponding modularity.
Compute the modularity $Q(PC)$ of the partition PC for the current individual.
$PG = PC$
$Q_{best} = Q(PC)$
repeat
  for $i = 1$ to n **do**
    Perform the globally best move (see Section III-G for details) of the ith node in PC to generate a new partition PN.
    Compute the modularity $Q(PN)$ of PN.
    if $Q(PN) > Q_{best}$ **then**
      $PG = PN$
      $Q_{best} = Q(PN)$
    end if
    if PG is not improved in the last k moves **then**
      break
    end if
  end for
  $PC = PG$
until no improvement of PC is found.

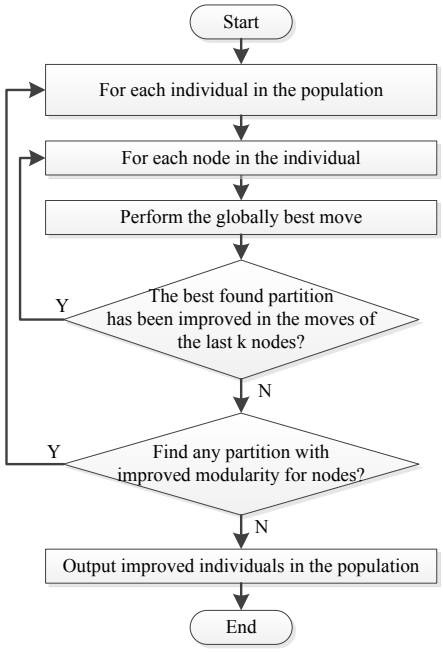

Fig. 5. The flowchart of the adapted Kernighan-Lin moving scheme.

increase of modularity, we do not move node i and make it stay in its original community. The above process is performed repeatedly through all nodes in randomized order and selects the best move for all nodes, until no further increase of modularity can be achieved for any node in the network.

G. Adapted Kernighan-Lin moving scheme

CoCoMi employs an adapted Kernighan-Lin (KL) moving scheme [23], [36] to optimize the partition of the whole network after the CC operation. The adapted KL moving scheme including an inner loop and an outer loop is implemented as described in Algorithm 3. In the inner loop, this scheme first performs the globally best move only once for each node of every individual in the population. Similar to the local best move of a node obtained by the local moving scheme in Section III-F, the globally best move of a node is performed as follows. For each node i in the network, we first find all its neighbors and then move node i from its community into the community of its each neighbor j to generate a new partition, after which the increase of modularity of the new partition is calculated. Subsequently, we compute all the increase of modularity for all neighbors of node i and then move node i into the community for which the increase of modularity of the new partition is maximum. In the above globally best move, each node is moved only once, but is not restricted to increase the modularity of the new generated partition for this move of the node which is a little different from the local best move in Section III-F. After all the nodes have finished the globally best move, the above process is then performed repeatedly from the best found partition (i.e., the partition with the maximum modularity) until the best found partition has not been improved in the moves of the last k nodes in which $k = 10\log_2 |n|$ and n is the number of nodes in the network since experimental results in [37] showed that terminating the inner loop at that time was much more efficient and rarely less effective. In the outer loop, each individual in the population performs the process in the inner loop iteratively until no any partition with improved modularity for nodes can be found. The schematic representation of the adapted KL moving scheme is shown in Fig. 5.

H. Recursive partitioning scheme

CoCoMi employs a recursive partitioning (RP) scheme [38] as follows to solve the resolution limit problem of the modularity [24].

After obtaining the partitioning of the network into communities, CoCoMi ignores the links among communities and considers each community as a disjoint subnetwork. Then, CoCoMi is recursively applied to optimize all these disjoint subnetworks separately to partition each subnetwork into smaller communities until all smaller communities are detected in this subnetwork. To determine when to terminate the recursive partitioning, we adopt the stop strategy proposed in [38] which works as follows. During the process of recursive partitioning of each subnetwork, we first record the modularity of the new partition obtained by CoCoMi. If the modularity Q is smaller than a threshold $Q_{\min}$, which indicates that the subnetwork has no strong sub-community structure, we then stop the recursive partitioning of the subnetwork. In our implementation, we use $Q_{\min} = 3.0$ since most real-world networks have $Q \leq 0.3$ [26]. However, some subnetworks might have modularity values larger than $Q_{\min}$ by chance [38] due to sparsity but not because of strong sub-community structure, therefore, the recursive partitioning is not necessary. To address this problem, we use the same Monte-Carlo method in [38] to estimate the statistical significance of the modularity

of the new partition as follows. We first generate N random networks in which the edges are randomly rewired but each node has the same degree as in the subnetwork [26], and then use CoCoMi to partition them to compute the mean and variance of the modularity of each random network. The modularity of the ith random network is denoted as Q_i. Subsequently, we calculate the $Z\text{-}score$ [38] of the modularity Q as follows:

$$Z\text{-}score = \frac{Q - \langle Q \rangle}{\sigma_Q}, \tag{6}$$

where $\langle Q \rangle$ and σ_Q are the mean and standard deviation of the modularity of all these previously generated random subnetworks Q_i, $i = 1, \cdots, N$. A high $Z\text{-}score$ means the modularity of the partition of the subnetwork is less likely to occur by chance, which indicates strong sub-community structure. According to [38], we adopt $Z\text{-}score = 2$ corresponding to a p-value of 0.05 as the cutoff, that is, if $Z\text{-}score$ is not smaller than 2, then we continue the recursive partitioning process; otherwise, we stop the recursive partitioning process.

IV. Benchmark Experiments

A. Experimental settings

In this section, the performance of CoCoMi is evaluated on nine well-known real-world networks and three synthetic networks. CoCoMi is implemented in MATLAB 7.0 and all the experiments are performed on Windows XP SP2 with a Pentium Dual-Core 2.5GHz processor and 2.0GB RAM.

The parameters in CoCoMi are set as follows: the population size is 30; the maximum number of generations is $g_{\max} = 100$; the mutation rate for the global network mutation operator is set to be 0.2; for the standard DE in which the "rand/1" mutation operator is employed, the scaling factor is $F = 0.9$.

For comparison, we implement three EA-based module identification algorithms (e.g., CCDECD [13], MOGA-Net [11] and Meme-Net [10]), and their number of function evaluations is set to be the same as that of CoCoMi. We also adopted an MATLAB implementation of the Newman's fast algorithm described in [27] which we refer to as Fast-Nm from [39] for comparison. Except for Fast-Nm which is a deterministic algorithm, all the other four stochastic comparison algorithms (i.e., CoCoMi, CCDECD, MOGA-Net and Meme-Net) are executed 30 runs in all experiments.

B. Benchmark complex networks

We used twelve widely used benchmark networks, including nine real-world benchmark networks and three synthetic complex networks to evaluate the performance of CoCoMi. These nine real-world complex networks can be grouped into three categories: 1) four well-known small-scale benchmark networks with known community structures, which provide gold-standards, i.e., normalized mutual information (NMI) [40] defined as Equation (7), for the evaluation of the performance of detecting natural communities of CoCoMi; 2) five medium to large scale benchmark networks for the evaluation of the scalability of CoCoMi. The characteristics of these nine networks are summarized in Table I. Moreover, we also use another three well-known synthetic benchmark networks [24] (i.e., a Ring network, a Y-shaped network and a Ring K4 network) to evaluate the effectiveness of CoCoMi for solving the resolution limitation problem [24].

TABLE I
THE CHARACTERISTICS OF 12 REAL-WORLD AND SYNTHETIC NETWORKS.

Network	#Vertices	#Edges	Type
Ring	30	40	synthetic
Y-shaped	50	424	synthetic
Ring K4	64	112	synthetic
Karate	34	78	social
Dolphin	62	159	social
Books	105	441	social
Football	115	613	social
C.*elegans*	453	2040	biology
Email	1133	5451	social
Erdös	6927	11850	co-author
PGP	10680	24316	social
Cond-Mat	27519	116181	co-author

C. Performance metrics

1) Normalized Mutual Information (NMI): NMI [40] is an information-theoretic measure of the agreement between two partitions. Suppose that A denotes the real partition and B denotes a predicted partition in the network, we can define a confusion matrix $\boldsymbol{N}$, where the rows correspond to the real communities defined in A, and the columns correspond to the predicted communities found in B. The element N_{ij} of $\boldsymbol{N}$ is the number of nodes in the real community i that appear in the predicted community j. Based on the above definition of $\boldsymbol{N}$, NMI is defined as follows:

$$NMI(A,B) = \frac{-2\sum_{i=1}^{c_A}\sum_{j=1}^{c_B} N_{ij} log(\frac{N_{ij}n}{N_{i.}N_{.j}})}{\sum_{i=1}^{c_A} N_{i.} log(\frac{N_{i.}}{n}) + \sum_{j=1}^{c_B} N_{.j} log(\frac{N_{.j}}{n})}, \tag{7}$$

where c_A is the number of real communities in A and c_B is the number of predicted communities in B; $N_{i.}$ is the sum over row i and $N_{.j}$ is the sum over column j of matrix $\boldsymbol{N}$, and n is the total number of nodes in the network.

From the Equation (7), we can see that if A is equal to B, NMI takes its maximum value of 1. If B is completely different from A, NMI is equal to 0.

2) Modularity Q: For those medium to large scale benchmark networks, since they do not have known community structures, the modularity defined as Equation (5) in Section III-C is used as the performance metric to measure the quality of the community structures detected in these networks. As mentioned in the introduction of Section I, modularity suffers from the so-called resolution limit problem [24]. Therefore, the modularity should be seen as an performance metric to evaluate the optimization performance on the modularity (fitness function), rather than the performance of finding natural community divisions of the network.

D. Results and discussion

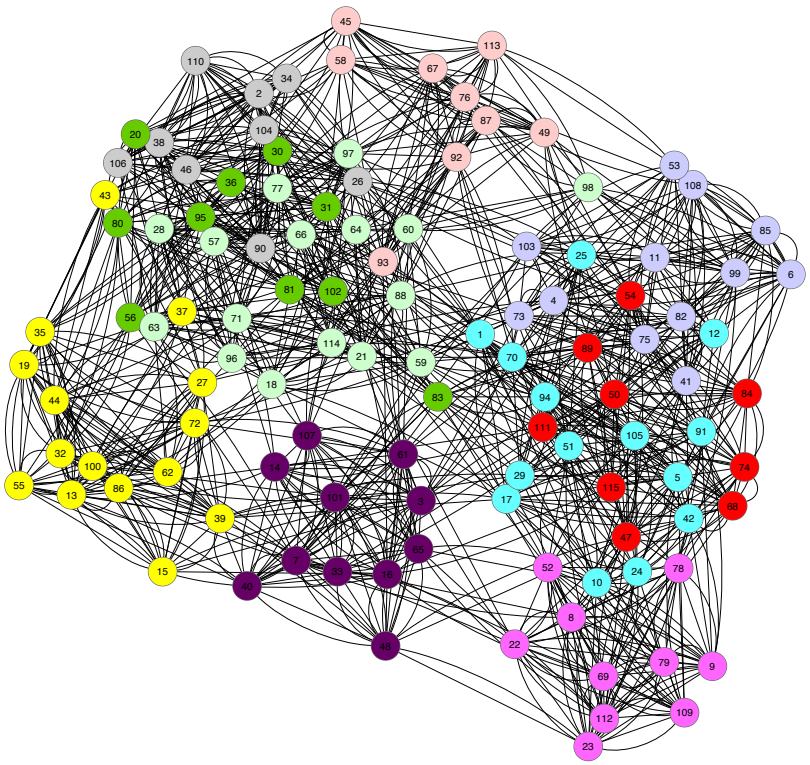

Fig. 6. The partition result of the football network.

1) Small-scale benchmark networks with known community structure: We first validate the performance of CoCoMi to detect natural communities on four small-scale benchmark networks with known community structures, i.e., the Zachary's karate club network [41] referred to as Karate network, the Dolphin network [42], the American college football network [41] referred to as Football network whose partition result is shown in Fig. 6, and the network of books on American politics [43] referred to as Books network.

We compare the performance of CoCoMi with that of other four modularity identification algorithms (i.e., Meme-Net [10], MOGA-Net [11], CCDECD [13] and Fast-Nm [27]). In the experiments, on these above four networks we executed CoCoMi 30 runs which is the same as that of the other three stochastic comparison algorithms (i.e., CCDECD, MOGA-Net and Meme-Net). The experimental results for these five algorithms are presented in Table II, in which the best, average and standard deviation values of Q and NMI (i.e., Q_{bst}, Q_{avg}, Q_{std}, NMI_{bst}, NMI_{avg} and NMI_{std}) are used to compare the performance of these five algorithms. In order to test the significant difference among results of these five algorithms, the statistical method called the analysis of variance (ANOVA) [44], [45], [46] is adopted in Table II. Note that for CCDECD, MOGA-Net, Meme-Net and Fast-Nm in Table II, we collected some results from literatures (i.e., [10], [11] and [13]), and run the codes of these four comparison algorithms to obtain the other results which could not be found from literatures.

From Table II, it can be seen that among all these five algorithms CoCoMi always obtained the best Q_{bst} and Q_{avg} values for the Karate, Dolphin, Football and Books networks. Moreover, with respect to the NMI's best value NMI_{bst} and average value NMI_{avg}, the results obtained by CoCoMi are equal or better than that obtained by the other four competitors (i.e., CCDECD, MOGA-Net, Meme-Net and Fast-Nm) on these Karate, Football, Dolphin and Books networks. From the statistical analysis by ANOVA, we can see that, for the Karate network, results of CoCoMi are significantly better than that of MOGA-Net and Fast-Nm. Since CoCoMi, CCDECD and Meme-Net obtain the same results on the Karate network, no significantly difference is detected by ANOVA among results of these three algorithms. Moreover, for the Dolphin, Football and Books networks, results of CoCoMi are significantly better than that of the other four comparison algorithms as indicated in Table II. Therefore, the above experimental results clearly show that CoCoMi can effectively identify modules in small-scale real-world networks.

2) Medium to large scale benchmark networks: In this section, five well-known median to large scale benchmark complex networks are used to further test the scalability of CoCoMi, i.e., the C. $elegans$ metabolic network [47] referred to as C. $elegans$ network, a university e-mail network [48] referred to as Email network, a network of users of the pretty good privacy (PGP) algorithm for secure information transactions [49] referred to as PGP network, the relationships between authors that shared a paper in cond-mat [50] referred to as Cond-Mat network and the Erdös collaboration network [51] referred to as Erdös network.

The performance of CoCoMi is compared with that of other five modularity identification algorithms (i.e., EO [8], PBD [9], LPA [52], SDJB [53] and SC [54]) on these above five

TABLE II
EXPERIMENTAL RESULTS OF THE COCOMI, MEME-NET, MOGA-NET AND FAST-NM ALGORITHMS FOR THE KARATE, DOLPHIN, FOOTBALL AND BOOKS NETWORKS. NOTE THAT THE BEST RESULT FOR EACH BENCHMARK NETWORK AMONG THE COMPARED ALGORITHMS IS HIGHLIGHTED IN BOLDFACE. THE ANALYSIS OF VARIANCE (ANOVA) IS PERFORMED AMONG RESULTS, AND RESULTS WITH ASTERISKS INDICATE THESE RESULTS ARE SIGNIFICANTLY DIFFERENT FROM RESULTS OF COCOMI.

Network	Algorithm	Q_{bst}	$Q_{avg} \pm Q_{std}$	NMI_{bst}	$NMI_{avg} \pm NMI_{std}$
Karate	CoCoMi	**0.420**	**0.420 ± 0.0000**	**0.687**	**0.687 ± 0.0000**
	CCDECD	**0.420**	**0.420 ± 0.0000**	**0.687**	**0.687 ± 0.0000**
	Meme-Net	**0.420**	**0.420 ± 0.0000**	**0.687**	**0.687 ± 0.0000**
	MOGA-Net	0.415	0.415 ± 0.0000*	0.602	0.602 ± 0.0000*
	Fast-Nm	0.381	0.381 ± 0.0000*	0.652	0.652 ± 0.0000*
Dolphin	CoCoMi	**0.5285**	**0.5267 ± 0.0031**	**0.930**	**0.884 ± 0.0282**
	CCDECD	0.5216	0.52078 ± 0.00026*	**0.930**	0.800 ± 0.040*
	Meme-Net	0.5191	0.5096 ± 0.0043*	0.586	0.569 ± 0.035*
	MOGA-Net	0.508	0.505 ± 0.009*	0.549	0.506 ± 0.046*
	Fast-Nm	0.496	0.496 ± 0.0000*	0.573	0.573 ± 0.0000*
Football	CoCoMi	**0.605**	**0.6042 ± 0.0011**	**0.932**	**0.898 ± 0.0076**
	CCDECD	**0.605**	0.60382 ± 0.00089*	0.930	0.891 ± 0.022*
	Meme-Net	0.603	0.5952 ± 0.010*	0.911	0.890 ± 0.033*
	MOGA-Net	0.522	0.515 ± 0.016*	0.798	0.775 ± 0.023*
	Fast-Nm	0.549	0.549 ± 0.0000*	0.762	0.762 ± 0.0000*
Books	CoCoMi	**0.5271**	**0.5266 ± 0.0012**	**0.560**	**0.559 ± 0.0018**
	CCDECD	0.5268	0.5256 ± 0.0012*	0.554	0.553 ± 0.0023*
	Meme-Net	0.5255	0.5222 ± 0.0029*	0.540	0.538 ± 0.0063*
	MOGA-Net	0.5207	0.518 ± 0.004*	0.544	0.536 ± 0.025*
	Fast-Nm	0.502	0.502 ± 0.0000*	0.531	0.531 ± 0.0000*

TABLE III
EXPERIMENTAL RESULTS OF THE COCOMI, LPA, SDJB, SC, PBD AND EO ALGORITHMS FOR THE C.*elegans*, EMAIL, ERDÖS, PGP AND COND-MAT NETWORKS. NOTE THAT THE BEST RESULT FOR EACH BENCHMARK NETWORK AMONG THE COMPARED ALGORITHMS IS HIGHLIGHTED IN BOLDFACE AND THE SYMBOL '-' INDICATES THAT THE RESULTS OF THE CORRESPONDING ALGORITHMS CANNOT BE FOUND FROM LITERATURES.

Network	CoCoMi	LPA	SDJB	SC	PBD	EO
C. *elegans*	**0.452**	**0.452**	**0.452**	0.450	0.4164	0.4342
Email	**0.583**	0.582	0.580	0.575	–	0.5738
Erdös	**0.718**	–	–	–	0.6817	0.6520
PGP	**0.886**	0.884	0.867	0.878	–	0.8459
Cond-Mat	**0.8129**	0.755	0.737	–	0.7251	0.6790

networks and CoCoMi is also executed 30 runs. We summarize the experimental results of these six algorithms in Table III, in which the best value of Q (i.e., Q_{bst}) are used as the metric. Note that in Table III the experimental results for LPA, SDJB, SC, EO and PBD algorithms are collected from literatures (i.e., [8], [9] and [36]) and the symbol '-' indicates that the results of these corresponding algorithms cannot be found from literatures.

From Table III, we can see that among all these six algorithms CoCoMi always obtained the best Q_{bst} value of 0.452, 0.583, 0.718, 0.886 and 0.8129 on the C.*elegans*, Email, Erdös, PGP and Cond-Mat networks respectively. Compared with the results of SC, PBD and EO algorithms on the C. *elegans* network, CoCoMi detected better partition with higher Q_{bst} which is the same as that of LPA and SDJB algorithms. On the other Email, Erdös, PGP and Cond-Mat networks, CoCoMi always found the best partitions with the highest Q_{bst} among all these six algorithms.

Moreover, we also compare the performance of CoCoMi with that of other four comparison algorithms (i.e., CCDECD, Meme-Net, MOGA-Net and Fast-Nm) on these above five networks (i.e., C.*elegans*, Email, Erdös, PGP and Cond-Mat networks), and summarize their results in Table IV, in terms of the best, average and standard deviation values of Q (i.e., Q_{bst}, Q_{avg} and Q_{std}). In addition, we also used the analysis of variance (ANOVA) [44] to determine significant difference among results of these five algorithms in Table IV. Note that in Table IV, we collected some results from literatures (i.e., [10], [11] and [13]) for these four comparison algorithms (i.e., CCDECD, MOGA-Net, Meme-Net and Fast-Nm), and generated the other results which could not be found from literatures by running the codes of these algorithms. From Table IV, we can see that CoCoMi can always achieve best partitions with the highest Q_{bst} and Q_{avg} among these five algorithms on these networks. Moreover, the statistical analysis by ANOVA also shows that results of CoCoMi are significantly better than that of the other four comparison algorithms on these five networks in Table IV. Thus, it can be concluded that CoCoMi has a very excellent scalability on module identification in real-world networks.

3) Resolution limit benchmark networks: To show the effectiveness of CoCoMi for solving the resolution limitation problem in modularity optimization, we also employ three well-known benchmark networks [24], i.e., a Ring network, a Y-shaped network and a Ring K4 network. The Ring network as shown in Fig. 7(a) includes 30 nodes and 40 edges. Moreover, this network is partitioned into 10 identical communities named K_3 and each one of them consists of 3 nodes and 3 edges. And the Y-shaped network as shown in Fig. 7(b) includes 50 nodes and 424 edges. From Fig. 7(b), we can see that this network is partitioned into 4 communities in which each one of the first two communities named K_{20}

TABLE IV
EXPERIMENTAL RESULTS OF THE COCOMI, CCDECD, MEME-NET, MOGA-NET AND FAST-NM ALGORITHMS FOR THE C.*elegans*, EMAIL, ERDÖS, PGP AND COND-MAT NETWORKS. NOTE THAT THE BEST RESULT FOR EACH BENCHMARK NETWORK AMONG THE COMPARED ALGORITHMS IS HIGHLIGHTED IN BOLDFACE. THE ANALYSIS OF VARIANCE (ANOVA) IS PERFORMED AMONG RESULTS, AND RESULTS WITH ASTERISKS INDICATE THESE RESULTS ARE SIGNIFICANTLY DIFFERENT FROM RESULTS OF COCOMI.

Network	Algorithm	Q_{bst}	$Q_{avg} \pm Q_{std}$
C.*elegans*	CoCoMi	**0.4520**	$\mathbf{0.4514 \pm 0.0004}$
	CCDECD	0.4507	$0.4486 \pm 0.0013^*$
	Meme-Net	0.4413	$0.4331 \pm 0.0038^*$
	MOGA-Net	0.4336	$0.4267 \pm 0.0079^*$
	Fast-Nm	0.4001	$0.4001 \pm 0.0000^*$
Email	CoCoMi	**0.5828**	$\mathbf{0.5820 \pm 0.0006}$
	CCDECD	0.5808	$0.5805 \pm 0.0013^*$
	Meme-Net	0.5596	$0.5563 \pm 0.0073^*$
	MOGA-Net	0.5272	$0.5196 \pm 0.0075^*$
	Fast-Nm	0.4796	$0.4796 \pm 0.0000^*$
Erdös	CoCoMi	**0.7180**	$\mathbf{0.7178 \pm 0.0001}$
	CCDECD	0.6980	$0.6873 \pm 0.0020^*$
	Meme-Net	0.6828	$0.6797 \pm 0.0093^*$
	MOGA-Net	0.6307	$0.6181 \pm 0.0103^*$
	Fast-Nm	0.6533	$0.6533 \pm 0.0000^*$
PGP	CoCoMi	**0.8859**	$\mathbf{0.8857 \pm 0.0002}$
	CCDECD	0.8842	$0.8838 \pm 0.0023^*$
	Meme-Net	0.8516	$0.8488 \pm 0.0035^*$
	MOGA-Net	0.8191	$0.8141 \pm 0.0035^*$
	Fast-Nm	0.7329	$0.7329 \pm 0.0000^*$
Cond-Mat	CoCoMi	**0.8129**	$\mathbf{0.8125 \pm 0.0003}$
	CCDECD	0.7706	$0.7703 \pm 0.0018^*$
	Meme-Net	0.7396	$0.7356 \pm 0.0086^*$
	MOGA-Net	0.7263	$0.7151 \pm 0.0105^*$
	Fast-Nm	0.6683	$0.6683 \pm 0.0000^*$

consists of 20 nodes and 190 edges, while each one of the last two communities named K_5 consists of 5 nodes and 20 edges. Similar to the Ring network, the Ring K4 network consists of 64 nodes and 112 edges. Moreover, it is partitioned into 16 identical communities and in each community there are 4 nodes and 6 edges.

In this section, we compare the performance of CoCoMi with that of the other four algorithms (i.e., CCDECD, Meme-Net, MOGA-Net and Fast-Nm). The results obtained by these four algorithms are summarized in Table V, in which the best, average and standard deviation values of the number of communities (denoted as g_{bst}, g_{avg} and g_{std} respectively) of the obtained partitions are used to compare the performance of these five algorithms and the analysis of variance (ANOVA) [44] is also employed to test significant difference among results.

From Table V, it can be seen that CoCoMi correctly and consistently detected all these 10, 4 and 16 communities in the Ring, Y-shaped and Ring K4 networks respectively in all runs. However, due to the resolution limit problem, the other four algorithms (i.e., CCDECD, Meme-Net, MOGA-Net and Fast-Nm) failed to detect all those true communities in some runs on these three networks, and achieved a partition in which some communities include two or more small true communities. In addition, from the statistical analysis by ANOVA, it indicates that results of CoCoMi are also significantly better than that of other three comparison algorithms (i.e., Meme-Net, MOGA-Net and Fast-Nm) on these three networks. Although no significant statistical difference exists between results of CCDECD and CoCoMi, CCDECD failed to find true partitions in several runs out of 30 runs on these three networks. Thus, we can conclude that CoCoMi overcomes the resolution limitation problems in modularity optimization in complex networks.

TABLE V
EXPERIMENTAL RESULTS OF THE COCOMI, CCDECD, MEME-NET, MOGA-NET AND FAST-NM ALGORITHMS FOR THE RING, Y-SHAPED AND RING K4 NETWORKS. NOTE THAT THE BEST RESULT FOR EACH BENCHMARK NETWORK AMONG THE COMPARED ALGORITHMS IS HIGHLIGHTED IN BOLDFACE. THE ANALYSIS OF VARIANCE (ANOVA) IS PERFORMED AMONG RESULTS, AND RESULTS WITH ASTERISKS INDICATE THESE RESULTS ARE SIGNIFICANTLY DIFFERENT FROM RESULTS OF COCOMI.

Network	Algorithm	g_{bst}	$g_{avg} \pm g_{std}$
Ring	CoCoMi	**10**	$\mathbf{10 \pm 0.0000}$
	CCDECD	**10**	9.9333 ± 0.2537
	Meme-Net	**10**	$9.2000 \pm 0.8469^*$
	MOGA-Net	7	$6.2333 \pm 0.4302^*$
	Fast-Nm	5	$5 \pm 0.0000^*$
Y-shaped	CoCoMi	**4**	$\mathbf{4 \pm 0.0000}$
	CCDECD	**4**	3.9000 ± 0.3051
	Meme-Net	**4**	$3.8333 \pm 0.3790^*$
	MOGA-Net	3	$3 \pm 0.0000^*$
	Fast-Nm	3	$3 \pm 0.0000^*$
Ring K4	CoCoMi	**16**	$\mathbf{16 \pm 0.0000}$
	CCDECD	**16**	15.9000 ± 0.3051
	Meme-Net	**16**	$15.3667 \pm 0.7649^*$
	MOGA-Net	10	$8.4667 \pm 0.8193^*$
	Fast-Nm	8	$8 \pm 0.0000^*$

TABLE VI
EXPERIMENTAL RESULTS OF THE COCOMI, COCOMI-NNG, COCOMI-NLS AND COCOMI-NKL ALGORITHMS ON THE EMAIL NETWORK. NOTE THAT THE BEST RESULT FOR EACH BENCHMARK NETWORK AMONG THE COMPARED ALGORITHMS IS HIGHLIGHTED IN BOLDFACE. THE ANALYSIS OF VARIANCE (ANOVA) IS PERFORMED AMONG RESULTS, AND RESULTS WITH ASTERISKS INDICATE THESE RESULTS ARE SIGNIFICANTLY DIFFERENT FROM RESULTS OF COCOMI.

Algorithm	Q_{bst}	$Q_{avg} \pm Q_{std}$
CoCoMi	**0.583**	$\mathbf{0.582 \pm 0.0006}$
CoCoMi-nNG	0.549	$0.541 \pm 0.0057^*$
CoCoMi-nLS	0.568	$0.561 \pm 0.0047^*$
CoCoMi-nKL	0.514	$0.501 \pm 0.0057^*$

E. Validation of schemes in CoCoMi

The CoCoMi algorithm employs three distinct schemes to optimise modularity for community detection, e.g., node grouping scheme (see Section III-E), local moving scheme (see Section III-F) and adapted Kernighan-Lin (KL) moving scheme (see Section III-G). To validate the efficiency of these above three schemes in CoCoMi, we designed three comparison algorithms. The first comparison algorithm was obtained by replacing the node grouping scheme in CoCoMi with the random grouping scheme as in CCDECD [13] and named as CoCoMi-nNG. By removing the local moving scheme in CoCoMi, we generated the second comparison algorithm termed as CoCoMi-nLS. The third comparison algorithm was constructed by deleting the adapted KL scheme in CoCoMi and termed as CoCoMi-nKL. We executed these three comparison algorithms 30 times on the above Email network [48],

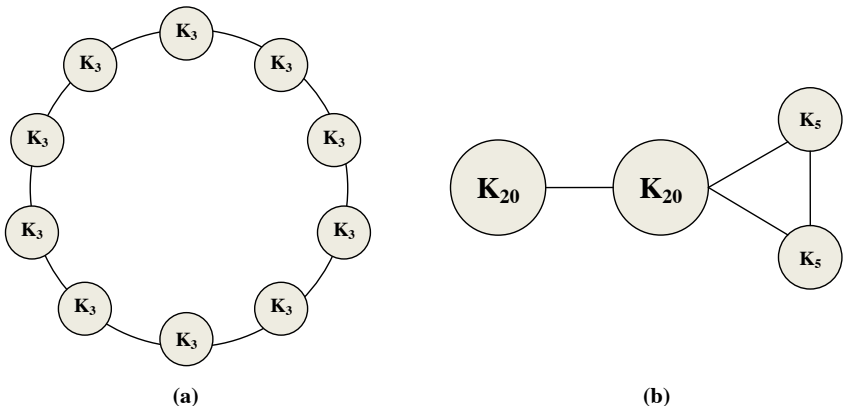

Fig. 7. (a) The Ring-shaped network. (b) The Y-shaped network.

and the experimental results and those results of CoCoMi are shown in Table VI in terms of the best, average and standard deviation values of Q (i.e., Q_{bst}, Q_{avg} and Q_{std}) and the significant difference among these results are also tested by the analysis of variance (ANOVA) [44].

From the Table VI, CoCoMi obtains the best Q_{bst}, Q_{avg} and Q_{std} compared with the other three algorithms on this Email network. Moreover, CoCoMi-nKL obtained the worst Q_{bst} and Q_{avg} among these four algorithms which shows the efficiency of the adapted KL scheme to improve the global search ability of CoCoMi. In addition, CoCoMi-nNG obtained the worse Q_{bst} and Q_{avg} than CoCoMi and CoCoMi-nLS, which demonstrates the efficiency of grouping a network into several subnetworks by using the node grouping schemes in CoCoMi. Furthermore, the bad Q_{bst} and Q_{avg} values obtained by CoCoMi-nLS indicate that the local search scheme can improve the local search ability of CoCoMi and increase the change to find the optimal partitions of communities in networks. The statistical analysis by ANOVA demonstrates that results of CoCoMi are also significantly better than that of the other three algorithms. Therefore, by using these above schemes, CoCoMi can detect community structures efficiently especially in large-scale complex networks.

V. Application to Glioma Tumour Disease Module Identification

"Disease module" is formally defined as a group of interacting components such as genes, proteins and metabolites that collectively contribute the development of disease in the biological network [55]. As shown in Fig. 8, the "disease module" is different from the other two distinct phenomena (i.e., the "topological module" and the "functional module"). A topological module shown in the Fig. 8(a) is a pure network property and represents a locally dense neighborhood in which nodes have a higher tendency to interact with each other than with nodes outside the neighborhood in a network. However, a functional module shown in the Fig. 8(b) represents the aggregation of nodes of similar or related function in a network. By using disease modules as biomarkers for disease diagnosis and prognosis, we can obtain better accuracy and reproducibility than those derived without network information. More importantly, by further investigation of disease modules we will be able to gain insights into the molecular mechanisms of disease, e.g., to identify the interacting cellular pathways or even the driver mutation that initialize the disease, which ultimately will lead to novel therapeutic targets.

Recently, many MI algorithms have been proposed for disease module identification [56], [57], [58], [59], [60], [61], [62], [63], [64]. According to local hypothesis, all cellular components in the same topological module are very likely to have the same molecular function and thus to be involved in the same disease [55]. Therefore, by identifying topological modules, we will be able to identify functional modules and then discover disease modules which overlap with topological modules.

Gliomas [65] are the most common brain tumour which make up 30% of all brain and central nervous system tumors and 80% of all malignant brain tumors. According to World Health Organization (WHO) classification system [66], gliomas can be classified into 4 grades, e.g., grades 1-4. The prognosis for patients with high grade, e.g., grade 4 gliomas (also called glioblastoma multiform) is generally poor, with less than a 12-month average survival after diagnosis. However, the prognosis of low grade gliomas is much more optimistic, with a median survival of 11.6 years [67]. However, 50% to 75% all these low grade glioma patients will inevitable progress to a higher grade and ultimately death. Is it possible to prevent low grade gliomas progress to high grades? To answer this question we need to understand the molecular mechanisms of glioma progression.

We apply CoCoMi to a Protein-Protein interaction network [68] to identify disease modules that can differentiate low grade glioma (grade II) and high grade Glioblastoma (grade IV). Our aim is to use these disease modules to unveil the molecular mechanisms of glioma progression.

A. Dataset and preprocessing

We downloaded the glioma gene expression dataset GSE4290 from NCBI GEO [69]. This dataset was collected by Henry Ford Hospital, which consists of 23 non-tumor samples and 157 tumor samples including 26 astrocytomas, 50 oligodendrogliomas and 81 glioblastomas. The mRNA expression

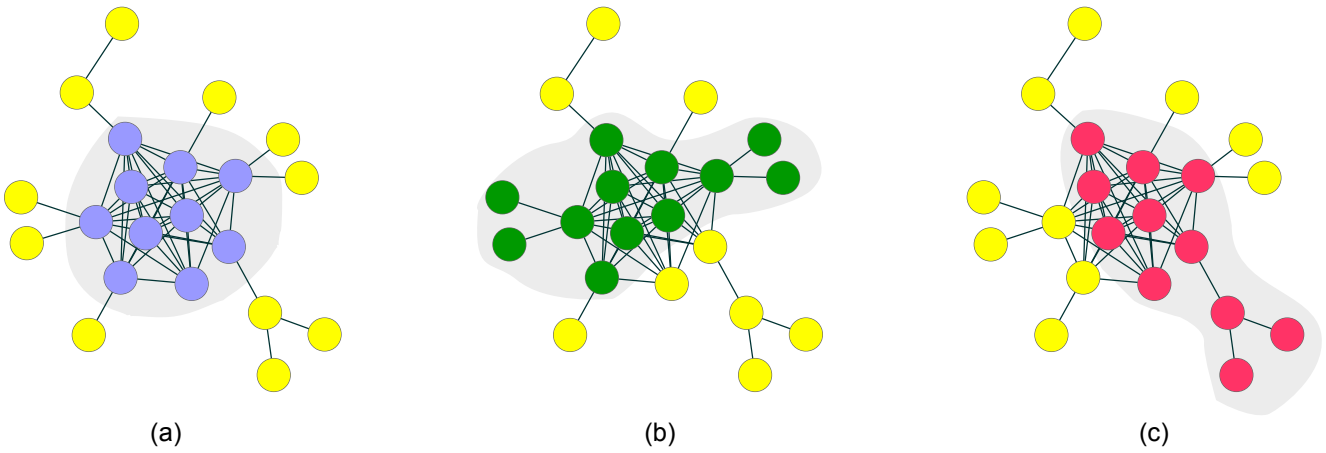

Fig. 8. (a) A topological module (shown as blue nodes). (b) A functional module (shown as green nodes). (c) A disease module (shown as red nodes).

profiling of the samples was conducted using the Affymetrix Human Genome U-133 plus 2.0 GeneChip. After preprocessing (i.e., normalization) the raw files, we compared the gene expression 81 grade IV astrocytoma or glioblastoma samples and 45 grade II astrocytoma samples. The comparisons were conducted using the R simpleaffy package. We defined significantly differently expressed (up and down-regulated) genes (SDEGs) as genes with fold changes greater than 1.5 and are significantly different between groups with a t-test p value < 0.001. In total, we selected 5756 genes as SDEGs. We then constructed a protein-protein interaction (PPI) network from the 5756 SDEGs using Michigan Molecular Interaction (MiMI) protein interactions databases [70], which merged and integrated a number of protein interactions databases such as BioGRID [71] and HPRD [72]. Those isolated SDEGs that do not have interactions with other SDEGs were filtered out. Several small networks with less than 10 nodes were also excluded. The resulting PPI network consists of 1423 nodes (SDEGs) and 3893 edges (PPI), and is called glioma grading PPI (GGPPI) network as shown in Fig. 9.

B. CoCoMi identifies modules with higher quality than other algorithms

Since we do not have any prior knowledge how the GGPPI network should be partitioned, we cannot use NMI to measure the performance of CoCoMi. However, the members of a high quality topological module in a biological network should have a similar or relevant biological function. Therefore, the quality of a module based on the biological function homogeneity can be evaluated based on Gene Ontology (GO) [73], which annotate the function of genes and group them into categories by some common biological properties. In order to give a more quantitative measure, we employ the functional linkage enrichment (FLE) score [74] which is defined as:

$$FLE = \sum_{i}^{N} (funsim_{avg,i} - funsim_{rand}), \qquad (8)$$

where i refers to the ith partitioned module, N is the total number of partitioned modules, $funsim_{avg,i}$ is the averaged $funsim$ score [75] of all gene pairs in the ith partitioned module, and $funsim_{rand}$ is the random $funsim$ score of a pair of genes in the genome. The $funsim$ score of a pair of genes is described in [75] and ranges from 0 to 1 in which a higher score represents stronger functional linkage between a pair of genes. Moreover, the $funsim$ score considers all GO terms associated with the two genes and the specificity of each GO term. According to the definition of the FLE score, if a partition of a network has a higher FLE score it will have more number of modules with enriched functionally related genes.

Using FLE, we compare the quality of the network partitions obtained CoCoMi with those of other algorithms, e.g., CCDECD, Meme-Net, MOGA-Net and Fast-Nm used in our previous experiments. We also use MCODE which is a popular module identification algorithm [76] as implemented as a plugin of Cytoscape [77]. Finally, we remove the recursive partitioning (RP) scheme from CoCoMi and obtain a comparison algorithm named as CoCoMi-nRP, to investigate its contribution to the disease module identification. As shown in Fig. 10(a), the partition of modules obtained by CoCoMi has the highest FLE score of 104.3709, which is higher than the partition of modules obtained by the other algorithms. It is interesting to note from Fig. 10(a) that, without recursive partitioning scheme, the FLE score of CoCoMi-nRP is far worse than that of CoCoMi, CCDECD, Meme-Net and MCODE but similar to that of Fast-Nm and MOGA-Net.

According to the definition of the FLE score, the increase of FLE score may be resulted from random partitioning when some modules enriched with functionally related genes are randomly partitioned into more modules [74]. Since we introduced the recursive partitioning scheme to solve the resolution limit problem, it is particularly necessary to check whether the higher FLE of the modules identified by our CoCoMi algorithm are due to the random partitioning or biologically plausible. To do this, we use CoCoMi without

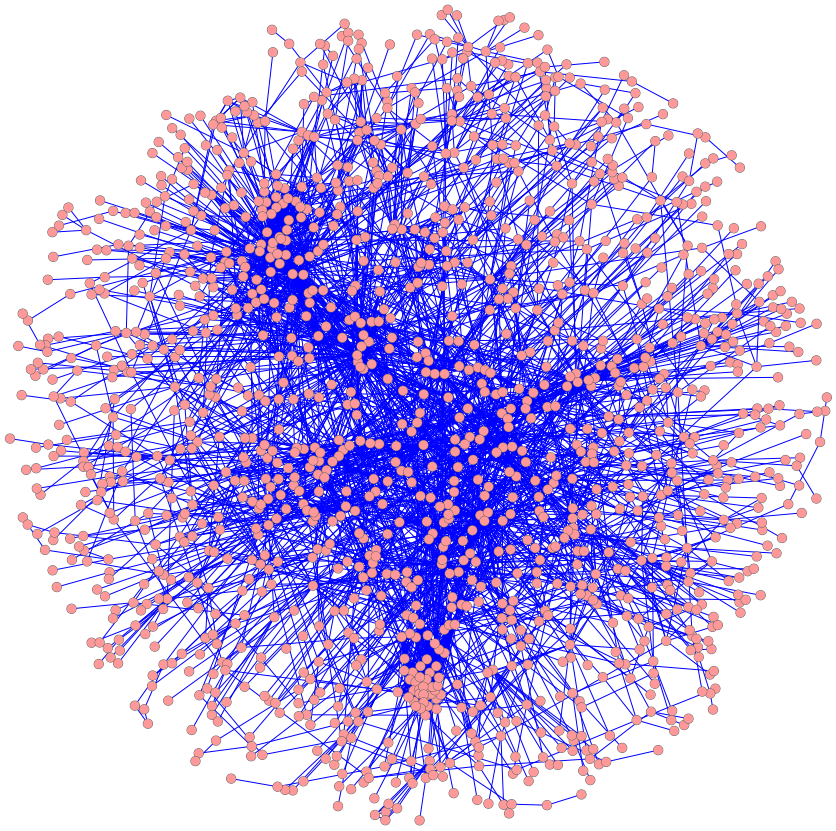

Fig. 9. The protein-protein interaction network that can differentiate grade II and grade IV gliomas.

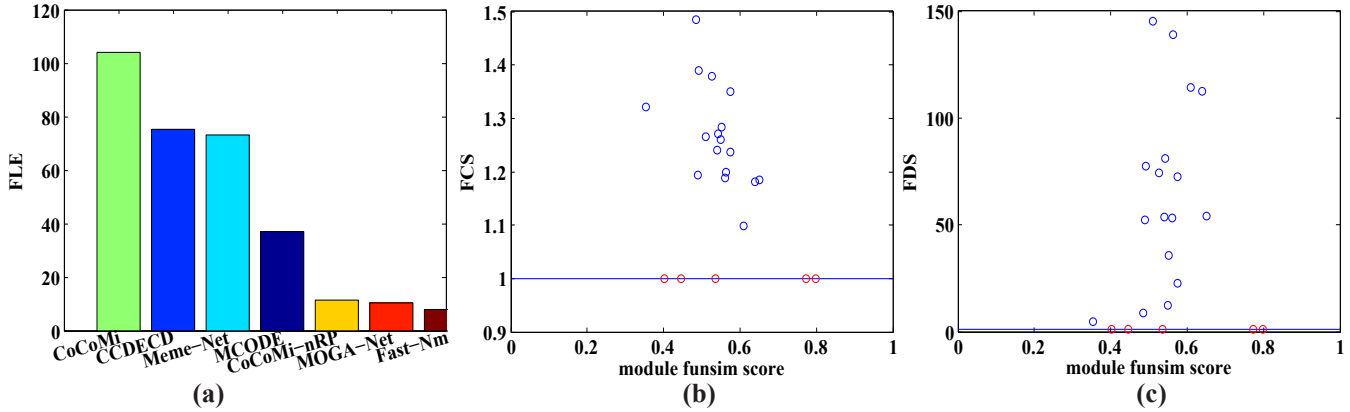

Fig. 10. (a) The FLE scores of partitions of modules obtained by the CoCoMi, CoCoMi-nRP and other algorithms on the GGPPI network. (b) The FCS of modules of CoCoMi-nRP on the GGPPI network. (c) The FDS of modules of CoCoMi-nRP on the GGPPI network. Red circles indicate modules that cannot be no further partitioned by the RP scheme in CoCoMi.

recursive partitioning scheme (i.e., CoCoMi-nRP) to identify modules as baseline modules, which represent optimal community structure in terms of modularity value. Using these baseline modules, we then adopt two additional scores, i.e., the functional cohesiveness score (FCS) and the functional distinctiveness score (FDS) [74] to quantify their biological relevance of further partitioning. Suppose a module m is one module in the partition obtained by CoCoMi-nRP, and after using the RP scheme it is further partitioned into K modules. Thus FCS is defined as follows:

$$FCS = \frac{\sum_{i}^{K} funsim_i}{K \times funsim_m}, \tag{9}$$

where $funsim_m$ is the averaged $funsim$ score between genes in the module m, and $funsim_i$ is the averaged $funsim$ score between genes in the module i, and K is the total number of newly generated modules from the module m. Similarly, FDS

is defined as follows:

$$FDS = \frac{\sum_{i \le K, j \le K, i \ne j} \left(\frac{funsim_i + funsim_j}{funsim_{i,j} + funsim_{j,i}} \right)}{K(K+1)/2}, \quad (10)$$

where $funsim_i$ and $funsim_j$ are the $funsim$ scores of module i and module j respectively, while $funsim_{i,j}$ is the averaged $funsim$ score between genes in module i and genes in module j, and $funsim_{j,i}$ equals to $funsim_{i,j}$.

According to the above definitions, the FCS evaluates the relative enrichment of functional-related genes in the new modules to that in the parent module, while the FDS assesses whether the newly generated modules have relatively distinctive functions between each other. Moreover, if a module includes several functionally distinctive smaller modules, then a successful partition for this module will result in both the FCS and FDS larger than 1, indicating that genes inside the new modules are more functionally cohesively related to each other while the genes between these new modules are functionally distinctive from each other. In contrast, a random partitioning for a module will result in both FCS and FDS equal to 1, while an unsuccessful partitioning for a module will result in both FCS and FDS smaller than 1.

Fig. 10(b) and (c) show the FCS and FDS of all 22 modules obtained by CoCoMi-nRP on the GGPPI network respectively. As we can see from Fig. 10(b), 5 modules marked by red circles have the same FCS of 1 and their corresponding FDS also have the same value of 1 that are also marked by red circles in Fig. 10(c). The reason is that these 5 modules are subjected to random partitioning and can not be further partitioned by the RP scheme in CoCoMi. Thus both the FCS and FDS of these 5 modules are equal to 1 which exhibits the excellent performance of CoCoMi-nRP even without using the RP scheme to some extent when identifying modules in the real-world biological networks. For all the other 17 modules subjected for further partitioning by the RP scheme in CoCoMi, both their FCS and FDS are larger than 1 as shown in Fig. 10(b) and Fig. 10(c). Therefore, we can conclude that the significant increase of the FLE score by the RP scheme in CoCoMi is resulted from successfully partitioning of the original modules obtained by CoCoMi-nRP into functionally distinctive ones, and CoCoMi by using the RP scheme is able to solve the resolution limit problem effectively when identifying modules in the real-world biological networks.

C. CoCoMi identified medically relevant disease modules

Since we are interested in the underlying molecular mechanisms of glioma progression, we need to analyse those modules that are most relevant to cancer progression. To this end, we selected eight modules with known cancer genes and further analyze them by using the Gene Ontology (GO) [78]. The detailed information of these eight modules are summarized in Table VII. The connections of these modules and their GO annotation are illustrated in Fig. 11.

Some of the disease modules in Table VII are supported by literatures. For example, gliomas usually show genetic aberrations of genes for cell cycle regulatory process. Indeed, we found module 3 corresponds to regulation of cell cycle. Previous studies also show two genes in this module, e.g., CDK4 and CDK6 (activation of cyclin-dependent kinases 4 and 6), occurs in the majority of glioblastoma multiforme (GBM) tumors [79]. The down regulation of CDKN2C (cyclin-dependent kinase inhibitor 2C), a cell growth regulator that controls cell cycle, has already been found to be associated with GBM [79]. CCND2 (cyclin D2) has also been found to play a critical role in GBM [80]. It is interesting to see module 3 is a small size module with only 5 genes of which 4 genes are known cancer genes. The only unknown cancer gene is SERTAD1 (SERTA domain containing 1), which has been found to play an essential role in developmental and pathological neuron death. Although not directly associated with high grade glioma, SERTAD1 might be an interesting gene for further investigations. It is worth mentioning that, this small but medically significant disease module cannot be detected by other modularity maximisation algorithms which suffer from the resolution limit problem.

In module 8, we also found a well known cancer gene EGFR (epidermal growth factor receptor) gene, of which the amplification and overexpression are a feature of GBM but are rare in low-grade gliomas [81]. It is interesting to find that MUC1 (Mucin 1, cell surface associated) and ERBB1 epidermal growth factor receptor) genes, which are overexpressed in breast cancer [82] are also in the module. Their roles in the progression of glioma are unclear and required further investigation.

Some disease modules are novel. For example, module 1 includes BRCA1 (breast cancer 1, early onset) and BRCA2 (breast cancer 2, early onset), which are known cancer genes for breast and ovarian cancers. Although BRCA1 and BRCA2 have not been confirmed as cancer genes for gliomas, recent research has shown genetic links between breast cancer and glioblastoma [83]. Indeed, researchers have proposed the idea to use PARP (poly (ADP-ribose) polymerase), a chemical inhibitor which was developed for breast or ovarian cancer patients carrying BRCA1 or BRCA2 genes mutations, to treat glioblastoma [84]. Similarity, module 8 with the Hox genes such as HOXD13, HOXA11 and HOXD11 is also novel. The roles of the Hox genes in oncogenesis of other cancers, e.g., Ovarian cancer, have been confirmed by many researchers [85], however, our findings indicate that they might also play an important role in the progression of glioma.

VI. Conclusion

To summarize, this paper has introduced the cooperative co-evolution framework with a novel node grouping scheme and local search operators into EA-based module identification to handle medium to large scale complex networks. In addition, we have also incorporated the recursive partitioning scheme to tackle the resolution limit problem. We have tested our CoCoMi on several benchmark real-world complex networks. Compared with other state-of-the-art algorithms, the experimental results have demonstrated that CoCoMi can effectively handle large scale complex network up to $27,519$ nodes. Our experimental results on a set of synthetic benchmark networks also show that CoCoMi has also successfully addressed the resolution limit problem.

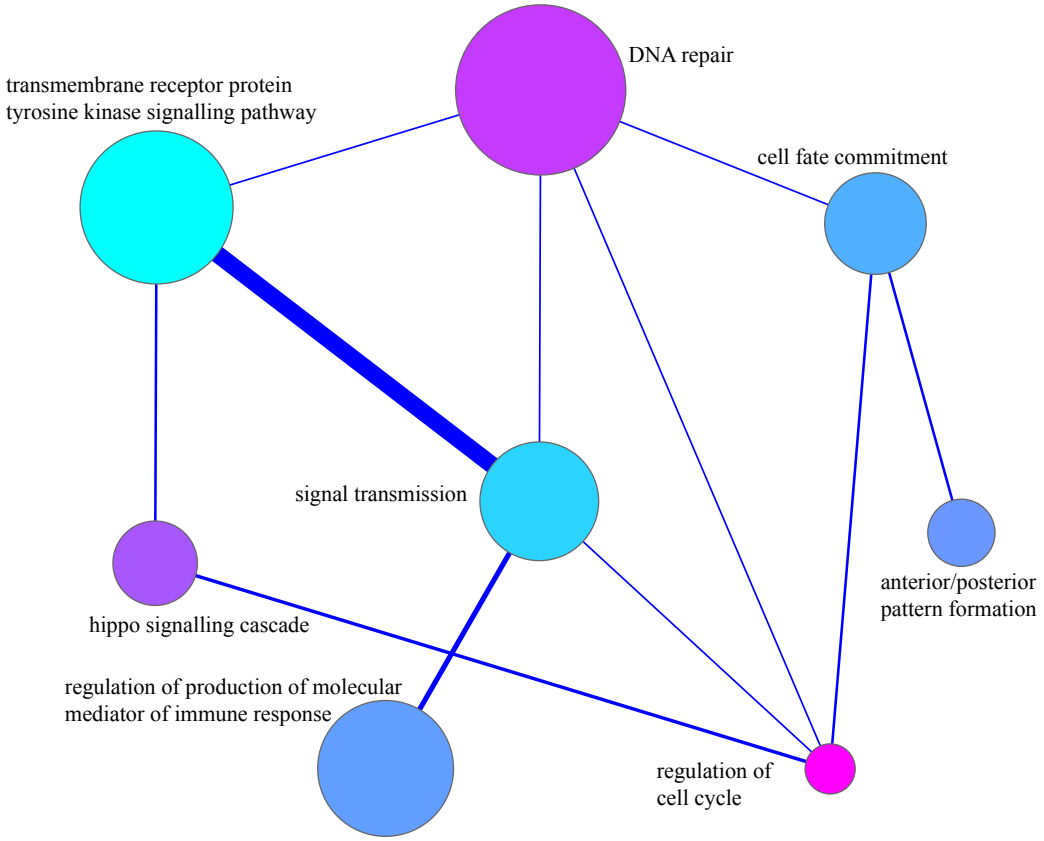

Fig. 11. Visualization of the eight disease modules identified by CoCoMi from the GGPPI network and their inter-module connectivity. Nodes represent identified modules, node size represents module size and node color represents (log-transformed) fold-change in average module gene expression level compared with normal patient samples (red - increase in average expression, green - decrease in average expression, lavender - no change in average expression). Edge widths are proportional to connectivity (i.e., number of interacting protein pairs) between module pairs.

We have also applied CoCoMi to glioma protein interaction networks to investigate the molecular mechanisms that underpin glioma progress from low to high grades. Our results show that disease modules identified using CoCoMi contain well-known cancer genes which are relevant gliomas. It is interesting to note that those unknown cancer genes in the same module and those novel disease modules might also play to important roles in glioma progression. Such results has shown that CoCoMi has the potential to open whole new areas for biological investigation that may lead to significant advances in knowledge of diseases such as glioma.

It is worth pointing out that the largest network used to evaluate our CoCoMi algorithm is Cond-Mat [50] with $27,519$ nodes and $116,181$ edges. Compared with the largest social networks available on Stanford Large Network Dataset Collection [86], Friendster social network, which has $65,608,366$ nodes and $1,806,067,135$ edges, our used largest network Cond-Mat is still relative small (although it is larger than the smallest network on Stanford Large Network Dataset Collection). How to scale up EA-based module identification algorithms to handle even larger complex networks is an open question to the researchers in the field.

VII. Acknowledgment

We would like to thank Paul and Yuanbi Ramsay for their financial support for Qiang Huang. Zexuan Zhu and Shan He are supported by the Royal Society International Exchanges 2011 NSFC cost share scheme (IE111069). Jing Liu, Ke Tang, Shan He and Xin Yao are supported by an EU FP7-PEOPLE-2009-IRSES project under Nature Inspired Computation and its Applications (NICaiA) (247619). Shan He is sponsored by The Royal Society International Exchanges project (BIR002) and Basic Research Program of Shenzhen (JCYJ20130401170306880).

References

[1] J. Scott, *Social network analysis: a handbook*. London: Sage Publications, 2000.
[2] C.-H. Yeh and C.-Y. Yang, "Social networks and asset price dynamics," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 387–399, 2015.
[3] A. C. Gavin and *et. al.*, "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, pp. 631–636, 2006.
[4] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, p. 066111, 2004.
[5] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 82–97, 2014.

TABLE VII
EXPERIMENTAL RESULTS OF COCOMI BY USING GO ON THE GGPPI NETWORK.

Module	Size	Genes	GO p-value	GO function	GO percentage	Number of known cancer genes	Names of known cancer genes
Module 1	26	RAD54L, BCAT1, RAD54B, IFI16,SHFM1, FANCD2, NPM1, USP1, TOP2A, H2AFX, NUFIP1, MSH6, RAD51AP1, UBE3A, RAD51, BRCA1, BCCIP, BRIP1, SWAP70, TGIF2, BRCA2, BARD1, ANTXR1, RBBP8, DDX39, RAD51L1	$2.6300e$-21	DNA repair	0.6800	7	FANCD2, NPM1, MSH6, BRCA1, BRIP1, BRCA2, RAD51L1
Module 2	17	PIK3R1, SPRED1, PLCG1, SNAP91, SHCBP1, TUB, MET, DOK1, LYN, SHC1, KIT, FCGR2B, FCGR2A, RET, GRAP, CALD1, WASF3	$1.5531e$-07	signal transmission	0.5000	5	PIK3R1, MET, KIT, FCGR2B, RET
Module 3	5	CDK4, CDKN2C, CCND2, SERTAD1, CDK6	$3.2679e$-07	regulation of cell cycle	0.6000	4	CDK4, CDKN2C, CCND2, CDK6
Module 4	10	HOXD13, HOXA11, HOXA7, HOXA10, HOXB13, HOXA5, HOXD11, HOXA2, HOXD4, MEIS1	$8.6793e$-14	anterior/posterior pattern formation	0.8000	3	HOXD13, HOXA11, HOXD11
Module 5	12	TEAD3, MYST4, TEAD2, LEF1, RUNX1, WWTR1, ELF4, MSX2, TEAD4, NRARP, YAP1, RUNX2	$1.3337e$-11	hippo signaling cascade	0.4545	3	MYST4, RUNX1, ELF4
Module 6	14	TWIST1, FAM46A, HES5, MYOD1, HOXA1, ZIC1, GLI3, AEBP1, EGLN2, NR2F2, ATOH1, CHIC2, TCF3, BCL11B	$8.9725e$-10	cell fate commitment	0.7692	3	CHIC2, TCF3, BCL11B
Module 7	20	NR3C2, HIVEP3, BIRC3, TIFA, NTRK2, CDCA3, TRAF2, TNFAIP3, TRAF5, HSPA4, TNFRSF12A, SOX9, ADCYAP1R1, SLC30A7, MALT1, TNFRSF4, CD40, ABLIM1, ITPK1, SIVA1	$1.1353e$-08	regulation of production of molecular mediator of immune response	0.6000	3	BIRC3, TNFAIP3, MALT1
Module 8	23	GALNT2, FGF13, RNF41, HOXC10, ERRFI1, RIN2, CPM, ALCAM, PKIA, FGF12, ERBB4, NRG3, EGF, MAPK8IP2, TP53RK, ADAM12, ERBB2, ARF4, DOK6, EGFR, SCAMP1, CISH, MUC1	$1.3986e$-07	transmembrane receptor protein tyrosine kinase signaling pathway	0.2727	3	ERBB2, EGFR, MUC1

[6] A. Bailey, M. Ventresca, and B. Ombuki-Berman, "Genetic programming for the automatic inference of graph models for complex networks," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 405–419, 2014.

[7] B. H. Good, Y. Montjoye, and A. Clauset, "Performance of modularity maximization in practical contexts," *Physical Review E*, vol. 81, no. 4, p. 046106, 2010.

[8] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, p. 027104, 2005.

[9] J. M. Pujol, J. Béjar, and J. Delgado, "Clustering algorithm for determing community structure in large networks," *Physical Review E*, vol. 74, no. 1, p. 016107, 2006.

[10] M. Gong, B. Fu, L. Jiao, and H. Du, "Memetic algorithm for community detection in networks," *Physical Review E*, vol. 84, no. 5, p. 056101, 2011.

[11] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex network," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 3, pp. 418–430, 2012.

[12] G. Jia, Z. Cai, M. Musolesi, Y. Wang, D. A. Tennant, R. Weber, J. K. Heath, and S. He, "Community detection in social and biological networks using differential evolution," in *Proceedings of the 6th international conference on Learing and Intelligent OptimizatioN (LION 6)*, vol. 7492, 2012.

[13] Q. Huang, T. White, G. Jia, M. Musolesi, N. Turan, K. Tang, S. He, J. K. Heath, and X. Yao, "Community detection using cooperative co-

evolutionary differential evolution," *The 12th International Conference on Parallel Problem Solving From Nature (PPSN XII)*, vol. 7492, pp. 235–244, 2012.

[14] W. Song, Y. Wang, H. X. Li, and Z. Cai, "Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 414–431, 2015.

[15] Y. Wang and Z. Cai, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 117–134, 2012.

[16] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.

[17] M. N. Omidvar, Y. Mei, X. Li, and Y. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2014.

[18] Y. Mei, X. Li, and Y. Yao, "Cooperative co-evolution with route distance grouping for large-scale capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 435–449, 2014.

[19] A. Hertz and D. Kobler, "A framework for the description of evolutionary algorithms," *European Journal of Operational Research*, vol. 126, no. 2000, pp. 1–12, 2000.

[20] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[21] X. Lai, Y. Zhou, J. He, and J. Zhang, "Performance analysis on evolutionary algorithms for the minimum label spanning tree problem," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 860–872, 2014.

[22] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[23] B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *Bell System Technical Journal*, vol. 49, no. 291-307, 1970.

[24] S. Fortunato and M. Barthélemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[25] G. Xu, L. Bennett, L. Papageorgiou, and S. Tsoka, "Module detection in complex networks using integer optimization," *Algorithms for Molecular Biology*, vol. 5, p. 36, 2010.

[26] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, 2004.

[27] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, p. 066133, 2004.

[28] R. Storn and K. Price, "Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," *Technical Report*, pp. TR–95–012, 1995.

[29] ——, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[30] A. S. Ruhul, M. E. Saber, and R. Tapabrata, "Differential evolution with dynamic parameters selection for optimization problem," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 5, pp. 689–707, 2015.

[31] E. Mezura-Montes, M. Miranda-Varela, and R. Gómez-Ramón, "Differential evolution in constrained numerical optimization: an empirical study," *Information Sciences*, vol. 180, no. 22, pp. 4223–4262, 2010.

[32] F. Neri and V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61–106, 2010.

[33] G. Jia, Y. Wang, Z. Cai, and Y. Jin, "An improved (μ + λ)-constrained differential evolution for constrained optimization," *Information Sciences*, vol. 222, pp. 302–322, 2013.

[34] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, "Parameter control in evolutionary algorithms: trends and challenges," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 167–187, 2015.

[35] M. Tasgin and H. Bingol, "Community detection in complex networks using genetic algorithm," in *Proceedings of the European Conference on Complex Systems (ECCS 2006)*, 2006.

[36] R. Rotta and A. Noack, "Multilevel local search algorithms for modularity clustering," *ACM Journal of Experimental Algorithmics*, vol. 16, no. 2, 2011.

[37] R. Rotta, "A multi-level algorithm for modularity graph clustering," Master's thesis, Brandenburg University of Technology, 2008.

[38] J. Ruan and W. Zhang, "Identifying network communities with a high resolution," *Physical Review E*, vol. 77, p. 016104, 2008.

[39] E. L. Martelot and C. Hankin, "Fast multi-scale detection of overlapping communities using local criteria," *Computing*, vol. 96, pp. 1011–1027, 2014.

[40] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics*, vol. 2005, no. 09, p. P09008, 2005.

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

[42] D. Lusseau and M. E. J. Newman, "Identifying the role that animals play in their social networks," in *Proceedings of the Royal Society of London Series B: Biological Sciences*, vol. 271, 2004.

[43] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.

[44] R. A. Fisher, *Statistical methods for research workers*, 13th ed. Hafner Publishing Company, New York, 1967.

[45] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour: a case study on the CEC2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

[46] G. Santafe, I. Inza, and J. A. Lozano, "Dealing with the evaluation of supervised classification algorithms," *Artificial Intelligence Review*, vol. 44, no. 4, pp. 467–508, 2015.

[47] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, pp. 651–654, 2000.

[48] R. Guimerá, 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, p. 065103(R), 2003.

[49] X. Guardiola, R. Guimerá, A. Arenas, A. Díaz-Guilera, D. Streib, and L. A. N. Amaral, "Macro- and micro-structure of trust networks," *arXiv:cond-mat/0206240*, 2002.

[50] M. E. J. Newman, "Scientific collaboration networks. i. network construction and fundamental results," *Physical Review E*, vol. 64, p. 016131, 2001.

[51] V. Batagel and P. D. F. Ion, "Some analyses of erdos collaboration graph," *Social Networks*, vol. 22, no. 2, pp. 173–186, 2000.

[52] X. Liu and T. Murata, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 389, pp. 1493–1500, 2010.

[53] Y. Sun, B. Danila, K. Josico, and K. Bassler, "Improved community structure detection using a modified fine-tuning strategy," *Europhysics Letters*, vol. 86, no. 2, p. 28004, 2009.

[54] P. Schuetz and A. Caflisch, "Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement," *Physical Review E*, vol. 77, p. 046112, 2008.

[55] A.-L. Barabási, N. Gulbahce, and J. Loscalzo, "Network medicine: a network-based approach to human disease," *Nature Reviews Genetics*, vol. 12, pp. 56–68, 2011.

[56] M. A. Lones, S. L. Smith, J. E. Alty, E. L. Stuart, K. L. Possin, D. R. S. Jamieson, and A. M. Tyrrell, "Evolving classifiers to recognize the movement characteristics of parkinson's disease patients," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 559–576, 2014.

[57] Y. Liu, D. A. Tennant, J. K. Heath, and S. He, "Disease module identification from an integrated transcriptomic and interactomic network using evolutionary community extraction," in *proceedings of the 17th Annual International Conference on Research in Computational Molecular Biology (RECOMB 2013)*. Beijing, China: Springer, 2013, pp. 823–830.

[58] Y. Liu, D. A. Tennant, Z. Zhu, J. K. Heath, X. Yao, and S. He, "DiME: A scalable disease module identification algorithm with application to glioma progression," *PLoS One*, vol. 9, no. 2, p. e86693, 2014.

[59] T. Gong, J. Xuan, C. Wang, H. Li, E. Hoffman, R. Clarke, and Y. Wang, "Gene module identification from microarray data using nonnegative independent component analysis," *Gene Regulation and Systems Biology*, vol. 1, pp. 349–363, 2007.

[60] X. Shen, L. Yi, Y. Yi, J. Yang, T. He, and X. Hu, "Dynamic identifying protein functional modules based on adaptive density modularity in protein-protein interaction networks," *BMC Bioinformatics*, vol. 16, no. Suppl. 12, p. S5, 2015.
[61] Y. Wang and X. Qian, "Functional module identification in protein interaction networks by interaction patterns," *Bioinformatics*, vol. 30, no. 1, pp. 81–93, 2014.
[62] P. Pei and A. Zhang, "A 'seed-refine' algorithm for detecting protein complexes from protein interaction data," *IEEE Transactions on Nanobioscience*, vol. 6, no. 1, pp. 43–50, 2007.
[63] J. Swafford, M. Nicolau, E. Hemberg, and A. Brabazon, "Comparing methods for module identification in grammatical evolution," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation (GECCO 2012)*. New York, NY, USA: ACM, 2012, pp. 823–830.
[64] J. Wang, Y. Zuo, Y. G. Man, I. Avital, A. Stojadinovic, M. Liu, X. Yang, R. S. Varghese, M. G. Tadesse, and H. W. Ressom, "Pathway and network approaches for identification of cancer signature markers from omics data," *Journal of Cancer*, vol. 6, no. 1, pp. 54–65, 2015.
[65] M. Jhanwar-Uniyal, M. Labagnara, M. Friedman, A. Kwasnicki, and R. Murali, "Glioblastoma: molecular pathways, stem cells and therapeutic targets," *Cancers*, vol. 7, no. 2, pp. 538–555, 2015.
[66] A. Tefferi, J. Thiele, and J. W. Vardiman, "The 2008 world health organization classification system for myeloproliferative neoplasms: order out of chaos," *Cancers*, vol. 115, no. 17, pp. 3842–3847, 2009.
[67] H. Ohgaki and P. Kleihues, "Population-based studies on incidence, survival rates, and genetic alterations in astrocytic and oligodendroglial gliomas," *Journal of Neuropathol and Experimental Neurology*, vol. 64, no. 6, pp. 479–489, 2005.
[68] A. Vinayagam, J. Zirin, C. Roesel, Y. Hu, B. Yilmazel, A. A. Samsonova, R. A. Neumüller, S. E. Mohr, and N. Perrimon, "Integrating protein-protein interaction networks with phenotypes reveals signs of interactions," *Nature Methods*, vol. 11, no. 1, pp. 94–99, 2014.
[69] T. Barrett, "Ncbi geo: archive for functional genomics data sets update," *Nucleic Acids Research*, vol. 1, no. D1, pp. D991–D995, 2013.
[70] M. Jayapandian, A. Chapman, V. Tarcea, C. Yu, A. Elkiss, A. Ianni, B. Liu, A. Nandi, C. Santos, P. Andrews, B. Athey, D. States, and H. Jagadish, "Michigan molecular interactions (mimi): putting the jigsaw puzzle together," *Nucleic Acids Research*, vol. 35, pp. D566–D571, 2007.
[71] C. Stark, B. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, "Biogrid: a general repository for interaction datasets," *Nucleic Acids Research*, vol. 34, pp. D535–D539, 2006.
[72] S. Peri, J. Navarro, R. Amanchy, T. Kristiansen, C. Jonnalagadda, V. Surendranath, V. Niranjan, B. Muthusamy, T. Gandhi, M. Gronborg, and et al., "Development of human protein reference database as an initial platform for approaching systems biology in humans," *Genome Research*, vol. 13, no. 10, pp. 2363–2371, 2003.
[73] J. A. Blake, "Ten quick tips for using the gene ontology," *PLOS Computational Biology*, vol. 9, no. 11, p. e1003343, 2013.
[74] S. Sun, X. Dong, Y. Fu, and W. Tian, "An iterative network partition algorithm for accurate identification of dense network modules," *Nucleic Acids Research*, vol. 40, no. 3, p. e18, 2012.
[75] A. Schlicker, F. S. Domingues, J. Rahnenführer, and T. Lengauer, "A new measure for functional similarity of gene products based on gene ontology," *BMC Bioinformatics*, vol. 7, p. 302, 2006.
[76] G. Bader and C. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, pp. 1–27, 2003.
[77] J. Morris, L. Apeltsin, A. Newman, J. Baumbach, T. Wittkop, G. Su, G. Bader, and T. Ferrin, "Development of human protein reference database as an initial platform for approaching systems biology in humans," *BMC Bioinformatics*, vol. 12, pp. 1–14, 2011.
[78] M. Ashburner and *et. al.*, "Gene ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.
[79] W. R. Wiedemeyer and *et. al.*, "Pattern of retinoblastoma pathway inactivation dictates response to cdk4/6 inhibition in gbm," *Proceedings of the National Academy of Sciences*, vol. 107, pp. 11 501–11 506, 2010.
[80] R. Koyama-Nasu and *et. al.*, "The critical role of cyclin d2 in cell cycle progression and tumorigenicity of glioblastoma stem cells," *Oncogene*, vol. 32, pp. 3840–3845, 2013.
[81] K. J. Hatanpaa, S. Burma, D. Zhao, and A. A. Habib, "Epidermal growth factor receptor in glioma: signal transduction, neuropathology, imaging, and radioresistance," *Neoplasia*, vol. 12, pp. 675–684, 2010.
[82] A. Baruch and *et. al.*, "The breast cancer-associated muc1 gene generates both a receptor and its cognate binding protein," *Cancer Research*, vol. 59, p. 1552, 1999.
[83] S. B. Elmariah, J. Huse, B. Mason, P. Leroux, and R. A. Lustig, "Multicentric glioblastoma multiforme in a patient with brca-1 invasive breast cancer," *The Breast Journal*, vol. 12, no. 5, pp. 470–474, 2006. [Online]. Available: http://dx.doi.org/10.1111/j.1075-122X.2006.00307.x
[84] A. Chalmers, "PARP inhibitors and glioblastoma – a match made in heaven?" *Oncology News*, vol. 5, pp. 182–184, 2011.
[85] N. Shah and S. Sukumar, "The Hox genes and their roles in oncogenesis," *Nature Reviews Cancer*, vol. 10, pp. 361–371, 2010.
[86] J. Leskovec. [Online]. Available: http://snap.stanford.edu/data/