ACTIVE MODULE IDENTIFICATION IN BIOLOGICAL NETWORKS

by

WEIQI CHEN

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
The University of Birmingham
August 2018

ABSTRACT

This thesis addresses the problem of active module identification in biological networks. Active module identification is a research topic in network biology that aims to identify regions in network showing striking changes in activity. It is often associated with a given cellular response and expected to reveal dynamic and process-specific information.

The key research questions for this thesis are the practical formulations of active module identification problem, the design of effective, efficient and robust algorithms to identify active modules, and the right way to interpret identified active module.

This thesis contributes by proposing three different algorithm frameworks to address the research question from three different aspects. It first explores an integrated approach of combining both gene differential expression and differential correlation, formulates it as a multi-objective problem, and solves it on both simulated data and real world data. Then the thesis investigates a novel approach that brings in prior knowledge of biological process, and balances between pure data-driven search and prior information guidance. Finally, the thesis presents a brand new framework of identifying active module and topological communities simultaneously using evolutionary multitasking, accompanied with a series of task-specific algorithm designs and improvements, and provides a new way of integrating topological information to help the interpretation of active module.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

This chapter gives an introduction of active module identification in biological networks as the research topic of this thesis and an overview of the whole thesis. The rest of this chapter is organised as follows. Section 1.1 discusses the general research background of active module identification. Section 1.2 presents the research questions this thesis is aiming at. Section 1.3 provides the outline of the subsequent chapters. Section 1.4 and Section 1.5 list the contributions and publication from this thesis.

1.1 Background

Active module identification is a research area in network biology, a discipline that applies knowledge and approaches in network theory to biological data and tries to reveal the underlining mechanisms of biological activities by abstract them using network models.

The increasing interest in network biology is driven by the fast development of high-throughput biological data collection technologies. Exponential amount of data accumulate year by year concerning the complete DNA sequence of an organism's genome, the information of an organism's RNA transcripts under certain conditions, the entire set of proteins produced by organism, the molecular metabolite profiles that reflects specific cellular process, and a bunch of other types of information measuring the biological components and processes. A main challenge for researchers is how to cope with this unprecedentedly huge amount of data and dig for the information that precisely reflect the targeted states, mechanisms or activities of the organisms against background noises.

Network biology offers a highly abstract model of networks to characterise various levels of biological systems and provides insights into the intrinsic characteristics of these systems through the utilisation of concepts and methodologies in graph theory [6, 5]. Graph theory has been applied to study many complex systems such as the social networks, the transportation networks, or the Internet. It has successfully shown that many complex systems share the same set of essential architectural features and behaviour patterns that help reveal sights into the evolution and operation mechanisms of the systems [60].

Studies on network biology mainly focus on the following aspects:

- **The construction of networks from biological data.** The quality of network representation, i.e. whether the constructed network faithfully reflects the activities and interactions among biological components represented by this network, directly decides the qualities of all following research based on it. Widely adopted types of biological networks are protein-protein interaction networks, metabolic networks, regulatory networks and gene co-expression networks. More studies construct networks by integrating existing interaction database with conditional specific gene expression profiles in order to construct more interesting and informative networks.

- **The structural features of biological networks.** Biological networks share a bunch of structural features and behaviour patterns with other complex systems. Research on these properties has revealed some insights into the relationships between biological components, the formation of biological systems, the reaction process in response to internal or external stimuli. Commonly studies structural features include degree distribution, clustering coefficient, shortest path and small-world property [79], scale-free property [4], modularity [65] and network robustness [1].

- **The interpretation of biological meaning from network analysis.** Studies on the network models of biological system eventually have to fall back to the interpretation of biological meaning. Whether an identified structural module associates with certain functional groups, whether an interesting path in the network corresponds to some metabolic pathway, or whether a change in the topological parameters of the network indicates a cellular response, all have to be validated. There have been a number of database storing the experimental validated or deduced information concerning the relationships, functions, pathways in biological systems, e.g. the BioGRID [9] for protein, chemical, and genetic interactions or the KEGG [41] as an encyclopedia of genes and genomes.

Active module, first defined and formulated by Ideker in 2002 [34] is a region in network that shows striking changes in molecular activity or phenotypic signatures associated with a given cellular response [54]. It is expected to reveal dynamic and process-specific information that is correlated with cellular or disease states.

A number of computational techniques have been developed to identify active modules, mainly falling in three categories: significant-area-search methods [34, 51, 87, 26, 15] that often formulate active module identification as a maximum scoring subgraph identification problem, diffusion flow and network propagation methods [84] that model the influence or information flow in biological network as the diffusion process of fluid or heat flow in a network of pipes, and clustering-based methods [71, 70, 52] that use biclustering to find a subset of genes only showing activity under a subset of experimental conditions.

The exact quantitative formulation of active module varies from problem to problem. It can be based on node score annotation [34], or edge score [26], or a combination of both [87, 51]. Statistics used to generate the score also differs. As the formulated problem is often NP-hard, heuristic optimisation [34, 33, 43, 56, 48, 45] that aims to find high scoring region but does not guarantee a maximum score is widely used as the search algorithm. Nevertheless, exact approaches [15, 92, 3] using mathematical programming have also been developed.

1.2 Research Questions

The following key research questions clarify the main objectives to be investigated and addressed by this thesis. The motivations and formal definitions of these objectives will be presented again in details in the following chapters where they are addressed with algorithms proposed by this thesis, accompanied with a series of experimental studies.

1.2.1 How to formulate the problem of active module identification?

As introduced above, whether network representation faithfully reflects the process and activity of biological system has an essential effect on the accuracy and reliability of following experimental results. Similarly, the formal formulation of an active module shall be very carefully designed to highlight the specific signals in target cellular response and decrease background noises. Many studies use p-values from differential expression analysis and formulate a scoring function based on a variety of different statistical models. Fold-changes are also widely used because of its simplicity yet often satisfactory results. In addition, information and knowledge from existing interaction database is incorporated into the pure data driven method by some research. The differences in the specific problem formulation are often a result of different definitions for active module under certain contents.

Research Question 1: How to build a practical formulation of active module identification problem that faithfully reflects the dynamic changes of cellular activities and helps reveal new insights compared to other existing methods?

1.2.2 How to design and improve algorithms to identify active module?

After a formal definition of the active module, the next step is to design an appropriate algorithm to identify it. Heuristic algorithms are common choices for solving this NP-hard problem. Details and problem specific modifications need to be carefully designed and improved for problems with different structures. For example, in active module identification methods using genetic algorithms, the representations of a solution and the genetic operators may vary for different problems. In simulated annealing, it is essential to define the state for a system. Besides, the accuracy, efficiency, scalability and robustness are often taken into consideration when assessing an algorithm.

Research Question 2: How to develop effective, efficient and robust algorithms to identify active modules that are truly biological meaningful?

1.2.3 What is the right way to interpret identified active module?

This question rises in the process of our research. A widely accepted way of interpreting active module is to apply functional annotation on it. It gives a description on functions enriched in given module in a hierarchical structure and the significance of each enrichment term. However, due to the complexity of biological system and the hierarchical structure of functional groups, such functional annotation can have limited results, mainly in the following two aspects.

- **Criteria to choose the most representative modules in network.** Many methods identify more than one active module at a time with similar scores, overlapping areas, and slightly differences in the biological interpretations. It is worth discussing on how to choose the most representative module as the final result.

- **Ambiguity in interpreting module with large size.** Functional annotation given by an relatively large size of active module is often too general and sometimes ambiguous as it contains a number of network members. There have been some research that try to control the size of the module by changing the parameters in network construction, scoring function formulation, or explicitly constraining the size in algorithm design. These methods, although have been proven to be effective, sometime can be unnatural or have difficulties in applying to other formulations.

This thesis also aims to find a better way for interpretation and avoid the above issues.

Research Question 3: What is the right way to interpret identified active module? Especially, how to choose the most representative module in a bunch of results? Is there any better way to deal with the size problem for identified active module other than setting a hard constrain?

1.3 Outline of the Thesis

This thesis is trying to answer the research questions listed in Section 1.2. The rest of this thesis is organised as follows.

Chapter 2 gives an introduction of biological networks and its structural properties as the research background of active module, and a review of widely used definitions and approaches for active module identification. Representative approaches in the three major categories of active module identification methods, i.e. significant-area-search methods, diffusion flow and network propagation methods and clustering-based methods are introduced, with a focus on their problem formulation and algorithm design. Different scoring strategies and algorithm development for significant-area-search methods are further reviewed.

Chapter 3 introduces the motivation and general issues for novel research presented in this thesis. Especially, it explains the motivation of formulating multi-objective problem for integrating differential expression and differential correlation associated with Chapter 4, of introducing prior information guidance in the traditional pure data driven active module identification methods associated with Chapter 5, and of incorporating topological community detection as a multitasking scheme associated with Chapter 6. It illustrates the issues encountered when we are trying to address the formulated problems.

Chapter 4 proposes an integrated approach for active module identification that consider both the differential expression of each gene and the differential correlation between genes. We adopts two classic measurements for active module detection, and incorporates the two objective functions using a multi-objective evolutionary algorithm. By formulate the problem as a multi-objective problem, this approach avoids the weight parameter for balance between objectives in traditional integrated methods, and provides meaningful results that otherwise cannot be detected using single measurement.

Chapter 5 propose a prior information guided active module identification approach aiming at identifying modules that are both active and enriched by prior knowledge. We formulate the active module identification problem as a multi-objective optimisation problem, design a novel constraint based on algebraic connectivity to ensure the connectivity of the identified active modules, and solve it using a modified approach based on NSGA-II. Experimental studies show that integrating knowledge of functional groups into the identification of active module is an effective method and provides a flexible control of balance between pure data-driven method and prior information guidance.

Chapter 6 proposes a novel algorithm framework of detecting active module and topological communities simultaneously using evolutionary multitasking. A series of task-specific algorithm designs and improvements have been made based on the original framework of evolutionary multitasking algorithm, including a unified genetic representation and problem-specific decoding methods for the two tasks, task-specific mutation operators with local search strategy, and an extra solution improvement step. The proposed algorithm is first applied on some classic community structured networks to test its performance on community detection, and then on biological networks to simultaneously run both tasks. Experimental studies show that the proposed algorithm is able to detect network divisions with values of modularity comparable or even better than classic community detection algorithms, and also able to identify active modules with considerably high scores. By mapping the community structure to the active module and further dividing the module into smaller fractions, this algorithm provides a new way to better interpreter the biological meaning of active module.

Chapter 7 givens conclusions for the thesis and some discussion on the future work.

1.4 Contribution of the thesis

By addressing the research questions listed above, this thesis presents the following contributions.

- **A multi-objective formulation of active module measurements that combines differential expression of each gene and the differential correlation between genes.**

- **A novel formulation of prior information guided active module that provides a flexible control of balance between pure data-driven method and prior information guidance.**

- **A multi-objective optimisation framework modified for the problem and uses a novel constraint to ensure the connectivity of the identified active modules.**

- **A novel framework of detecting active module and topological communities simultaneously using evolutionary multitasking with a series of task-specific algorithm designs and improvements.**

- **An inspiring way of integrating topological community information to help the interpretation of active module.**

1.5 Publication Resulting from the Thesis

- Published journal paper
  - Chen W, Liu J, He S. Prior knowledge guided active modules identification: an integrated multi-objective approach[J]. BMC systems biology, 2017, 11(2): 8.
  - This paper is associated with Chapter 5.

- Submitted paper
  - Chen W, Zhu Z, He S. Mumi: multitask module identification for biological networks. Submitted to IEEE Transactions on Evolutionary Computation.
  - This paper is associated with Chapter 6.
- Paper in preparation
  - Chen W, He S. Combined measurements for active module identification using multi-objective method.
  - This paper is associated with Chapter 4.

CHAPTER 2

LITERATURE REVIEW ON ACTIVE MODULE IDENTIFICATION

This chapter is a literature review on the research topics in biological networks and representative approaches for active module identification. The first section provides a quick review on the development and research interest in network biology and several essential structural features in biological networks that help reveal the dynamics and mechanism of biological systems. The second section introduces the concept of active module and further gives a concrete review on the mainstream categories of active module identification, especially the significant-area-search methods that includes heuristic search based algorithms and mathematical programming based algorithms.

2.1 Modular Structure in Biological Networks

2.1.1 The Development of Network Biology

With the development of high throughput data collection technologies, vast amounts of omics data that cover different species and different levels of biological activities have accumulated exponentially. There are genomics that analyse the complete DNA sequence of an organism's genome, transcriptomics that collect the information of an organism's RNA transcripts and thus generate related gene expression profiles under a given condition, proteomics that focus on the entire set of proteins produced by organism and involved in every biological activity, and metabolomics that study the molecular metabolite profiles that reflect specific cellular process. These varied omics data provide valuable information concerning the intrinsic mechanisms underlining biological processes. With the accumulation of large data sets, one of the most essential challenges for researchers is that how to properly interpret these data.

Techniques and methods have developed rapidly during the past several decades, both in high throughput data collection level and interpretation and analysis level. As a fast-growing interdisciplinary field, computational biology has gradually learned, explored and absorbed analysis approaches from many other disciplines, e.g. mathematical modelling, statistic inference, or computational simulation. Take gene expression data analysis as an example, methods have evolved from the simple single or multivariate statistical analysis, e.g., calculation of fold-change [80], identification of differential expressed genes [72], to integrated approaches that integrate prior knowledge and different data set [2]. As a research field driven by those integrated approaches, network biology has gained popularity recently years.

Network biology offers a highly abstract model of networks to characterise various levels of biological systems and provides insights into those system by taking advantages of network theory [6, 5]. The development of network biology is based on the awareness that as a complex system, biological interaction networks share many essential architectural features and behaviours with other complex systems that have been long studied [60], such as the social networks or the Internet. Graph theories that help reveal the formulation and evolution principles of these systems can also be applied to biological networks in the hope of discovering and characterising the functions and mechanisms behind biological activities.

2.1.2 Structural Features of Biological Networks

A network can be simply viewed as a collection of nodes with pair-wise interactions called edges among them. There are a bunch of different ways to model complex biological systems as networks. For the purpose of simplicity, in each type of networks there is only one or two specified types of molecular components selected as the nodes. Interactions or relations between the components are measured as edges. These interactions are often physical or chemical interactions like the interaction between two protein, or metabolic interactions between metabolites. A list of commonly used biological types and related public databases is shown in Table 2.1. Aside from direct construction of network using interaction databases, integrative approaches that combine existing interaction maps and experimental condition specific data are also widely adopted. Gene co-expression network [91, 78] for example, is consist of genes as nodes and edges indicating a significant co-expression relationship between genes, most commonly Pearson correlation. Other network construction methods are varied from calculating pair-wise correlation coefficient of expression data (correlation network [48]), filtering from existing interaction database (protein-protein interaction network [26, 87, 56, 51]), or integrated approaches based on both expression data and metabolic models (tissue specific metabolic network [75]).

Network Type	Node Type	Edge Type	Representative Databases
Protein-protein interaction networks	Protein	Binary protein-protein interactions	the MIPS database [53], the BioGRID [9], the STRING [77]
Metabolic networks	Metabolite	Metabolic and transport reaction	KEGG[41]
Regulatory networks	Protein or DNA	Protein-DNA interactions	UniPROBE [82], JASPAR [37]

Table 2.1: Biological networks and interaction databases

Modelling biological system as networks provides quantitative measurements and analysis approaches to study the activities and features of the system. Structural features that are commonly used to describe a network includes degree distribution, clustering coefficient, shortest path and small-world property [79], scale-free property [4], modularity [65] and network robustness [1]. Researches have shown that, despite the high complexity of a living organism, its network architecture follows a few simple universal laws that govern a broad range of network systems graph theory has been investigating into.

One of the most important discoveries is the scale-free property of cellular networks. Scale-free property describes a type of network whose node degree follows a power-law distribution, i.e. the the probability of a node with degree k follows $P(k) \sim k^{-\gamma}$ where γ is the degree exponent [4]. This property is proved to be the consequence of network expanding by connecting new node to its existing nodes with probability proportional to node degree. It has been observed in a wide range of complex networks including social networks, transportation networks, business networks and biological networks. Network with scale-free property tends to have highly connected nodes and form small groups of highly connected nodes called hubs. The phenomenon that scale-free property and hubs are commonly observed in biological networks provides sights and supports in the research of molecular evolution in biological system [85].

Another important characteristic of biological network is the modular structure. Cellular functions are carried out by functional unites called modules [30]. These modules are made up of different types of molecules and have relatively independent functions that arise from the interactions among their components. Through proper way of mapping, components in the same functional modules can be located in the same neighbourhood in biological network, forming a densely connected topological module where nodes are more likely to interact with each other than with nodes from outside the module. The reverse method is often used to help identify functional modules or disease modules [5] in network medicine, whose basic assumption is that the topological, functional and disease module overlap.

There are, however, some challenges confounding the analysis of biological network and its structural features. One major challenge comes from the complexity of biological system itself: how to properly convert data containing activities of a huge number of molecules into a model that nicely reflects the functions and changes the system is undergoing. Considering that there are often tens of thousands of molecules and interactions, carefully designed simplification or reduction is necessary so that the core components and functions can be preserved. Technological biases in high-throughput approaches also effects signal accuracy and generate false positives and false negatives, which triggers the arising of research and techniques in high-throughput data processing.

Although currently biological networks are not able to fully capture the diversity and dynamics of complex biological system[25], it is still one of the most promising and fast developing research area in modern biology. Many studies have been performed on the construction of networks from biological systems and the structural and functional features that may respond to related biological information.

2.2 Active Module Identification in Biological Networks

Modular structure is one of the essential characteristics that reveal information about the relationship and interaction among components in the network. In biological networks, modules are considered as the functional units of cellular process and organisation [30]. Varied definitions of module have been proposed and numerous methods have been developed to identify those modules [38, 32], all aiming to reveal essential biological mechanisms [54, 31]. Among them, active module detection is a successfully applied integrative approach.

Active module is a region in network that shows striking changes in molecular activity or phenotypic signatures, which is often associated with a given cellular response [54]. Such response-specific regions are expected to reveal dynamic and process-specific information that is correlated with cellular or disease states. In some literature active modules are alternatively described as network hotspots, or responsive subnetworks. In this thesis we will use the term active module.

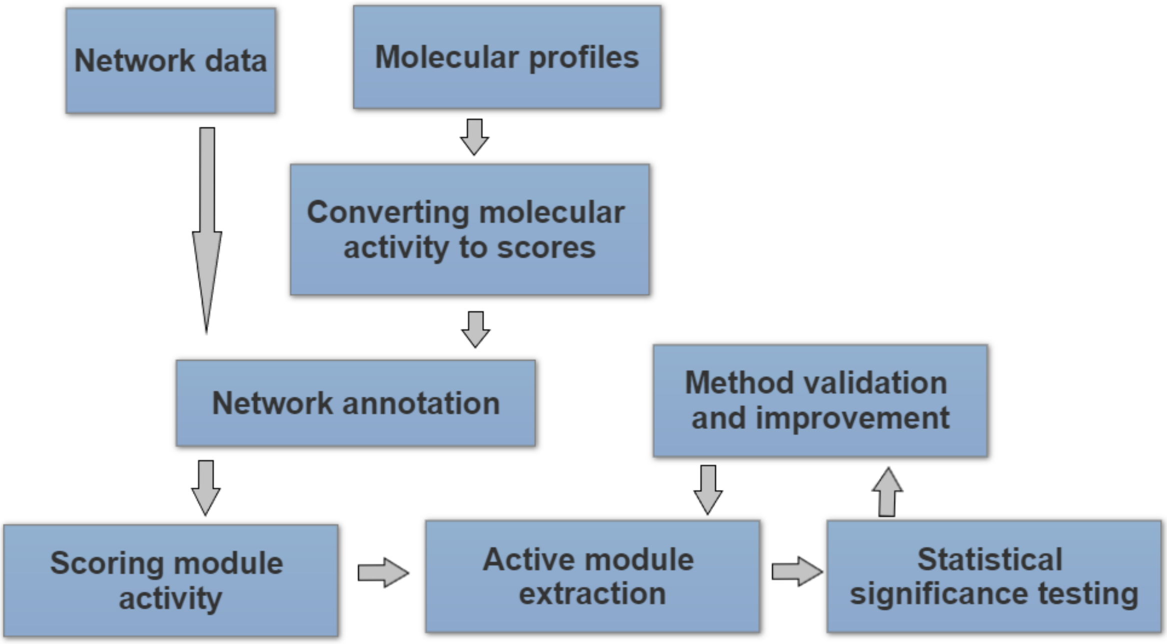

Network data
Molecular profiles
Converting molecular activity to scores
Network annotation
Method validation and improvement
Scoring module activity
Active module extraction
Statistical significance testing

Figure 2.1: A brief workflow for active module identification. This figure is redrawn from Figure 1 in reference [54].

2.2.1 Categories of Active Module Identification Methods

A number of computational techniques have been developed to identify active modules in biological network. Many of them have been packaged as convenient tools available for public use. A general procedural workflow for active module identification is shown in Figure 2.1. Network data usually comes from public interaction database like those listed in Table 2.1. Molecular profiles are incorporated to provide quantified information of molecular activities that can be converted into scores for network annotation. After network activity is annotated, algorithms are applied to the network for the identification of active modules based on a variety of strategies. The extracted modules are tested for statistical significance. Method validation and improvement is also performed in this step. After that, active modules that are statistical significant will be used for biological interpretation and analysis. Mainstream methods for active module identification can be roughly classified into three categories described as following.

2.2.1.1 Significant-Area-Search Methods

A typical significant-area-search method annotates the nodes or edges in network with scores indicating the level of molecular activity, formulates a scoring function to calculate the module score that is able to measure the overall activity of a selected network region, and finally applies a search strategy that identifies the region with optimised module score, indicating an active module. This is the type of methods that we will use for the research presented in this thesis. A detailed review of representative significant-area-search methods is presented in Section 2.2.2.

2.2.1.2 Diffusion-Flow and Network-Propagation Methods

This type of methods adopts the concepts of diffusion flow and network propagation. It assumes that the spread of information in biological network is analogous to the fluid or heat flow in a network of pipes. Thus in biological system, network flow is diffused from source nodes with high level of differential expression or known disease genes, flows outwards along network edges, and gets accumulated in certain regions. Regions accumulating the maximum flow, i.e. the maximum influence from neighbouring nodes, are then detected as active modules.

In one such method called HotNet [84] that is designed to detect significant mutated pathways in cancer, an influence graph is constructed by using a diffusion flow on the interaction network to define influence between gene pairs. The influence of gene g_s on gene g_i is calculated as the amount of fluid f_i^s when fluid is pumped into the source g_s at a constant rate, lost from each node at a constant first-order rate, and the system reaches the equilibrium. The diffusion process is related to certain random walks on graph. After computing the diffusion flow for all tested genes, an influence graph G_I is constructed where nodes are the set of tested genes, and the weight of an edge $e(g_j, g_k)$ is given by $w(g_j, g_k) = min(f_k^j, f_j^k)$ for all pairs of tested genes as the influence is not symmetric. After defining the influence measure, the method formulates the problem of finding a connected subgraph of k genes that are mutated in the largest number of samples as connected maximum coverage problem defined below.

Problem 2.1 (Connected Maximum Coverage Problem). *Given a graph G defined on a set of n nodes $V = \{v_1, v_2, ..., v_n\}$, a set I, a family of subsets $p = \{P_1, P_2, ..., P_n\}$, with $P_i \in 2^I$ associated to $v_i \in V$, and a value k, find the connected subgraph $C = \{v_{i1}, ..., v_{ik}\}$ with k nodes in G so that $| \cup_{j=1}^{k} P_{ij}|$ is maximised.*

In this case, graph G is the influence graph G_I, and subsets P_i is the sets of samples in which gene g_i is mutated. As the connected maximum coverage problem is NP-hard even for simple network, HotNet proposes a combinatorial algorithm that runs in polynomial time and gives $O(\frac{1}{r})$ approximation where r is the radius of the optimal solution. The detailed description of this algorithm is shown in Algorithm 2.1.

Algorithm 2.1: Combinatorial Algorithm for Connected Maximum Coverage Problem by HotNet

```
Input: Influence graph $G_I$, threshold $\delta$, size $k$
Output: Connected subgraph $C$ with $k$ nodes
Construct $G_I(\delta)$ by removing edges with weight $w < \delta$ from $G$ ;
$C \leftarrow \emptyset$ ;
for each node $v \in V$ do
    $C_v \leftarrow \{v\}$ ;
    for each node $u \in V \setminus \{v\}$ do
        $p_v(u) \leftarrow$ shortest path from v to u in $G_I(\delta)$
    end
    while $|C_v| < k$ do
        $l_v(u) \leftarrow$ set of nodes in $p_v(u)$;
        $P_v(u) \leftarrow$ elements of $I$ covered by $l_v(u)$ ;
        $P_{C_v} \leftarrow$ elements covered by $C_v$;
        $P_C \leftarrow$ elements covered by $C$;
        $u \leftarrow argmax_{u \in V \setminus C_v; |l_v(u) \cup C_v| \leq k}\{\frac{|P_v(u) \setminus P_{C_v}|}{|l_v(u) \setminus C_v|}\}$ ;
        $C_v \leftarrow l_v(u) \cup C_v$
    end
    if $|P_{C_v}| > |P_C|$ then
        $C \leftarrow C_v$
    end
end
return $C$
```

The HotNet algorithm is successful in detecting pathways that are known to play an essential role in cancers and is also able to identify additional pathways that have not yet previously reported as mutated.

2.2.1.3 Clustering-based Methods

The third group of methods simultaneously clusters network components and the corresponding conditions under which those components become active, based on a concept called biclustering. Biclustering is the clustering performed on the row and column dimensions of the data matrix simultaneously.

The motivation of applying biclustering algorithms for biological data analysis is that the results from standard clustering methods are sometimes limited because the activity of genes may not be correlated in all of the experimental condition [52]. A gene expression profile is usually presented in a data matrix whose rows correspond to genes and columns correspond to conditions. Analysis on expression data is either aiming at revealing the expression patterns of genes by comparing rows of the matrix, or expression patterns of sample conditions by comparing columns. Because there often exists a subset of genes that show compatible expression patterns under a subset of experimental conditions [70], biclustering algorithms that are able to detect submatrices have been broadly applied in finding these subgroups of genes or subgroups of conditions.

As summarised in a comprehensive survey [52], biclustering algorithms are particularly suitable for the following situations:

- Only a subset of genes are activated in target cellular process.
- Target cellular process is only activated in a subset of experimental samples or conditions.
- The activation state of multiple pathways that a single gene participate in may not be highly correlated across all samples.

Given a data matrix A with the set of rows X and set of columns Y, $A_{IJ} = \{I, J\}$ represents the submatrix of A where $I \subseteq X$ and $J \subseteq Y$ are the subsets of rows and columns.

A bicluster is the submatrix whose rows exhibit similar behaviour across columns, and columns exhibit similar behaviour across rows. The problem addressed by biclustering algorithms is given below.

Problem 2.2 (Biclustering Problem)**.** *Given a data matrix A, find a set of k biclusters $B_k = (I_k, J_k)$ such that each bicluster B_k satisfies some specific characteristics of homogeneity.*

The characteristics of homogeneity are differently defined by each approach. Some biclustering algorithms directly analyse numeric values in data matrix and try to find constant or coherent values, while some other algorithms are designed to find coherent evolutions across rows or columns instead of the exact values. As the complexity of the biclustering problem is NP-complete, heuristic search strategies are broadly used to address it.

One of the biclustering algorithms, cMonkey [71], defines the probabilities of each gene or condition belonging to a given bicluster as p-values based upon individual data likelihoods, which are then calculated from the correlation of gene expression, similarity of upstream sequences, and association network topology. The algorithm uses a variety of seeding methods to start the procedure of clustering, e.g. seeding with a single random gene, with an existing cluster or bicluster, with a highly connected node or with a motif. It iteratively improves a newly seeded bicluster by adding genes or conditions with high membership probability and dropping those with low membership probability. The workflow of this algorithm is shown in Algorithm 2.2.

2.2.2 Representative Significant-Area-Search Methods

This section gives a review of several representative significant-area-search methods for active module identification. As the problem of finding the maximal-scoring connected module has proven to be NP-hard (non-deterministic polynomial-time hard) [34], heuristic algorithms are broadly used to approximately search for high scoring modules than

Algorithm 2.2: Seeding and Annealing Based Biclustering Algorithm by cMonkey

```
Input: data matrix, membership probabilities for each gene and condition,
       maximum number of clusters $k_{max}$
Output: biclusters
$k \leftarrow 0$;
while $k \leq k_{max}$ and significant optimisation is still possible do
    seed a new bicluster;
    repeat
        search for motifs in bicluster;
        compute conditional probability that each gene or condition is a member of
          the cluster;
        perform moves sampled from the conditional probability;
    until the cluster does not change;
end
return biclusters detected
```

finding the maximally scoring module. Commonly used heuristic approaches are simulated annealing [34], greedy search [33], and evolutionary algorithm [43, 56]. Nevertheless, exact approaches that guarantees to identify maximally scoring module have also been explored and developed [15, 92, 3].

2.2.2.1 Heuristic Search Based Methods

The jActiveModule [34] method proposed by Ideker in 2002 is considered as the first to formulate active module search task into an optimisation problem. It takes p-value p_i representing the significance of expression change for each gene i and converts it to a z-value-score z_i through

$$z_i = \Phi^{-1}(1 - p_i) \tag{2.1}$$

where Φ^{-1} is the inverse normal cumulative distribution function. For randomly distributed data p-values are uniformly distributed between 0 and 1, thus z-scores follow a standard normal distribution. The aggregate z-score z_A for a given module A of k nodes is then given by

$$z_A = \frac{1}{\sqrt{k}} \sum_{i \in A} z_i \tag{2.2}$$

z_A also follows a standard normal distribution assuming z_i are independently drawn from a standard normal distribution, making modules of different sizes comparable to each other under this scoring system.

To calibrate z-score against the background distribution, this algorithm uses a Monte Carlo approach to estimate average scores μ_k and standard deviation σ_k for randomly selected modules of size k. The module score s_A of size k after background correction is calculated by

$$s_A = \frac{z_A - \mu_k}{\sigma_k} \tag{2.3}$$

A higher s_A score indicates a higher module activity. This scoring system can be extended for gene expression changes measured over multiple conditions by sorting z_A scores across all conditions, computing the significance of the j-th highest score through a binomial order statistic, and converting it back into a standard normal z-score.

After the formulation of active module score s_A, the jActiveModule uses an approach based on simulated annealing to find the maximal-scoring connected module. Simulated annealing is a heuristic algorithm that allows probabilistic transitions to an inferior state in order to avoid getting stuck in local optima. It is inspired by the cooling process of molten materials down to the solid state where in the process of seeking a minimum-energy state, there is a probability of transition from a lower energy state to a higher energy state that correlates with the energy gap and decreases when the temperature gets lower [76]. The search strategy used by jActiveModule is shown in Algorithm 2.3.

The jActiveModule method by Ideker is among the earliest to identify active modules on molecular interaction networks. In real work the simulated annealing based search converges relatively slow and is not easy find a satisfactory solution for large scale network. Besides in the literature it shows no guarantee for the connectivity of detected module. In the source code implemented as a jAcitveModule plug-in for network visualisation and analysis software Cytoscape, the algorithm maintains an additional HashMap storing nodes and their connected components, and checks whether the connectivity of subgraph G_w would be affected every time a node is to be toggled, which effects the running speed

Algorithm 2.3: Simulated Annealing Based Search Strategy by jActiveModule

```
Input: A graph $G = (V, E)$, a number $N$ of iterations, a temperature function $T_i$
       that decreases geometrically from $T_{start}$ to $T_{end}$
Output: A subgraph $G_w$ of $G$
Initialise $G_w$ by setting each $v \in V$ as active or inactive by probability 0.5 ;
for $i = 1$ to $N$ do
    Randomly select a node $v \in V$ and toggle its activation state ;
    Computer the background corrected score $s_i$ for current subgraph $G_w$;
    if $s_i > s_{i-1}$ then
        // if the state increases score, jump to it
        keep $v$ toggled
    else
        // if the state decreases score, jump to it with probability
           related to the gap of two scores
        keep $v$ toggled with probability $p = e^{\frac{s_i - s_{i-1}}{T_i}}$
    end
end
```

of the algorithm. Nevertheless, as the first research to formulate active module identification problem and present a feasible optimisation method to solve it, jAcitveModule has influenced a number of following research in the form of problem formulation and the choice of search strategy.

Another method adopts similar formulation and search strategy to identify active modules in protein-protein interaction network, but instead of annotating activity score to nodes, it uses edge score to measure the module activity [26]. In this method, the edge score $Score(e(x, y))$ for an edge $e(x, y)$ between two proteins x and y is calculated as

$$Score(e(x, y)) = Cov(x, y) = Corr(x, y)std(x)std(y) \tag{2.4}$$

where $Corr(x, y)$ is the Pearson correlation coefficient of the expressions levels between x and y. The overall expression variation $std(x)$ and $std(y)$ are used as the measurement of the differential expressions of corresponding genes. Then the aggregated score for a given subgraph $G_w = (V_w, E_w)$ is given by

$$T(G_w) = \sum_{e \in E_w} Score(e) \tag{2.5}$$

Similar to the background correction used by jActiveModule, this algorithm randomly samples subgraph of size k and estimates average scores μ_k and standard deviation σ_k for background subgraph. Eventually the standardised score for a subgraph G_w of size k is

$$Score(G_w) = \frac{T(G_w) - \mu_k}{\sigma_k} \tag{2.6}$$

This scoring method also guarantees that the scores of modules with different size follow the same distribution, and thus are comparable to each other. This method again uses simulated annealing based search algorithm.

In addition to optimisation methods based on node score or edge score, there are also formulations that combine both node and edge score [87, 51]. The COSINE method [51], proposed to identify active modules based on gene expression profiles, uses a scoring function that considers the differential expression of individual genes and the differential correlation of gene pairs together. It uses the F-statistic to measure the changes of gene expression as node scores, and the expected conditional F-statistic to calculate the changes in gene co-expressions across different groups as edge scores. Both node score and edge score are adjusted against background mean score and standard deviation same as the two methods described above. For a given subgraph $G = (V, E)$ of size k, COSINE calculates the activity score of G as

$$Score(G) = \lambda \frac{\sum_{e \in E} EdgeScore(e)}{\sqrt{\binom{k}{2}}} + (1 - \lambda) \frac{\sum_{v \in V} NodeScore(v)}{\sqrt{k}} \tag{2.7}$$

where $\lambda (0 \leq \lambda \leq 1)$ is a weight parameter to control the fraction of contributions from edge score and node score to the integrated score, and the denominator is an adjustment for the size of module.

COSINE formulates the highest scoring module identification problem as an optimisation problem of finding a binary vector of length $|V|$ to maximise the active module score $Score(G)$, where the i-th position of the vector being 1 represents that the corresponding i-th node is included in the module. It uses genetic algorithm to search for the module with highest score.

The genetic algorithm is a population based global search algorithm that gets inspiration from evolution and natural selection. During the reproduction of organisms, crossover occurs in parental chromosomes to generate offspring chromosomes that contain the inheritable genetic characteristics from both parents. Sometimes random mutations occur when passing chromosomes from one generation to the next generation, increasing the diversity of genotypes in the whole population. If one genotype is suitable for the current environment, the individual has higher probability to survive and reproduce, otherwise it is more likely to die without leaving any offspring. Through this selection pressure, the population of organisms evolves towards the direction of having genetic variations with high fitness to the environment.

Algorithm 2.4: Basic Structure of Genetic Algorithm

```
Input: population size $pop$, maximum generation $gen_{max}$
Output: solution
population ← initialisation of population with pop solutions ;
evaluate the fitness of every individual in population ;
$gen \leftarrow 0$ ;
while $gen \leq gen_{max}$ or other termination criteria are not satisfied do
    apply crossover and mutation to population to generate offspring ;
    evaluate the fitness of every individual in offspring ;
    intermediate-population ← Union(population, offspring);
    population ← selecting fittest individuals from intermediate-population ;
    $gen = gen + 1$;
end
return solution with highest fitness in population
```

For a given problem, genetic algorithm maintains a population of solution candidates as individual chromosomes. The representation form of solution depends on the characteristics of the problem. Problem-specific objective functions are defined to calculate the fitness of individuals. For each generation, genetic operators like crossover and mutation are applied to produce the offspring population. Selection is performed to preserve solutions with higher fitness. The algorithm iterates until the maximum number of generation is produced or other termination criteria are satisfied. The workflow of a basic genetic algorithm is given in Algorithm 2.4. In the design of genetic algorithms, the representation form of solution, the choice on genetic operators and the type of selection process vary from problem to problem and can have different effects on the performance of the algorithm.

There have been some criticism on genetic algorithm, mainly focused on its scalability with complexity and slow convergence for nontrivial problems [76]. In the condition when crossover and mutation operator cannot make good use of problem-specific structure, reproduction process generates a large proportion of inferior solution, which leads to slow convergence. Due to the population based strategy, it often has high space complexity and a high number of repeated fitness function evaluation. Nevertheless, it is still used in a broad range of optimisation problems.

2.2.2.2 Mathematical Programming Based Methods

In 2008, Dittrich and Klau [15] proposed an integer-linear programming based approach to find the optimal solution to the maximal scoring subgraph problem, which is the first exact approach for active module identification. This approach formulates an additive score for each node based on a beta-uniform mixture distribution model proposed by an earlier research [69] of approximating and partitioning the empirical distribution of p-values in microarray analysis. As we adopt the formulation of this score in our research, a further description and related equations are given in Section 4.1.1.

The scoring function used in this method is based on signal-noise decomposition where positive value indicates signal content and negative value indicates background noise. After assigning scores to each node, the score for measuring activity for a given module is simply the sum of scores of every individual node in the module. Thus the problem of active module identification in this context is defined as a maximum-weight connected subgraph problem stated below.

Problem 2.3 (Maximum-Weight Connected Subgraph Problem, MWCS)**.** *Given a connected undirected node weighted graph $G = (V, E, w)$ with weights $w : V \to R$, find a connected subgraph $T = (V_T, E_T), V_T \subseteq V, E_T \subseteq E$ that maximises the score $w(T) = \sum_{v \in V_T} w(v)$.*

Given an instance of MWCS problem $G = (V, E, w)$ with positive and negative node weights, let $w_{min} = min_{v \in V} w(v)$ be the lowest value of node weight, the instance can be transformed to an instance $G = (V, E, c, p)$ of prize-collecting Steiner tree problem shown below by setting node profits $p(v) = w(v) - w_{min}$ for all $v \in V$ and edge costs $c(e) = -w_{min}$ for all $e \in E$.

Problem 2.4 (Prize-Collecting Steiner Tree Problem, PCST)**.** *Gvien a connected undirected node and edge weighted graph $G = (V, E, c, p)$ with node profits $p : V \to R^{\geq 0}$ and edge costs $c : E \to R^{\geq 0}$, find a connected subgraph $T = (V_T, E_T), V_T \subseteq V, E_T \subseteq E$ that maximises the profit*

$$p(T) = \sum_{v \in V_T} p(v) - \sum_{e \in E_T} c(e) \tag{2.8}$$

The method has proved that a maximum-weight connected subgraph in $G = (V, E, w)$ corresponds to the optimal prize-collecting Steiner tree in its transformed version $G = (V, E, c, p)$. Thus after converting the MWCS problem into the well-known PCST problem, the method is able to solve it through mathematical programming that finds provably optimal solution to the active module identification problem.

2.3 Summary

This chapter has presented a review on the literature background of active module identification as the topic of this thesis. We have introduced the biological network as a main research topic in network biology, the fundamental structural features of biological network, and the active module as a region showing significant changes in molecular activity or phenotypic signatures in biological network. We have presented a series of works on active module identification, whose approaches are roughly divided into three categories: significant-area-search methods that formulate active module identification as a high scoring subgraph detection problem, diffusion-flow and network-propagation methods that model the flow of information or influence in network as fluid or heat diffusion process, and clustering-based methods that use biclustering to discover a subset of genes of a subset of conditions sharing similar expression patterns. A further detailed review on significant-area-search methods is given as it is the main type of method this thesis will use for novel algorithm frameworks proposed later. In significant-area-search, score annotation can be node score, edge score, or a combination of both. Heuristic algorithms like simulated annealing and genetic algorithms are often applied as the search strategy for high scoring subgraph. Exact methods that transform the active module identification problem into forms that are solved in mathematical programming are also developed.

CHAPTER 3

MOTIVATIONS AND GENERAL ISSUES OF ACTIVE MODULE IDENTIFICATION

Chapter 1 introduces the key research questions this thesis is aiming at, among which the first one is the formulation of active module identification problem that faithfully reflects the dynamic changes of cellular activities and helps reveal new insights compared to other existing method. In chapter 2, a variety of problem definitions and objective formulations for active module identification and their corresponding algorithms are reviewed. The differences in the design of these algorithm framework are often because of the differences in research interest and research questions. The detailed formulation of active module problem can be very flexible in a given context.

This chapter gives the research interest and intuitions behind the design of the algorithms proposed by this thesis. It explains the motivation of combining differential expression and differential correlation using a multi-objective optimisation approach in Chapter 4, motivation of introducing prior knowledge and using a multi-objective formulation in Chapter 5, and the motivation of incorporating topological communities under a multitasking scheme in Chapter 6. It also briefly introduces several general issues in designing an active module identification framework.

3.1 Motivation of Multi-Objective Formulation for Differential Expression and Differential Correlation

The idea of combining differential expression and differential correlation itself is not new. Back to 2011, COSINE [51] was proposed as one of the earliest to consider both active module measurements simultaneously. It used a weight parameter λ to balance between differential expression as node score and differential correlation as edge score, resulting in one combined objective function, which is then optimised through evolutionary algorithm. A simplified version of its objective function is shown as below. More details of objective function formulation proposed by COSINE can be found through equation 2.7 and its related contents.

$$Score(module) = \lambda * EdgeScore + (1 - \lambda) * NodeScore \tag{3.1}$$

However, although the choice of parameter λ is essential for objective function and final results, it's not easy to choose its value. COSINE designed a whole set of rules to choose λ that requires considerable amount of calculation and random sampling. Upon seeing the objective function, it came to us that the explicit weight parameter can actually be avoided using a multi-objective formulation.

A multi-objective problem is an optimisation problem that optimises multiple competing objective functions simultaneously. As these functions conflict with each other, there is seldom a single, perfect solution. Instead the optimisation of a multi-objective problem gives a set of alternative solutions whose objective function values can not be simultaneously improved any more. They are considered equivalent without further information concerning the priority of objectives [20]. Such a set is called the Pareto-optimal set, or the Pareto front. Pareto optimality means no objective value can be improved without degrading any other objective values. In the Pareto front, no solution is better than any other solution across all objectives. In practise, however, there can be a preference in the trade-offs between different objectives. On the other hands, all solutions in the Pareto front dominate any other solution that is not in Pareto front. A solution dominates another solution if it is not worse than the other in all objectives and has a least one objective that is strictly better.

From the objective function proposed by COSINE it's obvious that the two measurements for active module, i.e. node score and edge score, are conflicting. Thus a multi-objective formulation is feasible. The balance between node score and edge score can still be controlled as the multi-objective optimisation returns not a single best solution, but a Pareto front containing a series of "equally good" Pareto solutions that ranges from high node score to high edge score. Instead of setting weight parameter before the search, preference for the two kinds of activity measurements is fulfilled by selecting preferred solutions on Pareto front. Chapter 4 provides detailed description of this approach and experimental results.

3.2 Motivation of Introducing Prior Knowledge and Multi-Objective

The integration of prior knowledge is inspired by the concept of functional module which corresponds to a statistically significant segregation of nodes with similar or related functions in the same network neighbourhood[5]. Functionally related nodes tend to interact with each other and get activated simultaneously in a given cellular response, thus nodes known to be in the same or related functional group have higher potential to form an active module. On the other hand, considering that there are still genes whose function is not revealed clearly yet, it would be interesting to search for a module that is both active in terms of differential expression and enriched in particular functional groups. By investigating into genes assembled in such a module, potential relation among genes may be uncovered.

In chapter 5 we again use a multi-objective framework to formulate this problem. The reason why finding an active module enriched in prior knowledge is conflicting is that for a given cellular response, only a small fraction of functional groups related in this biological process get activated while a number of other function groups do not participate in a significant level. Thus purely enriching prior knowledge only gives a collection of existing functional pathways that do not contain much information for the cellular activity. Meanwhile, purely searching for high active score modules omits a fact that some transcription factors that serve as an important regulator in cellular response may not show very significant differential expression level. Similarly it is questionable whether a whole response related functional process that may have different level of differential expression can be discovered by a purely data driven method. Thus we combine the two conflicting objectives together to design a new framework as a prior knowledge guided active module identification.

We first formulate the problem as a multi-objective optimisation problem, which not only maximises the activity score as defined by Dittrich and Klau [15] but also maximises the prior knowledge contained in the active module. The intuition behind this multi-objective formulation is to use prior knowledge to guide the search of novel information from data, i.e., active modules. The Pareto solutions from this multi-objective optimisation problem are then the optimal trade-off between known knowledge and novel information.

In order to solve this multi-objective problem, we proposed a modified multi-objective genetic algorithm. One of the important details omitted in many papers of active module identification is how to ensure the connectivity of the solutions. Without this connectivity constraint, the optimal solution is trivial, i.e., the top genes with largest node scores. In order to ensure the connectivity of the identified active modules, we design a novel constraints based on algebraic connectivity. The algorithm is applied to a small molecular interaction network that was used by Ideker [34] and then applied to a large Protein-Protein Interaction (PPI) network constructed from microarray data on drug toxicity and resistance.

3.3 Motivation of Introducing Community Detection and Multitasking

Identifying modules from biological networks is important since modules can reveal essential mechanisms and dynamic processes in biological system. Existing algorithms focus on identifying either active modules or topological modules (communities), which represent active functional and topological units in the network, respectively.

Community detection is a mature research area that discovers the densely connected local regions in network which often reveal the topological property and member relationships of the network. There are also studies that try to identify putative disease modules through detecting topological core modules in biological network [48]. As the traditional active module purely depends on the differential expression data, introducing communities as topological information may help further in the identification and interpretation of active modules.

Active modules or topological modules often overlap with each other, which reveal the interactions between active functional units and topological units. These overlaps shed lights on the biological mechanisms that cannot be revealed by these two modules alone. Therefore, it is important to identify and study both active modules and communities. However, despite its importance, there are no existing methods to do so.

In Chapter 6 we propose a novel multitask module identification algorithm to detect active modules and communities simultaneously. By search for these two types of modules simultaneously, the algorithm can exploit their latent complementarities to obtain new insights into the dynamic biological mechanisms. We will investigate into the relationship between active module and community, and improve precision of the functional annotations by integrating structural information into active module.

Multifactorial evolution is an evolutionary multitasking paradigm that maintains multiple search spaces corresponding to different tasks that may or may not be independent to each other [28]. Multitasking generally does not impose any strict constraint on the relationships between tasks, which is different from multi-objective problem whose objectives conflict with each other. Multitasking is inspired by the observation that in the parallel execution of multiple tasks, every task contributes a unique factor to the evolution process of one singe population. An important design for multitasking dealing with cross-domain optimisation problems is a unified representation scheme. In evolutionary multitasking environment it is also necessary to have a method to compare the relative performance of individual solutions in a population.

A fundamental difference between multifactorial evolution and multi-objective optimisation is that the former aims at finding the global optima for each task while the latter attempts to resolve conflicts among competing objectives and results in a Pareto front with trade-offs between objectives. We use multitasking to solve the active module identification problem and community detection problem simultaneously as it does not have strict constraint on the relationship between the two tasks.

3.4 General Issues in Designing Active Module Identification Approach

There are several general issues to be clarified and addressed in the design of an active module identification method.

- **Scoring function.** The formulation of scoring function determines the formal problem definition of an approach. Scoring is often based on a statistic model for given data, such as the uniform distribution model for random data used by jActiveModule [34], or the beta-uniform mixture model for for p-values in differential expression analysis used by Dittrich and Klau [15]. An appropriate scoring function reflects the activity of module and decreases the influence from background noises. In the research we use the additive score formulated by Dittrich and Klau [15].

- **Connectivity.** A subgraph must be connected in order to be considered as one active module. When the connectivity is not directly guaranteed in the solution representation, additional designs are required to ensure it. As previously reviewed, jActiveModule maintains an additional HashMap storing nodes and their connected components, and checks whether the connectivity of current solution would be affected in each iteration of the algorithm. In an evolutionary algorithm based method [56], depth first search is performed every time crossover operator is performed on two solution to make sure the offspring solutions are connected. In the research we use two different strategies to ensure connectivity for the two proposed algorithms. In the algorithm presented by Chapter 5, a constraint is formulated based on algebraic connectivity. In another two algorithms, solution representation is decoded in a way that generates a connected module directly.

- **Size control.** As discussed before, a relatively large size of active module leads to ambiguity in interpreting through functional annotation. In the design of scoring function by Dittrich and Klau, positive score indicates the activity signal and negative score indicates background noise, thus by setting a threshold to control the proportion of positive and negative scores, the size of identified active module is controlled. In the second proposed algorithm, we further present a topological structure based method to divide large active module into fractions before performing functional annotation on it, which also nicely addresses the size problem.

CHAPTER 4

INTEGRATION OF NODE AND EDGE INFORMATION FOR ACTIVE MODULE IDENTIFICATION USING MULTI-OBJECTIVE APPROACH

In this chapter, an integrated approach of combining node and edge information is proposed for active module identification. It considers both the gene differential expression and differential correlation, formulate it into a multi-objective problem, and solves it using evolutionary algorithm. A beta-uniform-mixture model is used to estimate the distribution of p-values and generate node attributes for activity measurement. Probabilistic mutation is designed to accelerate the search process.

The algorithm is first applied to a series of simulated data. Performance comparison is made among the proposed algorithm and several other methods, showing that the algorithm is able to identify active module from data with mixed pattern distribution. It is then applied to real biological networks and successfully detects active module with related function.

4.1 Methods

The network G is represented as $G = (V, E)$ with $p_i \in (0, 1)$ for $v_i \in V$, $Corr(i, j) \in [0, 1]$ for $i, j \in V$, where V is the set of nodes, E the set of edges, p_v the assigned p-value from differential expression analysis for each node v, and $Corr(i,j)$ the Pearson correlation coefficient for the gene pair i and j. In the proposed algorithm there are two objectives for a given module A:

- Node score S_A indicating significant changes in gene expression for a given module, to be maximised during search.
- Edge score S_e indicating significant changes in gene correlation for a given module, to be maximised during search.

4.1.1 Formulation of Node Score

Microarray analysis studies showed that expression data can be effectively estimated by many mixture-model methods that divide genes into two or more groups, one group contains genes that are differentially expressed, and other(s) not differentially expressed [2]. Among those many methods, Pounds and Morris proposed a beta-uniform mixture (BUM) model that very accurately describes the distribution of a large set of p-values produced from an microarray experiment [69]. The BUM model considers the distribution of p-values as a mixture of a special case of beta distribution ($b = 1$) and a uniform(0, 1) distribution, with a mixture parameter λ. The p-values under the null hypothesis are assumed to have a uniform distribution. Under the alternative hypothesis the distribution of p-values will have a high density for small p-values and can be described by $B(a, 1)$.

A general beta distribution $B(a, b)$ is given by

$$f(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1} \tag{4.1}$$

where $\Gamma(.)$ denotes the gamma function. As $\Gamma(1) = 1$, the probability density function of BUM model is then reduced to

$$f(x|a,\lambda) = \lambda + (1-\lambda)ax^{a-1} \tag{4.2}$$

for $0 < x \leq 1$, $0 < \lambda < 1$ and $0 < a < 1$. Given a set of p-values the two parameters of BUM distribution λ and a can be calculated by maximum likelihood estimation.

Following the idea of Dittrich and Klau [15] to decompose signal component from background noise, an additive score to measure the significance of gene's differential expression is calculated as

$$\begin{aligned} S^{FDR}(x) &= \log \frac{B(a,1)(x)}{B(a,1)(\tau)} \\ &= \log \frac{ax^{a-1}}{a\tau^{a-1}} \\ &= (a-1)(\log x - \log \tau) \end{aligned} \tag{4.3}$$

where τ is a threshold to determine the significance of a p-value. In order to control the estimated upper bound of the false discovery rate (FDR) introduced by Benjamini and Hochberg [8], τ could then be selected to ensure that $FDR \leq \alpha$ for some predefined α using the following equation

$$\tau = (\frac{\hat{\pi} - \alpha\lambda}{\alpha(1-\lambda)})^{\frac{1}{(a-1)}} \tag{4.4}$$

where $\hat{\pi} = \lambda + (1-\lambda)a$, meaning the maximum proportion of the set of p-values that could arise from the null hypothesis.

After assigning score to each of the genes, the overall score for a given module A is then the summation of all genes' scores in it, given by

$$S_A = \sum_{x \in A} S^{FDR}(x) \tag{4.5}$$

4.1.2 Formulation of Edge Score

Edge-based scoring system follows the design proposed by Guo in reference [26]. Edge score $Score_e$ for a given edge e connecting nodes i and j is assigned as the covariance of expression levels between gene i and j. Thus we have

$$Score_e = Cov(i,j) = Corr(i,j)\sigma(i)\sigma(j) \tag{4.6}$$

The overall edge score for a given module A is simply the summation of $Score_e$ for all edges in the module. Monte Carlo approach is used to calibrate the edge score of a module against background distribution. For all possible number of edges m in a network, 10,000 edge sets of size m are randomly sampled from the entire network edge list and used to derive estimates mean μ_m and standard deviation σ_m for edge score. Eventually the corrected edge score for a given module A with m edges is given by

$$S_e = \frac{\sum_e Score_e - \mu_m}{\sigma_m} \tag{4.7}$$

and it also follows standard normal distribution.

4.1.3 Multi-objective Optimisation Algorithm as Search Strategy

As the search of highest scoring module is a NP-hard problem (see reference [34] for more details), we choose to use evolutionary algorithm as the optimisation strategy. In order to perform multi-objective optimisation to maximise both node score S_v and edge score S_e simultaneously, a multi-objective evolutionary algorithm modified from NSGA-II (non-dominated sorting genetic algorithm II, for more details see [14]) is applied as search strategy for module detection.

- **Solution representation**

  A solution is represented as a binary vector of length n, where $n = |V|$ is the size of network, i.e. total number of nodes. Adding or deleting a node is performed through simply flip one bit of the vector at corresponding site. Thus for the i-th individual L_i in population, we have

$$L_i = \{l_i^1, l_i^2, ..., l_i^n\} \tag{4.8}$$

where $l_i^j \in \{0, 1\}$ for $j = 1, 2, ..., n$.

To ensure the connectedness of module, the largest connected component among all the nodes labelled by 1 in the individual is calculated and selected as the module represented by this individual. Decoding procedure is described in Algorithm 4.1.

Algorithm 4.1: Decode Module from Individual Representation

```
Input: Individual $l$, adjacency matrix G of the whole network
Output: Connected node set of active module $V_S$
subgraph $A_l \leftarrow G(l, l)$ ;
get all the $k$ connected components $\{V_1, V2, ..., V_k\}$ in $A_l$ ;
$S = argmax_i\{|V_i|\}, i \in \{1, 2, ..., k\}$ ;
return $V_S$
```

- **Fitness function**

  Node score S_A and edge score S_e are used as two objectives. As the implementation of the algorithm is aimed at minimisation both objectives, scores calculated from above equations would be given an extra negative sign.

- **Genetic Operators** The search algorithm starts by initialising individuals as random binary vector drawn from uniform distribution. Uniform crossover is used to generate two child individuals from two parent individuals. Mutation is performed on the generated child individuals and is designed in a probabilistic way to add or delete nodes. It is inspired from simulated annealing that allows the solution to go to a less satisfactory state with some probability in order to avoid being trapped in local optima. The mutation operator contains two stages: probabilistic subgraph expanding and probabilistic negative weighted node deletion. In the first stage, it each tests whether adding a node would decrease individual's node score S_A and edge score S_e and make choices with probability. Details of the mutation is shown in Algorithm 4.2.

- **Parent selection**

  Binary tournament selection is applied for selecting parents to reproduce. In some cases when the population converges too fast, this step is skipped to decrease selection pressure, thus the whole population would be used for reproduction.

- **Sorting and replacement**

  The algorithm uses fast non-dominated sorting and crowding distance assignment proposed by Ref [14] to generate new population from the combined population efficiently and preserve solution diversity. As we have introduced in Section 3.1, multi-objecitve optimisation generates the so called Pareto front where no solution is better than any other solution across all objectives. Non-dominated sorting proposed by NSGA-II is a simple approach that compares each solution with all other solutions in given population, puts all non-dominated solutions into a Pareto front, then repeats the process in the rest of the population until all solutions are divided into different layers of solution front. Crowding distance is used to determine crowded solutions in the objective space and remove them when necessary in order to maintain diversity of solutions.

4.2 Experimental Studies

The proposed algorithm is aiming at identification of modules that are both showing activity in differential expression and in differential correlation. To better assess its performance, simulated data from multivariate normal distributions are generated and simulated networks are constructed. Several existing methods and the proposed algorithm are applied on these test networks and compared with each other using F-score measurement. Then the algorithm is applied to real biological network to investigate its ability

Algorithm 4.2: Probabilistic Mutation

```
Input: Individual $L_i$, adjacency matrix A of the whole network, a list of $S^{FDR}(v)$
       assigned to each node
Output: Individual $L_i$ after mutation
node set of subgraph S is given by $V_S \leftarrow \{V_j | L_i^j > 0\}, j = 1, 2, ..., n$ ;
$V_{neighbours} \leftarrow$ all neighbouring nodes of $V_S$;
// Stage 1: probabilistic subgraph expanding
for every node $v_j$ in $V_{neighbours}$ do
    // include node depending on node score
    if $S^{FDR}(v_j) \geq 0$ then
        // if the neighbour $v_j$ is assigned with positive $S^{FDR}(v_j)$,
           include it
        $L_i^j \leftarrow 1$
    else
        // if the neighbour $v_j$ is assigned with negative $S^{FDR}(v_j)$,
           include it with probability $exp(S^{FDR}(v_j))$
        if $exp(S^{FDR}(v_j)) > random()$ then
            $L_i^j \leftarrow 1$
        end
    end
    // include node depending on edge score
    New individual $L_{new} \leftarrow$ add node $v_j$ to $L_i$ ;
    if $S_e(L_{new}) > S_e(L_i)$ then
        // if edge score of new individual is higher, include node $v_j$
        $L_i \leftarrow L_{new}$ ;
    else
        // if edge score of new individual is lower, include node $v_j$
           with probability $exp(S_e(L_{new}) - S_e(L_i))$
        if $exp(S_e(L_{new}) - S_e(L_i)) > random()$ then
            $L_i \leftarrow L_{new}$ ;
        end
    end
end
// Stage 2: probabilistic negative weighted node deletion
update node set of subgraph S by $V_S \leftarrow \{V_j | L_i^j > 0\}, j = 1, 2, ..., n$ ;
for every node $v_j$ in $V_S$ do
    // if the node $v_j$ is assigned with negative $S^{FDR}(v_j)$, delete it
       with probability $1 - exp(S^{FDR}(v_j))$
    if $exp(S^{FDR}(v_j)) < random()$ then
        $L_i^j \leftarrow 0$
    end
end
return $L_i$
```

of revealing biologically meaningful results.

4.2.1 Experimental Data

4.2.1.1 Simulated Data

To better assess the performance of the algorithm, simulated data are generated following the design pattern of simulated data in COSINE [51]. For each condition, data for a total of 500 genes, each with 20 samples are generated. The simulated data are drawn independently from multivariate normal distribution with specified parameter setting. Let μ be the mean and ρ the Pearson correlation coefficient. Details for data sets are listed as following.

- **Data Set 1** $\mu = \rho = 0$ for all 500 genes.
- **Data Set 2** For genes 1 to 50, $\mu = 0.75, \rho = 0.6$. For other 450 genes, $\mu = \rho = 0$.
- **Data Set 3** For genes 1 to 50, $\mu = 0.75, \rho = 0$. For other 450 genes, $\mu = \rho = 0$.
- **Data Set 4** For genes 1 to 50, $\mu = 0, \rho = 0.6$. For other 450 genes, $\mu = \rho = 0$.
- **Data Set 5** For genes 1 to 25, $\mu = 0.75, \rho = 0.6$. For genes 26 to 50, $\mu = -0.75, \rho = 0.6$. $\rho = -0.6$ between any gene from 1 to 25 and any gene from 26 to 50. For other 450 genes, $\mu = \rho = 0$.

Data set 1 is the control group. Each of the data sets 2 to 5 is compared with control group through two-sample t-test to generate p-values for each node. Network is constructed based on the covariance matrix. Edges with weight less than 0.6 are deleted to generate a sparse network. The 4 networks are referred to as simulated network 2 to 5, depending on the data sets they are generated from.

4.2.1.2 Real Data

Transcriptional profiles of rodent hippocampal CA1 tissue during ageing and cognitive decline [55] are used for real data study. The source data come from research on rat that are behaviourally characterised on the Morris water maze. Rats are divided into 5 groups depending on their age, i.e. 3, 6, 9, 12 and 23 months. In our study we use 3-month group as the control group and 23-month group as the experimental group. Differential expression analysis between 23-month group and 3-month group is performed using the on-line tool GEO2R [58], with p-value adjustment set to Benjamini and Hochberg false discovery rate control. After deleting genes with adjusted p-value larger than 0.01, a set of differential expressed genes is generated. Network is constructed by computing the covariance matrix of the differential expressed gene list and removal of edges with weights less than 0.8. This netowrk is referred to as the rat network.

4.2.1.3 Performance Assessment

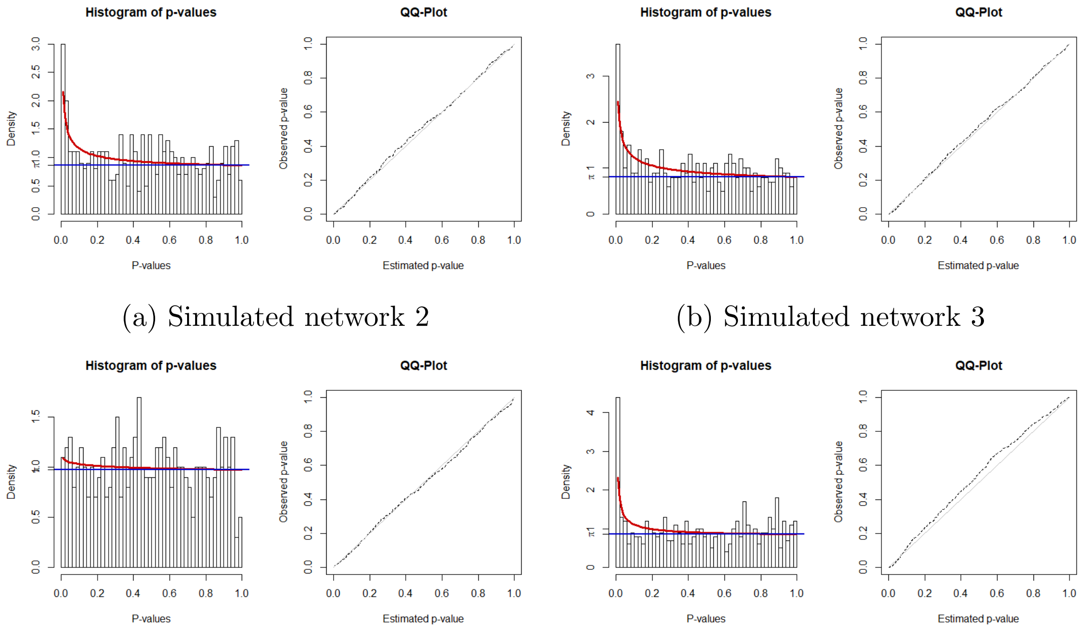
Performance of algorithm on simulated data is directly measured through recall, precision and the combined F-score. Let TP (true positive) be the number of correctly identified genes in active module, FN (false negative) be the number of genes in active module that algorithm fails to identify, and FP (false positive) be the number of genes that are not in active module but mistakenly identified as in it. Then the three measurements can be given by $recall = \frac{TP}{TP+FN}$, $precision = \frac{TP}{TP+FP}$ and $F - score = \frac{2 \times recall \times precision}{recall + precision}$.

Performance of algorithm on real data is not as straightforward as on simulated data. With no knowledge of the true active module, measurement based on true positive and false negative is no longer feasible. Instead, module detected in the real biological network is enriched with gene ontology annotation, indicating specific functions and processes the module is enriched with. This kind of assessment often requires related biological knowledge in order to tell if the enriched annotations match with the experimental condition.

4.2.2 Experimental Results

4.2.2.1 Method Comparison on Simulated Data

To estimate distribution for p-values in simulated networks, the parameters of BUM model a and λ are estimated by R package BioNet [7]. Figure 4.1 shows the fitted model for 4 networks. The distribution of p-values in network 4 differs from 3 other networks because it is the only one that has the same mean value $\mu = 0$ with control group. Nevertheless, all 4 networks show distributions that can be properly estimated by BUM model.



(a) Simulated network 2

(b) Simulated network 3

(c) Simulated network 4

(d) Simulated network 5

Figure 4.1: BUM model estimation on p-values for simulated networks. In each of the 4 sub-figures, left figure is a histogram of p-values with fitted beta-uniform-mixture model distribution. Blue line indicates the uniformly distributed noises and red line the signals as beta distribution $B(a, 1)$. Right figure is a Q-Q plot of the fitted distribution versus the empirical p-values.

The proposed algorithm is applied to the 4 simulated networks. Population size is set to be 50, and generation number 60. Generation is relatively small as the algorithm optimises very fast on simulated data. A typical optimisation process in simulated network 2 in shown in Figure 4.2. Clear Pareto fronts are shown in every 10 generations, indicating that the two objectives, node score S_A and edge score S_e are indeed conflicting with each other. Convergence of the algorithm can be told from the shape of convex hulls drawn by Pareto front and the origin point, which becomes stable at around 40 generations.

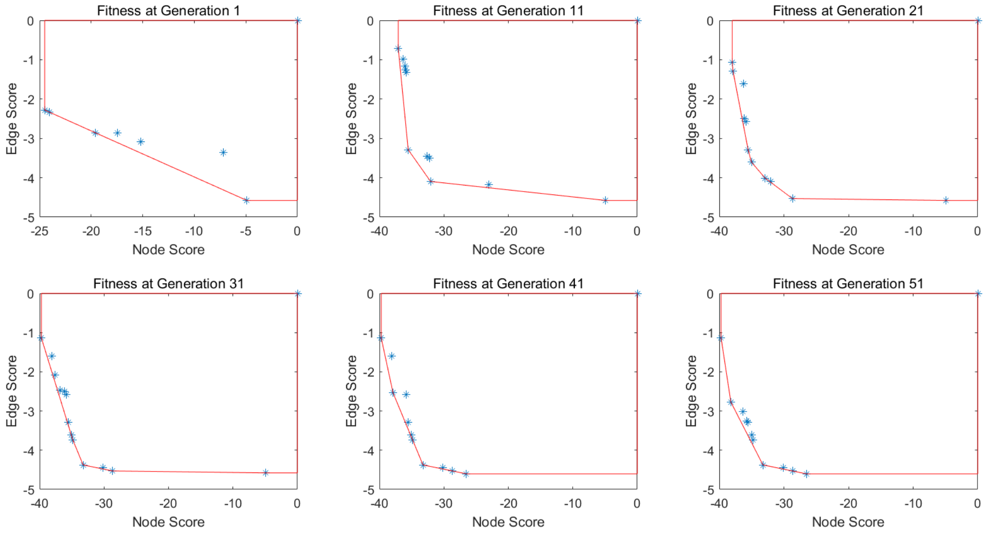


Figure 4.2: Pareto front in every 10 generations from applying proposed algorithm on simulated network 2. Stars represent the fitness of population at generation 1, 11, 21, 31, 41 and 51. Convex hulls drawn by the fitness points and the origin point show that the Pareto front becomes stable after 40 generations, indicating that the algorithm converges fast on simulated networks.

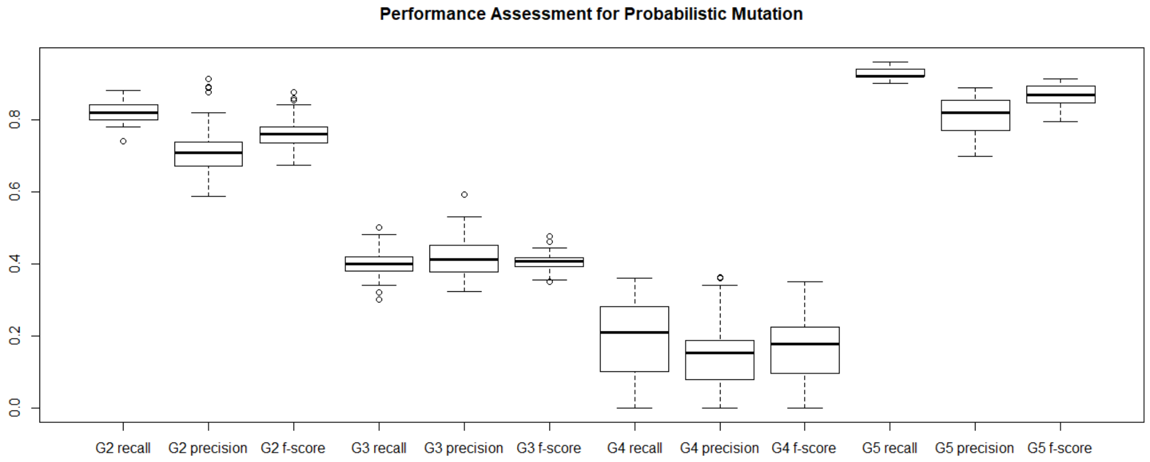
From the Pareto fronts generated by proposed algorithm, the first solution is selected to assess the algorithm's performance by calculating its recall, precision and F-score. The algorithm is repeated for 50 runs on each simulated network and the distributions of assessment results are shown through boxplots in Figure 4.3a. The proposed algorithm gets steady and high recall, precision and F-score on simulated network 2 and 5, proving its satisfactory performance to identify a combined feature of both differential expression and differential correlation. It is still able to detect differential expression alone to some extent, as shown by the boxplots for network 3. However for network 4 with only differential correlation and the same expression level, the proposed algorithm doesn't gain a good score for any of the three assessment. One possible reason is that the design of objective function for edge score doesn't well distinguish the module with differential correlations from the rest.

To show the necessity of probabilistic mutation operator described in Algorithm 4.2, a performance assessment experiment is made following exact the same way described above, using the same algorithm with the same parameters, only that its probabilistic mutation operator is replaced by a standard bitwise mutation. The results are shown in Figure 4.3b. It is clear that the performance of bitwise mutation is defeated by probabilistic one from network 3 to 5 as it has much lower scores and higher variances. Interestingly in network 2 the bitwise mutation is able to gain higher precision and has an overall similar performance with probabilistic mutation, showing that it can handle one single pattern of differential expression and differential correlation. Yet in network 5 with mixed patterns, it again doesn't perform as well as probabilistic one. The comparison provides strong evidence for the power of probabilistic mutation operator.

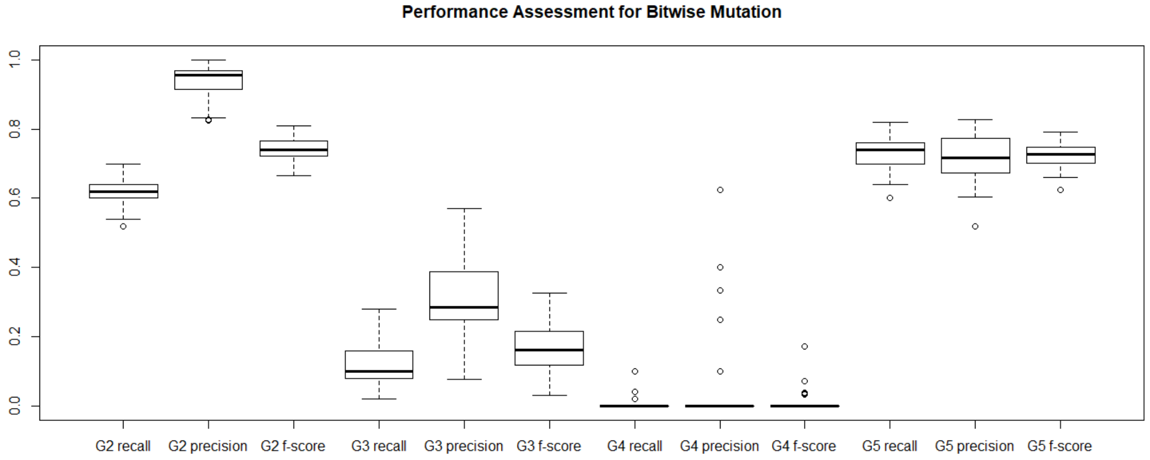
To make a comparison of proposed algorithm with other active module identification algorithms, the averaged values of assessment scores from the 50 runs are compared with the results from three methods, COSINE[51], jActiveModule [34] and another method that combines information from both nodes and edges denoted as Local [86]. The F-measurements data are directly from reference [51] as it uses the same way of data simulation. Performance comparison of the 4 algorithms on 4 simulated networks is shown in Table 4.1. From the table we can see that due to the complexity of varied data distributions, no singe algorithm is able to achieve best performance across all 4 networks for all 3 kinds of measurements. Nevertheless, the proposed algorithm shows averagely good performance, and has high performance in several slots highlighted in bold. Experiment on simulated data shows that the formulation of information from gene differential expression and differential correlation as multi-objective problem is feasible. It also proves that the proposed algorithm is able to identify active module from data with complicated distribution in a relatively reliable level.

4.2.2.2 Application on Real Data

The proposed algorithm is applied to the rat network, with parameters population size set to 100, and generation number 100. The Pareto front generated is shown in Figure 4.4.



(a) Performance of algorithm with probabilistic mutation operator.



(b) Performance of algorithm with bitwise mutation operator.

Figure 4.3: Comparison of performance for proposed algorithm on simulated networks with different mutation operators. Figure 4.3a is the performance of algorithm with probabilistic mutation operator and figure 4.3b is the one with standard bitwise mutation. G2 to G5 represents simulated network 2 to 5. Algorithms are repeated for 50 runs on each network. Performance assessment is done through selecting the first solution in generated Pareto front.

Table 4.1: Performance comparison of proposed algorithm and three other methods on simulated networks. Method jActiveModule is from reference [34]. Method Local is from reference [86]. Data for proposed algorithm is the mean value from 50 runs on each simulated network. Records for proposed algorithm are the averaged value from 50 runs.

Measurements	Networks	COSINE	jActiveModule	Local	proposed algorithm
Recall	2	0.98	1	0.82	**0.82**
	3	0.24	0.98	0.7	0.40
	4	0.86	0.86	0.06	0.19
	5	0.92	1	0.74	0.3
Precision	2	0.86	0.12	0.65	**0.72**
	3	0.92	0.13	0.69	0.42
	4	0.45	0.1	0.04	0.14
	5	0.61	0.125	0.84	**0.81**
F-score	2	0.92	0.21	0.73	**0.76**
	3	0.38	0.23	0.69	0.41
	4	0.59	0.18	0.05	0.16
	5	0.74	0.22	0.79	**0.86**

Unlike in simulated data where solutions are distributed on the Pareto front in a relatively even way, Pareto solutions from rat network are more sparse and unevenly distributed. The reason behind might be related to some intrinsic properties of the real data, which we will not further explore in this chapter.

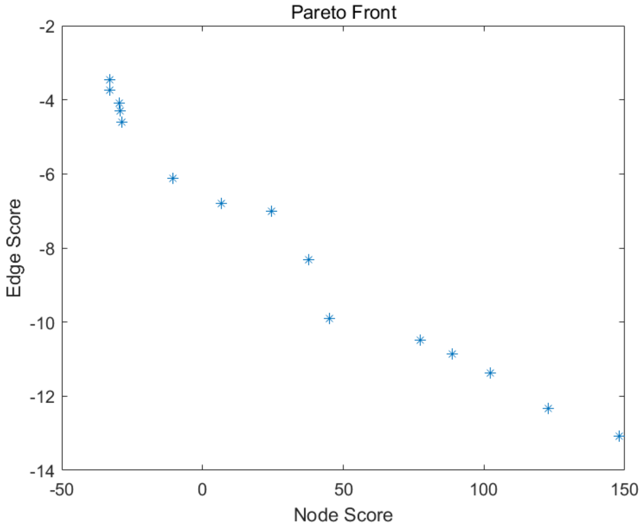


Figure 4.4: Pareto front from applying proposed algorithm on rat network.

In order to analyse the performance of proposed algorithm on real data, we select the solution on top left corn of Pareto front as an example. It is the extreme point with maximised node score S_A. Gene ontology analysis through the online tool of Gene Ontology Consortium [23] for biological process is performed on the decode module, shown in Table 4.2. As gene ontology (GO) terms are given in a hierarchical structure, for simplicity only the top level of GO terms are selected to display.

Because the ageing and cognitive decline process is very complicated and involves a broad range of functional units, the gene annotation for detected module, also contains considerable number of terms. Literature [39] has figured out that the brain ageing process is closely related to the sequential cascade change in metabolic alterations, inflammation, and down regulation of energy-dependent pathways that are necessary to sustain cognitive functions. Looking up the annotations in Table 4.2, we can find terms related to immune response, immune development, and positive or negative regulations on a series of pathways.

4.3 Summary

An integrated approach has been proposed for active module identification to combine information from both node weights and edge weights. The algorithm is motivated by the formulation of objective function COSINE that uses a weight parameter to balance between node and edge score. Our proposed algorithm formulates the problem in a multi-objective optimisation that avoids explicit use and redundant calculation for weight parameter and provides more flexibility in choosing multiple solutions on Pareto front. We have also designed a probabilistic mutation to accelerate the search process.

We first applied our algorithm and several other methods to a series of simulated data. Performance assessment through F-score comparison shows that the proposed algorithm has satisfactory performance in identifying active module from data with complex distributions. Then the algorithm is applied to a network constructed on rats brain ageing and cognitive decline data. The algorithm is able to identify active module with some biologically meaningful annotations. However, one drawback is that the number of gene

Table 4.2: Gene ontology results for the active module identified by proposed algorithm in rat network. This module has 55 nodes, with node score $S_A = 33.09$ and edge score $S_e = 3.47$. p-value gives the statistical significance of corresponding GO term's enrichment in the gene set after FDR correction.

Typical GO terms	p-value
positive regulation of type IIa hypersensitivity	1.59×10^{-02}
Fc receptor mediated stimulatory signaling pathway	2.01×10^{-02}
regulation of type I hypersensitivity	1.98×10^{-02}
platelet degranulation	2.61×10^{-02}
antigen processing and presentation of exogenous peptide antigen via MHC class II	8.61×10^{-05}
neutrophil activation involved in immune response	3.50×10^{-02}
Fc-gamma receptor signaling pathway	3.47×10^{-02}
Bergmann glial cell differentiation	3.91×10^{-02}
regulation of complement activation	3.30×10^{-02}
positive regulation of myeloid leukocyte cytokine production involved in immune response	4.92×10^{-03}
complement activation, classical pathway	9.02×10^{-03}
phagocytosis, engulfment	1.60×10^{-02}
platelet aggregation	2.20×10^{-02}
positive regulation of receptor-mediated endocytosis	3.30×10^{-02}
negative regulation of leukocyte apoptotic process	3.85×10^{-02}
positive regulation of phagocytosis	3.93×10^{-02}
regulation of blood coagulation	1.13×10^{-02}
negative regulation of T cell activation	2.21×10^{-02}
positive regulation of ERK1 and ERK2 cascade	2.02×10^{-02}
regulation of lymphocyte proliferation	2.31×10^{-02}
negative regulation of secretion	3.10×10^{-02}
innate immune response	1.98×10^{-03}
regulation of metal ion transport	8.59×10^{-03}
chemotaxis	3.88×10^{-02}
immune system development	2.27×10^{-02}
regulation of cell migration	1.33×10^{-02}
negative regulation of multicellular organismal process	2.77×10^{-03}
response to hormone	3.38×10^{-02}
cellular response to chemical stimulus	5.44×10^{-03}
regulation of molecular function	3.21×10^{-02}
animal organ development	2.37×10^{-02}

ontology terms is too large to have a very good and deep interpretation. This issue, commonly shared by a number of active module identification methods, will be further explored and addressed in Chapter 6.

CHAPTER 5

PRIOR KNOWLEDGE GUIDED ACTIVE MODULES IDENTIFICATION THROUGH MULTI-OBJECTIVE OPTIMISATION

In this chapter, a prior information guided active module identification approach is proposed to detect modules that are both active and enriched by prior knowledge. We formulate the active module identification problem as a multi-objective optimisation problem, which consists two conflicting objective functions of maximising the coverage of known biological pathways and the activity of the active module simultaneously. Network is constructed from protein-protein interaction database. A beta-uniform-mixture model is used to estimate the distribution of p-values and generate scores for activity measurement from microarray data. A multi-objective evolutionary algorithm is used to search for Pareto optimal solutions. We also incorporate a novel constraint based on algebraic connectivity to ensure the connectivity of the identified active modules.

Application of proposed algorithm on a small yeast molecular network shows that it can identify modules with high activities and with more cross-talk nodes between related functional groups. The Pareto solutions generated by the algorithm provides solutions with different trade-off between prior knowledge and novel information from data. The approach is then applied on microarray data from diclofenac-treated yeast cells to build network and identify modules to elucidate the molecular mechanisms of diclofenac toxicity and resistance. Gene ontology analysis is applied to the identified modules for biological interpretation.

Experiments showed that integrating knowledge of functional groups into the identification of active module is an effective method and provides a flexible control of balance between pure data-driven method and prior information guidance.

5.1 A Novel Framework of Prior Knowledge Guided Active Modules Identification Approach

The network G is represented as $G = (V, E)$ with $p_v \in (0, 1)$ for $v \in V$ where V is the set of nodes, E the set of edges, and p_v the assigned p-value from differential expression analysis for each node v. In the proposed algorithm there are two objectives and one constraint for a given module A:

- Active module score S_A indicating significant changes in gene expression for a given module, to be maximised during search.
- KEGG (Kyoto Encyclopedia of Genes and Genomes) pathway coverage score R_A for the number of covered metabolic pathway by genes in module, to be maximised.
- Algebraic connectivity to check whether a given subgraph is connected or not, used as a constraint to ensure connectivity.

5.1.1 Formulation of Active Module Score

For a given protein-protein interaction network with p-values indicating the gene differential expression level assigned to each node v, an additive score $S^{FDR}(v)$ based on beta-uniform distribution can be calculated using equation 4.3 (see Section 4.1.1 for more details). The active module score S_A is simply the summation of $S^{FDR}(v)$.

5.1.2 Assessment of Prior Knowledge Enrichment

KEGG is an integrated database of high level functions and utilities of biological systems [40]. KEGG PATHWAY is a collection of manually drawn pathway maps representing the knowledge on the molecular interaction and reaction networks. Mapping of pathway information mainly relies on molecular data set, especially large-scale data set such as genomics, transcriptomics, proteomics, and metabolomics. Genes involved in the same KEGG pathway are considered as functionally related to each other. In the experiment KEGG pathway coverage score R_A is formulated as the second objective to measure the enrichment of functional groups in a given module A.

The KEGG pathway information is retrieved from the KEGG REST-style entry for *Saccharomyces cerevisiae* (yeast) [42]. Each entry of the mapping data records one gene and its corresponding pathway. The records are then split into different groups labelled by the pathways. For the i-th pathway, V_i stands for the set of genes it contains. Given a module A with V_A as the set of genes, its KEGG pathway cover rate R_i over the i-th pathway is calculated as

$$R_i = \frac{|V_i \cap V_A|}{|V_i|} \tag{5.1}$$

meaning the percentage this pathway is covered by given module. The cover rate R_i is then compared with a threshold R_{ratio} to determine whether this pathway can be considered as enriched in the given module. The threshold shall be selected carefully. A too high value of R_{ratio} leads to a tiny group of connected pathways genes with positive active module score as the search could not expand to other area under such stringent condition. On the contrary, a very low R_{ratio} could not reflect the meaning for the second objective. In practice, R_{ratio} is set to a series of values for preliminary experiment. The results are analysed and compared to decide a suitable value. The total enriched pathway count R_A is given by

$$R_A = |\{R_i | R_i > R_{ratio}\}|, i \in P \tag{5.2}$$

where P stands for total number of pathways.

5.1.3 Algebraic Connectivity as A Constraint for connectivity

Let $G = (V, E)$ be an undirected edge weighted graph with node set $V = \{v_1, ..., v_n\}$ and non-negative weights $a_{ij} \geq 0$ for two nodes i and j. $a_{ij} = 0$ means there is no edge connection between the two nodes. The weighted adjacency matrix A of the graph is given by the matrix $A = (a_{ij})_{i,j=1,...,n}$ and $a_{ij} = a_{ji}$ for all $i, j = 1, ..., n$ as the graph is undirected. The degree of a node $v_i \in V$ is given by

$$d_i = \sum_{j=1}^{n} a_{ij} \tag{5.3}$$

The degree matrix D is defined as the diagonal matrix with the degrees $d_1, ..., d_n$ on the diagonal. The unnormalized Laplacian matrix L of graph G is then calculated as

$$L = D - A \tag{5.4}$$

Let $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ be the eigenvalues of the Laplacian matrix L, then the algebraic connectivity $\alpha(G)$ of the graph G is given by $\alpha(G) = \lambda_2$, i.e. the second-smallest eigenvalue of the Laplacian matrix of G. The algebraic connectivity $\alpha(G)$ is zero if and only if the graph G is not connected, otherwise it is greater than zero indicating that G is a connected graph.

Besides from being an indication of connectivity, $\alpha(G)$ has also been used as a measure of the robustness in complex networks [36]. It has been long proved that for a non-complete graph $\alpha(G) \leq v(G)$ and we also have $v(G) \leq e(G)$ [19] where $v(G)$ denotes for vertex connectivity as the minimal number of nodes whose removal would result in losing connectivity of the graph, and $e(G)$ denotes for edge connectivity defined in a similar way to vertex connectivity. However, there is also research showing that although $\alpha(G)$ is the lower bound of both vertex and edge connectivity, its relationship to the graph robustness to node and link failures is not trivial.

In the proposed algorithm algebraic connectivity $\alpha(G)$ is calculated for given subgraph and used as a constraint to ensure its connectivity.

After introducing the formulation of two objectives and one constraint, the overall problem of finding a prior knowledge enriched active module is defined as following.

Problem 5.1 (Prior Knowledge Enriched Active Module Identification Problem)**.** *Given a network $G = \{V, E\}$ where V denotes for the set of nodes, E denotes for the set of edges, $n = |V|$ is the number of nodes and each node $v_i \in V, i = 1, 2, ..., n$ is assigned with node weight $S^{FDR}(v_i)$ and labelled with zero, one or multiple pathway groups, find a subgraph $A = \{V_A, E_A\}, V_A \in V, E_A \in E$ so that both S_A and R_A are maximised, constrained to $\alpha(A) > 0$.*

5.1.4 Multi-objective Optimisation Algorithm as Search Strategy

In order to perform multi-objective optimisation to maximise both active module score and KEGG pathway coverage score simultaneously, a multi-objective evolutionary algorithm modified from NSGA-II (non-dominated sorting genetic algorithm II, see [14]) is applied as search strategy for module detection.

- **Solution representation**

  A solution is represented as a binary vector of length n, where $n = |V|$ is the size of network, i.e. total number of nodes. Adding or deleting a node is performed through simply flip one bit of the vector at corresponding site. Thus for the i-th individual L_i in population, we have

  $$L_i = \{l_i^1, l_i^2, ..., l_i^n\} \tag{5.5}$$

  where $l_i^j \in \{0, 1\}$ for $j = 1, 2, ..., n$, $l_i^j = 1$ means the j-th node is in the module.

- **Fitness function**

Active module score S_A and KEGG coverage score R_A are used as two objectives. As the implementation of the algorithm is aimed at minimisation both objectives, scores calculated from above equations would be given an extra negative sign.

- **Initialisation**

The search starts by randomly initialising a group of small cores in network. Nodes with high $S^{FDR}(x)$ scores are selected as seeds of potential modules to begin with. Number of seed nodes is decided by the population parameter for evolutionary algorithm. For instance, if population is set to 50, nodes with top 50 $S^{FDR}(x)$ scores are selected as seeds. In the case when the population size exceeds network size, every node will be selected as a seed. In initialisation stage, neighbouring nodes of a seed with positive scores are added to the module in which the seed represents. A detailed description is shown in Algorithm 5.1.

Algorithm 5.1: Initialisation Stage

```
Input: population number $pop$, adjacency matrix A of the whole network, a list of
       $S^{FDR}(v)$ assigned to each node
Output: Initialised population
$\{v_1, v_2, ..., v_n\} \leftarrow$ sort nodes by $S^{FDR}(v)$ in descending order ;
for $i \leftarrow 1$ to pop do
    $L_i \leftarrow$ zeros of length $n$ ;
    $k \leftarrow mod(i, n)$ ;
    $l_i^k \leftarrow 1$ ;
    node set $V_{neighbours} \leftarrow$ neighbouring nodes of seed $v_k$ with positive scores;
    for $v_j \in V_{neighbours}$ do
        // neighbouring nodes of a seed with positive scores are added
           to the module
        $l_i^j \leftarrow 1$ ;
    end
end
$population \leftarrow \{L_1, ..., L_{pop}\}$;
return population
```

- **Parent selection**

Binary tournament selection is applied for selecting parents to reproduce. In some cases when the population converges too fast, this step is skipped to decrease selection pressure, thus the whole population would be used for reproduction.

- **Reproduction**

  Single point crossover is applied to selected parents. Mutation is performed by adding neighbouring nodes with positive $S^{FDR}(x)$ score or in a pathway into current module each time. Offspring generated is added to parental population to form a combined population with twice the size, waiting to be sorted and selected.

- **Clearing procedure**

  An extra clearing procedure is applied after reproduction and before non-dominated sorting. The step is introduced because in practise the algorithm tends to generate a number of replicated solutions when converging towards global optima. However, considering the natural property of our optimisation problem, it is reasonable to obtain multiple optima, both those global on the non-dominated Pareto front and those dominated local optima, each representing the most significantly changed modules and modules that do not change that significantly, but still worth looking into. This procedure, inspired and simplified from Petrowski [68], detects replicated solution groups, preserves one copy, and resets all other individuals as infeasible solutions which will soon be eliminated after sorting and replacement. A detailed description is shown in Algorithm 5.2.

- **Sorting and replacement**

  The algorithm uses fast non-dominated sorting and crowding distance assignment as detailed in Ref [14] to generate new population from the combined population efficiently and preserve solution diversity.

- **Constraint handling**

  To ensure the connectivity of detected module after crossover, algebraic connectivity $\alpha(G)$ is used as a constraint. Solution with non-positive algebraic connectivity vio-

Algorithm 5.2: Clearing Procedure

```
Input: Current population, population size pop
Output: Population cleaned
population ← sort input population by $S_A$ in ascending order;
Mark individual $L_m \leftarrow L_1$ ;
for $i \leftarrow 2$ to pop do
    if $S_A(L_i) == S_A(L_m)$ then
        set $L_i$ as infeasible;
    else
        $L_m \leftarrow L_i$;
    end
end
return population
```

lates the constraint, indicating itself a disconnected subgraph and thus an infeasible solution. Replicated solutions are also marked infeasible in the clearing procedure. Infeasible solutions are dominated by all feasible solutions.

5.2 Experimental Studies

The proposed algorithm is aiming at identification of modules that are both showing activity in expression (i.e. having high active module scores) and enriched with metabolic pathways. To test its performance, two networks are selected as experimental networks. Network 1 is the network that Ideker used [34] to show the experimental results of jActiveModule. It has a relatively suitable size for visualisation, contains a portion of genes that show significant changes in expression level, and can be viewed as a nice model of yeast galactose utilization pathway. Network 2 is constructed from mapping differential analysis results to the whole Interactome Network. In such a network, a large number of unrelated pathway information would be included, which is a challenge for the proposed algorithm to target on the pathways that are most relevant to the active modules.

5.2.1 Construction of Experimental Networks

5.2.1.1 Network 1: A Small Molecular Interaction Network on Galactose Utilization Pathway

A small molecular interaction network once used by Ideker [34] is used as a test network. The molecular interaction networks visualisation software Cytoscape [74] provides jActiveModule as a plugin to find expression activated modules. The tutorial in Cytoscape App Store [35] provides samples data consists of a network edge list file as a model of the galactose utilisation pathway in yeast and a companion expression file contains p-values to describe the significance of each observed change in expression. p-values under condition labelled as *GAL80R* are extracted and overlaid to network file, resulting in a network with 330 genes.

5.2.1.2 Network 2: Yeast Drug Reaction Network Constructed from Differential Analysis and Interactome Mapping

Gene expression data on yeast's reaction to diclofenac is downloaded from GEO (NCBI Gene Exprssion Omnibus) database [88]. Diclofenac is a widely used analgesic drug that can cause serious adverse drug reactions [83]. Yeast is used as model eukaryote to capture the cellular changes under the treatment of diclofenac. The data provides the microarray expression for diclofenac-treated yeast cells and control cells, each with 5 replicates. Differential expression analysis between diclofenac-treated group and control group is performed using the on-line tool GEO2R [58], with p-value adjustment set to Benjamini and Hochberg false discovery rate control. After deleting genes with adjusted p-value larger than 0.05, a set of differentially expressed genes is generated for interactome mapping.

Protein-protein interaction data is download from BioGRID [9], an integrated and up-to-date public database that archives and disseminates genetic and protein interaction data from model organisms and humans. To be specific, the downloaded file is BIOGRID-ORGANISM-LATEST.tab2.zip that separates interactions into distinct files based on

Organism and was released on June 30, 2016. File for interactions of *Saccharomyces cerevisiae* is extracted for use. As the whole interaction data contains tens of thousands of proteins and millions of interaction records, a considerable amount of proteins have no corresponding records in given expression data or show no differential expression. Those proteins shall be excluded from the final network in order to avoid the waste of both computational resource and analysis attention. According to the filtering method applied by Muraro and Simmons [56], interactions containing at least one differentially expressed gene are selected as an attempt to include indirect interactions. The resulting network concerning yeast cellular reaction to diclofenac consists of 1803 nodes and 3356 edges.

Table 5.1: Parameters for experimental networks.

Parameters	Network 1	Network 2
nodes	330	1803
edges	359	3356
a	0.113	0.280
λ	9.07×10^{-2}	0.168
α (FDR)	1×10^{-4}	1×10^{-4}
τ	1.76×10^{-4}	7.71×10^{-6}
R_{ratio}	0.6	0.8

5.2.2 Experimental Results

5.2.2.1 Analysis of Network 1

To estimate distribution for p-values, the parameters of BUM model a and λ are estimated by R package BioNet [7]. Figure 5.1 shows the fitted model. As the majority of genes in yeast network have a very significant p-value, threshold τ is calculated at an extremely stringent FDR level as an attempt to control the size of detected module. Parameter details are shown in table 5.1.

As a benchmark, the jActiveModule method is applied to the network via Cytoscape plugin, generating 5 active modules by default. Figure 5.2 gives a visualisation of the network by Cytoscape, with detected active modules mapped on it. To understand the

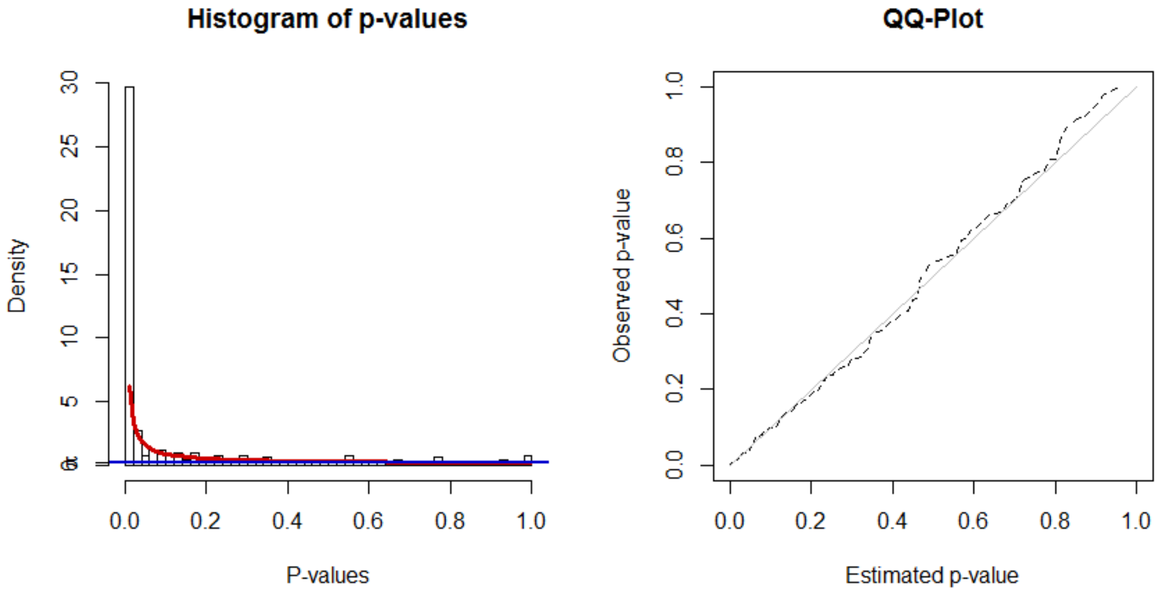


Figure 5.1: BUM model estimation on p-values in network 1. Left figure is a histogram of p-values with fitted beta-uniform-mixture model distribution. Blue line indicates the uniformly distributed noises and red line the signals as beta distribution $B(a, 1)$. Right figure is a Q-Q plot of the fitted distribution versus the empirical p-values for network 1.

biological function of modules, gene ontology (GO) annotation for biological process is applied to modules by enrichment analysis tools provided on Gene Ontology Consortium [13]. The tool only asks for a submission of gene list, GO type (biological process, molecular function, cellular component) and species. The results is shown in Table 5.2. Among the 5 modules, 3 modules are enriched in the GO term galactose catabolic process via UDP-galactose with p-values from 4.85×10^{-05} to 3.42×10^{-04}. Other 2 modules are too tiny to have accurate explanation.

The proposed algorithm is applied to network 1 with threshold $R_{ratio} = 0.6$ for KEGG pathway coverage score, resulting in a set of 13 Pareto solutions. As a feature for multi-objective optimisation, all the modules in the same Pareto front are equally good. No one out performs another. In order to show the difference of those modules in trade-offs between two objectives, we selected 3 modules from the 13 Pareto solutions:

- Module 1: the extreme point on the Pareto front with maximum active module score $S_A = 393.41$.

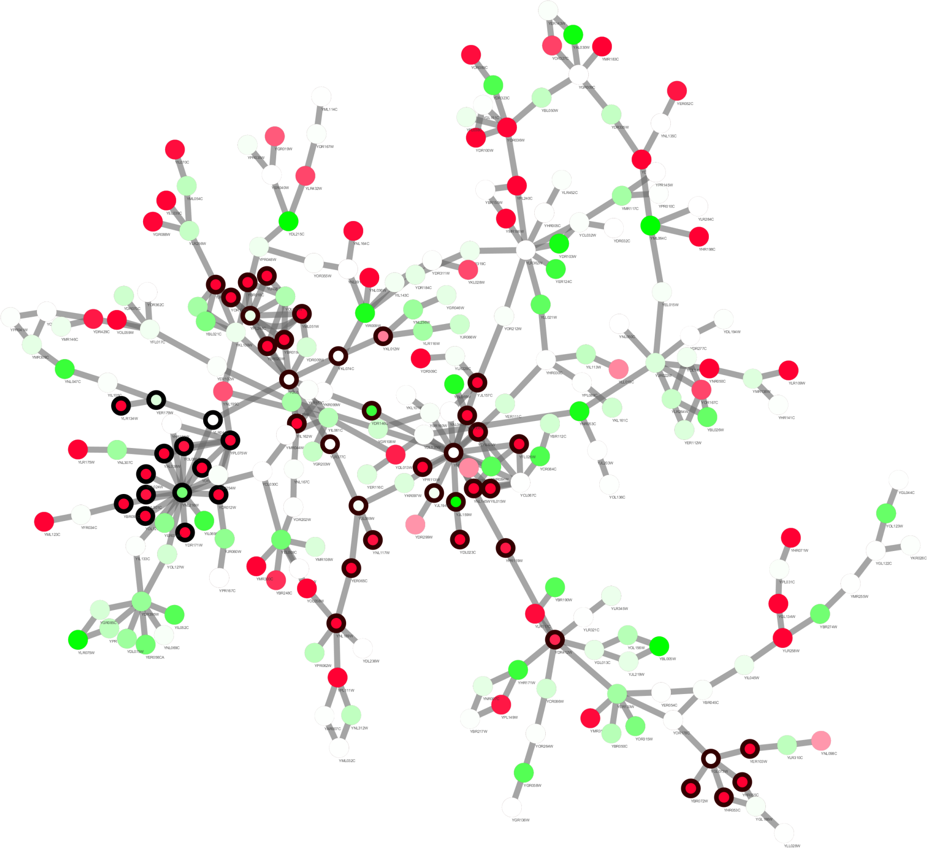


Figure 5.2: Network 1 with active modules detected by jActiveModule. Each node denotes for one gene. Node colour is a continuous mapping of the p-value generated from differential expression analysis. Red colour indicates a significant change with small p-value and green colour means no significant difference. The point where colour will change between red and green is set to the threshold $\tau = 1.76 \times 10^{-4}$ that is used as a parameter for the proposed algorithm. Colour of nodes near the changing point is white. Modules identified by jActiveModule are highlighted with black node border. Modules may overlap with each other.

- Module 2: at the knee point of the Pareto front, which represents the optimal trade-off between active score ($S_A = 268.96$) and KEGG pathway coverage score ($R_A = 19$)
- Module 3: the extreme point on Pareto front with maximum KEGG pathway coverage $R_A = 25$.

GO analysis for biological process is performed on the three modules. The results together with the objective function values are tabulated in Table 5.3. We also visualise

Table 5.2: Gene ontology results of modules detected by jActiveModule in network 1. S_A and R_A are the objective functions of active module score and KEGG pathway coverage score, respectively. The values are calculated by the proposed objective functions using the same parameters setting as the proposed algorithm. $\tau = 1.76 \times 10^{-4}$ and $R_{ratio} = 0.6$.

module	size	S_A	R_A	typical GO terms	p-value
1	26	250.39	1	galactose catabolic process via UDP-galactose	3.42×10^{-04}
				glycolytic fermentation to ethanol	2.72×10^{-03}
				amino acid catabolic process to alcohol via Ehrlich pathway	1.25×10^{-02}
2	5	58.21	0	response to heat	2.16×10^{-03}
3	16	270.79	2	galactose catabolic process via UDP-galactose	4.85×10^{-05}
4	18	169.89	2	galactose catabolic process via UDP-galactose	1.15×10^{-04}
				cellular carbohydrate metabolic process	3.27×10^{-02}
5	4	37.05	0	None	Not available

Modules 1 and 2 in Figures 5.3.

By comparing the results in Table 5.3 with those in Table 5.2, we found that Module 1 identified by the proposed algorithm have better active module score (S_A) and KEGG pathway coverage score (R_A) than all the modules found by jActiveModule algorithm. Such results indicate that by incorporating the prior knowledge, we can guide the algorithm to search areas in the network with more significant activity.

From Figure 5.3 and Table 5.3, we found that compared with jActiveModule that searches for small and separated modules, the proposed algorithm tends to identify a large connected subgraph. Even for Module 1 where the active module score is maximised, because of the integration of the prior knowledge, highly active areas are more likely to be connected together by intermediate nodes that might not be significantly differential expressed, but serve as a bridge for cross-talk between neighbouring functional areas.

By visualisation of those Pareto solutions (figures not shown), we found that as the solution on Pareto front moves from maximum active score to maximum pathway coverage score, such intermediate nodes appear with higher frequency. We also found that, as R_A gets higher, detected module expands from a small core area with high activity to a broad

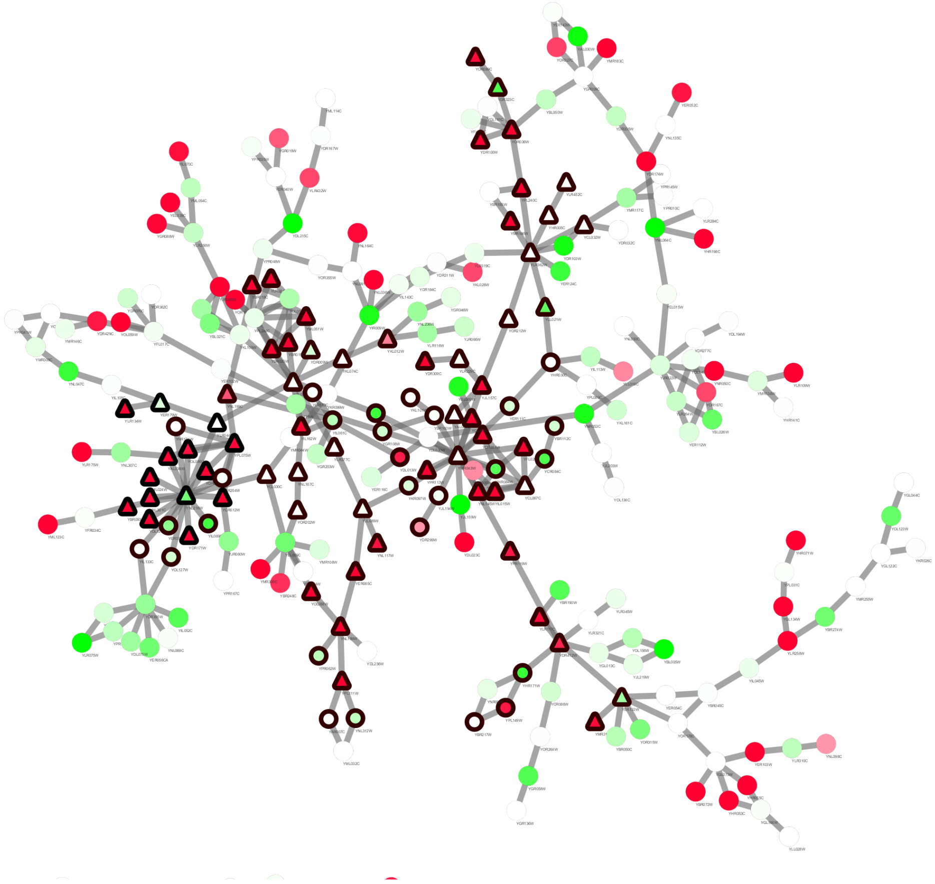


Figure 5.3: Visualisation of extreme point solution and knee point solution detected by the proposed algorithm in network 1. Module 1 is drawn by triangle shaped nodes and module 2 is highlighted with black border. Node colour is set in the same way as figure 5.2. Module 1 is the extreme point on Pareto front with maximised active score S_A. It contains the majority of red nodes that are connected densely, indicating high activity. Notice that compared to small separated modules identified by jActiveModule shown in figure 5.2, this module tends to connect small areas of red nods by including linkage nodes with white or light green colour. Although these intermediate nodes shows only modest changes in expression, they serve as bridges for cross-talk between functional groups, or as transcription factors that regulate other genes. Module 2 is the knee point of the Pareto front with optimal trade-off between S_A and R_A. Compared to module 1, this module expands broader as R_A gets higher.

Table 5.3: Gene ontology results of 3 modules on Pareto front detected by the proposed algorithm in network 1. Module 1 is the extreme point with maximised active score S_A. Module 2 is a balanced solution between S_A and R_A. Module 3 is the other extreme point with maximised pathway coverage score R_A.

module	size	S_A	R_A	typical GO terms	p-value
1	65	393.41	9	galactose catabolic process via UDP-galactose	5.15×10^{-03}
				negative regulation of mating-type specific transcription from RNA polymerase II promoter	1.21×10^{-02}
				glycolytic fermentation to ethanol	4.05×10^{-02}
				pheromone-dependent signal transduction involved in conjugation with cellular fusion	6.39×10^{-03}
				cellular carbohydrate metabolic process	4.16×10^{-02}
2	92	268.96	19	negative regulation of mating-type specific transcription from RNA polymerase II promoter	4.67×10^{-04}
				galactose catabolic process via UDP-galactose	1.63×10^{-02}
				regulation of transcription during mitosis	7.19×10^{-03}
				gluconeogenesis	1.84×10^{-04}
				glycolytic process	2.87×10^{-02}
				pyruvate metabolic process	4.20×10^{-02}
				response to pheromone involved in conjugation with cellular fusion	3.93×10^{-06}
3	126	181.3	25	negative regulation of mating-type specific transcription from RNA polymerase II promoter	1.80×10^{-03}
				galactose catabolic process via UDP-galactose	4.48×10^{-02}
				C-terminal protein lipidation	1.62×10^{-02}
				gluconeogenesis	1.36×10^{-03}
				ADP metabolic process	2.47×10^{-04}
				pyruvate metabolic process	7.73×10^{-05}
				response to pheromone involved in conjugation with cellular fusion	1.47×10^{-02}
				ribonucleoprotein complex assembly	5.31×10^{-03}

area with more varied functional groups while still keeping overall activity. This result indicates that by using prior knowledge, we are able to reveal underlying mechanisms that link different activities in the network.

While all the three modules are significantly enriched in the GO term "galactose catabolic process via UDP-galactose" (corresponding p-value 5.15×10^{-03}, 1.63×10^{-02} and 4.48×10^{-02}, respectively), annotations for Module 1 (the extreme point with maximum activity score S_A) are more densely related with galactose metabolic process. On the other hand, for Module 3 with maximum KEGG pathway coverage score R_A, core annotations remain the same while additional annotations concerning essential biological processes increases. However, it is worth noting that, all the additional annotations can be reasonably related to the cellular response to disturbance in galactose utilisation pathway.

The most interesting module is Module 2, which represents the optimal trade-off between prior knowledge and novel information from data. It is worth noting from Tables 5.3 and 5.2 that, even it is a knee point solution, Module 2 has a slightly worse S_A but much higher R_A than all the modules identified by JActiveModule. We can also observe from Table 5.3 that, module 2 has a range of slightly broader annotations concerning metabolic process of galactose, pyruvate and gluconeogenesis, which are highly relevant to galactose utilisation pathways [81].

5.2.2.2 Analysis of network 2

Parameters of BUM model a and λ to fit p-value distribution are estimated as shown in Figure 5.4. Threshold τ is calculated at given FDR level. See Table 5.1 details of parameters.

The proposed algorithm is applied to network 2 with threshold $R_{ratio} = 0.8$ for KEGG pathway coverage score, resulting in a set of 12 Pareto solutions. Solutions on the Pareto front are chosen for gene ontology analysis on biological process. Surprisingly, all identified modules shows a high consistency in the annotation on drug reaction, which ex-

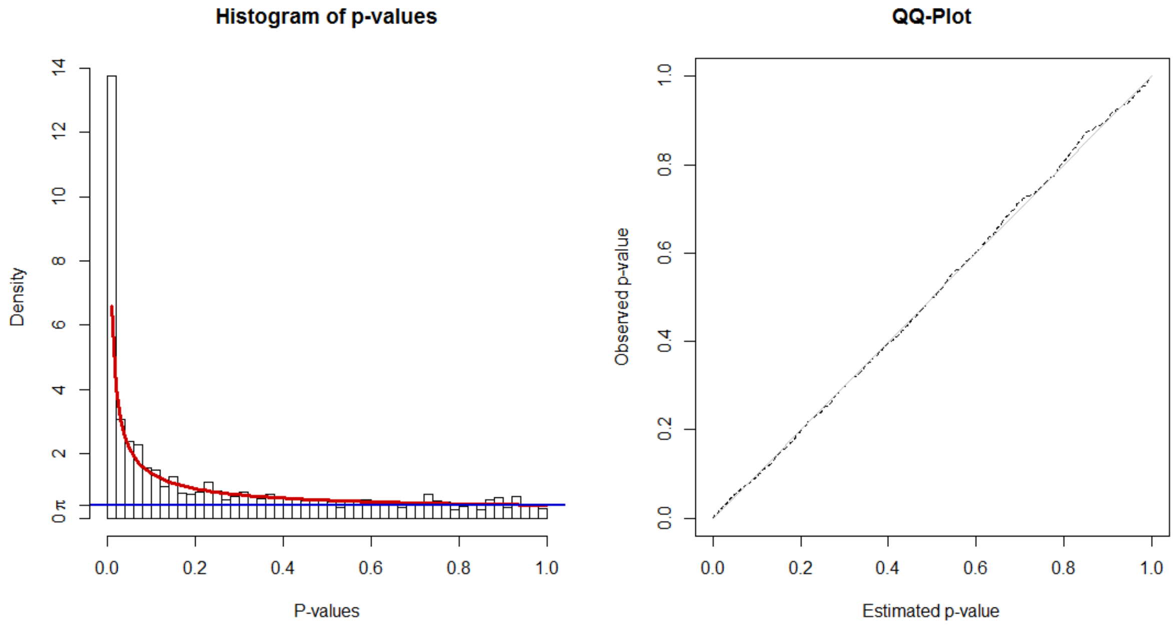


Figure 5.4: BUM model estimation on p-values in network 2. Histogram of p-values with fitted BUM model and a Q-Q plot of estimated and empirical distribution of p-values for network 2. As the network size increases, estimation becomes more accurate.

actly reflects the cellular response for yeast under the diclofenac treatment. Three genes (YDR406W, YOR153W and YOR153W, all act as ATP-binding transporter, for detailed functional explanation, see caption in Figure 5.5) that play an important role in the cellular reaction and resistance to drug treatment are discovered in all the 12 modules, indicating the accuracy and robustness of searching algorithm.

Similar to the analysis methods for results in network 1, 3 representative modules on Pareto front with different trade-off between active score S_A and pathway coverage score R_A are select for gene ontology annotation (see Table 5.4) and visualisation (Figure 5.5). From Table 5.4 we can see that as pathway score R_A increases, size of module increases and the annotation includes a larger range of biological processes. As drug reaction is considerably complicated response that involves a series of up or down regulation in related function groups such as protein kinase pathway, ribosome biogenesis, rRNA processing and zinc-responsive genes [83], the enriched annotation in modules with higher R_A provides a guidance of deciding which functional groups to look into as it combines both prior knowledge from existing interaction database and novel information from gene expression

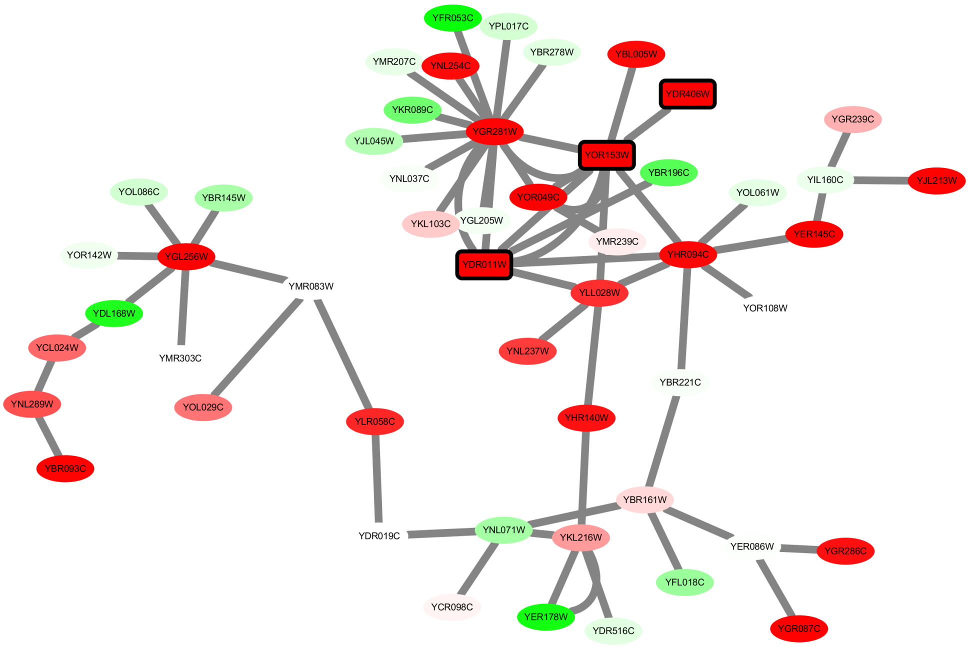


Figure 5.5: Visualisation of module 3 identified by the proposed algorithm in network 2. Each node represents for a gene. The setting for node colour is the same with figure 5.2. The turning point between red and green is set to the value $\tau = 7.71 \times 10^{-6}$. Three rectangle shaped nodes with black border are genes involved in drug export and are highly consistent in all modules. YDR406W is an ATP-binding cassette multi-drug transporter. YDR011W is a ATP-binding cassette transporter. YOR153W is also an ATP-binding cassette multi-drug transporter. The three genes serve as an important role in yeast's resistance to diclofenac.

data specific for given experimental conditions.

5.3 Summary

An integrated multi-objective approach has been proposed for active module identification. The algorithm is motivated by the idea that incorporating prior information into data-driven method would provide new insights into sophisticated biological processes. We have also designed an constraint based on algebraic connectivity to ensure the connectivity of the identified active modules.

We first applied our algorithm on a small molecular interaction network, which identified a set of Pareto solutions that represents different trade-off between prior knowledge

Table 5.4: Gene ontology results of 3 modules on Pareto front detected by the proposed algorithm in network 2.

module	size	S_A	R_A	typical GO terms	p-value
1	34	91.01	0	drug export	1.79×10^{-03}
				cellular response to drug	4.71×10^{-02}
2	39	57.56	4	drug export	2.84×10^{-03}
3	62	46.332	8	drug export	1.21×10^{-02}
				amino acid catabolic process to alcohol via Ehrlich pathway	8.65×10^{-09}
				ethanol metabolic process	3.71×10^{-06}
				NADH oxidation	3.73×10^{-03}
				glycolytic process	4.34×10^{-03}
				fermentation	1.40×10^{-02}
				macromolecule metabolic process	2.51×10^{-02}

and novel information from data. Gene Ontology analysis results show that it successfully identifies modules with relevant and reasonable biological interpretations. The algorithm was applied to the second network, The approach is then applied on a microarray data set from diclofenac-treated yeast cells and identify modules to elucidate the molecular mechanisms of diclofenac toxicity and resistance. The algorithm identifies accurate and consistent modules with biological function densely related to given cellular response, proving that the integrated approach for network construction is feasible and that the proposed algorithm is able to identify biologically meaningful modules in large scale network.

CHAPTER 6

SIMULTANEOUS DETECTION OF ACTIVE MODULE AND TOPOLOGICAL COMMUNITY THROUGH MULTIFACTORIAL EVOLUTION

In this chapter, a novel algorithm framework of detecting active module and topological communities simultaneously using evolutionary multitasking is proposed to improve the computational outcome of active module detection and help the interpretation of biological meaning. This algorithm uses an additive node score for measuring activity of module, and searches for network division through modularity maximisation.

A series of task-specific algorithm designs and improvements are made based on the original framework of evolutionary multitasking algorithm. We have developed a unified genetic representation and problem-specific decoding methods for the two tasks. Task-specific mutation operators are designed for individuals specialised in different tasks. Individuals talented in active module identification task are applied with probabilistic local search to approach the optimisation of active module score, and individuals specialised in topological community detection task are performed with community merging strategy as an imitation of classic fast bottom-up community detection algorithms. Uniform crossover is adopted to preserve the diversity of population, help explore solution space, and avoid being stuck in local optima. Extra solution improvement step is designed to further enhance the value of modularity by fixing the connectivity problem for detected communities.

The proposed algorithm is first applied on some classic community structured networks to test its performance on modularity optimisation and compare with some other published algorithms. It is then applied on a yeast molecular interaction network to simultaneously run both tasks.

Results show that the proposed algorithm has satisfactory performance on both tasks. It is able to detect network divisions with values of modularity comparable or even better than classic community detection algorithms. It also successfully identifies active modules with considerably high scores. By mapping the community structure to the active module and further dividing the module into smaller fractions, this algorithm provides a new way to better interpreter the biological meaning of active module. Gene annotation analysis results show that the fractions from one active module have more specific and clear meanings than the usually general and ambiguous interpretation for the whole active module.

6.1 Background

Topological modules, also known as communities, are locally dense neighbourhoods with more inner interactions than outside interactions. In biological networks, communities are used to approximate the functional units of cellular process and organisation [30]. This is because biological components exert functions by interacting with each other. This components organised as functional units that overlap with communities. Therefore, by identifying communities [38], we can identify (part of) functional modules [5]. Furthermore, by comparing functional modules at different disease stages, we can reveal essential biological mechanisms [48, 31, 54]. However, communities do not consider network activities such as gene expression of each gene in a biological network. With this drawback, communities cannot fully depict the dynamic mechanisms of biological systems.

Active modules, on the other hand, consider network activities. They have been proposed to depict the dynamic mechanisms of biological systems. An active module is a region (sub-network) in a biological network that show striking changes in molecular activity. These active modules are often associated with a given cellular response [54]. However, existing active module identification algorithms do not consider modular structure. As a result, it is difficult to associate the identified active modules with functional units. Hence, active modules cannot precisely depict the how functional units are activated. Because of this drawback, the dynamic mechanisms reveal by active modules are not accurate.

As mentioned above, both communities and active modules reveal some aspects of the underlying biological mechanisms. It is desirable to identify and study both of them, especially their overlaps, which are important to reveal the interactions between structure, activities and functions. Naively, we can apply algorithms to identify communities and active modules separately. However, we hypothesise that by identifying communities and active modules simultaneously, we are able to reveal new insights that cannot be achieved through identifying them separately. This hypothesis is based on the fact that these two types of modules overlap with each other in structure, and complement each other in illustrating functions and dynamics of networks. By searching both types of modules simultaneously, we could exploit their latent complementarities, which will lead to new insights.

To test our hypothesis, we propose a novel multitask module identification algorithm based on multifactorial evolution, which is a evolutionary algorithm that simultaneously solves multiple tasks that may or may not be interdependent [28]. Different from multi-objective evolution in which tasks are conflicting with each other, multifactorial evolution requires no prior knowledge of inter-task dependencies and doesn't aim at optimal trade-offs. During the evolutionary search each task contributes a special factor to the search space, encouraging the transfer of unique genetic materials between tasks, which makes multifactorial evolution more powerful than separately performing single task search.

To our best knowledge, this is the first evolutionary multitasking algorithm that can identify both communities and active modules simultaneously. We have designed a novel unified genetic representation for multiple tasks, problem-specific decoding scheme, and task specific genetic operators. Through experimental tests on both tasks we have proved that our method has satisfactory performance on both tasks. More importantly, it help us to gain insights into biological mechanisms that cannot be obtained by community detection and active module identification algorithms alone. Supplementary materials including MATLAB source code, formatted input data and experimental results are available via https://github.com/WeiqiChen/Mumi-multitask-module-identification.

6.2 Related Work

6.2.1 Evolutionary Multitasking

Evolutionary multitasking investigates into the implicit parallelism of evolutionary optimisation problems. An introductory study [28] on evolutionary multitasking shows that it allows for implicit transfer of genetically encoded information across multiple optimisation tasks. This process, also known as transfer learning, improves the efficiency and convergence speed of evolutionary multitasking on computationally expensive problems.

The idea of accelerating convergence via information transfer between objectives is not newly invented by multitasking. Previous researches on multi co-objective evolution [44] and memetic search [17, 18] have already shown that the knowledge transfer and reuse across objectives is able to improve search performance of evolutionary algorithm on computationally expensive problems. In the context of computational intelligence, memes are referred to as recurring patterns or knowledge embedded in computational representations [67]. In an early study [18] that formulates transfer learning as computational operators, knowledge learned in previous problem-solving process is transferred in the form of memes as building blocks, and helps accelerate future search. A similar study [47] on re-usable knowledge extraction proposes the concept of simultaneous problem learning that emphasises on the interaction between optimiser and problem learning.

Research on evolutionary multitasking is strongly triggered by the need in fast developing cloud computing industry where cross-domain optimisation must be handled with high efficiency. Different to traditional multi-objective optimisation that has one single search space, multitask optimisation is capable of dealing with multiple search spaces, each corresponding to an individual optimisation task [66]. Dependency among tasks are not required for multitasking. The essential point is that it handles cross-domain optimisation through a unified solution representation scheme [28, 66] for objectives across domain. Research [66] has shown that genetic operator applied to the unified genetic space is able to drive knowledge transfer between different optimisation tasks across domain, thus proven that evolutionary multitasking indeed works.

Evolutionary multitasking has a broad range of application that are not restricted to cloud computing or solely cross-domain multitasking. It has been applied to a series of classic combinatorial optimisation problems [89] as well as real world problems like manufacturing process design [29], neural network training [11], bi-level optimisation [27], etc. Nevertheless, it is still a new emerging field that has far not been fully explored. The development of more efficient evolutionary multitasking algorithms and further application to numerous complicated real world problems are promising and attractive future directions in this field.

6.2.2 Modularity Optimisation Methods in Community Detection

Modular or community structure is an essential structural property that reflects relationships among members of a network. It's also one of the most studied network features [60, 21]. A community is usually viewed as a densely connected region in network that has more inner connections than outer connections. Detection of community structures have been studied for many decades under different terminologies like graph partitioning, network division, hierarchical clustering, or block modelling [64].

One of the most successful methods is the modularity optimisation proposed by Newman and Girvan [65]. In their work, a scalar measurement called modularity, which is based on their previous work on assortative mixing [59], is formulated to assess the quality of a given division of an undirected network. They designed an algorithm that divides a network into communities by iterative removal of edge with the highest betweenness score in remaining edges. Edge with high betweenness score is a sign for bottleneck in traffic moving in a network, thus less likely to be located inside a community. Modularity, also denoted as Q, is then used as a guidance for choosing the number of communities a network should be divided into. The value of Q falls between 0 and 1. If the network division is no better than random, then Q = 0. An increase in the value of Q indicates a better community structure. In their experimental results it's clear that as the edges gradually get removed from network, forming different level of split of network, modularity typically first increases and then decreases. Usually there are only one or two peaks during the whole process, indicating the strongest community structure that ever appears.

The algorithm based on the removal of edges with high edge betweenness is able to find densely connected local areas and has been widely used. However, the major drawback of this algorithm is the high computational demand. In the worst case it runs in time $O(m^2n)$ on a network with m edges and n nodes, or $O(n^3)$ on a sparse network. Thus the application of this algorithm is limited to only small scale networks. To fix this problem, later on Newman proposed a fast algorithm for community detection that adopts a different strategy [62] and runs in time $O((m + n)n)$, or $O(n^2)$ on a sparse network. Opposed to the previous algorithm which is a top-down removal and division method, this fast algorithm starts from assigning each node in the network as an isolated community, and gradually join two communities together by the criteria that modularity Q can be increase the most or at least decrease the least. According to the experimental results on both computer generated and real world networks, this algorithm generates excellent results and can be thousands of times faster than previous one.

The bottom-up greedy search based agglomerative strategy is also adopted by another work using a different algorithm [12] for finding community structure in very large network. This algorithm takes advantage of some shortcuts in the optimisation problem. Instead of maintaining the adjacency matrix and calculating the change of modualrity when joining two communities, it only store and update a matrix of changes in modularity. It also maintains some sophisticated data structure to ensure the running speed. This algorithm runs in time $O(mdlogn)$ where d is the depth of the dendrogram describing the network. For many real-world networks that are sparse and have a hierarchical structure, the algorithm has approximately linear running time $O(nlog^2n)$. In addition, an other influential algorithm called fast unfolding [10] uses a similar bottom-up merging method, only that it is divided into two phases, each repeatedly merging individual nodes or existing small communities. It has been proven to be able to run in huge network with more than 100 million nodes.

Besides from those top-down and bottom-up search strategies, there is an algorithm that maximises the value of modulairty based on an two partitioning strategy [16]. This heuristic search algorithm starts by dividing the whole network into two random partitions with equal number of nodes. Every connected components in the partition is considered as one community. A fitness measuring the contribution of an individual node to the value of modularity is calculated depending on current community division. In each iteration node with the lower fitness is moved from one partition to the other. The algorithm terminates when the modularity could not be improved any more, indicating an optimal state with a maximised value of modularity.

Although there have been researches showing that community detection by modularity optimisation may suffer from resolution limit and thus fail to identify communities that are smaller than a scale depending on some network parameters [22], optimisation of modularity Q is still one of the most successful and widely used community detection methods. By simply mapping a weighted network to an unweighted multigraph, it is showed that modularity works on weighted network as well [61]. As exhaustive search for highest modularity in network is intractable, a variety of approximate methods [46, 73, 16, 10] have been applied to modularity optimisation. Among those methods Newman showed that the modularity can be expressed in terms of eigenvectors of a so called modularity matrix, and thus the optimisation of modularity is converted to the problem of finding the leading eigenvector of the modularity matrix and get a two-community partitioning according to the signs of elements in the eigenvector. Each partition is again divided through this method until all the signs in leading eigenvector is the same, indicating that current partition can no longer be dividied. The spectral algorithm [63] is highly efficient for community detection. It is later shown that spectral modularity maximisation is an optimal method for community detection using modularity approach [57].

6.3 A Novel Framework of Multifactorial Evolution for Active Module and Topological Community Detection

In this section we give a formal description of a novel algorithm framework to identify active modules and community structures simultaneously in a node weighted biological network. It is so far as we know the first to formulate the two widely studied problems into one multifactorial evolution paradigm. In the aim of achieving multitasking in the same algorithm scheme, we have developed a unified genetic representation acting as a general solver for the two different tasks and corresponding task-specific decoding method. We have also designed task-specific mutation and local search operators in order to improve the algorithm performance. A final solution modification strategy has been developed and applied to the output solution by the evolutionary algorithm to again enhance the values of objective functions and produce results of higher quality.

6.3.1 Basic Structure of Multifactorial Evolution

We use the technique proposed by Gupta et. al [28] to compare the fitness of individual solutions in a multitasking context. The core concepts of this technique is the definition of scalar fitness φ and skill factor τ for an individual. In the initialisation stage, every individual in the population is evaluated with respect to every optimisation task in the multitasking environment. For each task T_j an individual L_i has a factorial rank r_j^i corresponding to the rank of the individual's objective fitness for this task in the whole population. The lower number the rank is, the better performance individual shows in specified task. For K number of tasks an individual L_i is assigned with a list of K factorial ranks $\{r_1^i, r_2^i, ..., r_K^i\}$. The scalar fitness φ_i of individual L_i is based on its best rank among all the tasks, given by

$$\varphi_i = \frac{1}{min_{j \in \{1,...,K\}}\{r_j^i\}} \tag{6.1}$$

The skill factor τ_i of individual L_i is then given by

$$\tau_i = argmin_j\{r_j^i\}, j \in \{1, 2, ..., K\} \tag{6.2}$$

meaning the task individual L_i is most effective. A basic structure of multifactorial evolutionary algorithm is described in Algorithm 6.1. More details on concept definition and multifactorial evolutionary algorithm scheme can be found in reference [28].

Algorithm 6.1: Basic Structure of Multifactorial Evolutionary Algorithm

```
Population initialisation as current-pop ;
Evaluate every individual with respect to every optimisation task ;
Compute the skill factor $\tau$ of every individual ;
while stopping criteria not satisfied do
    Apply Crossover and Mutation on current-pop to generate offspring-pop ;
    Evaluate offspring individuals for selected optimisation tasks ;
    intermediate-pop $\leftarrow$ Union(current-pop, offspring-pop);
    Update the scalar fitness $\varphi$ and skill factor $\tau$ for every individual in
      intermediate-pop ;
    current-pop $\leftarrow$ fittest individuals in intermediate-pop
end
```

6.3.2 Definition of Tasks

This multitasking problem contains two tasks: identification of active modules and division of network into structural communities. For a given protein-protein interaction network with p-values indicating the gene differential expression level assigned to each node v, an additive score $S^{FDR}(v)$ can be calculated using equation 4.3 (see Section 4.1.1 for more details). The active module identification problem is formulated as following.

Problem 6.1 (Active Module Identification Problem)**.** *Given a network $G = \{V, E\}$ where V denotes for the set of nodes, E denotes for the set of edges, $n = |V|$ is the number of nodes and each node $v_i \in V, i = 1, 2, ..., n$ is assigned with node weight $S^{FDR}(v_i)$, find a connected subgraph $S = \{V_S, E_S\}, V_S \in V, E_S \in E$ so that $\sum_{v_i \in V_S} S^{FDR}(v_i)$ is maximised.*

For a given division on network with adjacency matrix A_{ij}, the modularity defined by Newman and Girvan [65] and modified to be suitable for edge weighted network is calculated as

$$Q = \frac{1}{2m} \sum (A_{ij} - \frac{k_i k_j}{2m}) \delta(c_i, c_j) \tag{6.3}$$

where

$$m = \frac{1}{2} \sum_{ij} A_{ij} \tag{6.4}$$

is the number of edges in the network, and the δ function $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise. c_i is the label of community to which node i is assigned in this division. k_i denotes the degree of the i-th node. The intuition of modularity Q is to measure the difference between edge density inside communities given a community division in the network and the same quantity for a network with the same community division but randomly distributed edges.

The community detection problem through modularity maximisation is formulated as following.

Problem 6.2 (Community Detection Problem)**.** *Given a network $G = \{V, E\}$ where V denotes for the set of nodes and E for the set of edges, divide the set of nodes V into m mutually exclusive subsets $\{V_1, V2, ..., V_m\}$, $V_i \in V, V_i \neq \emptyset$ for $i = 1, 2, ..., m$, and $\cup_{i=1}^{m} V_i = V, V_i \cap V_j = \emptyset$ for $i \neq j$, so that the value of modulairty Q is maximised.*

In the previous chapter, we have already shown how to solve problem 6.1 through a binary vector encoding scheme constrained by algebraic connectivity in a multi-objective evolutionary algorithm. In order to include problem 6.2 in an environment of evolutionary multitasking, we propose a new unified genetic representation as a general solver for both problems. For the purpose of simplicity, in the following content, especially in algorithm description, task 1 refers to active module identification task and task 2 community detection task.

6.3.3 A Unified Genetic Representation for Multiple Tasks and Problem-Specific Decoding Scheme

For a network $G = V, E$ of size $n = |V|$, an individual solution is encoded as an integer vector of length n, each integer representing the label of community to which corresponding node is assigned, i.e. for the i-th individual L_i in population, we have

$$L_i = \{l_i^1, l_i^2, ..., l_i^n\} \tag{6.5}$$

where $l_i^j \in \{0, 1, ..., n-1\}$ for $j = 1, 2, ..., n$, meaning the available label of communities ranges from 0 to $n - 1$.

In this algorithm algebraic connectivity is no longer used as a constraint to ensure the connectivity of detected active module. Instead the collection of positions assigned with positive integers is interpreted as a subgraph whose connected component S with highest $\sum_{v \in V_S} S^{FDR}(v)$ is identified as the active module this individual represents. This connected components finding based decoding scheme is inspired by the work of Li et al. [45]. Details of the chromosome decoding scheme for active module identification task is described in Algorithm 6.2.

In the context of community detection, individual L_i is interpreted in a different way. Integer in the j-th position of L_i denotes for the label of community to which the j-th node is assigned. During the whole evolutionary algorithm, connectivity for each community is not explicitly required in algorithm implementation, however the process of modularity

Algorithm 6.2: Chromosome Decoding Scheme for Task 1

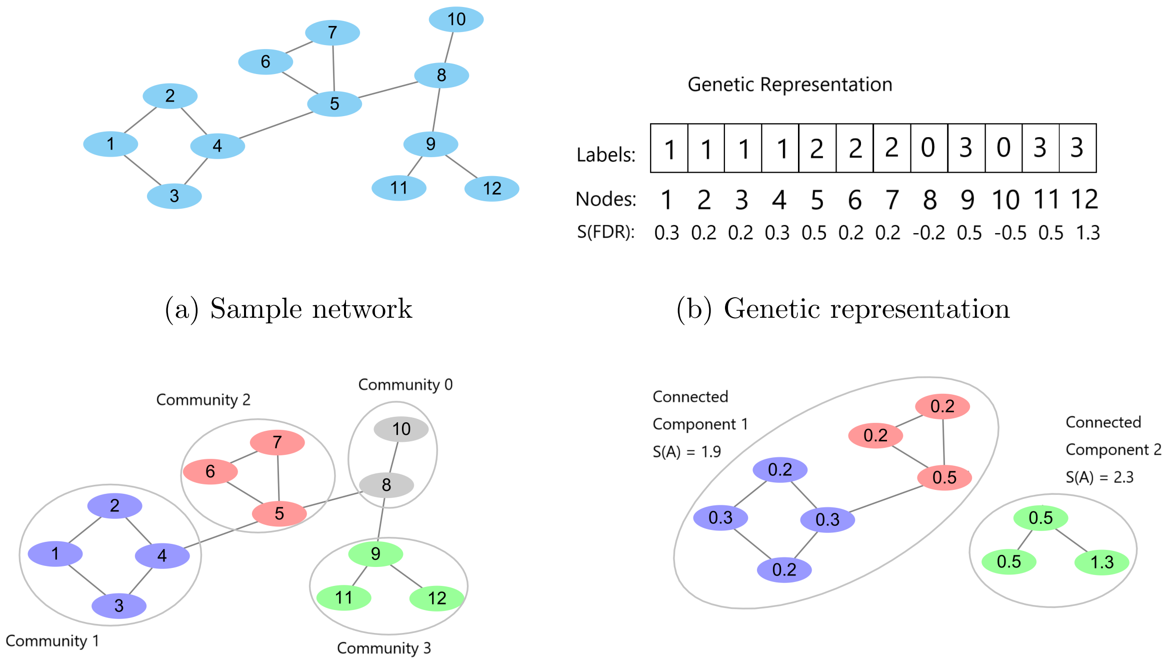
```
Input: Individual $L_i$, adjacency matrix A of the whole network, a list of $S^{FDR}(v)$
       assigned to each node
Output: Connected node set of active module $V_S$, active module score
        $\sum_{v \in V_S} S^{FDR}(v)$
binary vector $l \leftarrow L_i > 0$ ;
subgraph $A_l \leftarrow A(l, l)$ ;
get all the $k$ connected components $\{V_1, V2, ..., V_k\}$ in $A_l$ ;
$S_{max} \leftarrow$ negative infinity ;
$V_S \leftarrow \emptyset$ ;
for $j \leftarrow 1$ to $k$ do
    $S_j \leftarrow \sum_{v \in V_j} S^{FDR}(v)$ ;
    // get the active module score $S_j$ of the $j$-th connected components
        $V_j$
    if $S_j > S_{max}$ then
        $S_{max} = S_j$ ;
        $V_S \leftarrow V_j$ ;
    end
end
return $V_S$, $S_{max}$
```

maximisation implicitly drives the network division towards densely connected solutions. A detailed chromosome decoding scheme for community detection task is described in Algorithm 6.3.

Figure 6.1 gives a simple example of how to decode given chromosome representation for two tasks. In a network with 12 nodes and 13 edges (Figure 6.1a), the chromosome is encoded as $[1, 1, 1, 1, 2, 2, 2, 0, 3, 0, 3, 3]$ (Figure 6.1b). Visualisation of decoding scheme under two tasks can be seen in Figure 6.1c and 6.1d.

6.3.4 Task-Specific Mutation Operator

In order to improve the performance of the algorithm and provide better guidance in searching the solution space, we have mutation and local search operators specially designed for the two different tasks. Upon taking in an individual, the mutation operator first checks its skill factor to decide the task in which this individual is more effective, then it applies different mutation strategy accordingly.



(a) Sample network

(b) Genetic representation

(c) Decoding for network division

(d) Decoding for active module identification

Figure 6.1: A simple example of the chromosome encoding and decoding scheme for two tasks. Figure 6.1a: Visualisation of the sample network with 12 nodes and 13 edges. Figure 6.1b: The genetic representation of one individual, an integer vector of length 10, each integer representing the community label of corresponding node. The vector $S(FDR)$ gives the active node score. Figure 6.1c: Network is divided into 4 communities labelled from 0 to 3 according to the individual. Figure 6.1d: Subgraph formed by all nodes with non-zero labels. Nodes are labelled by active score $S(FDR)$. Module score $S(A)$ is calculated for each connected component in the subgraph. In this example, connected component 2 with a higher active module score $S(A) = 2.3$ is selected as the decoded active module.

Algorithm 6.3: Chromosome Decoding Scheme For task 2

```
Input: Individual $L_i$, adjacency matrix A of the whole network
Output: Network Division $\{V_1, V2, ..., V_m\}$, Modularity Q
// get labels of communities
labels $\leftarrow$ unique elements of $L_i$ ;
$m \leftarrow$ length(labels) ;
for $j \leftarrow 1$ to $m$ do
    label $\leftarrow labels(j)$ ;
    $V_j \leftarrow \emptyset$ ;
    for $k \leftarrow 1$ to $n$ do
        // get all the nodes in $j$-th community
        if $L_i^k == label$ then
            $V_j = V_j \cup v_k$
        end
    end
end
// Now network division represented by $L_i$ is decoded
calculate modularity Q for given network division $\{V_1, V2, ..., V_m\}$ ;
return $\{V_1, V2, ..., V_m\}, Q$
```

Individual specialised in active module identification goes through a subgraph expanding stage and a node deletion stage. In the first stage, neighbouring nodes with positive weight are added to the subgraph while those with negative weight also have probabilities to be included. In the second stage, negative weighted nodes in subgraph go through a similar probabilistic deletion process. A detailed description is shown in Algorithm 6.4.

Individual specialised in network division is applied with a completely different mutation strategy called random community merging. This mutation is an imitation of bottom-up merging strategy in quite a few community detection algorithms. In initialisation stage, every individual is assigned with a random permutation of integers 0 to $n - 1$, meaning that every node is the sole member of one of n communities. When such mutation strategy is applied to an individual, two connected communities are randomly selected to be joined together to form a new larger community. As evolution goes on, small communities are gradually merged into large communities, accompanied with a significant increase in modularity Q. In the late stage of evolution the value of Q becomes stable, indicating the algorithm has reached the optima in modularity maximisation task.

Algorithm 6.4: Apply mutation with local search steps to chromosomes specialised in task 1 (skill factor $\tau == 1$)

```
Input: Individual $L_i$, adjacency matrix A of the whole network, a list of $S^{FDR}(v)$
       assigned to each node
Output: Individual $L_i$ after mutation
// individual $L_i$ is more effective in task 1
node set of subgraph S is given by $V_S \leftarrow \{V_j | L_i^j > 0\}, j = 1, 2, ..., n$ ;
$V_{neighbours} \leftarrow$ all neighbouring nodes of $V_S$;
// get labels of communities
labels $\leftarrow$ unique elements of $L_i$ ;
// Stage 1: probabilistic subgraph expanding
for every node $v_j$ in $V_{neighbours}$ do
    if $S^{FDR}(v_j) \geq 0$ then
        // if the neighbour $v_j$ is assigned with positive $S^{FDR}(v_j)$,
           include it
        $L_i^j \leftarrow$ randomly select one label from labels (cannot be 0)
    else
        // if the neighbour $v_j$ is assigned with negative $S^{FDR}(v_j)$,
           include it with probability $exp(S^{FDR}(v_j))$
        if $exp(S^{FDR}(v_j)) > random()$ then
            $L_i^j \leftarrow$ randomly select one label from labels (cannot be 0)
        end
    end
end
// Stage 2: probabilistic negative weighted node deletion
update node set of subgraph S by $V_S \leftarrow \{V_j | L_i^j > 0\}, j = 1, 2, ..., n$ ;
for every node $v_j$ in $V_S$ do
    // if the node $v_j$ is assigned with negative $S^{FDR}(v_j)$, delete it
       with probability $1 - exp(S^{FDR}(v_j))$
    if $exp(S^{FDR}(v_j)) < random()$ then
        $L_i^j \leftarrow 0$
    end
end
return $L_i$
```

A detailed description of community merging mutation is shown in Algorithm 6.5.

Algorithm 6.5: Apply mutation with random community merging to chromosomes specialised in task 2 (skill factor $\tau == 2$)

```
Input: Individual $L_i$, adjacency matrix A of the whole network
Output: Individual $L_i$ after mutation
// individual $L_i$ is more effective in task 2
randomly select one community label $c_1$ in $L_i$ ;
node set of community $c_1$ is given by $V_{c1} \leftarrow \{v_k | L_i^k == c_1\}, k = 1, 2, ..., n$ ;
$V_{neighbours} \leftarrow$ all neighbouring nodes of $V_{c1}$;
if $V_{neighbours}$ contains community label different from $c_1$ then
    randomly select another community label $c_2$ in $V_{neighbours}$;
    // merge all nodes in community $c_1$ into community $c_2$
    $L_i(L_i == c_1) \leftarrow c_2$
end
return $L_i$
```

6.3.5 Uniform Crossover Operator

This algorithm uses uniform crossover to generate two child individuals from two parent individuals. Although uniform crossover has a higher probability to destroy community structures that are already detected than simple one-point or two-points crossover, in practise it is proven to be an effective way to preserve the diversity of population, help explore solution space, and avoid being stuck in local optima. Uniform crossover is also very simple to implement. Pseudocode is shown in Algorithm 6.6 below.

Algorithm 6.6: Uniform Crossover to generate two child individuals

```
Input: Parent individuals $L_1$, $L_2$
Output: Offspring individuals $Child_1$, $Child_1$
// randomly generate a binary mask vector with the same length of
   individuals
mask $\leftarrow RandomBinary(1, length(L_1))$ ;
// uniform crossover between two parent individuals
$Child_1 \leftarrow L_1$ ;
$Child_1(mask == 1) \leftarrow L_2(mask == 1)$ ;
$Child_2 \leftarrow L_2$ ;
$Child_2(mask == 1) \leftarrow L_1(mask == 1)$ ;
return $Child_1$, $Child_1$
```

6.3.6 Improvement of Output Solution

The multifactorial evolutionary algorithm scheme we used for solving Problems 6.1 and 6.2 is already able to provide satisfactory results in terms of objective evaluation. However, because the connectivity of communities is not explicitly required in the design of genetic representation, interpretation or algorithm implementation, the output solutions directly generated from the evolutionary algorithm sometimes still contain communities that are not connected. In a typical solution that fails to ensure connectivity there is one sole node separated from other community members. To solve this issue we designed an extra solution improvement step containing two stages. In the first stage, community with more than one connected components is assigned with new community labels for each of the extra components. This often results in communities with one sole node. Then in the second stage, this one node community is merged to its most frequent neighbouring community. Details of solution improvement is shown in Algorithm 6.7.

6.4 Experimental Studies

In this section, we give a few applications of our algorithm to real world networks. The first several experiments solely evaluate the algorithm's performance on community detection task because these networks have no features that can be formulated as an active module problem. Then we will apply the algorithm to biological network which allows for tackling community detection and active module identification simultaneously.

6.4.1 Modularity Optimisation Task

We first test the performance of our algorithm on modularity optimisation task to check whether the design of genetic representation and mutation scheme is suitable for this task. Some classic networks for community structure analysis are chosen as benchmark networks to compare the value of modulairty Q from this algorithm and other classic

Algorithm 6.7: Solution Improvement

```
Input: Individual $L_i$, adjacency matrix A of the whole network
Output: Individual $L_i$ after improvement
// Stage 1: assign disconnected community with different labels for
   each connected component
labels ← unique elements of $L_i$ ;
// new label starts from $n + 1$ to avoid overlap with all original
   labels
$newLabel \leftarrow n + 1$ ;
for $j \leftarrow 1$ to $length(labels)$ do
    get subgraph $A_j$ of the $j$-th community $labels(j)$ ;
    get all the $k$ connected components $\{V_1, V2, ..., V_k\}$ in $A_j$ ;
    if $k > 1$ then
        for $ii \leftarrow 2$ to $k$ do
            $L_i(V_k) \leftarrow newLabel$ ;
            $newLabel$ ++ ;
        end
    end
end
// Stage 2: merge one-node community to neighbouring community
labels ← unique elements of $L_i$ ;
for $j \leftarrow 1$ to $length(labels)$ do
    if $j$-th community has only one node $v_j$ then
        find community labels of all neighbouring nodes ;
        assign $v_j$ with the most frequent neighbouring community label ;
    end
end
return $L_i$
```

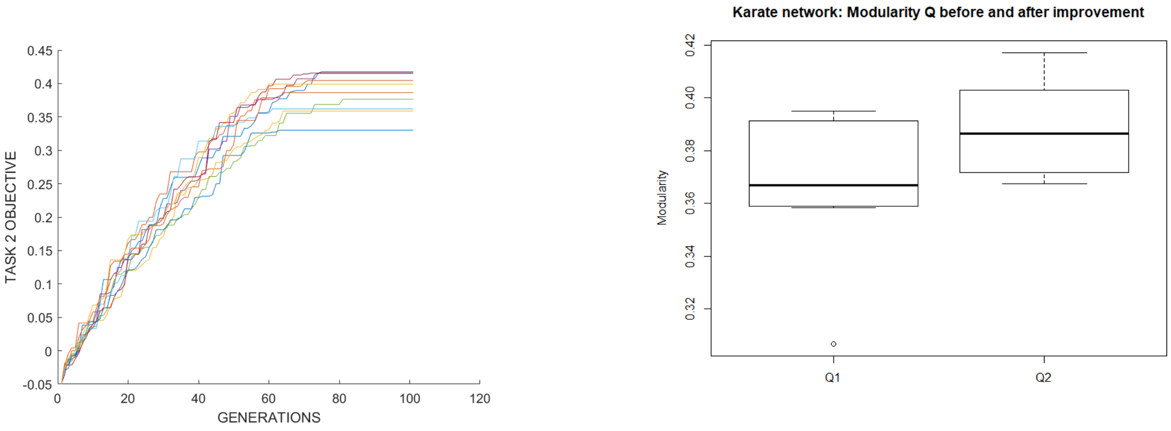
modularity optimisation algorithms. For those networks node weights are assgined with random values. Then the yeast molecular interaction network that has been tested in the previous chapter for active module identification is used for optimising both tasks. In this section the results for topological structure detection is analysed.

6.4.1.1 Experimental Results on Classic Community Detection Networks

The first experimental network for community detection comes from one of the classic studies in social network analysis. It is often named as Zachary's karate club network as it was first described by Wayne Zachary in the 1970s as a friendship network between 34 members of a karate club in a university [90]. Edges between members were based on their social interactions both inside and outside of the club. During his research period, a dispute accidentally appeared between the the club's administrator and main karate teacher, resulting in a split of the original club into two smaller clubs. Figure 6.3a shows the karate network and the two groups it was divided. This network has been widely used in community detection and modularity optimisation research as benchmark.

Figure 6.2a shows how the value of modularity Q changes after each generation. The algorithm is performed repeatedly for 10 times. It clearly shows the optimisation process during which the modularity gradually increases and becomes stable, although the final objective values slightly differ from each other due to the randomness of this algorithm. Figure 6.2b shows the value Q of the same group of solutions before and after solution improvement, proving that Algorithm 6.7 is able to improve the final solution.

From the 10 running results, solution with the highest post-improvement modularity is selected to visualise, shown in Figure 6.3b. Other than the two-group division in real world, a total of 4 communities are detected by proposed algorithm, with modularity Q equal to 0.4172. Notably this community structure is actually a further split based on the original division, with node 10 as an mistakenly classified exception. However, from the purely computational objective fitness view, modularity Q of the original network division is only 0.3715, a value lower than most Q values from the 10 runs of proposed algorithm.



(a) Maximisation process of modularity Q in 10 runs.

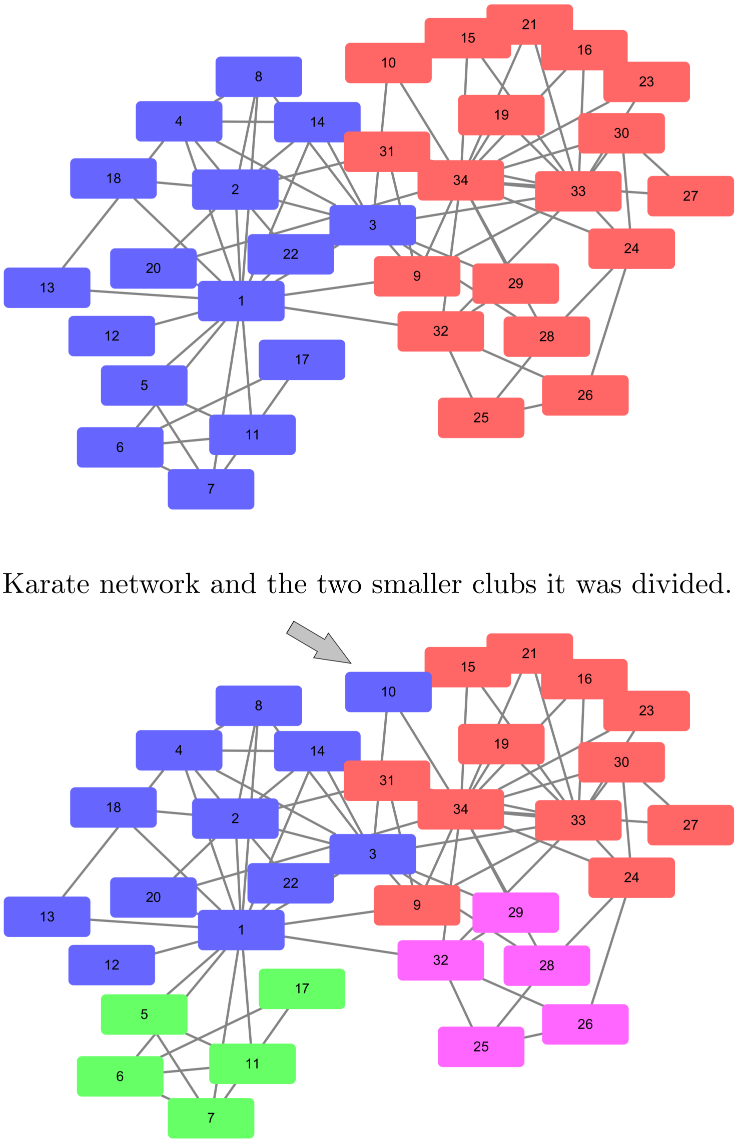
(b) Modularity Q before and after solution improvement.

Figure 6.2: Modularity Q of Zachary's karate club network. Figure 6.2a: 10 running records for value of modularity Q in karate network. Algorithm parameters are 100 populations and 100 generations. Figure 6.2b: Q1 group is the results directly generated from the 10 runs of multifactorial evolution, Q2 group is modularities of the same set of solutions after improvement.

As a comparison, modularities for the network divisions found by edge removal algorithm (Giran and Newman [24]) on this network is 0.401, by merging strategy based algorithm (Clauset et al. [12]) is 0.381, by extremal optimisation (Duch et al. [16]) is 0.419, and by Newman's spectral algorithm [63] is 0.419. Results have shown that the performance of proposed algorithm on karate network is comparable to classic modularity optimisation algorithms.

The second experimental network is the social network of 62 bottlenose dolphins living in Doubtful Sound, constructed by Lusseau [49, 50] for behavioral ecology and sociobiology research. Edges between two dolphins are constructed from observation of frequent association. Similarly, proposed algorithm is performed repeatedly for 10 times on this network. Maximisation process of modularity Q in the 10 runs and after solution improvement is shown in Figure 6.4.

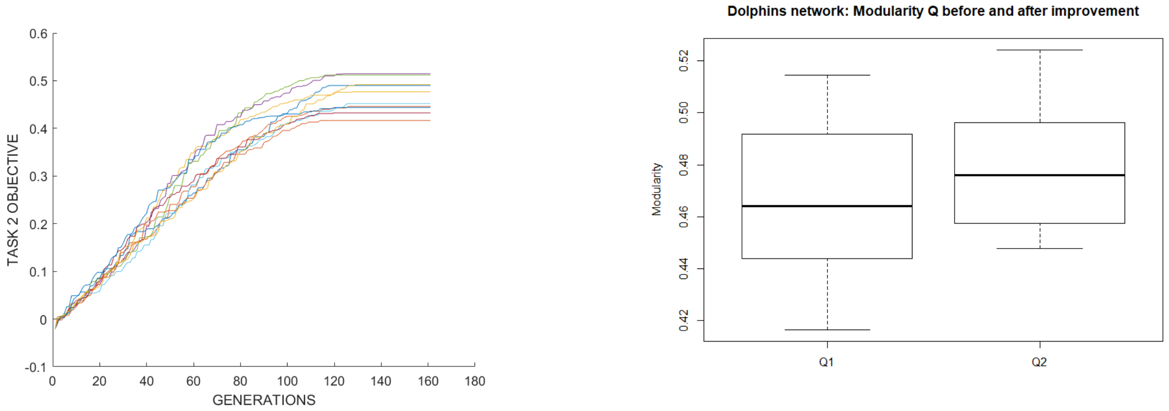
In the 10 running results, the solution with highest Q after solution improvement is not the one ranking highest before that. To better illustrate the effect of improvement step, we pick this solution and visualise its network division before and after improvement



(a) Karate network and the two smaller clubs it was divided.

(b) Community structure in karate network detected by proposed algorithm.

Figure 6.3: Visualisation of communities detected by proposed algorithm in Zachary's karate club network. Figure 6.3a: The social network between 34 individuals in the karate club studied by Zachary. The two different node colours represent two smaller clubs it was divided into. Figure 6.3b: Visualisation of the 4 communities detected by proposed algorithm. Communities are distinguished by different node colours. Node 10 pointed by grey arrow is the only node that is wrongly classified according to real world conditions.



(a) Maximisation process of modularity Q in 10 runs.

(b) Modularity Q before and after solution improvement.

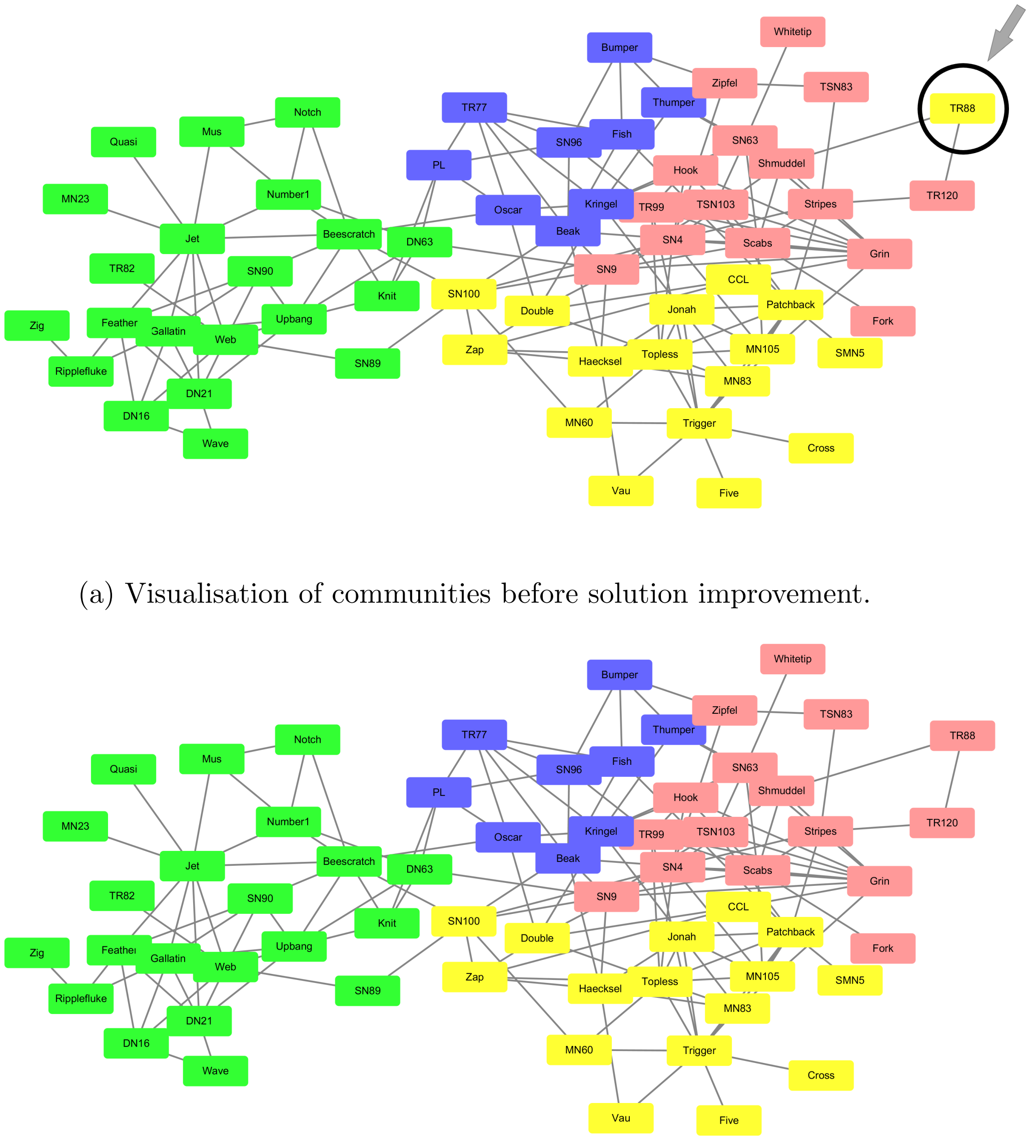
Figure 6.4: Modularity Q of bottlenose dolphins network. Figure 6.4a: 10 running records for value of modularity Q in dolphins network. Algorithm parameters are 200 populations and 160 generations. Figure 6.4b: Q1 group is the results directly generated from the 10 runs of multifactorial evolution, Q2 group is modularities of the same set of solutions after improvement.

in Figure 6.5. The value of modularity Q is 0.5118 before improvement, and is increased to 0.5242 after that. Modularity for the split on this network given by Newman and Giran [65] is also 0.52.

6.4.1.2 Experimental Results on Yeast Molecular Interaction Network

The yeast molecular interaction network once used by Ideker [34] is again used as an experimental network. It has 330 nodes assigned with p-values which are converted to the additive score $S^{FDR}(x)$ as a module activity measurement. Figure 6.6a shows the maximisation process of both active module score (denoted as task 1 objective) and modularity Q (task 2 objective) in 800 generations for 10 repeated runs. Effect on modularity Q by applying solution improvement to the same group of 10 solutions is shown in Figure 6.6b. The overall improvement for Q in this network is more significant largely due to the many small connected components.

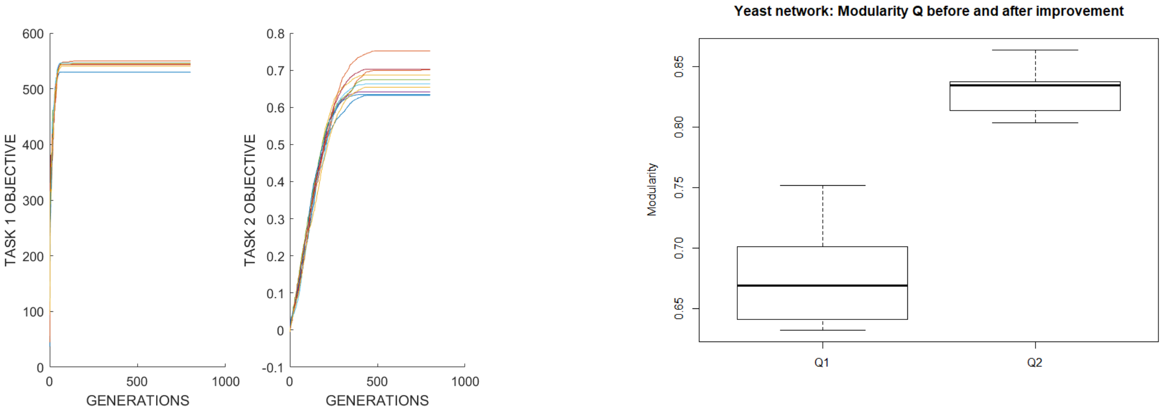
Figure 6.7 is the visualisation of network division with the highest modularity Q 0.8636 in the 10 runs of proposed algorithm on yeast network. In order to avoid having too many confusing colours in the network, node colours representing communities are manually se-



(a) Visualisation of communities before solution improvement.

(b) Visualisation of communities after solution improvement.

Figure 6.5: Visualisation of communities detected by proposed algorithm in dolphins network. Communities are distinguished by different node colours. Figure 6.5a: Visualisation of 4 communities detected directly from multitasking evolution process without solution improvement. Node TR88 pointed by grey arrow is separated from other members in the same community coloured in yellow, resulting in a disconnected community. Figure 6.5b: After solution improvement step, node TR88 is classified into its neighbouring community coloured in red, resulting in an increase of Q from 0.5118 to 0.5242.



(a) Maximisation process of active module score S_A and modularity Q in 10 runs.

(b) Modularity Q before and after solution improvement.

Figure 6.6: Modularity Q of yeast network. Figure 6.6a: 10 running records for value of active module score and modularity Q in yeast network. Algorithm parameters are 300 populations and 800 generations. Active module score (shown in the left) gets to its optima much faster than modularity Q (in the right), in about 100 generations, while Q gets stable after about 500 generations. Figure 6.6b: Q1 group is the results directly generated from the 10 runs of multifactorial evolution, Q2 group is modularities of the same set of solutions after improvement.

lected based on the rule that neighbouring communities that have at least one linking edge cannot be labelled with the same colour. Eventually a network division with 45 communities is carefully labelled by 5 colours. All of the 25 small connected components listed in the bottom of the figure are successfully classified as isolated communities. In the largest connected component, densely connected local areas are nicely classified into different communities. There are, however, three small communities that don't seem to be reasonably classified, shown in the figure by circles and arrows. Modularity Q can be further increased when those three communities are merged with their neighbouring communities. Nevertheless, this experiment on real-world biological network with a number of connected components again shows that the proposed algorithm has a satisfactory performance in identifying community structures.

To make a straightforward comparison between the performance of proposed algorithm and some other published algorithms, we use the clustering functions implemented in R igraph package to get the modularities values for network division found by different algorithms. The results are shown in Table 6.1. This comparison shows that the proposed

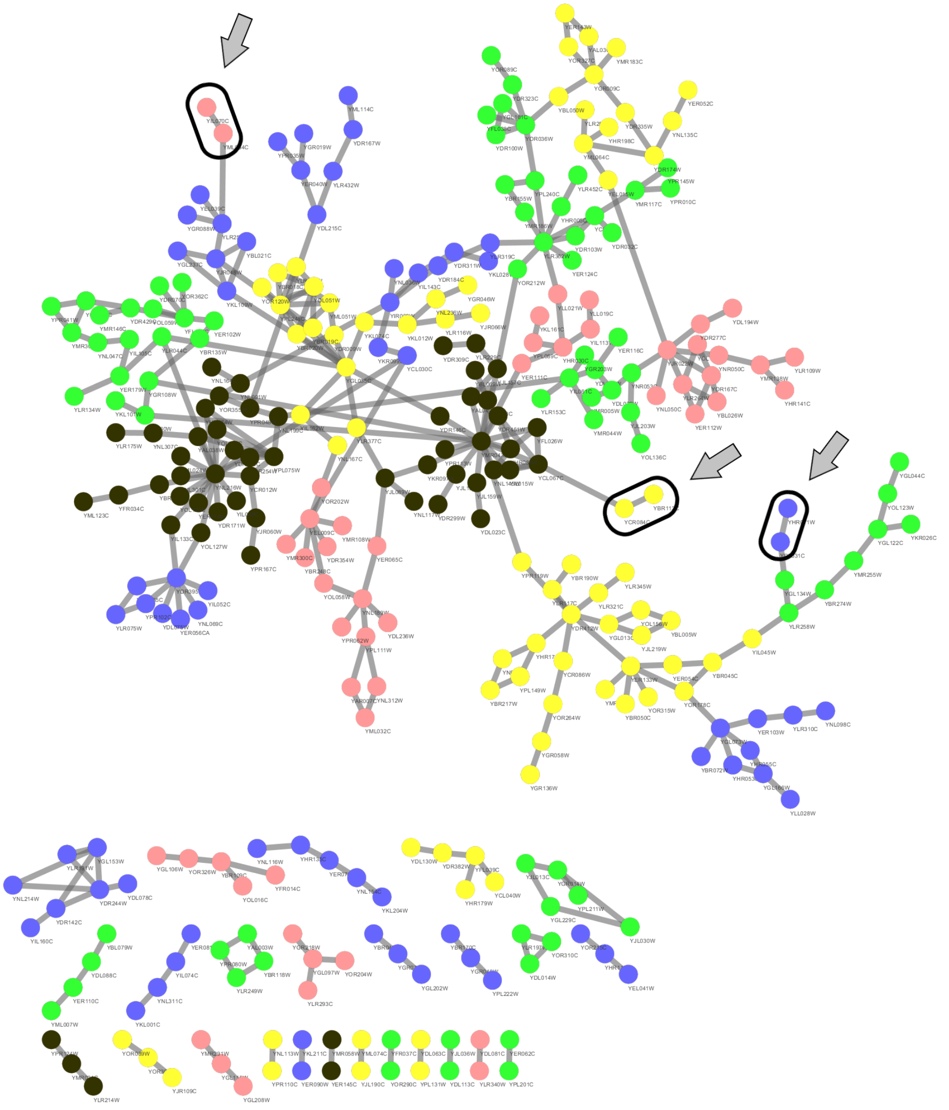


Figure 6.7: Visualisation of communities detected by proposed algorithm on yeast network. Nodes in neighbouring communities are filled with different colours. Modularity Q can be increased from 0.8636 to 0.8708 if the three tiny communities pointed by arrows are merged to their neighbouring communities.

algorithm is able to find network divisions with modularites comparable to the classic modularity optimisation algorithms.

Table 6.1: Comparison of modularities for the network division found by the proposed algorithm and some published algorithms. All the results for published algorithms are from running corresponding clustering functions implemented in igraph package in R. References and the function names of these algorithms are: GN [65], cluster_edge_betweenness; MNC [12], cluster_fast_greedy; Louvain [10], cluster_louvain; spectral [63], cluster_leading_eigen. Proposed algorithm is repeated for 10 times on each network, giving minimum, maximum and average modularity in the table. Clustering functions in R give stable results for repeated runs, thus only one single result is listed for each network.

network	size	GN	MNC	Louvain	spectral	proposed algorithm (10 runs)		
						min	max	average
Karate	34	0.4013	0.3807	0.4188	0.3934	0.3674	0.4172	0.3875
Dolphins	62	0.5194	0.4955	0.5158	0.4912	0.4478	0.5242	0.4796
Politics Books	105	0.9977	0.9977	0.9977	0.9977	0.9977	0.9977	0.9977
Football	613	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984	0.9984
Yeast	330	0.8784	0.8825	0.8827	0.8379	0.8308	0.8708	0.8502

6.4.2 Active Module Identification Task

This section shows the results of proposed algorithm on active module identification. The aim of our study is to demonstrate that our algorithm can not only identify high quality active modules but also provides more informative biological interpretation by combining the active modules with communities.

The maximisation process of active module score on yeast network in 10 runs is already displayed in Figure 6.6a. Active scores range from 529.8 to 549.9, all of those are significantly higher than the same scores for active modules identified by jActiveModule method (see Table 5.2 for details) of which the highest one is only 270. Figure 6.8 visualises the active module with the highest active module score among the 10 runs. Note that all the coloured nodes form one connected active module, with different colours indicating the labels of structural communities the nodes belonging. The network division is in accordance with the one shown in Figure 6.7. Labels of communities are also shown in the figure. By mapping nodes in active module to different communities and dividing it to smaller fractions we will demonstrate that the structural information helps get more accurate and specific biological interpretation for the identified module.

Gene ontology analysis through the online tool of Gene Ontology Consortium [23] for biological process is performed on the whole active module, shown in Table 6.2. As gene ontology (GO) terms are given in a hierarchical structure, for simplicity only the top level of GO terms are selected to display. Similar to the gene ontology results of modules detected by jActiveModule 5.2, our algorithm is also able to identify active modules relevant to the yeast galactose utilisation activities in the experiments, supported by GO terms like galactose metabolic process, ATP and ADP metabolic process, and pyruvate metabolic process. Other related activitiees such as catabolic process and response to stimulus are also found in the terms.

However, because the whole modules from both algorithms consist large number of genes, the GO annotation terms are too general, which cannot provide specific interpretation for the active modules. Because of the complexity of biological activities, the same group of metabolic components may play roles in a series of metabolic processes and have different annotation terms, which often causes ambiguous explanations for the active module. Next we will show one way to fix this problem through the utilisation of structrual property of network.

Table 6.2: Gene ontology results of the whole active module identified by proposed algorithm in yeast network. This module has 93 nodes and active module score equal to 549.9. p-value gives the statistical significance of corresponding GO term's enrichment in the gene set.

Typical GO terms	p-value
galactose metabolic process	3.64×10^{-05}
ADP metabolic process	5.62×10^{-03}
regulation of generation of precursor metabolites and energy	4.15×10^{-02}
pyruvate metabolic process	2.73×10^{-02}
ATP metabolic process	3.78×10^{-03}
carbohydrate catabolic process	1.48×10^{-02}
small molecule catabolic process	3.29×10^{-02}
energy derivation by oxidation of organic compounds	4.41×10^{-03}
cellular carbohydrate metabolic process	4.26×10^{-02}
response to abiotic stimulus	2.78×10^{-02}

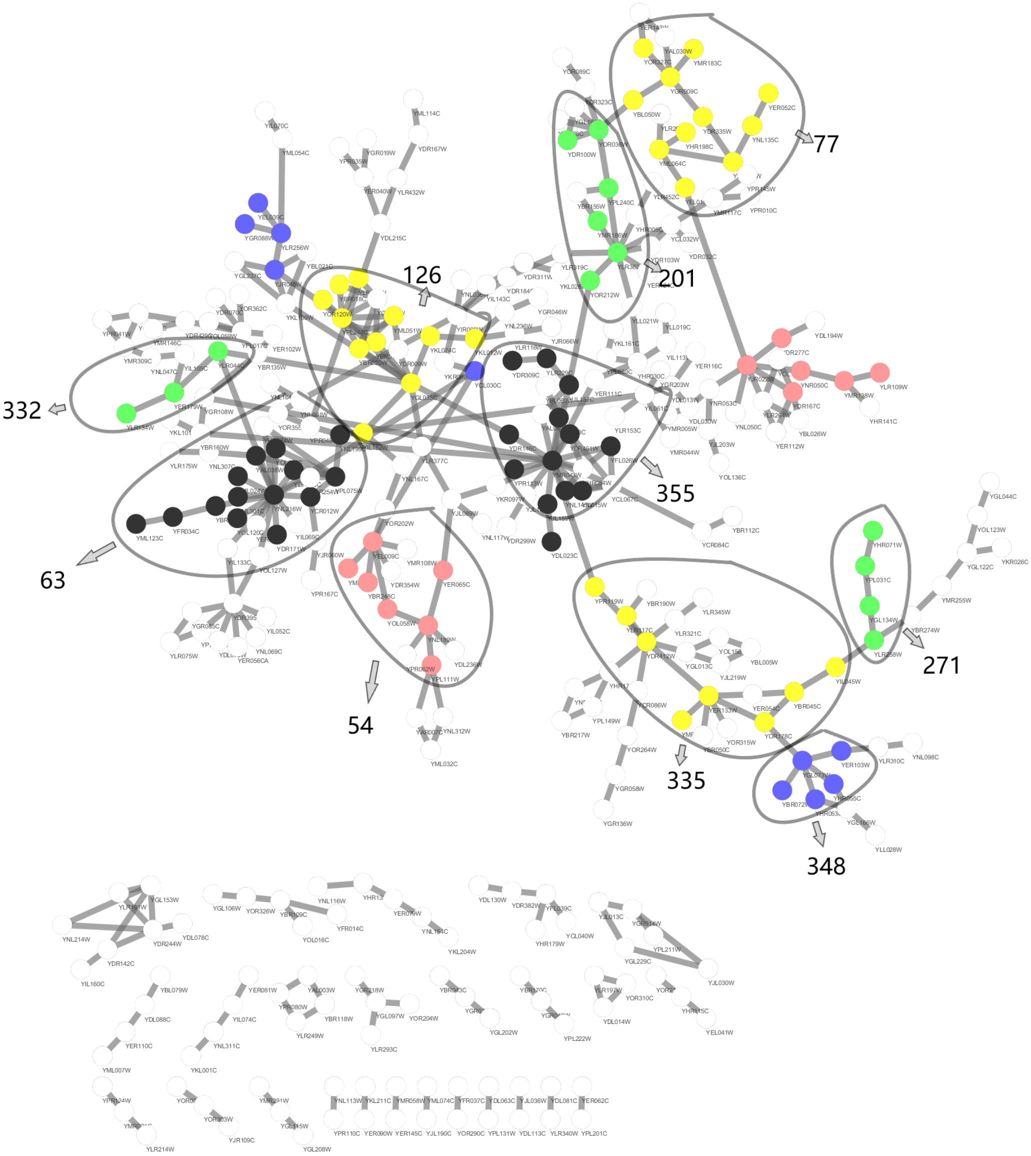


Figure 6.8: Visualisation of active module detected by proposed algorithm on yeast network. This one active module is shown by all coloured nodes, different colours indicating different structural communities. Nodes in active module that are classified in the same community are circled by curve and pointed with a grey arrow showing the label of community.

The main advantage of our algorithm is it can combine active modules and communities to divide the large active module into smaller fractions based on the community structure detected using proposed algorithm. Instead of performing GO annotation on the whole set of genes in the active module, we perform annotation on these fractions separately. Of all the 13 fractions, 3 of them have no significant functional enrichment, other 10 have annotation terms shown in Table 6.3. From this table it is easy to find that the functional annotation becomes more specific and clear. Every fraction has only one top level GO term or a small set of closely related terms. For example, the fraction 63 is specialised in glycolytic process, fraction 126 is targeted in galactose catabolic process via UDP-galactose, and fraction 54 in glutamine family amino acid metabolic process, all of those are highly relevant to galactose metabolic process. Other several fractions might not be directly related to the process, but serve as an assistance or as essential cellular activities, such as vesicle fusion from fraction 77, response to heat from fraction 348, and regulation of reproductive process from fraction 355. As a contrast, GO terms for whole active module shown in Table 6.2 contain a variety of metabolic process such as galactose, pyruvate, ATP, ADP, and carbohydrate catabolic process, yet could not further distinguish between these high level functions.

Our results demonstrated that, by combining the structure information, i.e., mapping nodes in active module to different communities and dividing it to smaller fractions, we can obtain more accurate and specific biological interpretation.

To investigate whether structure information, i.e., communities alone can provide the same accurate interpretation, we performed GO analysis on each of the 42 communities. Of all the 42 communities, 20 have no significant annotation. We selected three representative communities with significant annotations in Table 6.4. The results show that the annotations for each community are too general or ambiguous due to many mixed function terms. As a comparison, the active module fractions from our algorithm with the same labels have only one annotated function each as shown in Table 6.3. It is clear that communities cannot reflect the biological activity the system is going through accu-

Table 6.3: Gene ontology results for fractions in active module divided by community structure in yeast network. It uses the same label set as shown in Figure 6.8. Size gives the number of nodes in each fraction. Only top level GO terms are selected and displayed in the table.

label	size	Typical GO terms	p-value
54	7	glutamine family amino acid metabolic process	5.95×10^{-04}
63	14	glycolytic process	2.20×10^{-02}
77	11	vesicle fusion	$5.93{\times}10^{-04}$
126	11	galactose catabolic process via UDP-galactose	1.11×10^{-04}
201	6	box C/D snoRNP assembly	4.77×10^{-02}
271	4	negative regulation of macroautophagy	4.74×10^{-03}
		negative regulation of glycogen biosynthetic process	4.74×10^{-03}
		negative regulation of sequence-specific DNA binding transcription factor activity	7.47×10^{-03}
332	3	romatic amino acid family catabolic process to alcohol via Ehrlich pathway	3.74×10^{-03}
		L-phenylalanine catabolic process	5.38×10^{-03}
		glycolytic fermentation to ethanol	5.38×10^{-03}
		tryptophan catabolic process	1.49×10^{-02}
		branched-chain amino acid catabolic process	1.81×10^{-02}
335	8	regulation of protein dephosphorylation	5.67×10^{-04}
		glycogen metabolic process	6.91×10^{-03}
		regulation of mitotic sister chromatid segregation	4.68×10^{-02}
348	5	response to heat	2.88×10^{-03}
355	13	regulation of reproductive process	4.21×10^{-03}

rately. The main reason is that communities fail to incorporate activities, e.g., differential expression information to reveal the essential function changes of the system.

Table 6.4: Representative gene ontology results for communities in the yeast network. It also uses the label set generated directly form original results. Size gives the number of nodes in each fraction. As a contrast, the three fractions labelled as 54, 63 and 77 contain only one precisely described ontology term in Table 6.3. GO terms for all communities are shown in supplementary table.

label	size	Typical GO terms	p-value
54	15	urea cycle	3.14×10^{-02}
		heteroduplex formation	4.61×10^{-02}
		telomere maintenance via recombination	1.58×10^{-02}
		glutamine family amino acid metabolic process	6.34×10^{-03}
		alpha-amino acid biosynthetic process	2.99×10^{-03}
		aromatic compound biosynthetic process	1.35×10^{-02}
		heterocycle biosynthetic process	1.26×10^{-02}
		organic cyclic compound biosynthetic process	1.56×10^{-02}
63	28	cellular response to phosphate starvation	1.01×10^{-02}
		egulation of glycolytic process by positive regulation of transcription from RNA polymerase II promoter	1.72×10^{-02}
		gluconeogenesis	1.62×10^{-04}
		glycolytic process	1.95×10^{-05}
		cytoplasmic translation	2.23×10^{-04}
77	14	urea cycle	1.99×10^{-02}
		’de novo’ pyrimidine nucleobase biosynthetic process	1.31×10^{-02}
		arginine biosynthetic process	1.39×10^{-02}

6.5 Summary

A multifactorial evolution algorithm framework for detecting active module and topological communities simultaneously has been proposed in this chapter. We have introduced the motivation of inducing topological structure information in active module identification, the algorithm designs and improvements developed specific for the tasks, including a unified genetic representation and task-specific decoding scheme, mutation operator with local search or community merging operations targeting individuals specialised in different tasks, and an extra solution improvement step.

Experimental results have shown that the proposed algorithm is able to achieve high objective values for both tasks. Functional annotation further shows that mapping community to the identified active module produces smaller fractions that have more precise biological interpretations.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This chapter concludes the thesis and discusses some future work related to the work presented in this thesis.

7.1 Conclusions

This thesis is dedicated to active module identification in biological networks. To be more specific, it aims to address three key research questions introduced in Chapter 1: How to build a practical formulation of active module identification problem that faithfully reflects the dynamic changes of cellular activities and helps reveal new insights? How to design effective, efficient and robust algorithms to identify active module? What is the right way to interpret identified active module? The thesis proposes three novel algorithm frameworks to answer the research questions from three different aspects. A brief review of the thesis content is given as following.

Chapter 2 gives an introduction on research background of active module and reviews representative algorithms for active module identification, mainly focusing on the problem definition and algorithm design. The formulation of the problem is highly flexible and subject to change in specified context or for special research interest.

Chapter 3 explains the motivations of the novel research work to be proposed and figures out several general issues to be addressed when designing algorithms. We explain the intuition of introducing prior knowledge to reveal new insights and why it is a conflicting objective with pure data driven active module identification. We introduce community detection which often reveal the topological property and member relationships of the network, and why multifactorial evolution is used in the context. We briefly describe the general issues of scoring function, connectivity and size control in designing active module identification algorithms.

Chapter 4 proposes an integrated approach for active module identification through balancing between differential expression of each gene and the differential correlation between genes. In order to answer the research questions, we formulate differential expression as node score, differential correlation as edge score, and combine them using a multi-objective approach.

Chapter 5 proposes a novel prior information guided active module identification approach. In order to answer the research questions, we build a formulation of prior knowledge enriched active module that is able to reflect the activity of cellular process and reveal intermediate genes that serve as bridges for cross-talk between neighbouring functional areas. Due to the conflicting of two objectives, we formulate the identification of target module as a multi-objective optimisation problem and solve it effectively using modified algorithm based on NSGA-II. We select modules with different trade-offs between two objectives on the Pareto front to perform functional annotation and show that the algorithm is able to identify biologically meaningful modules in both small and large scale networks.

Chapter 6 proposes a novel algorithm framework of detecting active module and topological communities simultaneously. In order to answer the research questions, we formulate the problem as an evolutionary multitasking problem and develop a series of task-specific algorithm designs and improvements for the problem. We present a new way to better interpreter the biological meaning of active module by mapping the community structure to the active module and further dividing the module into smaller fractions.

7.2 Future Work

This final section briefly discusses several aspects of potential future work based on the research presented by this thesis.

- **Modification on scoring function.**

  Although the proposed algorithms are applied on molecular interaction networks or integrated protein-protein interaction networks, the problem definition and algorithm framework can be easily extended to other types of biological networks as there is no strict constraint on the type of biological network. There are, however, some issues to consider. One important assumption for the active module score we use is that the p-values annotated for each node in the network follow a beta-uniform mixture distribution. In the experimental networks the quantile-quantile plots of empirical and estimated p-value distribution prove that it is a good match. It is not always the case in other types of networks. In gene co-expression network with some cutting off threshold, the proportion of nodes with low p-values is increased as the threshold becomes stringent, thus beta-uniform model is not that suitable. Besides, some types of networks are edge weighted instead of node weighted. In these cases, we need to consider a new formulation of scoring functions.

- **Acceleration of algorithm speed.**

  A major criticism on genetic algorithm is the scalability with complexity and slow convergence, mainly due to its population based strategy and a large proportion of inferior solutions generated by genetic operators that cannot make good use of problem-specific structure. For networks with hundreds of nodes, the proposed algorithms are able to give satisfactory results in a few hundred generations. When dealing with large scale networks with thousands of nodes, the algorithms take much longer generation to converge as the search space grows quickly. Improvement of the algorithm speed can be considered by adding data structures to store intermediate variables in order to reduce the times of solution evaluation or useless genetic operation, or by choosing a different optimisation framework.

- **Improvement of validation methods.**

  A common problem of research in an interdisciplinary field is that we often have to face the challenges from two sides. In the development of computational tools for understanding complex biological system, no matter how far we can push the boundary of algorithm efficiency and performance score, eventually we will go back to the real biological meaning. Gene ontology annotation is a widely used method to explain the actual sense of module as a collection of genes. Aside from it, there is plenty of room for other various validation methods. How would the different linkages of gene pairs inside a module effect its function? Would the exploration of interactions between modules bring any new insights? In chapter 5 we have incorporated pathway information into the active module, can we use knowledge from a bunch of other biological information database for search guidance or validation as well?

- **Formulation of active module identification in evolving networks.**

  There have been a number of works on active module identification in static networks, but a limited number of studies on evolving network which is consist of multiple slices of networks corresponding to different time point. The dynamics and consistency of active modules in an evolving network is interesting as the network is not limited to a snapshot of biological system, but contains time course data that can reflect certain cellular process from beginning to end. Current methods for analysing evolving network often simply apply existing static network analysing methods to each slice of the network, and match the results across layers. It is worth developing specific methods for handling active module identification in evolving networks directly.

LIST OF REFERENCES

[1] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *nature*, 406(6794):378–382, 2000.

[2] David B Allison, Xiangqin Cui, Grier P Page, and Mahyar Sabripour. Microarray data analysis: from disarray to consolidation and consensus. *Nature reviews genetics*, 7(1):55–65, 2006.

[3] Christina Backes, Alexander Rurainski, Gunnar W Klau, Oliver Müller, Daniel Stöckel, Andreas Gerasch, Jan Küntzer, Daniela Maisel, Nicole Ludwig, Matthias Hein, et al. An integer linear programming approach for finding deregulated subgraphs in regulatory networks. *Nucleic acids research*, 40(6):e43–e43, 2011.

[4] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[5] Albert-László Barabási, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, 2011.

[6] Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113, 2004.

[7] Daniela Beisser, Gunnar W Klau, Thomas Dandekar, Tobias Müller, and Marcus T Dittrich. Bionet: an r-package for the functional analysis of biological networks. *Bioinformatics*, 26(8):1129–1130, 2010.

[8] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, 57:289–300, 1995.

[9] BioGRID (The Biological General Repository for Interaction Datasets). `https://thebiogrid.org/`. Accessed May 2016.

[10] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[11] Rohitash Chandra, Abhishek Gupta, Yew-Soon Ong, and Chi-Keong Goh. Evolutionary multi-task learning for modular training of feedforward neural networks. In *International Conference on Neural Information Processing*, pages 37–46. Springer, 2016.

[12] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[13] Gene Ontology Consortium et al. Gene ontology consortium: going forward. *Nucleic acids research*, 43(D1):D1049–D1056, 2015.

[14] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[15] Marcus T Dittrich, Gunnar W Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Müller. Identifying functional modules in protein–protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–i231, 2008.

[16] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical review E*, 72(2):027104, 2005.

[17] Liang Feng, Yew-Soon Ong, Meng-Hiot Lim, and Ivor W Tsang. Memetic search with interdomain learning: A realization between cvrp and carp. *IEEE Transactions on Evolutionary Computation*, 19(5):644–658, 2015.

[18] Liang Feng, Yew-Soon Ong, Ah-Hwee Tan, and Ivor W Tsang. Memes as building blocks: a case study on evolutionary optimization+ transfer learning for routing problems. *Memetic Computing*, 7(3):159–180, 2015.

[19] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[20] Carlos M Fonseca and Peter J Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary computation*, 3(1):1–16, 1995.

[21] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[22] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[23] Gene Ontology Consortium. `http://geneontology.org/`. Accessed Apr 2016.

[24] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[25] Andrew M Gross and Trey Ideker. Molecular networks in context. *Nature biotechnology*, 33(7):720–721, 2015.

[26] Zheng Guo, Yongjin Li, Xue Gong, Chen Yao, Wencai Ma, Dong Wang, Yanhui Li, Jing Zhu, Min Zhang, Da Yang, et al. Edge-based scoring and searching method for identifying condition-responsive protein–protein interaction sub-network. *Bioinformatics*, 23(16):2121–2128, 2007.

[27] Abhishek Gupta, Jacek Mańdziuk, and Yew-Soon Ong. Evolutionary multitasking in bi-level optimization. *Complex & Intelligent Systems*, 1(1-4):83–95, 2015.

[28] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Multifactorial evolution: toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 20(3):343–357, 2016.

[29] Abhishek Gupta, Yew-Soon Ong, Liang Feng, and Kay Chen Tan. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE transactions on cybernetics*, 47(7):1652–1665, 2017.

[30] Leland H Hartwell, John J Hopfield, Stanislas Leibler, Andrew W Murray, et al. From molecular to modular cell biology. *Nature*, 402(6761):C47, 1999.

[31] Shan He, Guanbo Jia, Zexuan Zhu, Daniel A Tennant, Qiang Huang, Ke Tang, Jing Liu, Mirco Musolesi, John K Heath, and Xin Yao. Cooperative co-evolutionary module identification with application to cancer disease module discovery. *IEEE Transactions on Evolutionary Computation*, 20(6):874–891, 2016.

[32] Qiang Huang, Thomas White, Guanbo Jia, Mirco Musolesi, Nil Turan, Ke Tang, Shan He, John K Heath, and Xin Yao. Community detection using cooperative coevolutionary differential evolution. In *International Conference on Parallel Problem Solving from Nature*, pages 235–244, 2012.

[33] Taeyoung Hwang and Taesung Park. Identification of differentially expressed subnetworks based on multivariate anova. *BMC bioinformatics*, 10(1):1, 2009.

[34] Trey Ideker, Owen Ozier, Benno Schwikowski, and Andrew F Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18(suppl 1):S233–S240, 2002.

[35] JActiveModules in Cytoscape App Store. `http://apps.cytoscape.org/apps/jactivemodules`. Accessed October 2015.

[36] A Jamakovic and Piet Van Mieghem. On the robustness of complex networks by using the algebraic connectivity. *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, pages 183–194, 2008.

[37] The JASPAR database. `http://jaspar.binf.ku.dk/`. Accessed Sep 2017.

[38] Guanbo Jia, Zixing Cai, Mirco Musolesi, Yong Wang, Dan A Tennant, R Weber, John K Heath, and Shan He. Community detection in social and biological networks using differential evolution. *Learning and Intelligent Optimization*, pages 71–85, 2012.

[39] Inga Kadish, Olivier Thibault, Eric M Blalock, Kuey-C Chen, John C Gant, Nada M Porter, and Philip W Landfield. Hippocampal and cognitive aging across the lifespan: a bioenergetic shift precedes and increased cholesterol trafficking parallels memory impairment. *Journal of Neuroscience*, 29(6):1805–1816, 2009.

[40] Minoru Kanehisa and Susumu Goto. KEGG: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

[41] KEGG: Kyoto Encyclopedia of Genes and Genomes. `http://www.genome.jp/kegg/`. Accessed Apr 2016.

[42] KEGG REST-style entry for *Saccharomyces cerevisiae*. `http://rest.kegg.jp/link/sce/pathway`. Accessed March 2016.

[43] Martin Klammer, Klaus Godl, Andreas Tebbe, and Christoph Schaab. Identifying differentially regulated subnetworks from phosphoproteomic data. *BMC bioinformatics*, 11(1):1, 2010.

[44] Minh Nghia Le, Yew Soon Ong, Stefan Menzel, Chun-Wei Seah, and Bernhard Sendhoff. Multi co-objective evolutionary optimization: Cross surrogate augmentation for computationally expensive problems. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE, 2012.

[45] Dong Li, Zhisong Pan, Guyu Hu, Zexuan Zhu, and Shan He. Active module identification in intracellular networks using a memetic algorithm with a new binary decoding scheme. *BMC genomics*, 18(2):209, 2017.

[46] Shuzhuo Li, Yinghui Chen, Haifeng Du, and Marcus W Feldman. A genetic algorithm with local search strategy for improved detection of community structure. *Complexity*, 15(4):53–60, 2010.

[47] Dudy Lim, Yew-Soon Ong, Abhishek Gupta, Chi Keong Goh, and Partha Sarathi Dutta. Towards a new praxis in optinformatics targeting knowledge re-use in evolutionary computation: simultaneous problem learning and optimization. *Evolutionary Intelligence*, 9(4):203–220, 2016.

[48] Yunpeng Liu, Daniel A Tennant, Zexuan Zhu, John K Heath, Xin Yao, and Shan He. Dime: a scalable disease module identification algorithm with application to glioma progression. *PloS one*, 9(2):e86693, 2014.

[49] David Lusseau. The emergent properties of a dolphin social network. *Proceedings of the Royal Society of London B: Biological Sciences*, 270(Suppl 2):S186–S188, 2003.

[50] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[51] Haisu Ma, Eric E Schadt, Lee M Kaplan, and Hongyu Zhao. COSINE: COndition-SpecIfic sub-NEtwork identification using a global optimization method. *Bioinformatics*, 27(9):1290–1298, 2011.

[52] Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45, 2004.

[53] The MIPS Mammalian Protein-Protein Database. http://mips.helmholtz-muenchen.de/proj/ppi/. Accessed Sep 2017.

[54] Koyel Mitra, Anne-Ruxandra Carvunis, Sanath Kumar Ramesh, and Trey Ideker. Integrative approaches for finding modular structure in biological networks. *Nature Reviews Genetics*, 14(10):719–732, 2013.

[55] Transcriptional profiles of rodent hippocampal ca1 tissue during aging and cognitive decline GSE9990. https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9990. Accessed Oct 2017.

[56] Daniele Muraro and Alison Simmons. An integrative analysis of gene expression and molecular interaction data to identify dys-regulated sub-networks in inflammatory bowel disease. *BMC bioinformatics*, 17(1):1, 2016.

[57] Raj Rao Nadakuditi and Mark EJ Newman. Graph spectra and the detectability of community structure in networks. *Physical review letters*, 108(18):188701, 2012.

[58] GEO2R. http://www.ncbi.nlm.nih.gov/geo/geo2r/. Accessed May 2016.

[59] Mark EJ Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.

[60] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[61] Mark EJ Newman. Analysis of weighted networks. *Physical review E*, 70(5):056131, 2004.

[62] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[63] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[64] Mark EJ Newman. Communities, modules and large-scale structure in networks. *Nature physics*, 8(1):25, 2012.

[65] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[66] Yew-Soon Ong and Abhishek Gupta. Evolutionary multitasking: a computer science view of cognitive multitasking. *Cognitive Computation*, 8(2):125–142, 2016.

[67] Yew-Soon Ong, Meng Hiot Lim, and Xianshun Chen. Memetic computationpast, present & future [research frontier]. *IEEE Computational Intelligence Magazine*, 5(2):24–31, 2010.

[68] Alain Pétrowski. A clearing procedure as a niching method for genetic algorithms. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 798–803. IEEE, 1996.

[69] Stan Pounds and Stephan W Morris. Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values. *Bioinformatics*, 19(10):1236–1242, 2003.

[70] Amela Prelić, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.

[71] David J Reiss, Nitin S Baliga, and Richard Bonneau. Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. *BMC bioinformatics*, 7(1):280, 2006.

[72] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

[73] Ronghua Shang, Jing Bai, Licheng Jiao, and Chao Jin. Community detection based on modularity and an improved genetic algorithm. *Physica A: Statistical Mechanics and its Applications*, 392(5):1215–1231, 2013.

[74] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research*, 13(11):2498–2504, 2003.

[75] Tomer Shlomi, Moran N Cabili, Markus J Herrgård, Bernhard Ø Palsson, and Eytan Ruppin. Network-based prediction of human tissue-specific metabolism. *Nature biotechnology*, 26(9):1003–1010, 2008.

[76] Steven S Skiena. *The algorithm design manual*, volume 1. Springer Science & Business Media, 1998.

[77] STRING: functional protein association networks. `https://string-db.org/`. Accessed Sep 2017.

[78] Joshua M Stuart, Eran Segal, Daphne Koller, and Stuart K Kim. A gene-coexpression network for global discovery of conserved genetic modules. *science*, 302(5643):249–255, 2003.

[79] Jeffrey Travers and Stanley Milgram. The small world problem. *Phychology Today*, 1:61–67, 1967.

[80] Virginia Goss Tusher, Robert Tibshirani, and Gilbert Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, 98(9):5116–5121, 2001.

[81] John L Tymoczko, Jeremy M Berg, and Lubert Stryer. *Biochemistry: a short course*. Macmillan, 2011.

[82] The UniPROBE (Universal PBM Resource for Oligonucleotide Binding Evaluation) . `http://the_brain.bwh.harvard.edu/uniprobe/`. Accessed Sep 2017.

[83] Jolanda S van Leeuwen, Nico PE Vermeulen, and J Chris Vos. Involvement of the pleiotropic drug resistance response, protein kinase c signaling, and altered zinc homeostasis in resistance of saccharomyces cerevisiae to diclofenac. *Applied and environmental microbiology*, 77(17):5973–5980, 2011.

[84] Fabio Vandin, Eli Upfal, and Benjamin J Raphael. Algorithms for detecting significantly mutated pathways in cancer. *Journal of Computational Biology*, 18(3):507–522, 2011.

[85] Andreas Wagner. How the global structure of protein interaction networks evolves. *Proceedings of the Royal Society of London B: Biological Sciences*, 270(1514):457–466, 2003.

[86] Yong Wang and Yu Xia. Condition specific subnetwork identification using an optimization model. *Optimization and Systems Biology*, pages 333–340, 2008.

[87] Yu-Chao Wang and Bor-Sen Chen. Integrated cellular network of transcription regulations and protein-protein interactions. *BMC Systems Biology*, 4(1):1, 2010.

[88] NCBI Gene Exprssion Omnibus - GSE29331. `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE29331`. Accessed Apr 2016.

[89] Yuan Yuan, Yew-Soon Ong, Abhishek Gupta, Puay Siew Tan, and Hua Xu. Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with tsp, qap, lop, and jsp. In *Region 10 Conference (TENCON), 2016 IEEE*, pages 3157–3164. IEEE, 2016.

[90] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[91] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1), 2005.

[92] Xing-Ming Zhao, Rui-Sheng Wang, Luonan Chen, and Kazuyuki Aihara. Uncovering signal transduction networks from high-throughput data by integer linear programming. *Nucleic acids research*, 36(9):e48–e48, 2008.