Multi-resolution functional summarization and alignment of biological network models

Seah, Boon Siew

2014

Seah, B. S. (2014). Multi-resolution functional summarization and alignment of biological network models. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/60641>

<https://doi.org/10.32657/10356/60641>

MULTI-RESOLUTION FUNCTIONAL SUMMARIZATION AND ALIGNMENT OF BIOLOGICAL NETWORK MODELS

Seah Boon Siew

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTATION AND SYSTEMS BIOLOGY (CSB)

SINGAPORE-MIT ALLIANCE
NANYANG TECHNOLOGICAL UNIVERSITY

Feb 17, 2014

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Seah Boon Siew

Feb 17, 2014

Acknowledgments

I would like to thank my supervisor **Assoc. Prof. Sourav S. Bhowmick** (NTU) and my co-supervisor **Prof. C. Forbes Dewey, Jr.** (MIT) for their guidance and support. I have learned from them a multitude of skills, including research, writing and presentation skills. They have taught me that there is no substitute for professionalism and attention to detail.

I also want to thank my colleagues Ms. Chua Huey Eng, Dr. Naveen Kumar Balla and Dr. Lakshmi Venkatraman for their technical (and personal) discussions. Moreover, I wish to show my appreciation to the following people: Dr. Erwin Leonardi, Dr. Andrew Koo, Dr. Shiva Ayyadurai, Assoc. Prof. Sun Aixin, Mr. Truong Ba Quan, Asst. Prof. Li Hui and Mr. Fajar Ardian. Also thanks to Mr. Lai Chee Keong and Mr. Loo Kian Hock for providing great technical support. I am much indebted to Singapore-MIT Alliance for the research scholarship.

Great thanks to my family for their love and patience. Finally, I am grateful to my wife Koo Khai Nee. This dissertation could not have been completed without her ceaseless encouragement and support.

Abstract

The desire to study biology from a systems perspective has led to an emergence of new science – biological networks analysis. Biological network models biological entities (*e.g.*, proteins and genes) and their relationships (*e.g.*, physical and genetic interactions) to characterize their cooperative activity within a system. With the rapid growth of network data, the *information overload* problem arises and human interpretation of such data becomes impossible. Hence, there is urgent need to construct methods for large-scale functional visualization of biological networks to understand the mechanics of biological systems.

In this dissertation, we aim to build frameworks that allow biologists to rapidly visualize the processes that govern biological systems via: 1) functional organization *within* a biological network (intra-system processes), and 2) functional relationships *between* biological networks (inter-system processes). Drawing on well-founded principles in data mining, systems biology and bioinformatics, we propose a *multi-resolution* and *multi-perspective* analysis paradigm to address both objectives. We propose the **FUSE** algorithm that systematically summarizes a protein-protein interaction (PPI) network in a multi-resolution fashion. **FUSE** summaries visualize not only the functional structure and organization within a network but also the relationships between processes. In particular, **FUSE** summaries of a network are multi-resolution and depict the functional landscape of the biological system at multiple levels of detail (**FUSE** for biological networks is analogous to **Google Maps** for geographic landscapes). Following that, we

extend **FUSE** to support a quantitative network model that more accurately depict the behavior of a biological system. We develop **DiffNet**, which constructs summaries of differential gene interaction networks (dE-MAP networks) to automatically visualize the *functional differential* regions that undergo “rewiring” after environmental change. We propose the **FACET** algorithm that summarizes a PPI network in a multi-perspective manner. This is based on the fact that a biological system can be seen from different functional perspectives (*e.g.*, components in a PPI network can be organized by localization, process, disease, etc.) The **FACET** algorithm automatically identifies unique and orthogonal functional landscapes of the network. Finally, we propose the **DualAligner** algorithm that characterizes conserved functional relationships between PPI networks via *network alignment*. Network alignment aligns two or more PPI networks to obtain conserved regions. **DualAligner** performs multi-resolution alignment not just at fine detail (alignment between biological entities), but also at coarser, high-level detail (alignment between functional regions). We tested our proposed algorithms on real-life biological datasets and demonstrated its superiority over current state-of-the-art methods.

Contents

Acknowledgments	iii
Abstract	iv
List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Challenges	4
1.2 Contribution	7
1.3 Outline	9
2 Background	12
2.1 Proteins: The Building Block of Life	12
2.2 Protein-protein Interactions (PPI)	14
2.3 Methods to Analyze Protein-protein Interactions	16
2.3.1 Yeast Two-Hybrid (Y2H)	17
2.3.2 Tandem Affinity Purification (TAP)	17
2.3.3 Bimolecular Fluorescence Complementation (BIFC)	18
2.3.4 Noise in High-throughput Screening Methods	19
2.4 Protein-protein Interaction Databases	19
2.5 Annotating the Roles of Proteins and Their Interactions	21
2.5.1 The Structure of Gene Ontology	22
2.6 Summary	25
3 Related Work	27

3.1	Graph Clustering of PPI Networks	27
3.1.1	Problem Definition of PPI Network Clustering . . .	30
3.1.2	Overview of PPI Network Clustering	30
3.1.3	Heuristics-based Algorithms	31
3.1.4	Complete Enumeration Algorithms	36
3.1.5	Random Walks and Message Passing Algorithms . . .	38
3.1.6	Flow Based Algorithms	41
3.1.7	Graph-cut and Hierarchical Clustering Algorithms . .	43
3.1.8	Other Algorithms	47
3.1.9	Detecting Structurally Loose Modules	49
3.1.10	Summary of PPI Network Clustering Algorithms . .	49
3.2	Network Alignment of PPI Networks	51
3.2.1	Overview of PPI Network Alignment Algorithms. . .	52
3.2.2	Dynamic Programming Algorithms	55
3.2.3	Seed and Expand Algorithms	56
3.2.4	Random Walk Algorithms	64
3.2.5	Integer Linear Program Algorithms	68
3.2.6	Summary of Network Alignment Algorithms	69
3.3	Summary	70
4	FUSE: Towards Multi-Level Functional Summarization of Protein Interaction Networks	73
4.1	Motivation	74
4.2	Overview	79
4.3	Related Work	80
4.4	The Functional Summarization Problem	82
4.4.1	Functional Summary of PPI	83
4.4.2	Problem Statement	86
4.5	The Algorithm FUSE	90
4.6	Experimental Results	95

4.6.1	Evaluation Metrics	95
4.6.2	FUSE vs Graph Clustering Methods	98
4.6.3	Cluster Quality Comparison	100
4.6.4	Function Representativeness Comparison	102
4.6.5	Qualitative Evaluation	104
4.6.6	Effects of User-Defined Parameters	106
4.6.7	Statistical Significance	108
4.6.8	Effect of Annotation Loss	111
4.6.9	Runtime and Scalability	112
4.7	Case Study on AD Network	113
4.8	Inferring Functional Cluster Hubs	116
4.9	Automatic Differential Summarization of dE-MAP networks .	119
4.10	Problem Formulation	123
4.10.1	Functional Subgraphs in a Differential Network . . .	126
4.10.2	The DiffNet Algorithm	126
4.11	Results	127
4.11.1	Weakness of independently clustering positive and negative edges of differential network.	133
4.11.2	Effect of parameter α	133
4.11.3	Running time.	134
4.11.4	Effect of annotation loss on differential summary con- struction.	136
4.12	Software Availability	137
4.13	Conclusions	138
5	FACETS: Multi-faceted Functional Decomposition of Pro- tein Interaction Networks	139
5.1	Motivation	139
5.2	Related work	142
5.3	Problem Statement	143

5.3.1	Terminology	143
5.3.2	Multi-faceted Functional Decomposition Problem . .	144
5.3.3	Problem Definition	147
5.4	FACETS Algorithm	149
5.5	Results	153
5.5.1	Experiment settings	153
5.5.2	Experiment results	154
5.5.3	Statistical Significance of FACETS clusters	159
5.5.4	Running time.	160
5.5.5	Varying parameters of graph clustering methods yields delta differences	161
5.6	Case study: Human autophagy system.	161
5.7	Comparison with GO DAG	163
5.8	Software Availability	165
5.9	Conclusion	165
6	DualAligner: Protein-protein Interaction Network Align- ment via Dual Alignment Strategy	166
6.1	Motivation	167
6.2	Problem Formulation	172
6.2.1	Terminology	172
6.2.2	Region-to-Region Alignment	173
6.2.3	Function-Constrained Network Alignment Problem .	175
6.2.4	The DualAligner Algorithm	178
6.3	Results	181
6.4	Software Availability	189
6.5	Summary	189
7	Conclusions and Future Work	190
7.1	Summary	190

7.2	Future Work	192
7.2.1	Extending Beyond Functional Information	192
7.2.2	Quantitative Network Summarization	193
7.2.3	Quantitative Network Alignment	194
7.2.4	Scalable Multi-resolution Network Alignment	194
7.3	Conclusions	194
A	Differential Functional Summarization	196
A.0.1	Differential Summarization Problem	199
A.0.2	Proof on Remainder Subgraphs	202
B	List of Publications	204
	References	206

List of Figures

1.1	Summary of important processes in the global Arabidopsis PPI network	3
1.2	Information overload in large PPI Networks	5
1.3	Overview of dissertation contributions.	9
2.1	The central dogma of molecular biology	13
2.2	Stable vs. Transient Interactions	15
2.3	Yeast-two-hybrid to detect protein-protein interactions . . .	16
2.4	Tandem Affinity Purification	18
2.5	Subset of the Gene Ontology Directed Acyclic Graph	23
3.1	An example of a PPI network modeled as graph.	28
3.2	An example of a clustering of a PPI network.	29
3.3	Illustration of the MCODE algorithm.	34
3.4	Overlapping versus Non-overlapping Clustering.	35
3.5	Full versus Partial Coverage Clustering.	36
3.6	An illustration of CFinder algorithm	37
3.7	Illustration of shared neighbors and collapse procedure in NeMo algorithm	44
3.8	The VI-Cut algorithm illustrated with a toy example	45
3.9	An example of a network alignment between two PPI networks.	52
3.10	The PathBLAST algorithm	56
3.11	Local vs Global PPI Network Alignment.	57
3.12	Illustration of NetworkBLAST algorithm.	59

3.13	Duplication/Divergence model used in the MaWISh algorithm	60
3.14	The CAPPI algorithm	63
3.15	Illustration of IsoRank algorithm	66
4.1	Functional summary (FSG) of the AD network for $k = 30$ (cluster size indicated in brackets).	74
4.2	FSG of the AD network ($k = 10$).	77
4.3	Toy Example Illustration	78
4.4	Structure and attribute considerations in graph summariza- tion.	79
4.5	Functional clusters associated with the p53 protein.	84
4.6	Illustration of <i>MapProfit</i> procedure.	94
4.7	Cluster quality of FUSE vs graph clustering-based approaches (precision)	97
4.8	Cluster quality of FUSE vs graph clustering-based approaches (recall)	97
4.9	Cluster quality of FUSE vs graph clustering-based approaches (F-score)	98
4.10	Cluster quality of FUSE vs graph clustering-based approaches (FUSE)	98
4.11	Function representativeness (precision)	102
4.12	Function representativeness (recall)	103
4.13	Functional summarization of DNA <i>S.cerevisiae</i>	105
4.14	Effect of k on summary SIC.	106
4.15	Effect of k on summary coverage.	107
4.16	Effect of b and d	108
4.17	Running times of FUSE (in sec.).	109
4.18	Effect of annotation loss.	112
4.19	Scalability of FUSE versus annotation size.	113
4.20	Scalability of FUSE versus vertex size.	114

4.21	Connectivity of functional clusters in <i>H. sapiens</i> network . .	117
4.22	The dE-MAP network that arises from two static E-MAP net- works under different conditions	120
4.23	Differential summary of de-MAP network	123
4.24	Illustration of DiffNet	124
4.25	Functional summary of stable modules.	128
4.26	Comparison with general graph clustering algorithms.	129
4.27	Effect of parameter k on DiffNet.	131
4.28	Effect of interaction noise.	132
4.29	Effect of α on DiffNet.	134
4.30	Independently clustering positive and negative edges of dif- ferential network.	134
4.31	Running time of DiffNet.	135
4.32	Running time of DiffNet at varying network density.	135
4.33	Effect of annotation loss on differential summary construc- tion.	136
4.34	Effect of MCL inflation parameter.	136
5.1	Illustration of multi-faceted PPI network decomposition. . . .	140
5.2	Illustration of the FACETS algorithm	148
5.3	Comparison between the decomposition similarities of FACETS , other methods, and gold standard decompositions.	156
5.4	Effect of noise on FACETS algorithm.	157
5.5	Effect of initial starting point versus noise on FACETS algo- rithm.	158
5.6	Rate of convergence of FACETS algorithm	159
5.7	Running time of FACETS algorithm.	159
5.8	Varying parameters of clustering methods.	161

5.9	Multiple facets (subset) illustrating the functional organization of the human autophagy network under different perspectives.	162
6.1	Illustration of DualAligner	168
6.2	Hard constraint versus soft constraint.	170
6.3	Selected region-to-region alignments showing highly conserved subgraphs between human and yeast networks. Note that there may not necessarily be an optimal protein-to-protein alignment between the subgraphs.	182
6.4	Performance of DualAligner (human vs yeast) protein-to-protein alignment; alignment of each method may not have the same coverage. With DualAligner , one can adjust the trade-off between alignment quality and coverage. The dashed vertical line indicate the portion of the DualAligner alignment that is aligned from region-to-region alignment.	183
6.5	Performance of DualAligner (fly vs yeast).	186
6.6	Effect of λ_s showing its role in controlling the trade-off between topology (EC) and sequence (bit-score) conservation. .	187
A.1	A toy differential network of gene interactions.	199

List of Tables

2.1	Selected Protein-protein Interaction Databases.	20
3.1	Overview of PPI network clustering techniques.	32
3.2	Summary of PPI network clustering techniques.	50
3.3	Overview of PPI network alignment approaches.	54
3.4	Summary of PPI network alignment techniques.	69
4.1	Notations in FUSE	82
4.2	Summary of datasets used.	95
4.3	Summary of DNA <i>S.cerevisiae</i> obtained through Cluster+Enrich (9 single member clusters are excluded).	109
4.4	The p-value significance of FUSE clusters.	110
4.5	High-degree CC functional clusters in the <i>H. sapiens</i> sum- mary ($k = 400$).	116
4.6	High-degree BP functional clusters in the <i>H. sapiens</i> sum- mary ($k = 400$).	118
5.1	Datasets used in FACETS	153
5.2	Comparison between facets of the <i>H. sapiens</i> PPI network ($n = 6$).	154
5.3	The p-value significance of FACETS clusters.	160
5.4	Comparing GO terms at a particular level in the GO DAG. .	164
6.1	Datasets.	181
6.2	Running times (in seconds).	188

Chapter 1

Introduction

For decades, scientists have studied the components of living systems in isolation [1]. For instance, in a study of proteins, genes, or even biological pathways, the component of interest is first removed from its true environment and then studied by observing its individual properties. This approach has served the research community well, and has been – and still is – an extremely effective technique at uncovering the properties of the components at a detailed level. However, its limitations are also apparent in recent times [2]. While effective at studying their behavior and properties in isolation, the behavior of isolated components often cannot be trivially extrapolated to groups of components when put together. For instance, the behavior of proteins *in vitro* often contradicts the behavior *in vivo* [1]. Proteins often play multiple roles (*moonlighting*), and the processes in which they take part are contextual and dynamic [3]. Even biological processes themselves do not operate in isolation; instead, they are a well orchestrated cooperation among multiple processes. An extreme example is that of social organisms (*e.g.*, ants) and their social structure needed to survive and operate together [4].

In light of this, the approach of viewing biological systems from a broader, global perspective is an increasingly attractive enterprise [5, 2, 6]. Rather than modeling components in an isolated, reductionist manner, the cooperative activity of a group of components is modeled as well. The

CHAPTER 1. INTRODUCTION

systems-based paradigm looks at not just the individual components, but also their activity and relationships as a cooperative whole [5]. The most well-known method to model biological systems in this manner is through *biological networks* (graphs) [7]. The analysis of biological networks shall be the main focus of this dissertation.

A biological network is modeled as a graph $G = (V, E, w)$, where V is a set containing the components of the network, $E \subseteq V \times V$ is a set containing the pairwise relationships between the components, and $w : E \rightarrow \mathbb{R}$ is a real-valued weight function that assigns weights to each $e \in E$. It lays out the structure of the components and their relationship and enables mathematical analysis to be performed on this structure. The most common class of biological networks is the *protein-protein interaction network* (PPI network) [7]. Here, V is the set of proteins in the PPI network to be modeled, E is the set of physical interactions among the proteins in V , and w is a function that models the strength or confidence of the interactions. Another example is a biological network model of pathway-pathway interactions [8]. One toy example of a pathway-pathway interaction network is a network of interactions between cell-cycle, apoptosis, and DNA repair pathways. In this case, V is the set of pathways, E is the set of pathway-pathway relationships, and w maps the relationship strength between pathways. Many other classes of biological networks exist. They may range from neuronal and disease networks to mRNA networks, transcriptional regulatory networks, and DNA-protein interaction networks. Although there are unique properties that defines each class of network, many common network properties emerge among them, and consequently, many analytical methods that apply to one class of network can be transferred to other classes of networks.

The desire to study biological systems from a global perspective has led to an emergence of new science—biological networks analysis. With network

Figure 1.1: Summary of important processes in the global Arabidopsis PPI network. (Taken from [9])

analysis, one may uncover key system-wide properties and behaviors of the biological system that reductionist methods could not. In the seminal paper by Barabasi *et al.* [10], the authors discovered the *scale-free* distribution of biological networks and proposed the preferential attachment model of proteins. Many other subsequent studies reveal other important models and properties of PPI networks using networks analysis, including the party and date hub model of proteins [11] and the evolutionary models of PPI networks [12]. Despite their importance to systems level biology, there is still a gap between networks analysis that searches for general properties and *functional analysis*¹ of networks needed by the average biological researcher. While the aforementioned studies uncover general properties of proteins and their interactions, others may still wish to interpret networks

¹Network functional analysis is the analysis of the underlying biological roles and function of the network (and its subnetworks).

in a more specific and concrete manner. We justify this with an example. Consider an analysis of the the Alzheimer's Disease PPI network [13]. Network analysis may reveal interaction distribution and characteristic of Alzheimer's Disease from an general viewpoint – for example, the degree distribution of the proteins in the network – but a typical researcher studying the Alzheimer's Disease PPI network may also want to look for more concrete patterns and observations. For instance, one may wish to look a summary of most important functional processes and their relationships (*e.g.*, the relationship between transport and apoptosis processes) that take part in the network. Figure 1.1 illustrates an example of a summary that organizes the PPI network hairball into groups of functionally related proteins.

1.1 Challenges

The complexity of modeling biological processes from a systems viewpoint gives rise to several challenges. The first challenge associated with networks analysis is the amount data required. The information needed to perform a global perspective analysis is daunting. To illustrate this, consider a simple case of 100 proteins in a biological system. While there are only 100 proteins to study in isolation, the networks approach studies not just the 100 proteins, but also potentially 10000 pairwise relationships among them. Even in this simple case, the combinatorial complexity of networks analysis literally increases the complexity of the study by several orders of magnitude. If the cost of acquiring 10000 relationship data is prohibitive, then systems-based study is not even close to feasible. Fortunately, recent advances in high-throughput technologies such as yeast-two-hybrid have play a massive impact in enabling such studies [14].

The second important challenge is noise. A natural consequence of large scale automated techniques like two-hybrid screening is the high rate of false

Figure 1.2: Information overload in large PPI Networks. (Taken from <http://navicluster.cb.k.u-tokyo.ac.jp/cs/>)

positives and false negatives [15]. It is important for any analytical tools to take noise into account to guard against spurious predictions. The approach proposed in [16], for example, takes into account the noisy nature of high-throughput data in inferring the predicting interactions from heterogeneous sources.

Finally, the third challenge is information overload. The deluge of data from high-throughput experiments comes at a cost. A biologist may find data provided by interaction datasets in its raw form overwhelming (see Figure 1.2). The difficulty of analyzing and interpreting such complex dataset is called *information overload*. As such, a major challenge to biologists is to make sense out the intertwining hairball of information contained in large PPI networks. One may wish to find ways to extract summarized

CHAPTER 1. INTRODUCTION

information about the PPI network. Alternatively, one may wish to find ways to compare several PPI networks to identify conserved regions or patterns. This may allow one to distinguish regions of the network that undergo significant changes in its diseased state compared to its normal state, for example.

Given the above challenges, a multitude of algorithms have been proposed [17, 18, 19, 20]. We highlight two important classes of such network analysis algorithms that is relevant to this dissertation – network clustering and network alignment [21]. Network clustering aims to identify densely interacting regions of the network. The clustering process assists in summarizing the PPI network and also to reveal interesting functional predictions regarding the cluster. In [22], network clustering is applied on global yeast PPI network to uncover the landscape of important functional modules within the network including protein complexes. Network alignment, on the other hand, is analogous to sequence alignment [23]. It compares two or more PPI networks to identify highly conserved regions among the networks. In [24], network alignment is performed on PPI networks of several species to uncover the conserved network regions, allowing the authors to construct a putative ancestral networks that underlie these PPI networks. In Chapter 3 we will survey these algorithms and discuss their strengths and limitations. Recall the large number of interaction and interactor attributes provided by biological networks data. When confronted with such a deluge of data, biological researchers are still limited in their ability to manually interpret and analyze the PPI networks together with their protein attributes. Each protein may be annotated with hundreds, if not thousands, of attribute information. Together with the large number of proteins and their interactions, interpreting these data as whole can be a daunting task.

1.2 Contribution

This dissertation focuses on making sense out of this deluge of PPI network data. We propose several algorithms that address the limitations of current network clustering and alignment techniques. We aim to bridge the gap between the complexity of large scale protein-protein interaction network data and concise interpretability demanded by the typical biological scientist.

In particular, we focus on *attributed* PPI networks, which is an extension of general PPI networks. Instead of modeling the proteins in the network as homogeneous entities, in attributed networks we endow each component with attributes (such as functional annotations). Here attributed PPI networks refer to networks whose proteins are not treated as homogeneous nodes, but entities that have attributes associated with it. For instance, a protein can have GO term annotations as its attributes. The richness provided by this extended graphical model introduces additional challenges to their analysis (for example, the high dimensionality of protein attributes), but at the same time, it opens the door for opportunities to yield novel findings. Few studies have considered networks analysis under on attributed PPI networks. We shall describe later how such networks can be obtained, and how their analysis using our proposed methods are advantageous to standard approaches. Towards addressing existing limitations, especially with regard to information overload, we present the following contributions:

- **Functional summarization of PPI networks.** We present the first algorithm to specifically construct birds-eye functional maps of the underlying PPI network at multiple levels of detail (*multi-resolution*). Unlike graph clustering algorithms that focuses on finding strongly coherent subgraphs, our summarization algorithm is focused on ensuring the modules are representative of the function described by the summary and the entire map is representative of the

underlying PPI network. Functional summarization allow researchers to overcome information overload problem associated interpretation of large-scale PPI networks, allowing visual interpretation of the functional components and their interactions that underlie a PPI network. We show how functional summarization can be adapted to uncover functional regions of a PPI network that under are significantly different two different conditions (for example, cancer vs normal cells).

- Multi-faceted summarization of PPI networks.** We present the first algorithm to provide *multiple perspective* summaries of the underlying PPI network (*multi-perspective*). Each summary represents a *facet* of the network that represents a particular functional organization, and the set of summaries presented are *functionally orthogonal*. We define functionally orthogonal summaries as a set of summaries, such that each summary represents a unique functional organization of the network that is different from the rest of the summaries. Existing graph clustering methods, on the other hand, generally present a single clustering view of the PPI network. We show quantitatively that adjusting the parameters of these methods do not yield significantly distinct clusterings from a functional viewpoint. The approach proposed here, on the other hand, is specifically designed to construct a set of unique summaries.
- Region-to-region network alignment that preserves function.** We present an algorithm that aligns PPI networks in a manner that is function preserving. Instead of focusing on aligning individual proteins of the networks, our algorithm aligns both functionally conserved regions as well as their constituent individual proteins (*multi-resolution*). The functional region conservation alignments along with individual protein alignments establishes consistency to the alignment, such that both function and sequence homology are preserved.

Figure 1.3: Overview of dissertation contributions.

1.3 Outline

The rest of this dissertation is organized as follows:

In Chapter 2 we present the elements that serve as background for the remaining chapters of the dissertation. In particular, we focus on the modeling of biological systems as networks. We discuss how interaction data is acquired, and we discuss several knowledge-bases that provide a wealth of interaction and functional information. We also discuss Gene Ontology and gene function annotations [25], which provide controlled annotation describing function and activity of genes, gene products (including proteins) and their interactions.

Chapter 3 introduces two key network analysis classes pertinent to this dissertation – network clustering and network alignment [21]. For both classes, we present a survey of existing methodologies together with an evaluation of the strengths and limitations of current tools, especially with respect to their ability to assist biologists in interpreting complex biological networks.

In Chapter 4, we introduce **FUSE**, a novel methodology that constructs functional summaries of any PPI network. The central goal of this ap-

proach is to provide researchers a summary of PPI network from a functional, birds-eye perspective. The summary reduces the complex “hairball” of PPI data into concise functional subgraphs that, along with their interactions, represent a compressed functional representation of the underlying PPI network. In addition, we provide an ability to control the granularity of this summary, with which we use to construct multiple layers of birds-eye view summaries with varying complexity. We formulate the summarization process as a profit maximization problem, and we uncover important properties of this formulation that allows us to propose an effective solution to the problem. We demonstrate the effectiveness of the method. We also show that our method compares favorably to existing strategies. Finally, we show the role of **FUSE** in summarizing pertinent functional differences between two **E-MAP** networks under contrasting conditions.

Chapter 5 discusses **FACETS**, an algorithm we propose that addresses the limitation of existing graph clustering methods that present only *one* clustering perspective of the PPI network. Instead, it recognizes that most PPI network can be organized in multiple ways. To this end, we present a novel algorithm that extends the capability of **FUSE** to construct an atlas of summaries, each presenting a *unique functional perspective* of the underlying PPI network.

Finally in Chapter 6, we discuss a novel algorithm that utilizes the technologies in **FUSE** to network alignment. Analogous to sequence alignment that identifies conserved sequence regions, network alignment analysis aligns two or more PPI network to uncover the conserved PPI subgraph regions. We propose a novel methodology that expands on the concept of network conservation to not just conservation between individual proteins among PPI networks, but also functional conservation of subgraphs or clusters among PPI networks.

Figure 1.3 illustrates the key contributions of this dissertation. While

CHAPTER 1. INTRODUCTION

the **FUSE** algorithm summarizes a network into a single organization view, **FACETS** constructs a multi-view summary of the network, allowing deeper functional overview of the network. **DiffNet**, on the other hand, extends **FUSE** by considering the functional summarization of a network under *multiple conditions*. All of these methods consider the functional organization within a single network. In Chapter 6, the **DualAligner** approach extends the concepts in **FUSE** to analyze relationships between PPI networks.

Chapter 2

Background

This chapter provides an overview of key topics that serve as background for the rest of the dissertation. First, we discuss the roles of proteins in living organisms. This is followed by a brief discussion on protein-protein interactions and methods for analyzing them. Finally, we touch on the roles of databases, ontologies and annotations in proteins and their interactions.

2.1 Proteins: The Building Block of Life

The basic building block of all living organisms is the cell. The cell itself is a complex machinery – within it a plethora of processes and parts that govern the mechanisms of the cell [26]. Microtubules, tubular shaped scaffolds of the cell, provide not only shape and structure, but also act as tracks for transporting cellular cargoes. Mitochondrions are the molecular engines of the cell, generating fuel to power cellular machines. These are just a few examples of cell parts that regulate the cell machinery.

The various parts of the cell work in tandem to regulate *biological processes* – functionalities performed within the cell that control its behavior and state, depending on its internal and external environment. For example, the cell cycle is a biological process that controls the growth and replication of itself. Transport processes cargo cell parts within the cell, as well as exporting cargoes out of the cell and importing cargoes into it.

CHAPTER 2. BACKGROUND

Figure 2.1: The central dogma of molecular biology. (Image by Dhorspool at en.wikipedia)

Homeostatic processes regulate the equilibrium of chemical concentration in the cell to a desirable optimum.

Remarkably, the machines that run biological processes of cells are largely performed by one class of molecules called *proteins* [27]. A protein is made up of a linear sequence of amino acids that are folded into a 3D structure. Informally, one can think of proteins as strings of words formed by an alphabet of amino acids. There are 20 “canonical” amino acids in eukaryotes [28]. Each amino acid exhibits distinct chemical properties (such as polarity and hydrophobicity) and also physical properties (such as mass), giving the 3D structured protein its character and behavior. The roles of proteins are many and varied. For instance, the protein **actin** lends structural integrity to cells. Enzymes are a special class of proteins that catalyze chemical reactions. Signaling proteins like **Ras** act as messengers that amplify and distribute signals from a stimuli.

Given the significance of proteins, this begs an important question: what directs their construction and regulation? *Genetic information* is the

CHAPTER 2. BACKGROUND

information required for construction of proteins. The central dogma of molecular biology [29] states that genetic information flows from deoxyribonucleic acid (DNA) to oxyribonucleic acid (RNA) to protein (Figure 2.1). Essentially, the DNA (a sequential chain of polymers called nucleotides) serves as blueprint for the construction of proteins. The sequence of nucleotides in DNA encodes the necessary information for protein construction, which is then transcribed into RNA before translated into proteins. Regions of the DNA that *directly* encode the construction of proteins are called *genes*. Beyond serving as blueprint for protein construction, DNA and RNA also encode information that guides regulation of proteins, for instance:

- amount of proteins produced (expression level)
- signals to start or stop production (gene activation or suppression)
- signals to modify proteins, affecting their behavior and interaction (protein modification)

2.2 Protein-protein Interactions (PPI)

Protein, DNA, RNA and other biological molecules do not work in isolation; they cooperate with other proteins to perform a particular biological activity. Two molecules that cooperate to perform a particular function are said to be *interacting*. It is the combination of these molecules and their interactions, and not the molecules alone, that characterize the mechanisms of a biological process. We wish to note that although the rest of the dissertation largely focuses on proteins, the concepts that we will discuss may extend to other molecules. Genes, DNAs, RNAs and other entities are also major drivers of a biological process. Interactions are typically grouped by their molecule types:

CHAPTER 2. BACKGROUND

Figure 2.2: Stable vs. Transient Interactions. (Part of Image Taken from <http://herkules.oulu.fi/>)

- **Protein-protein interactions** – cooperation between proteins to drive biological processes
- **Gene regulatory interactions** – interplay of genetic information to regulate protein expression level
- **Metabolic interactions** – cooperation between enzyme proteins to convert a substrate molecule into product molecule through several catalysis reactions

In this dissertation, major focus is placed on the class of protein-protein interactions, although most of the concepts covered apply to other classes of interactions as well.

Protein-protein interactions can be *stable* or *transient* [30]. In *stable* protein-protein interactions, a group of proteins form permanent protein-protein interactions to perform a biological role. A group of such stably interacting proteins is called a *protein complex*. An example of protein complexes is the **V-ATPase** (Figure 2.2(a)). Multiple protein subunits combine to form the **V-ATPase** enzyme that transports protons across membranes [31]. In *transient* protein-protein interactions, two proteins associate with each other briefly to perform biological activity before disassociating.

CHAPTER 2. BACKGROUND

Figure 2.3: Yeast-two-hybrid to detect protein-protein interactions. (Taken from The Science Creative Quarterly)

These interactions regulate a significant portion of biological processes. The interactions occur when a region of one protein complements the region of another, forming non-covalent bonds like hydrogen bonds, Van der Waals forces and hydrophobic bondings. A common surface region is the **leucine zipper** [32], a 3D structural motif in proteins with hydrophobic regions that allow two proteins with complementing zipper motifs to “zip” together. Typically, transient interactions only occur under conditions that promote their interaction, for instance the phosphorylation state of the proteins involved, the protein conformation state or their localization. Figure 2.2(b) shows transient interaction between UBI4 and PEX12; physical interaction occurs only during ubiquitination.

2.3 Methods to Analyze Protein-protein Interactions

Given the importance of protein-protein interactions in characterizing the mechanisms of a biological process, biologists have developed a range of experimental methods to detect and predict interactions between proteins. We describe several pertinent ones below.

2.3.1 Yeast Two-Hybrid (Y2H)

The yeast-two-hybrid method relies on activating the transcription of a *reporter gene* to detect interaction between two proteins [33]. Reporter genes are typically genes with easily observable phenotype. Figure 2.3 summarizes the concept behind Y2H. In Y2H, biologists engineer the two tested proteins such that when these two proteins interact, transcription of the reporter gene is activated, and thus, if the reporter gene phenotype is sufficiently expressed, one can deduce that the two proteins interact. To this end, Y2H uses two types of protein domains: the DNA-binding domain (BD) and the activation domain (AD). The BD and AD domains must be brought together proximally to bind and form a transcription activator, which then activates reporter gene transcription. Given two proteins, the BD domain is fused to one protein (called the bait) and the AD domain is fused to the remaining protein (called the prey). If these two proteins interact, the two domains are brought together proximally and activates reporter gene transcription. Commonly used reporter genes (and their promoter) include *HIS3*, *URA3* and *lacZ*. For example, the *lacZ* reporter gene when activated causes the yeast cell to express β -galactosidase, which can be detected by the formation of blue colored yeast colonies. A strong advantage of this method is its scalability and Y2H can easily be used to screen thousands of proteins for interactions, giving rise to high-throughput experiment technologies.

2.3.2 Tandem Affinity Purification (TAP)

The tandem affinity purification method identifies protein-protein interaction by incorporating a TAP tag to the target protein, followed by fishing for other proteins that interact with the tagged protein [35]. Figure 2.4 illustrates the TAP method. The TAP tag comprises two Immunoglobulin G (IgG) binding domains and a Calmodulin-binding peptide (CBP). In TAP,

CHAPTER 2. BACKGROUND

Figure 2.4: Tandem Affinity Purification. (Taken from [34])

the biologist engineers a fusion protein by fusing the TAP tag to the target protein. Next, the fusion protein, together with any other proteins attached to it, is isolated using beads coated with IgG. The biologist then applies the TEV cleavage enzyme to cleave the TAP tag from fusion, leaving behind the target protein plus the CBP domain bounded to the bead. A second isolation step is then applied using Calmodulin-coated beads. Here, the biologist obtains the final product of target protein, CBP and attached proteins that are interacting with the target protein. Finally, the end products are analyzed via mass spectrometry or SDS-PAGE [36]. The two step purification process minimizes the amount of contaminants obtained.

2.3.3 Bimolecular Fluorescence Complementation (BIFC)

Bimolecular fluorescence complementation is another protein-protein interaction screening strategy that relies on a reporter protein [37]. In this

method, the reporter protein is fluorescent, allowing it to be easily detected and located using tools such as flow cytometry. A reporter protein, the yellow fluorescent protein (YFP) for instance, is designed as two complementary fragments (YN and YC). Given two candidate proteins, the biologist separately fuses each fragment to the candidate proteins. When these two proteins interact, the two fragments will be brought to close proximity, encouraging them to re-attach and re-assemble into the YFP reporter protein. The fluorescent reporter protein can then be screened through a variety of techniques including flow cytometry.

2.3.4 Noise in High-throughput Screening Methods

Rapid high-throughput protein-protein interaction screening methods, however, suffer from significant noise and coverage issues. For instance, the *false negative* rate, defined as the probability of interacting protein detected as negative, could be as high as 70-90 percent with Y2H data [15]. This imply that there is a significant *coverage* gap (coverage here refers to the ratio between the number detected interactions and the number of actual interactions in the network). Moreover, high-throughput protein-protein interaction screening methods also suffer from relatively high *false positives* [38], which is defined as the probability of non-interacting protein detected as positive.

2.4 Protein-protein Interaction Databases

Advancements in protein-protein interaction screening methods has enabled the capability of generating large scale interaction data. Therefore, it is important to catalog and store these datasets to allow rapid and convenient access. We discuss several public databases that catalog key protein-protein interaction datasets. Table 2.1 lists several well known knowledge-bases with significant protein-protein interaction datasets. The

Table 2.1: Selected Protein-protein Interaction Databases.

Database	Reference
Human Protein Reference Database (HPRD)	[39]
Biological General Repository for Interaction Datasets (BioGRID)	[40]
Database of Interacting Proteins (DIP)	[41]
Kyoto Encyclopedia of Genes and Genomes (KEGG)	[42]
Biomolecular Interaction Network Database (BIND)	[43]
The MIPS Mammalian Protein-Protein Database	[44]
STRING: functional protein association networks	[45]
REACTOME	[46]
IntAct	[13]
BioCyc	[47]
BioCarta Pathways	[48]
PHOSIDA	[49]
Phospho-ELM	[50]
DOMINE: a database of protein domain interactions	[51]

STRING database [45] hosts a large collection of predicted and known protein-protein interactions. In addition, the STRING database links key information about the gene that codes for the interactor proteins, including their DNA sequence, biological annotations, co-occurrence, and co-expression data. The Kyoto Encyclopedia of Genes and Genomes (KEGG) database [42] is a resource of manually curated pathway datasets. The KEGG database is especially notable for its large collection of metabolic pathways for bacterial microbes. Important signaling pathways for a variety of organisms are also hosted in the KEGG database. The REACTOME database [46] hosts detailed biological pathways specifically for the human species. As is the KEGG database, pathways in the REACTOME database are manually curated and handcrafted. The IntAct database [13] stores a large amount of protein-protein interaction datasets submitted by individual labs. The datasets can range for a several protein-protein interactions per dataset to several hundred thousands of interactions per dataset. The Munich Information Center for Protein Sequences (MIPS) database [44] is

noted for its repository of protein complexes. Other significant databases hosting protein-protein interaction datasets are the Human Protein Reference Database (HPRD) [39], Biological General Repository for Interaction Datasets (BioGRID) [40], Database of Interacting Proteins (DIP) [41].

Apart from general protein-protein interaction resources, several web resources hosts context-specific datasets that focuses on a particular biological topic of interest. For example, the PHOSPIDA [49] and Phospho-ELM [50] knowledge-bases contain protein phosphorylation sites information, which can be used to deduce their interacting partners. DOMINE [51] is a database of protein domain-domain interactions. Apart from molecular function specific datasets, disease specific datasets are also abundant. The IntAct database contains a number of disease-related protein-protein interaction datasets that include Alzheimer's, cancer and cerebellar ataxia.

2.5 Annotating the Roles of Proteins and Their Interactions

With the growth of biological literature on the roles of proteins, groups of proteins, as well as their interactions, the need to annotate these information in a structured manner becomes pertinent. The Gene Ontology (GO) [25] is developed as a standard for providing a structured *ontology* describing attributes of genes and gene products (including proteins). An ontology is a set of controlled concepts (*GO terms*) and their relationships that models the domain. In Gene Ontology, the concepts describe the roles of the genes and their products, while the concept relationships connect the various concepts in Gene Ontology. For example, the **activation of protein kinase activity** concept can be used to annotate the MAPK protein, giving it that particular function. Now the concept relationships in GO may provide additional inferences to this concept. If suppose Gene Ontology states that **activation of protein kinase activity** is

CHAPTER 2. BACKGROUND

a type of **regulation of protein phosphorylation**, then one can reason that MAPK protein also has the attribute of **regulation of protein phosphorylation**.

The role of Gene Ontology as controlled vocabulary also resolves ambiguity in word descriptions. Functional descriptors that describe the role and function of proteins in the literature can be ambiguous, redundant and domain specific [52]. For instance, the gene names **CDC28**, **Cdc28p** or **cdc-28** all refer to the same biological entity. With a controlled vocabulary, computation methods can infer functional roles of proteins in a consistent manner.

Gene Ontology Annotation (GOA) database [53] stores associations of genes and proteins to GO terms. GO term annotation consists of manual and automatic approaches. In manual annotation, a domain expert or curator who is aware of the functional description of the gene or protein annotate that protein with the relevant GO terms. The automatic approach, on the other hand, predicts and infers the GO terms relevant of the protein via a multitude of machine learning techniques including literature mining and graph-based inferencing tools. The Online Mendelian Inheritance in Man (OMIM) database [54] supplies important annotations regarding diseases associated with human proteins.

2.5.1 The Structure of Gene Ontology

The Gene Ontology is modeled as a directed acyclic graph (DAG) and is divided into 3 major domains: *biological process*, *cellular component* and *molecular function*. The total number of GO terms in the GO DAG exceeds 30000.

The **biological process** domain contains GO terms describing the functional processes in cells, tissues, organs and organisms that proteins may take part in. The Gene Ontology defined a biological process as “a recognized series of events or molecular functions” with a defined beginning

CHAPTER 2. BACKGROUND

Figure 2.5: Subset of the Gene Ontology Directed Acyclic Graph.

and end. A biological process GO term may describe the process itself, or it may describe an encompassing process that is made up of subprocesses. For instance, the biological process term **apoptosis** describes cell apoptosis pathways in the cell. Thus, if the p53 protein is annotated with **apoptosis** GO term, then one can infer that p53 protein participates in cell apoptosis. The GO term **cell cycle** may describe the cell cycle process which itself is made up of several subprocesses, such as M-phase cell cycle and G-phase cell cycle subprocesses. In Gene Ontology, a process term may be connected to its parent via **is_a** and **part_of** relationships; the former describes that the process is an instance of the parent process, while the latter describes that the process is only a part of the parent process.

The **cellular component** domain contains GO terms describing the parts of the cell and its extracellular environment. Cellular components may be anatomical structures or macromolecular complexes. In Gene Ontology, a protein annotated with a cellular component GO term is said to be *located in* or is a *subcomponent of* the component described by the term. For example, the GO term **mitochondrial ribosome** describes the

CHAPTER 2. BACKGROUND

mitochondrial ribosomal components. Proteins like ribosomal protein L41 may be annotated with such GO terms.

Finally, the **molecular process** contains GO terms pertaining to an elemental activity of a protein. Activities here include any function performed by the proteins like catalysis, binding, phosphorylation, and other enzymatic roles. For example, the GO term **phosphorylation** describes the molecular activity that a protein may perform, which in this case is phosphorylation activity. A protein may be annotated with multiple activities. This is because proteins itself may participate in multiple functions. Protein kinases like PKC are known to have such capabilities and could be annotated with this term.

Figure 2.5 depicts a part of the GO DAG. Formally, the Gene Ontology for each domain is modeled as a directed acyclic graph $D = (V_{go}, E_{go})$ where V_{go} denotes the set of GO terms and E_{go} – the set of pair relationships between GO terms in V_{go} – denotes the set of GO term relationships. Here, an edge $(v_1, v_2) \in E$ represents a parent-child connection between two GO terms $v_1 \in V_{go}$ and $v_2 \in V_{go}$. The ordered set $\Delta = \langle \Delta_1, \Delta_2, \dots, \Delta_d \rangle$ is a topological sort of D . Each Δ_i represents a single GO term. We assume that a protein node $v \in V_i$ is annotated with a set of GO terms $D_v \subset \Delta$. The indicator function of terms annotated in node v is given by $I_{\{x \in D_v\}} : \Delta \rightarrow \{0, 1\}$ such that $I_{\{x \in D_v\}} = 1$ if $x \in D_v$ and 0 if otherwise.

The root node absorbs all GO terms of its descendants, *i.e.*, each descendant GO term is a or is part of the root node. The root nodes of biological process, cellular component and molecular function domains are **biological process**, **cellular component** and **molecular function**, respectively. As the GO DAG branches from the root node, the specificity of the functional description increases. Thus, one can utilize GO DAG and its associated annotations to group proteins by their function or parts in a hierarchical, nested manner. For example, in Figure 2.5, if proteins **MAPK**,

MAPKK, and MAPKKK are annotated with the `intracellular signaling process` term, then these proteins are also part of the `signal transduction`, `cell communication`, `cellular process` and `biological process`.

Gene Ontology and its annotations has been applied to a large number of bioinformatics approaches [55]. A pertinent usage is in gene expression analysis studies [56]. Typically, groups of genes which are either significantly up-regulated or de-regulated are identified using techniques such as gene clustering and enrichment analysis [57]. Then, the GO annotations are utilized to identify the over-expressed functional roles of these groups of proteins. An example of algorithms of such nature is the `MAPPFinder` [58], which looks for genes that are significantly deregulated using the GO annotations.

2.6 Summary

This chapter can be summarized as follows:

- Proteins, DNAs, RNAs and other biological molecules work in tandem to regulate biological processes. Cooperating molecules that perform a particular function are said to be interacting, and their interactions can be either transient or stable.
- A range of experimental methods have been developed to detect and predict interactions between proteins in a high-throughput manner. Among them are `Y2H`, `TAP` and `BIFC`.
- Advancement in protein-protein interaction screening methods has led to large scale interaction datasets. Several public databases serve as important repositories of such datasets, including `STRING`, `KEGG` and `REACTOME`.
- The Gene Ontology (GO) is developed as a standard for providing a structured ontology describing attributes of genes and gene prod-

CHAPTER 2. BACKGROUND

ucts (including proteins). Gene Ontology Annotation (GOA) database stores associations of genes and proteins to GO terms. GO term annotations are useful as functional descriptions of a gene or protein.

Chapter 3

Related Work

This chapter presents an overview of key research work related to this dissertation. We begin by discussing work related to analyzing the functional organization *within* a PPI network by *graph clustering*. Then, we discuss work related to the analysis of functional relationships *between* PPI networks via *network alignment*.

3.1 Graph Clustering of PPI Networks

Graph clustering methods aim to answer to following: given a PPI network, what are the “significant” *modules*¹ within it? Can one decompose the PPI network into an organization of modules? As discussed earlier, functional roles of a group of proteins and their interactions can be further organized into complexes, pathways and biological processes. Many binary protein-protein interactions are the underlying participants of biological processes and pathways. It is therefore relevant to reconstruct the pathway and processes of a biological system via PPI datasets. The typical approach is to manually identify these functional component via literature survey and manual analysis. This approach, however, scales poorly with the rapid growth of experimental data. Graph clustering algorithms could be potential solution to automatic discovery of such complexes and processes via module detection.

¹as one shall see later, modules are typically defined as densely connected subgraphs

Figure 3.1: An example of a PPI network modeled as graph.

A protein-protein interaction (PPI) network can be modeled as a graph $G = (V, E, \omega)$ that contains a set of vertices V representing proteins and a set of edges E representing interactions. The function ω assigns each interaction $e \in E$ a weight that represents its interaction strength or confidence. A biological pathway or module is a collection of molecular interactions that work together to achieve a particular functional objective in a biological process. An example of such pathway is the MAPK signaling network, a collection of interacting proteins that act as messengers to amplify and distribute signals from stimuli to intended destinations. Thus, one can model these functional pathways and modules as subgraphs of G that share a particular functional objective. Figure 3.1 shows an example of such a subgraph. We model all PPI networks as undirected graphs. We define a *functional module* (or functional cluster) as a graph $C_i = (V_C^i, E_C^i)$, where C is the vertex-induced subgraph of G by the set of proteins V_C^i . Suppose $C = \{C_1, C_2, \dots, C_d\}$ is the set of all functional modules in G . In general, there exists $C_i, C_j \in C$ s.t. $V_C^i \cap V_C^j \neq \emptyset$ (i.e., the functional modules in G are not “disjoint”; instead, they may overlap). Functional modules may

Figure 3.2: An example of a clustering of a PPI network.

be pathways, complexes or other biological processes.

Graph clustering (or network clustering) algorithms are methods that analyze the topological properties of a PPI network to identify and predict potential functional modules. In graph clustering, a PPI network G is analyzed to identify graph *clusters*, subgraphs of G that exhibit significant *clustering properties*. For instance, a graph clustering algorithm may seek for graph clusters with clustering property of having many edges within the graph clusters and few edges between the graph clusters. The graph clusters are then predicted to be functional modules on the basis of their clustering properties. Figure 3.2 illustrates a clustering of a PPI network, showing the **RSC complex** and **SWI/SNF complex** proteins grouped into distinct clusters. The shape of a protein node indicates an assignment of the protein into either **RSC complex** cluster (circle nodes) or **SWI/SNF complex** cluster (square nodes). Note that the number of protein-protein interaction connections within a cluster is much higher than those which go between clusters. This chapter discusses the key works in graph clustering of PPI networks. We begin by defining the graph clustering problem. Then, we provide an overview of key graph clustering approaches. Finally, we discuss several representative methods, highlighting their respective strengths and

weaknesses.

3.1.1 Problem Definition of PPI Network Clustering

Formally, a clustering of G aims to partition V into a set of clusters $C = \{C_1, C_2, \dots, C_d\}$ such that clustering property objective function $f : C \rightarrow \mathbb{R}$ is maximized, *i.e.*, the algorithm identifies $\operatorname{argmax}_C f(C)$. Typically, $f(\cdot)$ rewards clusters that exhibit many within-cluster edges and few between-cluster edges.

As discussed in Chapter 2, pathways and processes in a system do not work in isolation; instead they work in tandem to coordinate the functionalities of the cell. Moreover, it is possible to organize these processes into even higher-order processes, forming a hierarchy of biological processes.

The *density* of a subgraph or cluster $C = (V_C, E_C)$ is given by the ratio of the number of edges in C over the maximum number of possible edges in C :

$$\operatorname{Density}(C) = \frac{2|E_C|}{|V_C| \times (|V_C| - 1)} \quad (3.1)$$

As one shall see later, many clustering algorithms utilize cluster density to identify *modules* – subgraphs with density that exceed a specific *density threshold*. The density of a cluster can be weighted. In that case, the *weighted density* of C is given by:

$$\operatorname{WeightedDensity}(C) = \frac{2 \sum_{e \in E_C} \omega(e)}{|V_C| \times (|V_C| - 1)} \quad (3.2)$$

3.1.2 Overview of PPI Network Clustering

Table 3.1 presents an overview of work in network clustering of PPI networks. We organized the related work by clustering approach. Here, we identified several major categories of clustering approaches:

- Heuristic-based Algorithms
- Complete Enumeration Algorithms

CHAPTER 3. RELATED WORK

- Random Walks and Message Passing Algorithms
- Flow Based Algorithms
- Graph-cut and Hierarchical Clustering Algorithms
- Other Algorithms

Heuristic-based algorithms utilize a greedy heuristic to identify clusters. *Complete Enumeration* algorithms enumerate all possible subgraphs with density exceeding a specified threshold. *Random walk-based* methods model the graph clustering problem as identifying the stationary distribution of a random walk model. In *Flow-based* algorithms, clustering is achieved by a series of flow “expansions” and “contractions” to identify clusters with high intra-cluster flows and weak inter-cluster flows. On the other hand, *Graph-cut and Hierarchical Clustering* algorithms utilize graph theoretic properties to identify clusters. Finally, *Other* algorithms contain algorithms that do not belong to any of the above categories. The rest of this section discusses the features that distinguish these methods. We shall also outline the algorithms driving several representative methods.

3.1.3 Heuristics-based Algorithms

Heuristics-based algorithms find densely network regions by searching heuristically for potential cluster regions using an iterative greedy seed and extend strategy. **MCODE** identifies densely connected clusters based on a seed and extend heuristic. In this approach, a weighing scheme is introduced that searches for dense local neighborhood regions. Given a PPI network $G = (V, E)$, the **MCODE** algorithm is as follows:

1) Stage 1: Vertex weighting. Let the 1-neighborhood of a protein $u \in V$ be the subgraph $N(u) = (V_u, E_u)$ induced by the vertex u and its immediate neighborhood. For each $v \in V$, **MCODE** identifies the highest

Table 3.1: Overview of PPI network clustering techniques.

Heuristics Algorithms	
MCODE	[17]
Restricted Neighborhood Search Clustering Algorithm (RNSC)	[59]
ClusterONE	[60]
SPICi	[61]
Complete Enumeration Algorithms	
Clique Detection	[62]
CFinder	[63]
Clustering-based on Maximal Cliques (CMC)	[64]
Dense Module Enumeration (DME)	[65]
Random Walks and Message Passing Algorithms	
Affinity Propagation (AP)	[66]
Nibble	[67]
RRW	[68]
Flow Based Algorithms	
TRIBE-MCL	[18]
Multi-level Regularized-MCL (MLR-MCL)	[69]
Soft R-MCL (SR-MCL)	[70]
Graph-cut and Hierarchical Clustering Algorithms	
Metis	[71]
Tree-Snipping	[72]
VI-Cut	[73]
NeMo	[74]
HAC-ML	[75]
DBHT	[76]
Other Algorithms	
Ensemble clustering framework	[77]
MOD-ILP	[78]

CHAPTER 3. RELATED WORK

k – *core* number of the 1-neighborhood of v . A k -*core* is a graph $G_k = (V_k, E_k)$ such that for all $v_k \in V_k$, $d(v_k) \leq k$ where $d(v_k)$ is the degree of v_k . The highest k -*core* number of the 1-neighborhood of v is then defined as the largest number k such that the subgraph is k -*core*. Furthermore, MCODE determines for the 1-neighborhood of v its *density*, given by:

$$\sigma(N(v)) = \frac{2|E_v|}{|V|(|V| + 1)} \quad (3.3)$$

Given $\sigma(N(v))$ and the highest k -*core* number associated with v (denoted by k), the weight of vertex v is assigned as $w(v) = k\sigma(N(v))$. This weight boosts neighborhoods with high density and also rewards clusters that exhibit strong “clique-like” structure.

2) Stage 2: Molecular complex prediction. Equipped with the vertex weights, MCODE finds complexes in a greedy *seed and extend* manner. It starts with the highest weighted vertices as seeds. Following that, the seed is expanded by including neighbors having weight exceeding a user-specified threshold. This parameter is known as the *vertex weight percentage (VMP)* parameter. The expansion stops once no vertices satisfies the threshold parameter and the complex can no longer be grown. The algorithm then proceeds with the next remaining highest weight vertex as new seed.

3) Stage 3: Post-processing. In this step, complexes are pruned when they do not have at least a 2-*core*. A ‘fluff’ operation is also introduced to increase the size of complexes. The resulting complexes are then ranked and scored based on their clustering densities.

Figure 3.3 illustrates the algorithm with a toy PPI network. The 1-neighborhood of a protein v in the network is highlighted. Observe that the k -core of the 1-neighborhood is 3, because the lowest degree among the proteins in it is 3. Additionally, the density of the neighborhood is 0.47, thus $w(v) = 3 \times 0.47$. The rightmost subfigure shows the seed expansion process after selecting the highest weighted seed vertex. The cluster is grown one node at a time, until no vertices can satisfy the VMP parameter.

CHAPTER 3. RELATED WORK

Figure 3.3: Illustration of the MCODE algorithm.

Another popular greedy search algorithm is the *Restricted Neighborhood Search Clustering* Algorithm (RNSC) [59] method. This method is a cost-based partitioning algorithm that first constructs a random clustering, then iteratively moves nodes from one cluster to another in a randomized fashion to improve a specified cost function.

Given the growth of PPI data, there is a need to handle large PPI networks in a scalable manner. The SPICi method [61] is introduced to handle the computation complexity of clustering large networks. SPICi grows a module, one at a time, from a seed comprising a pair of proteins. To identify a seed, it identifies the node with the highest sum of edge weights connected to the node (*support*) followed by a binned selection process that identifies the best pair of nodes as seed. Following seed selection, a module is formed from the seed by greedily adding an adjacent (unclustered) node with the highest support score. Nodes are added so long the overall module density and/or remaining highest support exceeds their respective density threshold. Once a module is identified, the subgraph is removed from the PPI network and the process continues to identify remaining modules. The SPICi method has a time complexity $O(V \log V + E)$ and a space complexity $O(E)$, allowing it scale to large PPI networks.

The ClusterONE method [60] detects overlapping clusters in a PPI net-

Figure 3.4: Overlapping versus Non-overlapping Clustering.

work using a greedy seed and extend heuristic. The approach starts with a single protein node and then greedily adds or removes nodes to find a new group of proteins that show greatest improvement in *cluster cohesiveness*. Following that, the extent of overlap among the candidate clusters is evaluated, and cluster merging is performed selectively. This approach demonstrated clustering superiority over a variety of methods, including popular methods such as RRW, RNSC, AP, and MCL [60]. An advantage of ClusterONE is the ability to not just find overlapping clusters, but also clusters that may be contained in another cluster.

Figure 3.4 highlights the difference between a clustering algorithm that constructs non-overlapping clusters versus one that constructs overlapping clusters. Observe that in overlapping clusters, different modules are allowed to share the same nodes in the network. Non-overlapping clustering, on the other hand, constructs modules that share no nodes.

All heuristic-based methods, however, have strong likelihood of converging to a local minimum. On the other hand, these methods generally allow identification of overlapping clusters that better reflect the moonlighting property of proteins (as described in Chapter 1). These methods also rely purely on graph structure to identify functional clusters. Apart from that, they generate clusterings that are *partial coverage*. A partial coverage method finds a set of locally dense subgraphs of G , and this set of

Figure 3.5: Full versus Partial Coverage Clustering.

dense subgraphs does not cover the entire graph. Among partial coverage methods, it is generally advantageous to have one which achieves a high coverage score. The advantages of partial coverage methods are similar to those enjoyed by local sequence alignment methods. Clusters obtained are often significantly dense subgraphs, and irrelevant clusters that do not meet the objective function are automatically ignored. Hence, clusterings obtained using partial coverage methods are more amenable to human interpretation.

Figure 3.5 shows the contrast between a clustering algorithm that is *full coverage* versus *partial coverage*. Observe that in full coverage clustering, every node in the network belongs to at least one cluster. This is not the case with partial coverage clustering, which does not guarantee that every node is clustered.

3.1.4 Complete Enumeration Algorithms

The CFinder method [63] identifies a set of *k-clique modules* in a PPI network. *k*-cliques correspond to *k* node complete subgraphs of *G* with a maximum density of 1. The algorithm first extracts all complete subgraphs (cliques) of the PPI network. From these set of cliques, a *clique-clique overlap matrix* is built. This matrix is then used to identify *k*-clique com-

CHAPTER 3. RELATED WORK

Figure 3.6: An illustration of CFinder algorithm. (Image taken from [63])

munities by setting to 0 all diagonal entries in the matrix with value less than k and all off-diagonal entries with value less than $k - 1$. Following that, connected components in the matrix are identified as a k -clique community. The clique search method is NP-complete. Figure 3.6 illustrates the clique-clique overlap matrix constructed on a toy PPI network. Here $k = 4$ and each row/column corresponds to a k -clique in the network. Observe the setting of the values of the matrix based on the extent of overlap among the cliques.

Extending the idea of clique enumeration to more general graphs, the DME method [65] enumerates *all* subgraphs that satisfy a minimum density threshold (*i.e.*, modules). This approach models the search process by a tree. The root of the tree is an empty set, while any children node in the tree is a superset of that node's parent, and for any path from the root to the leaf, the *module density* is monotonically decreasing. Here, the module density refers to the average pairwise weights of the edges in a module.

CHAPTER 3. RELATED WORK

By enforcing density guarantee in the search tree, the **DME** method prunes all unnecessary explorations during the search process while exhaustively enumerating all sufficiently dense modules. Although the **DME** method can find all dense modules, it is computationally expensive. As such, it is applicable to only relatively small networks.

The Clustering-based on Maximal Cliques (**CMC**) [64] method first generates all maximal cliques of a PPI network; this is followed by a series of steps that merges highly overlapping cliques. This approach yields a set of densely interacting cliques that are fairly non-redundant. It is also shown to be less sensitive to parameters compared to **MCL**. One weakness of **CMC**, however, is that it identifies only clusters that correspond to a clique topology.

A common theme among complete enumeration algorithms is exhaustive search. While the upside of exhaustive search is the ability to identify all relevant modules within a PPI network, the downside is the high computation cost involved. These algorithms are NP-complete. Therefore, they are limited to relatively small PPI networks.

3.1.5 Random Walks and Message Passing Algorithms

Another well known strategy to cluster PPI network is to model the graph as a random walk model, and then after performing a series of random walks, identify a set of clusters.

Nibble is an approach that relies on a modified random walk strategy [67]. It finds clusters with low *conductance* (to be defined later), which informally refers to clusters that have many edges within it and few edges going out of the clusters. A strong advantage of this method is its scalability – **Nibble** runs in nearly-linear time in the size of the cluster outputs.

The *conductance* of a set S is given by:

$$\Theta(S) = \frac{\sigma(S)}{\min(\text{vol}(S), 2m - \text{vol}(S))} \quad (3.4)$$

CHAPTER 3. RELATED WORK

where $\sigma(s) = |\{(u, v) | u \in S, v \notin S\}|$; $vol(S) = \sum_{v \in S} deg(v)$ and $m = |E|$. The volume of S $vol(S)$ is also written as $\mu(S)$. The *conductance* of G is then given by:

$$\Theta_G = \min_{S \subset V} \Theta(S) \quad (3.5)$$

Also, **Nibble** defines the following vectors defined on a vertex set S :

$$\chi_S(u) = \begin{cases} 1 & \text{for } u \in S \\ 0 & \text{otherwise} \end{cases} \quad (3.6)$$

$$\varphi_S(u) = \begin{cases} d(u)/mu(S) & \text{for } u \in S \\ 0 & \text{otherwise} \end{cases} \quad (3.7)$$

Given a graph G , **Nibble** first constructs its adjacency matrix A :

$$A(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E, u \neq v \\ k & \text{if } u = v \text{ and } u \text{ has } k \text{ self loops} \\ 0 & \text{otherwise} \end{cases} \quad (3.8)$$

The random walk matrix is defined as $M = (AD^{-1} + I)/2$ where D is the diagonal matrix of node degrees. The distribution of the random walk given the start seed v after t steps is then $p_t = M^t \chi_v$. **Nibble** also introduces the *truncation* operation on p_t given by:

$$[p]_\epsilon(u) = \begin{cases} p(u) & \text{if } p(u) \geq d(u)\epsilon \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

which truncates every $q_t(u)$ less than $d(u)\epsilon$ to zero.

Nibble runs in iterations. Starting at a seed vertex v , at each iteration, a random walk is performed followed by a truncation operation. After a few steps, the distribution of the truncated random walk $[p_t]_\epsilon$ can be used to identify a cut with low conductance. If a clustering with desirable clustering score occurs in one of the steps, the algorithm terminates and the clustering is set as output. Otherwise, the iterative procedure is continued until a predefined maximum number of steps is reached. If reached, no desirable clustering is obtained.

The Repeated Random Walk (RRW) method [68] clusters PPI networks using a random walk with restart methodology. An advantage of the RRW

CHAPTER 3. RELATED WORK

approach is its ability to admit overlapping clusters. First, **RRW** constructs the transition matrix P from $G = (V, E)$ and edge weight function $f : E \rightarrow \mathbb{R}$. Then, for each $v \in V$, it computes the stationary distribution vector associated with v as starting node, defined by:

$$m[v] = \alpha s + (1 - \alpha) P^T m[v] \quad (3.10)$$

where s is the start vector such that v is the starting node; and α is the *restart probability* parameter, which defines the probability that the walk restarts at the starting vector s . Additionally, the stationary distribution x_C of a set of proteins $C = \{v_1, v_2, \dots, c_k\}$ is given by $x_C = \sum_{v \in C} m[v]$.

The **RRW** algorithm then proceeds as follows: 1) For each $v \in V$, set $C = \{v\}$, and expand C by identifying proteins that exhibit strong transition probabilities from x_C and adding them to C . The expansion terminates if the next added protein score is below λ percentage of the previously added protein. 2) Given the set C for each $v \in V$, post-processing is performed to remove highly overlapping clusters.

An approach that is similar to random walk is the message-passing based *Affinity Propagation* (**AP**) method [66]. Affinity Propagation searches for *exemplars*, representative proteins of clusters. First, every protein $v \in V$ is flagged as exemplars. For every protein $i \in V$ and an exemplar $v \in V$, denote by $r(i, v)$ the *responsibility* of v given i . Intuitively, responsibility reflects how likely v is the exemplar of protein i . It is defined by:

$$r(i, v) = s(i, v) - \max_{u \neq v} \{a(i, u) + s(i, u)\} \quad (3.11)$$

The **AP** method also defines *availability* as follows:

$$a(i, v) = \min\{r(v, v) + \sum_{u \in \{u | u \neq v\}} r(u, v)\} \quad (3.12)$$

Initially, all availabilities are set to zero. Messages in form of availabilities and responsibilities are then passed among neighbors and exemplars until

CHAPTER 3. RELATED WORK

convergence is achieved. It has been shown that **AP** underperformed **MCL** in clustering of PPI networks [79].

While both **AP** and **Nibble** do not admit overlapping clusters, the **RRW** method is one random walk based method that allows overlapping clusters. Again, all the above are purely topology driven clustering methods. None of these methods utilize the wealth of annotation data to compute important functional clusters. Typically, random walk and message-passing based clustering methods construct a partitioning on the PPI network, implying that the clustering is full coverage.

3.1.6 Flow Based Algorithms

One of the most widely used graph clustering algorithm is Markov Clustering (**MCL**) [18]. This approach partitions a PPI network into subgraphs using a flow-based approach. Given a PPI network $G = (V, E)$ with a function $f : E \rightarrow \mathbb{R}$ that gives each pair of proteins their BLAST E-value scores, **MCL** first constructs a weight transition matrix given by:

$$W[i, j] = I((i, j))f(i, j) \quad (3.13)$$

where $I(e)$ is the indicator function such that $I(e) = 1$ if $e \in E$ and $I(e) = 0$ otherwise. Given the weight transition matrix, normalization is performed to obtain the column-wise transition probability matrix:

$$M[i, j] = \frac{W[i, j]}{\sum_x W[i, x]} \quad (3.14)$$

The **MCL** clustering algorithm simulates the convergence and expansion of flows by iteratively alternating the following two steps until convergence: 1) the *expansion operator* and 2) the *inflation operator*. In the expansion operator, the transition matrix M is raised to the power of m :

$$M_t[i, j] = (M_{t-1}[i, j])^m \quad (3.15)$$

Intuitively, this step can be thought of as transforming M_{t-1} to a transition probability matrix of all random walks over m steps. In the inflation

CHAPTER 3. RELATED WORK

operator, the matrix takes its Hadamard power over $r > 1$ followed by renormalization. This corresponds to an entry-wise power and normalization:

$$\Gamma_r M_t[i, j] = \frac{M_{t-1}[i, j]^r}{\sum_x M_{t-1}[i, x]^r} \quad (3.16)$$

Since entry-wise transition probabilities are raised to a power of $r > 1$, entries with high transition probabilities are favored (*i.e.*, inflated) while entries with low transition probabilities are suppressed, thus favoring densely connected regions.

The MCL approach, however, may generate clusterings with *imbalanced clusters*, where clusters may have significantly different sizes. The occurrence of singleton clusters is one side-effect of having imbalanced clusters. To this end, the MLR-MCL algorithm [69] is proposed to construct more balanced clusters by augmenting the MCL method.

The MCL approach is also a partitioning algorithm that constructs non-overlapping clusters. Another extension of MCL addresses the non-overlapping nature of MCL clusters by introducing a MCL-based clustering strategy that creates overlapping clusters. Here, the authors propose the SR-MCL method [70] that extends the MCL approach by iterative re-execution of the underlying MCL clustering while ensuring the clusters are different. A post-processing is then applied to remove uninformative, redundant clusters, and the final set of overlapping clusters is obtained.

All flow based clustering methods are full coverage. A limitation of these methods is their inability to utilize the rich information provided by annotations. These annotations can be used to guide the clustering process to identify clusters that are compatible with known information. We shall discuss in the following section several algorithms that utilize annotations.

3.1.7 Graph-cut and Hierarchical Clustering Algorithms

The **Metis** algorithm [71] is a graph-cut based partitioning algorithm that looks for a clustering that minimizes the edge cut between clusters, while at the same time ensuring all clusters are of roughly the same size. The Kernighan-Lin objective is utilized to identify such a clustering [80].

The **NeMo** algorithm [74] is unique for incorporating the notion of indirect connections. Rather than observing direct interactions between two proteins u and v , **NeMo** predicts the association between them based on the idea of shared neighbors – two proteins are deemed highly connected if they share most of their neighbors. Suppose one wishes to cluster the graph $G = (V, E)$. Given the premise of shared neighbors, **NeMo** computes for all protein pairs $(a, b) \in V^2$ the log odds score r_{ab} :

$$r_{ab} = \ln \frac{P(s_{ab}|\hat{\lambda})}{P(s_{ab}|\bar{\lambda})} \quad (3.17)$$

where $\bar{\lambda}$ is the Poisson parameter of s_{ab} under the null hypothesis (the null hypothesis is that the number of shared neighbors between a and b is from a random network); $\hat{\lambda}$ is the Poisson parameter of s_{ab} under the alternative hypothesis (the alternative hypothesis is that the number of shared neighbors between a and b is greater than expected from a random network); and s_{ab} is the observed number of shared neighbors between a and b .

Given the log odds scores r_{ab} , **NeMo** then proceeds to identify clusters using a hierarchical agglomerative clustering approach. Both *complete-linkage* and *single-linkage* strategy are considered. Node pairs are processed greedily based on their log odds scores. **NeMo** only groups pairs having expected number of shared neighbors greater than by chance. To illustrate the concept of shared neighbors in **NeMo**, Figure 3.7 shows the shared neighbors between two nodes (indicated by triangles). Here, the two nodes depicted are sharing three neighbors. The greater the fraction of shared neighbors

CHAPTER 3. RELATED WORK

Figure 3.7: Illustration of shared neighbors and collapse procedure in NeMo algorithm

between two proteins a and b , the larger their r_{ab} score. A *collapse* procedure is also introduced to prune insignificant structures from the result. Given the hierarchical tree T , any internal node p having children m and n such that n is a leaf node and m is an internal node, **NeMo** collapses the edge between p and m . Figure 3.7 also shows the collapse procedure, where internal node m is collapsed.

Typically, a clustering is obtained from a hierarchical clustering tree by “cutting” the tree at a particular level. For instance, given a binary tree of five levels with 2^5 leaf nodes, a clustering with 4 clusters can be obtained by grouping the leaf nodes by their level 3 ancestors. The core idea of **Tree-Sniping** [72] is the following proposition: rather than cutting at a single level, snip the tree at selected edges at different levels. With the added flexibility, **Tree-Sniping** can pick and choose snips that maximizes the compatibility of the clusters with its constituent proteins’ annotation.

Let T be the hierarchical clustering tree obtained from a graph clustering of G . For each node $v \in T$ and snips l and k , define $\min\text{Mis}(v, l, k)$ as the minimal number of misclassified leaves (protein label not compatible with cluster label) when v is labeled as l and there are k snips in the subtree rooted at v . Also, define $\min\text{Num}(v, k) = \min_l \min\text{Mis}(v, l, k)$. Then for

CHAPTER 3. RELATED WORK

Figure 3.8: The VI-Cut algorithm illustrated with a toy example. (Image taken from [73])

each k and l , $\min\text{Mis}$ is defined as the following recursive function:

$$\min\text{Mis}(v, l, k) = \min \begin{cases} \min\text{Mis}(\text{left}, l, r) + \min\text{Mis}(\text{right}, l, k - r) & 0 \leq r \leq k \\ \min\text{Mis}(\text{left}, l, r) + \min\text{Num}(\text{right}, k - r - 1) & 0 \leq r \leq k - 1 \\ \min\text{Num}(\text{left}, r) + \min\text{Mis}(\text{right}, l, k - r - 1) & 0 \leq r \leq k - 1 \\ \min\text{Num}(\text{left}, r) + \min\text{Num}(\text{right}, k - r - 2) & 0 \leq r \leq k - 2 \end{cases} \quad (3.18)$$

To this end, the recursive function is solved via dynamic programming method that traverses the tree from bottom-up. The **Tree-Sniping** method does not scale well with the dimensionality of the labels. Given the high dimensionality of GO annotations, the misclassified labels will dominate the scores and mask the relatively fewer conserved labels. Experiments described in [72] are applied on one to three labeled genes. For instance, only three GO terms are manually selected for the clustering experiments. **Tree-Sniping** also performs best when the annotations largely form a partition. Consider for example the biological process GO term and a PPI network labeled with biological process related GO annotations. With **Tree-Sniping**, no proteins are considered misclassified under this overarching GO term, and as such the single large cluster associated with biological process is considered the optimal solution.

In similar fashion to **Tree-Sniping**, **VI-Cut** [73] is a tree-sniping approach that relies on the *variational information* metric to generate clusters that “matches” with MIPS annotations of the proteins. Suppose one wishes to cluster the graph $G = (V, E)$ with annotations D . The **VI-Cut** algorithm first obtains a hierarchical clustering tree T on G as input (this tree can be

CHAPTER 3. RELATED WORK

obtained using any hierarchical clustering techniques). Given T , **VI-Cut** determines the cuts of the tree such that the variational information score is minimized. Let C_K be a clustering on the graph G and D be a clustering of V by their MIPS annotation of the proteins. The variational information distance between two clusterings is defined as:

$$VI(C_K, D) = H(C_K) + H(D) - 2I(C_K, D) \quad (3.19)$$

where $H(C_K)$ are the entropies of C_K and D , respectively; and $I(C_K, D)$ is the mutual information between C_K and D . Intuitively, the entropies measure the amount of uncertainty or information in each clustering, while $I(C_K, D)$ measures how much information is shared among C_K and D . Thus, $VI(C_K, D)$ measures how much uncertainties are encoded in C_K given that D is known. In **VI-Cut**, this measure is minimized so that the graph clusters generated agree well with the proteins with GO annotations in D . The authors show that $VI(C_K, D)$ is equivalent to:

$$VI(C_K, D) = \sum_{x \in C_K} q(x) \quad (3.20)$$

where $q(x) = p(x) \log p(x) - 2 \sum_{d \in D} p(x, d) \log p(x, d)$. Here, x denotes a node in the hierarchical decomposition tree and $p(x)$ is the probability that a protein with an annotation belongs to x . Also, $p(x, d)$ is the joint probability that protein with an annotation belongs to x and has annotation d . Any cut that is made should minimize $VI(C_K, D)$. **VI-Cut** computes this via dynamic programming by solving the following recursive problem:

$$CutDist(x) = \min \left\{ \begin{array}{l} q(x) \\ \sum_{y \in \text{Children}(x)} CutDist(y) \end{array} \right. \quad (3.21)$$

Figure 3.8 demonstrates the **VI-Cut** algorithm. Annotations of the proteins are indicated by symbols. Starting from the root of the hierarchical clustering tree (cluster of all proteins), the cut is made such that the match between the clusterings and the symbols are as close as possible. Figure 3.8A shows the first level cut and Figure 3.8B shows the second

level cut. The **VI-Cut** approach scales poorly with the dimensionality of the annotations per protein, which makes it unsuitable for application of graph clustering on the richly attributed GO annotation data.

DBHT [76] is a graph theoretic based clustering strategy that searches for topological embeddings in a PPI network. To this end, a *Planar Maximally Filtered Graph* is obtained that the authors demonstrate desirable properties for cluster extraction. The strength of DBHT lies in its deterministic nature and its ability to identify clusters without a-priori parameters.

A strength of hierarchical methods like **VI-Cut** and **NeMo** is the potential of constructing a hierarchy of clusterings. This allows analysis of clusterings at multiple levels of granularities. Most graph theoretic based methods, however, do not admit overlapping clusters, which more closely resemble real life biological complexes. Hierarchical clustering methods are full coverage, but methods based on maximal cliques and topological embeddings like DBHT are partial coverage.

3.1.8 Other Algorithms

In the ensemble clustering framework proposed by [77], a range of independent clusterings are obtained and combined to construct a single *consensus clustering*. The intuition of ensemble clustering is that the combined clustering may yield high confidence clustering even in the presence of noise. In this approach, Principal Component Analysis (**PCA**) is first performed to reduce the dimensionality of the ensemble clustering problem, leading to more scalable clustering analysis.

The **MOD-ILP** algorithm [78] casts the PPI network clustering as an Integer Linear Program (ILP) problem. In **MOD-ILP**, the *modularity* of a subgraph of G is proposed to measure the clustering objective score of a clustering. Given a PPI Network G and a clustering S , the modularity of

CHAPTER 3. RELATED WORK

S is defined as:

$$q(G, S) = \sum_{u,v \in V} \frac{I_{uv} - k_u k_v / 2|E|}{1 - x_{uv}} \quad (3.22)$$

where $I_{uv} = 1$ if $(u, v) \in E$ or 0 otherwise; k_u is the degree of u ; and $x_{uv} = 1$ if u and v belongs to the same cluster in S and 0 if otherwise. Note that the above objective function rewards groups of proteins with strong co-connections when they are placed in the same cluster.

The ILP problem is then posed as finding the clustering S that maximizes the modularity objective function:

$$\max \sum_{u,v \in V} \frac{I_{uv} - k_u k_v / 2|E|}{1 - x_{uv}} \quad (3.23)$$

$$s.t. \quad x_{uv} + x_{vw} \leq x_{uw} \quad \forall u, v, w \in V \quad (3.24)$$

$$x_{uv} \in \{0, 1\} \quad (3.25)$$

The first constraint enforces the transitive property of cluster membership, while the second constraint arises from the combinatorial nature of the problem. An interesting novelty of MOD-ILP is its ability to general an ensemble of clusterings, as opposed to other clustering methods that produce only a single clustering of G . The above formulation, however, only admits one possible clustering of the network. Suppose the first clustering obtained is S_0 . MOD-ILP iteratively generates a new clustering S_t where $t > 0$ from the past clustering S_{t-1} by imposing a “uniqueness” constraint, which forces the next set of results to be different:

$$S_{t-1} \cdot (1 - S_t) \leq d_{merge}^0 \quad (3.26)$$

$$(1 - S_{t-1}) \cdot S_t \leq d_{split}^0 \quad (3.27)$$

where d_{merge}^0 and d_{split}^0 are real-valued parameters that defines the degree of difference required. Using the above formulation, MOD-ILP constructs a set of clusterings that can be seen as an ensemble of near-optimal solutions, and these ensembles were utilized to study the robustness of the modularity landscape.

3.1.9 Detecting Structurally Loose Modules

Proteins in signalling pathways are structurally loose, but share important functions. An important limitation of existing PPI network methods is the emphasis on cluster density in their clustering objective function. MOD-ILP defines an objective function to maximize the structural modularity of clusters. The MCL-based approaches utilize a sequence of expansion and inflation steps that results in groups of densely connected regions. The MCODE heuristic screens for clique-like structures in a given PPI network. The ClusterONE, SPICi and RNSC methods similarly finds clusters satisfying high subgraph density/cohesiveness. CFinder and CMC enumerate clique structures, while DME enumerates all subgraphs exceeding a minimum density. The DBHT method extracts cluster structures by finding topological embeddings in the network, and NeMo aggregates nodes having many shared neighbors. The Nibble method utilizes the conductance objective function that is based on having many edges within clusters as opposed to going out of clusters. The random walk methods RRW and AP similarly find groups of densely connected subgraphs. Finally, graph-cut algorithms minimize the edge cut between clusters. In general, the common objective among these methods is to find dense network regions. These methods assume that a functional module (*e.g.*, protein complex) corresponds to a strongly connected subgraph. Only the VI-Cut and Tree-Sniping methods go beyond using dense structure to identify functional modules. However, both methods cannot support high-dimensional GO annotations.

3.1.10 Summary of PPI Network Clustering Algorithms

Table 3.2 summarizes the network clustering approaches by the following properties:

CHAPTER 3. RELATED WORK

Table 3.2: Summary of PPI network clustering techniques.

Algorithms		Overlapping	Scalability	Exhaustive	Full Coverage
MCODE	[17]	Yes	Medium	No	Low
RNSC	[59]	No	Medium	No	High
ClusterONE	[60]	Yes	Medium	No	No
SPICi	[61]	No	High	No	Yes
CFinder	[63]	Yes	Low	Yes	No
CMC	[64]	Yes	Low	No	Yes
DME	[65]	Yes	Low	Yes	No
AP	[66]	No	Medium	No	Yes
Nibble	[67]	No	High	No	Yes
RRW	[68]	Yes	Medium	No	Yes
TRIBE-MCL	[18]	No	Medium	No	Yes
MLR-MCL	[69]	No	Medium	No	Yes
SR-MCL	[70]	Yes	Medium	No	Yes
Metis	[71]	No	Medium	No	Yes
Tree-Snipping	[72]	No	Medium	No	Yes
VI-Cut	[73]	No	Medium	No	Yes
NeMo	[74]	No	Medium	No	No
HAC-ML	[75]	No	Medium	No	Yes
DBHT	[76]	No	Medium	No	Yes
Ensemble	[77]	No	Low	No	Yes
MOD-ILP	[78]	No	Medium	No	Yes

Algorithms	Annotation
MCODE	[17] No
RNSC	[59] No
ClusterONE	[60] No
SPICi	[61] No
CFinder	[63] No
CMC	[64] No
DME	[65] No
AP	[66] No
Nibble	[67] No
RRW	[68] No
TRIBE-MCL	[18] No
MLR-MCL	[69] No
SR-MCL	[70] No
Metis	[71] No
Tree-Snipping	[72] Yes
VI-Cut	[73] Yes
NeMo	[74] No
HAC-ML	[75] No
DBHT	[76] No
Ensemble	[77] No
MOD-ILP	[78] No

CHAPTER 3. RELATED WORK

- *Overlapping* column indicates whether the algorithm can identify overlapping modules or otherwise
- *Scalability* column qualitatively suggests the algorithm’s capability to scale to larger PPI networks
- *Exhaustive* column indicates whether the algorithm can identify all modules in the PPI network that satisfy the clustering criteria specified by the algorithm
- *Complete Coverage* column identifies algorithms that can form clusters on all nodes in the PPI network (*i.e.*, full coverage)
- *Annotation* column identifies algorithms that are annotation aware, that is, the clustering process can be guided by available annotation information encoded within PPI networks.

It is generally advantageous to have an algorithm that admits overlapping communities (while controlling redundancies), has high scalability, is exhaustive, admits clustering with complete coverage, and is annotation-aware. We note that no single algorithm enjoys all of the above strengths. Additionally, we note that the performance of these algorithms cannot be quantified with just these properties alone, as many algorithms have their own unique defining characteristics and strengths. For instance, the **NeMo** algorithm is notable for its ability to incorporate indirect interaction information based on shared neighbors. **ClusterONE** is notable for outperforming many well established methods in reconstructing known complexes.

3.2 Network Alignment of PPI Networks

Network Alignment methods seek functional relationships between the networks by identifying conserved PPI subgraphs. One observes that network alignment for PPI networks is analogous to protein sequence alignment

Figure 3.9: An example of a network alignment between two PPI networks.

for amino acid sequences. Formally, given two PPI networks G_1 and G_2 , a network alignment between them is a mapping $f : V_1 \rightarrow V_2$ (network alignment may have multiple definitions). The mapping f is bijection if there is a one-to-one correspondence between proteins in V_1 and V_2 . Furthermore, the mapping f represents a subgraph isomorphism iff f is a bijection and $(x, y) \in E_1$ implies $(f(x), f(y)) \in E_2$. Give a subgraph $C_1 \subset G_1$ and $C_2 \subset G_2$, if there exists a subgraph isomorphism between C_1 and C_2 , one observes that these subgraph regions are topologically conserved. In Figure 3.9, the dashed lines represent a network alignment between the human and fruit fly PPI networks. The alignment shows the extent of network conservation of the MAPK pathways between the two species. In this case, the mapping f is a bijection but is not subgraph isomorphic.

3.2.1 Overview of PPI Network Alignment Algorithms.

Because the strict definition of subgraph isomorphism may yield only a small subset of topologically conserved alignment, network alignment strategies seek to find subgraph regions that are topologically conserved (or approximately so). Typically, an objective function $F_{G_1, G_2} : f \rightarrow \mathbb{R}$ is introduced to measure the alignment quality based on how well it identifies

CHAPTER 3. RELATED WORK

topologically conserved regions. The network alignment algorithm then finds $\operatorname{argmax}_f F(f)$.

We motivate the network alignment problem with a simple example shown in Figure 3.9. Consider two PPI networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. Also consider edge weight functions $f_1 : E_1 \rightarrow \mathbb{R}$ and $f_2 : E_2 \rightarrow \mathbb{R}$ associated with G_1 and G_2 respectively. The edge weight functions typically represent the confidence of the interactions.

Table 3.3 summarizes representative work relevant to PPI network alignment (organized by algorithmic approach). These methods are organized by the following:

- (i) *Dynamic Programming*. These methods find an exact solution to the alignment problem.
- (ii) *Seed and Expand*. Seed and expand heuristics utilize greedy algorithms to identify an alignment between networks.
- (iii) *Random Walk*. The rationale for random walk is the following: the overall similarity of a pair of proteins depends not just among themselves, but also the overall similarities of their *neighborhood*. Here, the neighborhood of a protein $v \in V$ in G is defined as the set of proteins adjacent to v , denoted by $N(v)$. Problems formulated in this manner – also known as the **PageRank** problem — is modeled by finding the stationary distribution of a transition matrix.
- (iv) *Integer Linear Programming (ILP) relaxation*. ILP methods pose the network alignment problem as an integer programming problem. Then, a relaxation is made to permit a tractable solution using well known convex optimization algorithms.

CHAPTER 3. RELATED WORK

Table 3.3: Overview of PPI network alignment approaches.

Seed and Expand Algorithms	
NetworkBLAST	[19]
NetworkBLAST-M	[81]
MaWish	[82]
Græmlin	[83]
CAPPI	[84]
MI-GRAAL	[85]
C-GRAAL	[86]
PINALOG	[87]
NetAligner	[88]
SPINAL	[89]
Dynamic Programming Algorithms	
PathBLAST	[23]
Pinter <i>et al.</i>	[90]
Random Walk Algorithms	
IsoRank	[20]
IsoRankN	[91]
Integer Linear Program Algorithms	
Cross-species analysis of biological networks by Bayesian alignment	[92]
Natalie	[93]

3.2.2 Dynamic Programming Algorithms

Among the earliest network alignment method is PathBLAST [23]. The PathBLAST method identifies conserved linear pathways between two PPI networks. First, a *global alignment graph* is constructed. The global alignment graph, denoted by $G_A = (V_A, E_A)$, is a graph where each node $v \in V_A$ represents a pair of proteins, one from each network. An edge $e \in E_A$ may be classified as one of the following: 1) *direct* – when two protein pairs $u, v \in V_A$ have direct matching interactions between the proteins; 2) *gap* – when one pair has direct interaction while the other pair has an indirect interaction; or 3) *mismatch* – when the protein pairs have mismatching interactions. Figure 3.10b illustrates a global alignment graph induced from the pair of PPI networks in Figure 3.10a. The set of node pairs V_A in this example is $\{(A, a), (B, b), (D, d), (F, f)\}$. Note how the classification (gap, mismatch and direct) of each $e \in E_A$ is derived. To align the two PPI networks, PathBLAST first generates $5L!$ acyclic subgraphs by random edges removal. Then, a dynamic programming approach is applied to score linear paths in the acyclic graphs. The highest-scoring path of length L is then identified. One may note that the algorithm is $O(L!)$. The two aligned paths are scored as follows:

$$S(P) = \sum_{v \in P} \log \frac{p(v)}{p_r} + \sum_{e \in P} \log \frac{q(e)}{q_r} \quad (3.28)$$

where $p(v)$ is the probability that given $v = (i, j) \in V_A$, the proteins i and j are truly homologous; $q(e)$ is the probability that the interaction e is true positive; and p_r and q_r correspond to the probabilities under the null model.

Dynamic programming strategies entail an exhaustive search to enumerate all search possibilities (with culling to remove redundant searches). While dynamic programming methods find an exact solution, these algorithms are NP-complete, implying that only very small networks are prac-

Figure 3.10: The PathBLAST algorithm. (Image taken from [23])

ticle. Usually, input networks are restricted to PPI networks with vertices of less than 50 proteins.

3.2.3 Seed and Expand Algorithms

Greedy seed and expand strategies dominate the body of network alignment methods. The seed and expand strategy is most popular because of its efficiency and effectiveness. In this approach, a small, highly conserved subgraph is first identified as seed. Because seeds are small, finding highly conserved subgraphs can be done optimally in a tractable manner. Once seeds have been discovered, greedy expansion is performed to grow the seed and increase the coverage of the alignment.

In general, seed and expand methods construct a *local alignment* between PPI networks. A local alignment does not map every protein from one network to another. Instead, it identifies small subgraphs that are well-conserved between the networks. This is analogous to local sequence alignment of amino acid and DNA sequences. Figure 3.11 illustrates the difference between a local alignment, which only aligns well-conserved regions, and a global alignment, which aligns every pair of proteins in both networks.

Figure 3.11: Local vs Global PPI Network Alignment.

An advantage of local network alignment is performance, as the algorithm does not need to find an optimal global mapping (involving a much larger search space). Another advantage is interpretability, since well-conserved subgraphs are typically meaningful, whereas in global network alignment, not all mappings may make sense given the noise in high-throughput experiment datasets.

NetworkBLAST [19] proposed a unified model that improves upon the methods introduced in **PathBLAST** by generalizing to other network structures, including both linear paths and network clusters. **NetworkBLAST** first constructs a *network alignment graph*. Suppose one wishes to align k networks from k species. In **NetworkBLAST**, up to $k = 3$ networks are supported. A network alignment graph is the graph $G_A = (V_A, E_A)$ such that V_A is the set of “supernodes” and E_A is the set of edges connecting the supernodes. A supernode $v \in V_A$ is a node corresponding to a set of k proteins, each taken from one of the k species. Additionally, the k proteins of $v \in V_A$ must satisfy a sequence similarity equivalence relationship. The k proteins are then said to be *sequence-similar*. To characterize E_A , two supernodes $u, v \in V_A$ are deemed to interact, *i.e.*, $(u, v) \in E_A$, if the following rules are met: 1) one pair directly interacts while other pairs are at

CHAPTER 3. RELATED WORK

most distance two away; 2) all pairs are exactly distance two away; or 3) at least $\max\{2, k-1\}$ pairs directly interact. Figure 3.12 shows an illustration of a (partial) network alignment graph constructed on two PPI networks (fly and yeast). A supernode (indicated by a rounded rectangle) is a node from each PPI network that are homologous (indicated by a dashed line). Two supernodes are deemed to interact depending on the number of direct interactions going between the supernodes, as specified earlier. We observe in the figure that one pair of super nodes has two direct interactions, while another has only one.

Given the network alignment graph defined above, **NetworkBLAST** identifies subgraphs of G_A corresponding to conserved subnetworks. Here, conserved subgraph identification is performed using a heuristic. First, a seed of size 3 is constructed greedily, adding a node at a time to maximally improve the current score. Then, nodes are added or removed, one at a time, such that the overall score is improved each time. The seed nodes, however, must not be removed. Subgraphs up to 15 nodes are discovered this way. Finally, a filtering step removes highly redundant subgraphs. To score the candidates during subgraph identification, a probabilistic model is proposed. Let u, v be two proteins in a network $G = (V, E)$. Let T_{uv} be the event that u and v interact and F_{uv} be the event that they do not. Let O_{uv} be the event that an experimental observation of their interaction exists. Let p_{uv} be probability that u and v interact when drawn from the null model. Given a subset of vertices U , and assuming independence, the likelihood ratio score is given by:

$$\mathcal{L}(U) = \sum_{(u,v) \in E} \log \frac{\beta P(O_{uv}|T_{uv}) + (1 - \beta)P(O_{uv}|F_{uv})}{p_{uv}P(O_{uv}|T_{uv}) + (1 - p_{uv})P(O_{uv}|F_{uv})} \quad (3.29)$$

The **NetworkBLAST** method, however, do not incorporate evolutionary models of protein-protein interactions during alignment. To this end, **MaWISh** [82] introduces a network alignment methodology that is more consistent with evolutionary models of PPI networks. **MaWISh** exploits the

Figure 3.12: Illustration of NetworkBLAST algorithm.

duplication and divergence model of PPI network evolution to identify conserved network subgraphs. Given two networks $G(U, E)$ and $H(V, F)$, the score of an alignment $\mathcal{A}$ is defined as:

$$\sigma(\mathcal{A}) = \sum_{m \in \mathcal{M}} \mu(m) - \sum_{n \in \mathcal{N}} \nu(n) - \sum_{d \in \mathcal{D}} \delta(d) \quad (3.30)$$

where $\mu(\cdot)$ scores matches, $\nu(\cdot)$ penalizes mismatches and $\delta(\cdot)$ penalizes duplication. Figure 3.13b depicts a simple alignment between the two toy PPI networks in Figure 3.13a. The solid edge between the pairs (u_1, v_1) and (u_3, v_3) is characterized by a *match* model, increasing $\mu(\cdot)$ score. Meanwhile, the dotted edge between the pairs (u_3, v_3) and (u_4, v_2) is characterized by a *mismatch* model, increasing $\nu(\cdot)$ penalty. The two proteins u_1 and u_2 represent a *duplication* model, increasing $\delta(\cdot)$ penalty. To identify an alignment that maximizes the above objective function, an alignment graph $G_A = (V_A, E_A)$ is first constructed in similar fashion to **NetworkBLAST** and **PathBLAST**. Additionally, an edge weight function $w : E_A \rightarrow \mathbb{R}$ is introduced to represent orthology information between the supernodes. This way, the problem of identifying conserved subgraphs can be formulated as a maximum weight induced subgraph problem (MWIS). Since finding the

CHAPTER 3. RELATED WORK

Figure 3.13: Duplication/Divergence model used in the MaWISh algorithm. (Image taken from [82])

optimal solution to an MWIS problem is NP-complete, a greedy heuristic is proposed instead. MaWISh first identifies a small maximally weighted subgraph as seed graph. The seed graph is then grown greedily by maximally increasing the alignment score. This step is iterated until no more positive weighted edge can be added.

Previous multiple network alignment methods like NetworkBLAST and MaWISh fail to scale to alignment of more than 3 networks. As such, the Græmlin [83] alignment approach to multiple network alignment is proposed, which is demonstrated by its capability to align up to 10 microbial networks. Given k PPI networks with each network indexed by i , we have $G_i = (V_i, E_i)$. Let $f : E_i \rightarrow \mathbb{R}$ denote the *interaction score function* that associates each interaction with a confidence score. Græmlin performs multiple network alignment in a pairwise manner. Like previous methods, an alignment between a pair of networks is modeled as a global alignment graph; “supernodes” described earlier are termed *equivalence class sets*. Suppose one wishes to align a pair of networks G_i and G_j . In the first phase, Græmlin constructs high confidence seeds, referred to as *d-clusters*. To this end, for each node v in every network, a *d-cluster* is constructed by adding to v its *d*-nearest neighbors. After constructing *d*-clusters for both

CHAPTER 3. RELATED WORK

pair of networks, d -clusters from one network are compared to the other in a pairwise manner. Given a pair of d -clusters D_1 and D_2 , a sum of all pairwise scores of nodes within the clusters is computed. Then, among all pairs of d -clusters, **Græmlin** filters clusters that score higher than a threshold T . These filtered d -clusters now serve as seeds. Following seed construction, a greedy expansion is performed by adding to the seeds new node pair alignments that maximizes the current alignment score. The expansion continues until the alignment score cannot be improved anymore. Other heuristics are also included to control for computation scalability. To score an alignment of PPI networks, **Græmlin** proposes two scoring methods:

- **Node scoring method:** Nodes are scored by their sequence homology to determine whether the nodes match. The scoring method also account for evolutionary mutations in the network, such as deletion, duplication and insertion events.
- **Edge scoring method:** The edges are also scored for their matches. Several scoring matrices are introduced, including module, pathway and complex scoring metrics.

To handle multiple network alignment, **Græmlin** aligns the closest pairs of network (by evaluating all pairwise network alignment scores). Following that, successive PPI networks are aligned one-by-one against the aligned set of networks. This approach is analogous to the progressive sequence alignment techniques in DNA or protein sequence alignment [83].

Another approach based on the phylogeny history of proteins is **CAPPI** [84]. Given k PPI networks indexed by an index set $I = \{1, \dots, k\}$, we have $G_i = (V_i, E_i)$. Figure 3.14 illustrates the **CAPPI** algorithm. **CAPPI** begins by clustering proteins in $\bigcup_{i \in I} V_i$ based on their sequence homology distance. To this end, the **TRIBE-MCL** clustering algorithm is applied. Once the vertex sets are partitioned into clusters of homologous proteins, **CAPPI** reconstructs the phylogeny tree of proteins within each cluster. **CAPPI**

CHAPTER 3. RELATED WORK

assumes that each protein group in a cluster originates from a common ancestral protein. The tree reconstruction is achieved via the *Neighbor Joining* algorithm. Given the ancestral tree of each cluster, an ancestral graph denoted by $G_{1,0} = (V_{1,0}, E_{1,0})$ comprising the ancestral root node from each cluster is constructed. A weight function $f : E_{1,0} \rightarrow \mathbb{R}$ maps the connection strength between two ancestral root nodes. The weights are determined using the interaction patterns of the underlying descendant proteins (which is given by the network edges in E_i). Given the ancestral graph, densely connected ancestral nodes are then identified. In this case, a simple thresholding parameter, denoted by t , is set and all edges with weights below f are removed, revealing the conserved clusters. Apart from the root ancestral graph $G_{1,0}$, ancestral graphs of further evolutionary events can be predicted. Let $G_{i,j}$ denote the ancestral graph of the i -th species and the j -th duplication or speciation event that evolved from $G_{1,0}$. At each step, $G_{i,j}$ is a single mutation of the $G_{i,j-1}$ graph based on a duplication or speciation event. A probabilistic model is developed to determine interactions in $G_{i,j}$ from its parent graph $G_{i,j-1}$.

While the previous approaches rely on only a single modality of similarity measure, the **MI-GRAAL** [85] global PPI network alignment approach integrates multiple sources of similarity measures without user-specified parameters. In **MI-GRAAL**, five similarity measures are utilized:

- Graphlet degree signature distance
- Relative degree difference
- Relative clustering coefficient difference
- Relative eccentricity difference
- BLAST E-value for protein sequence similarity

Given two PPI networks G_1 and G_2 , **MI-GRAAL** first constructs five matrices corresponding to pairwise scores of each similarity measure. Denote

CHAPTER 3. RELATED WORK

Figure 3.14: The CAPPI algorithm. (Image taken from [84])

for each measure i by the matrix M_i . Each matrix acts as independent confidence *agents* that votes the confidence of alignment between proteins u and v . The alignment confidence, denoted by $C(u, v)$, between proteins u and v is defined as:

$$C(u, v) = \sum_i \text{conf}_i(u, v) \quad (3.31)$$

where $\text{conf}_i(u, v)$ is the fraction of elements in u -th row of matrix M_i that are strictly greater than $M_i[u, v]$. With the aggregated confidence scores defined, the **MI-GRAAL** algorithm computes the alignment between G_1 and G_2 . Every protein pair $(u, v) \in V_1 \times V_2$ is ordered by their confidence scores $C(u, v)$ and placed in a priority queue. Then, the algorithm adds the best scoring pair (u_*, v_*) to the alignment. Following that, a k -th neighborhood is constructed for both u_* and v_* . A k -th neighborhood of v is the set of nodes that are at distance $d \leq k$ from v . These neighborhoods are then aligned to each other by solving the *Maximum Weight Bipartite Matching*

CHAPTER 3. RELATED WORK

problem. The best scoring and neighborhood matching steps are repeated until no more protein pairs can be aligned.

NetAligner [88] incorporates evolutionary distances to determine conserved interactions. The method first constructs a seed of homologous proteins, following which connecting vertices are extended by evaluating both mismatches and gap. A significant strength of this method is speed. Results demonstrate that **NetAligner** outperforms representative network alignment methods including **IsoRank** and **NetworkBLAST** while requiring very little computation time.

Unlike the aforementioned approaches that rely only on protein sequence similarity measure, **PINALOG** [87] combines both the similarities of protein sequence and protein function to compute an alignment between two PPI networks. To this end, the GO function similarity and sequence similarity are obtained independently and then combined using a linear combination of the independent similarities weighted by a factor θ . Following that, the Hungarian method is used to identify the mapping between the networks, of which the top 15% is used as seed. Finally, expansion is performed to complete the alignment. The combination of function and sequence allows **PINALOG** to outperform competing methods that include **MCODE**, **MCL**, **IsoRank**, and **MI-GRAAL**.

3.2.4 Random Walk Algorithms

Unlike seed and expand methods, random walk methods construct a *global alignment*. **IsoRank** [20] introduces an approach that performs a global alignment of two PPI networks. The alignment problem is formulated as an eigenvalue problem that is analogous to the PageRank method. Suppose one wishes to align two networks $G_1 = (V_1, E_1, w_1)$ and $G_2 = (V_2, E_2, w_2)$. The **IsoRank** method assumes that the alignment strength between two proteins $i \in V_1$ and $j \in V_2$ is given by a linear combination of the following factors: 1) the protein sequence homology between i and j ; 2) the alignment

CHAPTER 3. RELATED WORK

strengths among their neighborhood proteins. Formally, this is defined as a recursive function:

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{w_1(i, u)w_2(j, v)}{Z(u)Z(v)} R_{uv} \quad (3.32)$$

where R_{ij} is the alignment strength between proteins i and j ; $w_k(x, y)$ is the weight of the edge in $(x, y) \in E_k$; $N(x)$ denotes the neighborhood proteins of x ; $Z(u)Z(v)$ are weight normalization factors. In the special case where the edge weights are uniform (*i.e.*, $w(x, y) = 1$ iff there is an interaction between x and y) the above equation simplifies to:

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)||N(v)|} R_{uv} \quad (3.33)$$

When evaluating all protein pairs, the equation can be rewritten in matrix form as follows:

$$R = \sum R_{ij} \quad (3.34)$$

$$R = AR \quad (3.35)$$

$$A[i, j][u, v] = \frac{1}{|N(u)||N(v)|} I((i, u), (j, v)) \quad (3.36)$$

where $I((i, u), (j, v))$ is the indicator function such that $I = 1$ if and only if $(i, u) \in E_1$ and $(j, v) \in E_2$; otherwise, $I = 0$. To incorporate protein sequence homology, $R = AR$ is rewritten as:

$$R = \alpha AR + (1 - \alpha)E \quad (3.37)$$

where $E[i, j]$ is the normalized BLAST similarity score between proteins i and j .

In both cases, solving the above equations to determine R is equivalent to identifying the eigenvalues of A . To this end, **IsoRank** utilizes the well-known *power method*. Once R is obtained, **IsoRank** extracts the alignment mapping by proposing two approaches: 1) One-to-one mappings, obtained

CHAPTER 3. RELATED WORK

Figure 3.15: Illustration of IsoRank algorithm. (Image taken from [20])

via assigned methods like Hungarian method. 2) Many-to-many mappings, obtained via a greedy heuristic that relies on seed and extend.

Figure 3.15 illustrates the IsoRank algorithm with an example. The figure shows two toy networks with their nodes labeled a to e for the first network, and a' to e' for the second. The matrix depicts the pairwise similarities between the proteins in both networks. These values are utilized to compute the recursive functions $R_{aa'}$ to $R_{dd'}$, which in turn are used to formulate the eigenvalue problem.

IsoRankN [91] extends the original IsoRank method by proposing a more scalable approach to aligning multiple PPI networks. Similar to IsoRank, a PageRank-styled alignment strength vector, denoted by R , is identified by finding a solution to the eigenvalue problem. Once the alignment strength vector R is identified, IsoRankN proposes a spectral graph theory based method to infer a global alignment between multiple PPI networks. Unlike IsoRank and its computationally expensive greedy search strategy, the spectral graph method – called *star-spread method* in IsoRankN – scales better when dealing with multiple network alignment.

IsoRankN first constructs for each $v \in V$ a *star* S_v , which is defined as the set of neighbors of v with edge weights exceeding a user-defined

CHAPTER 3. RELATED WORK

threshold. The stars are then ordered by their sum of weights, and for each star S_v , a highly weighted neighborhood $S_v^* \subset S_v$ is identified. Such highly weighted neighborhood is measured using the *conductance score*. Given a set of vertices $S \subset V$, let the conductance of S be defined as:

$$\Theta(S) = \frac{\sigma(S)}{\min(\text{vol}(S), 2m - \text{vol}(S))} \quad (3.38)$$

where $\sigma(s) = |\{(u, v) | u \in S, v \notin S\}|$; $\text{vol}(S) = \sum_{v \in S} \text{deg}(v)$ and $m = |E|$. The conductance of S is the ratio of edge cuts needed to separate S from the rest of G over the smaller of either: 1) number of edges induced by S on G or 2) the number of edges induced by $V \setminus S$ on G . Intuitively, conductance reflects the “clusteredness” of S , because if few edges are needed to cut off S , then it may well be that S is clustered among themselves and is almost-disconnected from the rest of the network. Thus, low conductance is a good property for identifying clusters.

Identification of these low-conductance sets can be efficiently computed via approximate *personalized PageRank vector*. For each S_v , its personalized PageRank vector $P(\gamma, v)$ is defined as:

$$P(\gamma, v) = \gamma X_v(v) + (1 - \gamma)P(\gamma, v)W \quad (3.39)$$

where $\gamma \in [0, 1)$ is the parameter that controls the probability that the random walk restarts at v and W the transition matrix of a lazy random walk on S_v . The solution to this recursive equation is found using the approximate personalized PageRank algorithm, which finds highly weighted neighborhood S_v^* of each star S_v . Finally, the *star merging* step is performed to assign other members not determined previously. Given two stars S_v^* and S_u^* , merge them if every member of $S_v^* \setminus v$ has u as neighbor and every member of $S_u^* \setminus u$ has v as neighbor. The *spectral partitioning* and *star merging* steps are performed one star at a time, with each star taken from the ordered set of stars described earlier.

3.2.5 Integer Linear Program Algorithms

Integer Linear Program Algorithms cast PPI network alignment as an integer linear program problem. Similar to random walk methods, the method admits a global alignment between PPI networks. **Natalie** [93] proposes a *Lagrangian relaxation approach* to solve the problem approximately. Unlike the heuristics proposed in random walk and seed and expand methods, **Natalie** provides a performance guarantee to their solution. The alignment problem can be cast as a non-linear program as follows:

$$\max \sum_{(i,j) \in V_1 \times V_2} \sigma(i,j) + \sum_{(i,j) \in V_1 \times V_2} \sum_{(k,l) \in V_1 \times V_2} \tau(i,j,k,l) x_{ij} x_{kl} \quad (3.40)$$

$$\text{s.t.} \quad \sum_{(i,j) \in \delta(v)} x_{ij} \leq 1 \quad \forall v \in V_1 \cup V_2 \quad (3.41)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i,j) \in V_1 \times V_2 \quad (3.42)$$

where $\sigma(i,j)$ is the weight of the similarity between proteins i and j ; $\tau(i,j,k,l)$ is the score of the pairs of proteins (i,k) matched to (j,l) ; $\delta(v)$ is the set of edges incident to v and x_{ij} is an indicator of whether protein i is aligned to protein j . Linearization of the above is then given by the following Integer Linear Programming problem:

$$\max \sum_{(i,j) \in V_1 \times V_2} \sigma(i,j) + \sum_{(i,j) \in V_1 \times V_2} \sum_{(k,l) \in V_1 \times V_2} \tau(i,j,k,l) y_{ijkl} \quad (3.43)$$

$$\text{s.t.} \quad \sum_{(i,j) \in \delta(v)} x_{ij} \leq 1 \quad \forall v \in V_1 \cup V_2 \quad (3.44)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i,j) \in V_1 \times V_2 \quad (3.45)$$

$$y_{ijkl} \leq x_{ij} \quad (3.46)$$

$$y_{ijkl} \leq x_{kl} \quad (3.47)$$

$$y_{ijkl} \in \{0, 1\} \quad \forall (i,j,k,l) \in (V_1 \times V_2)^2 \quad (3.48)$$

$$(3.49)$$

Now with the ILP problem, a Lagrangian relaxation approach is applied to solve the ILP problem approximately.

Table 3.4: Summary of PPI network alignment techniques.

Algorithms		Evolutionary	Scalability	Multiple	Coverage
NetworkBLAST	[19]	No	Low	No	Local
NetworkBLAST-M	[81]	Yes	Medium	Yes	Local
MaWish	[82]	Yes	Medium	No	Local
Græmlin	[83]	Yes	Medium	Yes	Local
CAPPI	[84]	Yes	Medium	Yes	Local
MI-GRAAL	[85]	No	Medium	No	Global
C-GRAAL	[86]	No	Medium	No	Global
PINALOG	[87]	No	Medium	No	Global
NetAligner	[88]	No	Medium	No	Local
SPINAL	[89]		Medium	No	Local
PathBLAST	[23]	No	Low	No	Local
Pinter <i>et al.</i>	[90]	No	Medium	No	Local
IsoRank	[20]	No	Medium	No	Global
IsoRankN	[91]	No	Medium	Yes	Global
Bayesian alignment	[92]	Yes	Medium	No	Local
Natalie	[93]	Yes	Medium	No	Global

Algorithms		Integrative	Annotation
NetworkBLAST	[19]	No	No
NetworkBLAST-M	[81]	No	No
MaWish	[82]	No	No
Græmlin	[83]	Yes	No
CAPPI	[84]	Yes	No
MI-GRAAL	[85]	No	No
C-GRAAL	[86]	No	No
PINALOG	[87]	Yes	Yes
NetAligner	[88]	No	No
SPINAL	[89]	No	No
PathBLAST	[23]	No	No
Pinter <i>et al.</i>	[90]	No	No
IsoRank	[20]	No	No
IsoRankN	[91]	No	No
Bayesian alignment	[92]	No	No
Natalie	[93]	Yes	No

3.2.6 Summary of Network Alignment Algorithms

Table 3.4 summarizes the network alignment approaches by the following properties:

- *Evolutionary* column indicates whether the alignment is based on

evolutionary principles

- *Scalability* column qualitatively suggests the algorithm’s capability to scale to larger PPI network
- *Multiple* column indicates whether the algorithm can align more than three PPI networks (ability to scale to many networks)
- *Coverage* column indicates whether the algorithm aligns the PPI networks globally or locally
- *Integrative* column identifies algorithms that can integrate multiple similarity measures during the alignment process
- *Annotation* column identifies algorithms that are annotation aware, whereby the alignment process can be guided by available annotation information encoded within PPI networks

The ideal network alignment algorithm would exhibit the following strengths: evolutionary aware, highly scalable, ability to scale to multiple networks, ability to support both global and local alignment, integrative, and annotation aware. Clearly, no existing algorithm enjoys all of these strengths. Instead, one chooses the best algorithm based on specific user requirements.

3.3 Summary

Compared to existing work, our research is novel in the following ways:

- In elucidating the functional organization within a PPI network, existing graph clustering approaches overwhelmingly emphasize on finding structurally dense subgraphs over finding subgraphs with coherent attributes (*i.e.*, functionally coherent regions). However, attribute coherence is key to forming meaningful, interpretable functional modules. In a PPI network, groups of proteins (vertices) that share a

CHAPTER 3. RELATED WORK

common vertex property can form a meaningful cluster representing a specific biological function. Otherwise, clusters with inconsistent vertex properties, even if structurally well-connected, may not simply summarize into one functionally interpretable cluster. Furthermore, functional modules within PPI networks are not always structurally dense. Proteins in signalling pathways, for instance, are structurally loose, but share important functions. Such groups of proteins often have significant biological implications despite their loose structure, and should be present in any summary of the underlying network. Existing approaches fail to capture such structures. Finally, because the annotations that describe proteins and their functions are high-dimensional, finding the right choice of attribute coherent groupings is combinatorial and non-trivial. In Chapter 4, we present a novel data-driven algorithm called FUSE (**F**unctional **S**ummary **G**enerator) that addresses the aforementioned these limitations of existing work. Given a PPI network, it generates a multi-resolution *functional summary graph* (FSG) (at different levels of abstraction) that best represents the higher-order functional abstraction of the PPI network by simultaneously evaluating interaction and annotation data.

- Existing graph clustering methods generate *only a single optimal functional decomposition* of a PPI network. Consequently, a PPI network can only be decomposed and viewed from a single perspective, whereas in reality there are often multiple different perspectives (decompositions) associated with the functional organization of the underlying network, all of which are distinct and equally valid. For example, a PPI network may be decomposed into dense regions of the network that correspond to the decomposition of protein complexes. However, this network can also be organized from another perspective. It can be organized by the types of signaling pathways involved

CHAPTER 3. RELATED WORK

or by their localization, and such organization could be markedly different from the complex-based decomposition. Clearly, the possibility of uncovering multiple, distinct functional decompositions is real especially in the context of large-scale networks. In Chapter 5, we propose a novel algorithm called **FACETS** that discovers a set of functionally unique decompositions from a PPI network, portraying alternative views of the functional landscape of the network. Each decomposition represents a distinct interpretation of how the network can be functionally decomposed and organized.

- In elucidating the functional relationships between PPI networks via network alignment, existing methods face a key problem. Sequence similarity are only relevant to a subset of highly conserved proteins, leaving significant network regions poorly specified by sequence homology. Structural information from PPI also suffers either from high false positive rate of current high-throughput experiments or from false negatives due to incomplete data. This implies that in significant regions of the network, high confidence alignment of proteins is not possible, even biologically misleading. Local network alignment addresses this by completely ignoring low confidence mappings, while global network alignment pairs all proteins regardless. In Chapter 6, we propose the **DualAligner** algorithm that utilizes a dual alignment strategy in a multi-resolution manner, where coarse grained region-to-region alignments are first made followed by detailed protein-to-protein alignment (hence the term dual alignment).

Chapter 4

FUSE: Towards Multi-Level Functional Summarization of Protein Interaction Networks

This chapter focuses on the first contribution of the dissertation – the **FUSE** algorithm. **FUSE** constructs higher level functional summary that summarizes the underlying PPI network to obtain a concise, interpretable representation of the network. We first motivate the problem of information overload concerning PPI networks. Then, we demonstrate the role of **FUSE** in addressing the information overload issue of analyzing large scale PPI networks. We evaluate the performance of **FUSE** on several real-world PPIS. We also compare **FUSE** to state-of-the-art graph clustering methods with GO term enrichment by constructing the biological process landscape of the PPIS. Our experimental results demonstrate that **FUSE** is highly effective in constructing higher order functional maps with superior accuracy and representativeness compared to these state-of-the-art graph clustering methods. Using AD network as our case study, we further demonstrate the ability of **FUSE** to quickly summarize the network and identify many different processes and complexes that regulate it. We analyze the topological features of the functional landscape of human PPI that leads us to the identification of *functional hubs* (clusters of proteins that act as hubs). Finally, we demonstrate the role of **FUSE** in summarizing pertinent functional dif-

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.1: Functional summary (FSG) of the AD network for $k = 30$ (cluster size indicated in brackets).

ferences between two E-MAP networks under contrasting conditions. Part of this chapter is published in [94].

4.1 Motivation

With advances in high throughput experimental biology, the number of large scale protein interaction networks (PPI) have grown rapidly. At the same time, collaborative efforts to annotate proteins and genes using Gene Ontology [95] (GO) annotations has generated detailed attributes that describe these entities. Knowledge-bases with GO annotations, such as UniprotKB [96], provide a wealth of annotation data at different levels of specificity. Recall from Section 2 that GO provides standardized annotations that describe various attributes of a gene or protein, including localization attributes, molecular function, and the biological processes it participates in. As proteins may involve in multiple roles and functions,

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

GO attributes associated with a protein or a gene can be high-dimensional.

The amount of information contained within large biological networks can often overwhelm researchers, making systems level analysis of PPIs a daunting task. As majority of function annotation and high throughput or curated interaction data are encoded at protein or gene level, higher-order abstraction maps such as complex-complex or process-process functional landscapes, are often unavailable. However, availability of such information is invaluable as it not only allows one to ask questions about the relationships among high-level modules, such as processes and complexes, but also allows one to visualize higher order patterns from a bird’s eye perspective.

For instance, consider the Alzheimer’s Disease (AD) related PPI in *IntAct* [97]. An AD interaction network can be studied at different levels of organization, from broad-level process-process interactions to in-depth complex-complex interactions. Such maps would reveal higher-level patterns that otherwise would have been invisible. The objective here is not to study a process associated with AD in isolation, but instead focus on the interplay of related processes in tandem to identify the causative mechanisms of AD. For example, one might ask the following questions: How do signaling pathways implicated for AD associate with one another? How do proteins related to transportation play a role in AD, and how are they associated with bioenergetics? A bird’s-eye view of the functional landscape of AD network may provide answers to these questions. An example is shown in Figure 4.1 (detailed in Section 4.7). Observe that the associations between signaling pathways (*A28*, *A14*, *A18*, *A21*, and *A16*) are depicted in the summary. It is worth mentioning that it is extremely difficult to answer the aforementioned questions by simply looking at a large PPI containing large number of proteins and interactions as nodes. This problem is further exacerbated by the high-dimensional nature of PPI; each protein

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

may have hundreds of annotation attributes. *It is therefore crucial to have some form of summarization that maps higher-order information of the underlying PPI.* Fortunately, the modular nature of biological networks—either structurally or attribute wise—lends itself to the possibility of building such a summary. We shall also demonstrate the utility of functional summarization to observe the dynamics of networks under environmental and contextual changes. To this end, functional summarization can be modified to form differential functional summaries that shows a summary of the functional landscape that respond to contextual changes.

Although tools to abstract high-level and functional information from gene lists have been proven to be key to analyzing high throughput data [98], similar tools that automatically abstract and *summarize* PPIS at multiple resolutions to provide high level views of functional landscape of PPIS are still lacking. At first glance, it may seem that state-of-the-art graph clustering techniques [17, 99, 74, 78, 100] can be used for generating high quality summaries of PPIS as these techniques have been successful in identification of novel protein function and protein complexes. Intuitively, a biological network can be decomposed into modules—groups of vertices sharing a common function—that are then collapsed into a representative node to form a summary graph of the underlying network. Depending on the granularity of the decomposition, summaries of various level of detail can be formed. Despite the benefits of graph clustering, these techniques suffer from the following key weaknesses that make them less suitable for building high quality higher order functional summaries of PPIS.

Firstly, several existing graph clustering approaches [17, 99, 101, 74] overwhelmingly emphasize structure cohesiveness over attribute coherence. However, in practical applications of PPI summarization, attribute coherence is key to forming meaningful, interpretable modules. In PPI, groups of proteins (vertices) that share a common vertex property can form a

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.2: FSG of the AD network ($k = 10$).

meaningful cluster that represents a particular biological function. Otherwise, clusters with inconsistent vertex properties, even if structurally well-connected, may not simply summarize into one functionally interpretable cluster. Secondly, majority of existing graph clustering techniques form non-overlapping partitions [17, 101, 74]. Consequently they cannot be used to generate high-quality summary as “interactors” in biological processes and pathways are likely to overlap [102]. Thirdly, these techniques typically focus on identifying dense subgraphs from a graph. However, higher-level clusters in PPIS are not always structurally dense. Proteins in signaling pathways, for instance, are structurally loose, but share important functions. Such groups of proteins often have significant biological implications despite their loose structure, and should be present in any summary of the underlying network. Finally, because the annotations that describe proteins and their functions are high-dimensional, finding the right choice of attribute coherent groupings is combinatorial and non-trivial.

Figure 4.4 contains examples of both optimal and sub-optimal clustering summarization of biological graphs. In Figure 4.4(a), an optimal summarization decomposes the graph into clusters A and B, both of which

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.3: (a) A toy example of PPI network. (b) A set of *functional clusters* of the network in (a). (c) Suppose a 3-node summary is required ($k = 3$). **FUSE** explores the functional clusters of the PPI network to identify the 3-node functional summary that best partition and represent the underlying network. This functional summary graph (FSG) depicts the functional landscape of the PPI network in 3 nodes. (d) A 5-node partition ($k = 5$) and its corresponding FSG.

have consistent attributes and cohesive structure. Although the underlying graph is a clique, choosing a cluster that encompasses all vertices—as shown in Figure 4.4(b)—would be sub-optimal, because vertex attributes within the cluster would then be inconsistent. Consequently, one could not extract a common, biologically meaningful function that represents the cluster. Figure 4.4(c) is also less optimal compared to 4.4(a), because despite having attribute consistent clusters, intra-cluster cohesiveness of the vertices are weak.

Let us consider another scenario. Figure 4.4(d) shows a graph that is partitioned in clusters E and F. In the absence of dense structure and coher-

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.4: Structure and attribute considerations in graph summarization.

ent attributes, biological graphs may be partitioned based on its attributes. Proteins in signaling pathways, for instance, are structurally loose, but share important functions. Such groups of proteins often have significant biological implications despite their loose structure, and should be present in any summary of the underlying network. Finally, Figure 4.4(e) shows an example of poor summarization, as despite having consistent attributes, the clusterings have inadequate coverage. This makes the subsequent summary less representative of the underlying graph.

4.2 Overview

We present a novel data-driven algorithm called **FUSE** (**F**unctional **S**ummary **g**enerator) that addresses the aforementioned challenges (Sections 4.4 and 6.2.4). Given a PPI, it generates a *k*-node functional summary graph (FSG) that best represents the higher-order abstraction of the PPI by simultaneously evaluating interaction and annotation data. We argue that a “good” functional summary of a network is not merely a graph of all function-function relationships, but a graph that *reduces* details of the original PPI to form a

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF
PROTEIN INTERACTION NETWORKS

subset of interconnected *functional clusters*. A *functional cluster* represents a subnetwork of proteins that shares a common function. In particular, the functional summary graph must simultaneously satisfy the following requirements: (a) the summary is at a *specific level* (k nodes) of detail, (b) the summary is representative of the original network, and (c) redundancies are minimized. Specifically, **FUSE** exploits *Minimum Description Length* principle [103] to generate the “best” summary by maximizing *information gain* while satisfying the level of details constraint. Figure 4.3 depicts examples of functional summaries generated by **FUSE**. Figure 4.1 and 4.2 depict a 30-node and a 10-node FSGs of the AD network, respectively, generated by **FUSE**.

The goal of **FUSE** is not only to generate a higher level functional summary that is representative of the underlying PPI, but also to generate a k -node functional map whose visual complexity (determined by k) permits user analysis. With close to 30000 terms in the Gene Ontology (GO), interaction network of 30000 functional modules will not be a useful summary, as it is just as daunting as the original PPI, if not more. **FUSE** addresses this challenge by enabling generation of summaries that are small and understandable.

4.3 Related Work

Functional landscape of an underlying protein interaction network has been explored in [104]. The approach the authors used, however, rely on manual short listing of 229 biological processes for analysis. While this approach makes visualization permissible, it neither scale with the growing number of annotations, nor does it fully utilize the large number of annotations available. Additionally, the processes that are relevant depends on the context of the network.

Traditionally, functions pertaining to a list of genes is extracted through

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

functional enrichment techniques, which identifies statistically over-represented functions within a set of proteins or genes [98]. Such approaches are designed to identify enriched functions that describe the dataset as whole. In contrast to **FUSE**, they do not utilize the interaction data encoded within PPI, which is key to understanding how the processes and complexes cooperate to govern a particular function.

Graph clustering methods, on the other hand, identify functional clusters based on the underlying assumption that the topology of interacting proteins can be mined to identify protein clusters [17, 99, 101, 74]. Cluster function can then be inferred and annotated by finding enriched annotations within the cluster. While such methods have been proven effective for identification of complexes, they are less suitable for identifying higher level functional clusters, such as biological processes and pathways, where interactors within them are likely to overlap [102, 105]. Interactions within a process are also not necessarily cohesive. CFinder [106] locates overlapping communities based on structure of the network, but ignores the wealth of functional knowledge already encoded in GO annotation data. While most graph clustering techniques rely solely on network topology, several recent techniques utilize annotation information when clustering the networks [78, 107, 100, 108]. However, these techniques form non-overlapping partitions. Additionally, with the growing amount of annotation data, the attribute space of the nodes in an interaction network is high dimensional as a single protein may be linked to hundreds of annotations. However, these state-of-the-art approaches are not designed for clustering high-dimensional attributes of GO annotated interaction networks. For instance, in [78], a “semantic” distance function is used to measure semantic similarities between nodes with multiple **MIPS** complex annotations. The *curse of dimensionality* limits the applicability of such an approach on GO annotations [109]. To the best of our knowledge, *no existing method*

Table 4.1: Notations in FUSE.

Symbols	Description
G	Input PPI graph
$\Theta_G = (S, F)$	Functional summary graph where S and F are sets of nodes and interactions, respectively
ω	edge weight
Δ	Set of GO terms
S_Δ	Set of functional clusters induced from Δ
$C(u)$	Functional cluster representing the function u
$\phi^{C(u)}$	Structural information content of cluster $C(u)$
$c^{C(u)}$	Size deviation cost
k	Summary complexity parameter
b	Information budget parameter
d	Redundancy penalizing parameter
β	Significance cut-off parameter

directly addresses our need for generating overlapping clusters from high-dimensional attributed graphs. Note that existing subspace clustering approaches that allow overlapping subspace clusters typically produce a huge number of clusters that are difficult to interpret [110].

Lastly, the high dependency on interaction topology makes graph clustering ineffective for many context specific networks. Although there are many networks associated with diseases, there are few, if any, with complete interaction knowledge available. The high probability of false positive interactions may also occur. This hampers accurate identification of cohesive clusters.

4.4 The Functional Summarization Problem

In this section, we formally introduce the functional summarization problem. We begin by defining some terminology that we shall be using in the sequel. A summary of notations used in this chapter is given in Table 4.1.

A protein interaction network (PPI) $G = (V, E)$ contains a set of vertices V , representing proteins, and a set of edges E , representing interactions.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

An edge has a positive real weight ω that represents its interaction strength. Given a GO directed acyclic graph (DAG), denoted as D , the ordered set $\Delta = \langle a_1, a_2, \dots, a_n \rangle$ is a topological sort of D , where a_i represents a single GO term. The *term association vector* of $v \in V$, denoted by Δ_v , is defined as $\Delta_v = \langle a_1(v), a_2(v), \dots, a_n(v) \rangle$, $a_i(v) \in \{0, 1\}$, such that $a_i(v) = 1$ if and only if the term a_i or its descendants are associated with protein v . Otherwise, $a_i(v) = 0$. Note that Δ_v indicates GO terms that are associated with v .

4.4.1 Functional Summary of PPI

Given a PPI $G(V, E)$, a *functional summary graph* (FSG) is an undirected graph $\Theta_G(S, F)$ that models the set of higher-order *functional clusters* S and their interactions F that underlie the PPI. A *functional cluster* is a subgraph of G that shares a particular function/role based on the structure and attribute properties of the subgraph and its constituent proteins. Functional clusters may include complexes, processes, and signaling pathways. A pair of functional clusters may be connected by a web of protein interactions. If the number of interactions are significantly large, then we say that the pair of clusters are *associated*. An FSG Θ_G thus captures higher order modules that comprise the PPI and their interconnections. We now define these concepts formally.

Definition 4.1 (Functional Cluster) Let $V(a_i) \subseteq V$ denote the set of vertices in G such that $v \in V(a_i)$ if and only if $\Delta_v[a_i(v)] = 1$. The *functional cluster* of $a_i \in \Delta$, denoted by $C(a_i) \subseteq G$, is the subgraph of G that is induced by $V(a_i)$.

Note that $V(a_i)$ represents the set of vertices of G that are associated with term $a_i \in \Delta$. We treat $C(a_i)$ as a vertex as well. We may also call a functional cluster as *functional subgraph* when we wish to emphasize the fact that it is a graph. Fig. 4.3(b) shows a subset of the possible functional

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.5: Functional clusters associated with the p53 protein.

clusters of the PPI in Fig. 4.3(a). Every node in a cluster must share a particular function or attribute. For instance, nodes in functional cluster `cytosol` share the `cytosol` term.

Given that a protein could be annotated with multiple GO terms, there are a multitude of ways to form functional clusters. To illustrate this, let us restrict ourselves to several GO terms associated with the p53 protein, namely ‘p53 binding’, ‘metal ion binding’ and ‘protease binding’ terms. Fig. 4.5 shows a toy network of functional clusters formed using the ‘p53 binding’, ‘metal ion binding’ and ‘protease binding’ terms, respectively. A ‘p53 binding’ functional cluster, for example, is constructed by first taking all proteins sharing the ‘p53 binding’ term (indicated in the figure as shaded nodes). Following that, the subgraph induced by these proteins forms the ‘p53 binding’ functional cluster.

The three functional clusters represent several alternative ways which proteins can be grouped together based on their shared function. Recall in Section 3.1 that a good cluster exhibits significant clustering properties. For instance, proteins in a cluster should be densely interacting. Using the same intuition, we assess the clustering property of the three functional clusters using subgraph density (see Section 3.1.1). The ‘p53 binding’

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

subgraph is most suitable as a cluster, given that the induced subgraph formed using proteins sharing this function has the highest subgraph density. Conversely, functional cluster formed using ‘metal ion binding’ has the lowest subgraph density. This illustration demonstrates that although a gene or protein could have multiple GO term annotations, not all of these terms are useful for forming a functional cluster. It is therefore important to identify and distinguish terms that are suitable for forming functional clusters from those which are not. We shall see later in the next section how a suitable set of functional clusters can be selected to represent the entire PPI network.

Definition 4.2 (Functional Summary Graph (FSG)) *A functional summary graph of the underlying protein interaction network $G(V, E)$, Θ_G , is defined as $\Theta_G = (S, F, P_i, \alpha)$, where S is a set of functional clusters and F is a set of edges that links the functional clusters. Let oc_{uv} be the number of interactions connecting proteins in $C(u)$ and $C(v)$. Let P_i be the probability density function of observing oc_{uv} or more number of interactions between $C(u)$ and $C(v)$. Let β be a significance cut-off parameter (user-defined). Then, $(C(u), C(v)) \in F$ if and only if $P_i(X > oc_{uv}) \leq 2\beta/|S|^2$. The bijection $\alpha : 1, 2, \dots, m \leftrightarrow S$ is an ordering of S .*

Observe that the aforementioned definition of functional summary includes additional constructs and rules for determining whether two functional clusters are associated. We elaborate on this further. Given a PPI $G(V, E)$, the expected probability of observing an interaction between two randomly drawn protein pair is given by $p_i = \frac{2|E|}{|V|(|V|-1)}$. Let $(C(u), C(v))$ be a functional cluster pair such that members of both clusters were randomly drawn from V . If proteins v_1 and v_2 are randomly drawn from $C(u)$ and $C(v)$, respectively, then the expected probability of observing a positive interaction between them would also be p_i . Let $n = |C(u)||C(v)|$. Based on the independent and identically distributed variable (*iid*) assumption, we

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

model the probability of observing oc (the number of interactions between $C(u)$ and $C(v)$) as the probability of observing oc positive interactions after n *iid* trials, representing n pairwise interaction trials between proteins in $C(u)$ and $C(v)$. Hence, the probability of oc or more positive interactions between $C(u)$ and $C(v)$ can be modeled using a binomial distribution:

$$P_i(X > oc) = \sum_{i=oc}^n \binom{n}{i} p_i^i (1 - p_i)^{n-i} \quad (4.1)$$

This p – *value* is used to assess the *association significance* between a pair of functional clusters. Given a set containing k clusters, association significance between $\frac{1}{2}k(k-1)$ pairs of clusters would have to be tested. To this end, we applied Bonferroni correction to account for multiple testing. Given the *significance cut-off* β , a pair of functional clusters is *significantly associated* if

$$P_i(X > oc) \leq 2\beta/k(k-1) \approx 2\beta/k^2 \quad (4.2)$$

Observe that although we have adopted a simple model to assess cluster-cluster association, the aforementioned definition is general enough to encompass more sophisticated association models.

Example 1 *Fig. 4.3(d) shows an FSG consisting 5 functional clusters. Any edge between two functional clusters exists when $P_i(X > oc_{uv}) \leq 2\beta/|S|^2$, implying that more edges connect proteins between the functional clusters than expected in random.*

4.4.2 Problem Statement

The functional summarization problem is the problem of finding Θ_G that best represents the underlying PPI subject to a *summary complexity constraint*. To model this problem, we propose a profit maximization model that aims to find $\Theta_G = (S, F, P_i, \alpha)$ by maximizing information profit under a budget constraint. Every protein $i \in V$ is assigned a non-negative

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF
PROTEIN INTERACTION NETWORKS

information budget b , which represents the information it contains. Let S_Δ be the set of functional clusters induced from Δ . Every functional cluster $C(u) \in S_\Delta$ is assigned a non-negative *structural information value* $\psi^{C(u)}$ (to be defined later), which represents the amount of structural information contained within the functional subgraph. When a functional cluster $C(u)$ is added to the summary, for every protein $i \in V(u)$, a portion of b is taken out and added to summary information gain. This represents new information added to the summary. The amount to take depends on $\psi^{C(u)}$.

Imposing information budget b limits the amount of information a protein can provide. A parameter $0 \leq d \leq 10$ is also introduced to penalize redundancy. By doing so, repeated representation of a protein i yields reduced information gain, modeling diminishing returns. Based on this profit model, we construct the set of functional clusters that maximizes profit while satisfying the constraints.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Definition 4.3 (Functional Summarization Problem) Let K_i be a set of functional clusters such that $C(u) \in K_i$ if and only if $i \in C(u)$. For every $C(u) \in S_\Delta$, let $\psi^{C(u)}$ be the structural information value of $C(u)$. Given a protein interaction network $G(V, E)$ and user-defined parameters b , d and k , the *functional summarization problem* constructs a k -cluster FSG $\Theta_G = (S, F, P_i, \alpha)$ that satisfies the following optimization problem:

$$\begin{aligned}
 & \text{maximize } \sum_{i \in V} \sum_{j=1}^{|S|} p(i, j) \\
 & \text{where} \\
 & b(i, m) = \begin{cases} \frac{d}{10}(b(i, m-1) - p(i, m-1)) & \text{if } m > 1, \\ & \alpha_S(m-1) \in K_i \\ b(i, m-1) & \text{if } m > 1, \\ & \alpha_S(m-1) \notin K_i \\ b & \text{if } m = 1 \end{cases} \quad (4.3) \\
 & \text{and} \\
 & p(i, m) = \begin{cases} \psi^{\alpha_S(m)} & \text{if } b(i, m) \geq \psi^{\alpha_S(m)} \text{ and } \alpha_S(m) \in K_i \\ b(i, m) & \text{if } b(i, m) < \psi^{\alpha_S(m)} \text{ and } \alpha_S(m) \in K_i \\ 0 & \alpha_S(m) \notin K_i \end{cases} \\
 & \text{subject to} \\
 & |S| = k \\
 & S \subset S_\Delta
 \end{aligned}$$

Here, $p(i, j)$ serves as a store of profit obtained each time protein i is selected in one of the cluster at the j -th iteration. The map $\alpha_S(m)$ serves as an index set to assign the m -th iteration taken to its associated subgraph K of iteration m . The functions $b(i, m)$ the remaining budgeted profit that can be taken per protein, and it can be derived recursively from its preceding $b(m-1)$. We elaborate on how the *structural information value* $\psi^{C(u)}$ is assigned. A functional cluster $C(u)$ and its protein constituents share a common function u . Thus, vertices in the subgraph are considered homogeneous attribute wise. However, it does not imply that the functional subgraph is structurally cohesive (dense). Proteins having common function u may still be weakly interacting. This may be due to the fact that u itself may indicate a general function (e.g., ‘protein binding’) which is a common attribute to a large number of proteins that do not interact with

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

each other. We argue that structurally cohesive functional clusters contain more information than those which are loosely interconnected. The argument for this is that clusters that have higher than expected cohesiveness will have higher information content because of the lower probability of observing a random cluster having the same cohesiveness. However, we make the following exception – a functional cluster with lower than expected cohesiveness is not deemed structurally informative.

Since the optimization problem must choose among a set of functional clusters, we are not concerned about the actual p-value of observing a subgraph having such interaction density. Instead, we only need a measure that allows us to compute the relative ranking of the functional clusters by their information content. Such simplification leads to much greater computation efficiency. We define the *structural information value* of a functional cluster $C(u)$ as follows.

Definition 4.4 (Structural Information Value) Let ω_{ij} be the edge weight of $(i, j) \in E$. The *structural information value* of a functional cluster $C(u)$, denoted by $\psi^{C(u)}$, as $\psi^{C(u)} = \rho^{C(u)}$ where

$$\rho^{C(u)} = \frac{\sum_{i,j \in C(u)} E_{ij}}{|C(u)|} \quad (4.4)$$

At first glance, it may seem that the structural information value should be defined as $\psi^{C(u)} = \rho^{C(u)} - \rho^{\text{random}}$ where ρ^{random} is the *expected structural density* of a random cluster. However, we ignore ρ^{random} for the following reason. $\rho^{C(u)}$ is the *ratio association* [111] score of $C(u)$, a standard graph clustering objective we adopt that indicates the structural density of $C(u)$. In scale-free and Erdős–Rényi graphs, the self-information $-\log P(\psi^{C(u)})$ is a positive non-decreasing function of $\psi^{C(u)}$ for $\psi^{C(u)} > 0$. Hence, $\psi^{C(u)}$ can be used to compare the self-information between two functional clusters without having to determine the probability density function of the interaction distribution of a subgraph. Given $a_i, a_j \in \Delta$, $C(a_i)$ is deemed

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

more informative than $C(a_j)$ if and only if $\psi^{C(a_j)} > \psi^{C(a_i)}$ and $\psi^{C(a_j)} > 0$. If both $\psi^{C(a_j)}$ and $\psi^{C(a_i)}$ are negative, it does not matter whether one is more informative than the other, since both have structural density less than that of random networks. As such, for symmetry, we also deem that $C(a_i)$ is *more informative* than $C(a_j)$ if and only if $\psi^{C(a_j)} > \psi^{C(a_i)}$ for $\psi^{C(a_j)} \leq 0$. Therefore, when comparing the structural density between two clusters, ρ^{random} can be omitted from $\psi^{C(u)}$ and $\psi^{C(u)}$ is simply $\rho^{C(u)}$.

Example 2 Suppose we wish to summarize the PPI in Fig. 4.3(a) into a 3-node summary ($k = 3$). If clusters **apoptosis**, **receptors**, and **TGF-beta** are chosen—instead of the clusters in Fig. 4.3(c)—we can see that the profit obtained is suboptimal. Information budget for proteins **b, c** are depleted due to redundancy, while information budget for proteins **d, e, g, i** are untapped. In contrast, functional summary in Fig. 4.3(c) is relatively more optimal, as not only the set of clusters maximizes profit through superior coverage and minimal redundancy, but it also maximizes profit through higher structural information (e.g., the cluster **transcription** is structurally dense compared to **apoptosis**).

4.5 The Algorithm FUSE

The profit maximization problem is a variation of the *budgeted maximum coverage problem* [112], which is an NP-hard problem. To permit a tractable solution, let us first consider a straightforward greedy approach. The initial FSG is an empty graph. Given the input protein interaction network G , $\psi^{C(u)}$ for each functional cluster $C(u) \in S_\Delta$ are computed. The algorithm then iteratively selects the functional cluster that leads to greatest increase in net profit of the summary. Each time a functional cluster $C(u)$ is selected, the FSG and budget information $b(i)$ for every protein $i \in V(u)$ is updated. Once k clusters has been selected, the algorithm terminates by generating the FSG.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Algorithm 1 Algorithm FUSE

Input: $G, \Delta, D, k, b, d, \beta$
Output: Θ_{min}

- 1: Let $S = \text{empty set}$
- 2: Let $B_{\text{map}} = \text{set of pairs } (i, b) \text{ for each } i \in V$
- 3: Assign $\psi^{C(u)}$ and $c^{C(u)}$ for each $C(u) \in S_{\Delta}$
- 4: $i = 0$
- 5: **while** $i < k$ **do**
- 6: $(C_{\min}, B_{\text{map}}) = \text{MapProfit}(S_{\Delta}, B_{\text{map}}, d, |V|, k)$
- 7: Remove $C_{\min}$ from S_{Δ}
- 8: Add $C_{\min}$ to S
- 9: $i = i + 1$
- 10: **end while**
- 11: **for** $C(i), C(j) \in S$ **do**
- 12: **if** $C(i) \neq C(j)$ and $P_i(X > oc_{C(i)C(j)}) \leq 2\beta/|S|^2$ **then**
- 13: Add edge $(C(i), C(j))$ to F
- 14: **end if**
- 15: **end for**

A major weakness of the aforementioned approach is that it tends to be “overenthusiastic” in selection of functional clusters during early iterations. Functional clusters that are too large or too small may be selected at early iterations resulting in very poor cluster choices at later iterations due to limited information budget and summary size (k) constraint. Hence, our proposed algorithm adds a *complexity cost* to each chosen cluster. Given graph size $|V|$ and summary size k , the *expected cardinality* of a functional cluster in the summary is defined by $E[|C|] = \frac{|V|}{k}$. Then the *size deviation cost*, denoted as $c^{C(u)}$, is defined as the square of the deviation of $|C(u)|$ from $E[|C|]$. That is, $c^{C(u)} = \left(|V(u)| - \frac{|V|}{k}\right)^2$. Observe that the greater the difference between $|V(u)|$ and $E[|C|]$, the less likely it is to be part of a summary of k -granularity. Clusters whose size deviates too much from the expected cardinality are penalized and therefore less likely to be selected. This reduces the chance of having too less or too much information budget remaining during the later iterations of the greedy heuristic.

The aforementioned intuition is realized in **FUSE** as outlined in Algorithm 1. It consists of three phases, namely, the *initialization* phase, the

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Algorithm 2 The *MapProfit* procedure.

Input: $S_\Delta, B_{\text{map}}, d, |V|, k$
Output: $C_{\text{min}}, B_{\text{map}}$

```

1: Let  $p_{\text{max}} = 0$ 
2: for  $C(u) \in S_\Delta$  do
3:   Let  $B_{\text{temp}} = B_{\text{map}}$ 
4:   Let  $p = 0$ 
5:   for  $i \in V(u)$  do
6:     Let  $(i, b(i)) \in B_{\text{temp}}$  and  $p(i) = b(i) - \psi^{C(u)}$ 
7:     if  $p(i) > 0$  then
8:        $p = p + \psi^{C(u)}$ 
9:        $b(i) = b(i) - \psi^{C(u)}$ 
10:    else
11:       $p = p + b(i)$ 
12:       $b(i) = 0$ 
13:    end if
14:  end for
15:   $c^{C(u)} = \left( |V(u)| - \frac{|V|}{k} \right)^2$ 
16:   $p = p - c^{C(u)}$ 
17:  if  $p_{\text{max}} < p$  then
18:     $p_{\text{max}} = p$ 
19:     $C_{\text{min}} = C(u)$ 
20:  end if
21: end for
22: for  $i \in V_{\text{min}}$  do
23:   Let  $(i, b(i)) \in B_{\text{map}}$  and  $p(i) = (d/10)(b(i) - \psi^{C(u)})$ 
24:   if  $p(i) > 0$  then
25:      $b(i) = (d/10)(b(i) - \psi^{C(u)})$ 
26:   else
27:      $b(i) = 0$ 
28:   end if
29: end for
30: return (  $C_{\text{min}}, B_{\text{map}}$  )

```

greedy iteration phase, and the *summary graph construction* phase. In the initialization phase (Lines 1-3), $\psi^{C(u)}$ and $c^{C(u)}$ for each functional cluster $C(u) \in S_\Delta$ are computed. The greedy iteration phase (Lines 4-10) involves iterative addition of functional clusters into S in a greedy manner as described above. The best candidate functional cluster for the current round (C_{min}) is determined through the subroutine **MapProfit** (Line 6). This step also maintains the information profit of S and the remaining informa-

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

tion budget of every v in G through a persistent *profit map* (B_{map}). $C_{\min}$ is then removed from the candidate pool S_{Δ} and added to the solution set S (Lines 7-8). Finally, the summary graph construction phase (Lines 11-15) computes F to generate the FSG $\Theta_{\min}$.

The **MapProfit** procedure is outlined in Algorithm 2. In order to identify the best candidate cluster of the current iteration round, it evaluates every cluster in the candidate pool by evaluating its profit gain potential (Lines 1-21). First, the amount of information to extract from each protein's information budget pool ($b(i)$) is computed (Lines 7-13). Next, the potential profit gain is adjusted to compensate for the complexity cost (Lines 15-16). After $C_{\min}$ is found, the profit map is recomputed to commit the changes made to the information budget map due to the selection of $C_{\min}$ (Lines 21-29).

Theorem 4.1 Algorithm **FUSE** takes $O(|S_{\Delta}|^2|V|^2)$ time in the worst case.

Proof: In the initialization phase, $\psi^{C(u)}$ for each $C(u) \in S_{\Delta}$ has to be computed. Each $C(u)$ may contain up to $|E|$ edges and $|V|$ vertices. In Algorithm 1, $\psi^{C(u)}$ for each $C(u) \in S_{\Delta}$ takes $O(|E|)$ time. Thus, the total complexity for this procedure is $O(|E||S_{\Delta}| + |V||S_{\Delta}|)$ time.

In the greedy iteration phase, the **MapProfit** subroutine defined in Algorithm 2 is evaluated k times. In Algorithm 2, lines 2-21 require $O(|S_{\Delta}||V|)$. Lines 22-29 require $O(|V|)$ time. Thus, Algorithm 2 takes $O(|S_{\Delta}||V| + |V|)$ time. The iteration phase, as such, takes $O(k|S_{\Delta}||V| + k|V|)$ time in total.

Finally, the summary graph construction phase involves pairwise significance evaluation of the resultant functional cluster set. This involves evaluation of all edges between k -pairwise functional clusters of the summary. Each significance $P_i(X > oc_{uv})$ test requires a single-pass evaluation of edges connecting a pair of clusters. At worst case, this takes $O(|E|)$ time. The summary graph construction phase therefore require $O(k^2|E|)$ time.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.6: Illustration of *MapProfit* procedure.

The **FUSE** algorithm, as whole, takes $O(|E||S_\Delta| + |V||S_\Delta| + k|S_\Delta||V| + k|V| + k^2|E|)$ time. In the worst case scenario of $|E| = |V|^2$ and $k = |V|$, the algorithm takes $O(|S_\Delta||V| + |S_\Delta||V|^2 + |V|^2 + |V|^4)$ time, implying a polynomial time complexity at worst possible case.

Example 3 Consider as an example the summarization of the PPI in Fig. 4.3(a). Lines 1-3 in Algorithm 1 constructs the candidates shown in Fig. 4.3(b) and computes, for each candidate $C(u)$, its structural information value $\psi^{C(u)}$ and cost $c^{C(u)}$. Following that, the modified greedy iteration phase selects k candidates by profit maximization (Lines 4-10 in Algorithm 1). Fig. 4.3(c) and (d) show examples of functional subgraphs selected following the greedy iteration phase. Finally, the edges depicted in Fig. 4.3(c) and (d) that indicate the functional relationship between the functional subgraphs are computed in Lines 11-15 in Algorithm 1.

Fig. 4.6 further illustrates the *MapProfit* procedure in Algorithm 2. Fig. 4.6(a) shows a toy PPI network with each protein assigned an initial information budget of $b = 2$. Fig. 4.6(b) shows the selection of the **cell-cycle** functional subgraph with structural information value $\psi^{C(u)} = 1$. Observe that the information budget remaining for each affected protein

Table 4.2: Summary of datasets used.

Dataset	#nodes	#edges	Source
<i>H. sapiens</i>	9181	34624	HPRD [113]
<i>S. cerevisiae</i>	4768	177299	IntAct [97]
<i>D. melanogaster</i>	3114	6472	IntAct
Alzheimer’s disease (AD)	177	1038	IntAct

is updated accordingly. Fig. 4.6(c) shows the remaining information budget when another functional subgraph (*apoptosis* with $\psi^{C(u)} = 1$) is selected. Finally Fig. 4.6(d) depicts the summary of functional subgraphs selected.

4.6 Experimental Results

We have implemented **FUSE** in Scala and Java. We now present the experiments conducted to evaluate the performance of **FUSE** and report the results obtained. The PPI datasets employed in this study are shown in Table 4.2. *Biological Process* (BP), *Molecular Function* (MF), and *Cellular Component* (CC) GO annotations are used. Unless specified otherwise, we set $\beta = 0.01$, $b = 3$, and $d = 0$ in order to balance *coverage* and *redundancy* of the functional summaries. We assign all edge weights be 1.0. All experiments were run on a 1.66GHz Intel Core 2 Duo T5450 machine, with 3GB memory, and a 250GB SATA disk.

4.6.1 Evaluation Metrics

We used the *coverage* metric to evaluate the fraction of the annotated protein interaction network covered by a summary. A functional summary with high coverage is desirable because it is more representative of the underlying interaction network than a summary with low coverage. The

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF
PROTEIN INTERACTION NETWORKS

coverage of a functional summary Θ is defined as:

$$coverage(\Theta) = \frac{\left| \bigcup_{C(u) \in S_\Theta} V(u) \right|}{\left| \bigcup_{C(u) \in S_\Delta} V(u) \right|} \quad (4.5)$$

The coverage is the ratio of the total number annotated proteins in the summary over the total number of annotated proteins in the protein interaction network.

The *redundancy* metric is the average number of functional clusters each protein belongs to. This is an indicator of the amount of cluster overlap in the summary. Redundancy of Θ is defined as:

$$redundancy(\Theta) = \frac{\sum_{C(u) \in S_\Theta} |V(u)|}{\left| \bigcup_{C(u) \in S_\Theta} V(u) \right|} \quad (4.6)$$

A summary Θ with no overlapping clusters will have lowest possible redundancy value of 1, where every protein is assigned to exactly one cluster. A summary with high redundancy is undesirable, because a summary with many highly overlapping clusters is less intuitive and more complicated.

The following well-known evaluation metrics are also used – *precision* and *recall*. These are well known statistical measures to indicate accuracy and completeness. Precision, a measure of exactness, is defined as:

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \quad (4.7)$$

Recall, a measure of completeness, is defined as:

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \quad (4.8)$$

If a cluster $C(i)$ is assigned with the function i , then any protein $p \in C(i)$ that is not annotated with i or its descendants is deemed a false positive. If

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.7: Cluster quality of **FUSE** vs graph clustering-based approaches (precision)

Figure 4.8: Cluster quality of **FUSE** vs graph clustering-based approaches (recall)

$p \in C(i)$ is annotated with i or descendants, it is a true positive. Likewise, a protein $p \in V$ that is annotated with i but not in $C(i)$ is deemed a false negative. Here, proteins without annotations are not taken into consideration.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.9: Cluster quality of **FUSE** vs graph clustering-based approaches (F-score)

Figure 4.10: Cluster quality of **FUSE** vs graph clustering-based approaches (FUSE)

4.6.2 FUSE vs Graph Clustering Methods

Dataset. Currently, there does not exist any gold standard to compare functional summaries of PPIS. Typically, biological graph clustering approaches use **MIPS** complex annotations [114] as gold standard data for testing cluster quality. These annotations, however, are limited to complexes and not for other functional clusters like pathways. GO annota-

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

tion data is also used as gold standard for evaluating clustering algorithms. As our approach utilizes attributes of GO, using GO annotations as gold standard evaluation may lead to results biased in favor of **FUSE**. Instead, we obtained a *different* set of curated attributes as gold standard—the *molecule class* annotations from HPRD—which is distinct from GO attributes. Note that these annotations are only available in the *H. sapiens* dataset. Consequently, we use this dataset for the comparative study. To create a gold standard *reference summary*, we generated a network from subgraphs induced from the HPRD network using nodes grouped by their *molecule class* attribute, signifying the molecular functional groups within the network. Subgraphs from five functional groups corresponding to subgraphs of proteins classified as **G protein coupled receptor**, **Protease inhibitor**, **RNA binding protein**, **Cytoskeletal associated protein**, and **Calcium binding protein** are extracted and merged to form the reference summary network (747 nodes, 959 edges). **FUSE** and state-of-the-art graph clustering methods are then evaluated on this network to determine whether the graph can be partitioned and summarized to reconstruct the gold standard functional groups.

We compare the performance of **FUSE** with four state-of-the-art graph clustering methods for life sciences applications, namely Markov clustering (**MCL**) [115], **MCODE** [17], and **nemo** [74]. We also compare **FUSE** with **csv** [101], a recent cohesive subgraph visualization method. Note that in order to obtain higher order modules of a PPI, the current approach is to first use an existing graph clustering method on the network to generate the clusters followed by function assignment. For example, in Krogan *et al.* [115], the global yeast PPI is first clustered using **MCL** to generate non-overlapping clusters. Then, each cluster is compared against **MIPS** complex annotations [114] and the complex annotation with the greatest overlap is assigned to represent the cluster.

4.6.3 Cluster Quality Comparison

We first emphasize on the qualities of an ideal summarization. First, the generated clusters have to be representative of the underlying graph, which implies that coverage of the clustering should be sufficiently high. Second, *attribute purity* [116] of the clusterings should correspond to the functional groups that were merged *a priori*. This can be determined through the purity of the `molecule class` attribute within the proteins in each cluster. Each functional group should also be well-represented. We use *precision*, *recall*, and *F-measure* to quantify these features. For each cluster, we determine the `molecule class` functional group that best matches the cluster. The *purity* of that cluster is then defined as the proportion of nodes in the cluster that belong to the best matching group. As a functional group may be represented by several smaller clusters, we define *recall* for each functional group as total coverage of the functional group among the clusters that best matches that functional group. Then, the *precision* of a clustering is defined as the average purity among all clusters. The *recall* of a clustering is defined as the average recall among all functional groups. Lastly, the *F-measure* ($\frac{2*precision*recall}{precision+recall}$) provides an overall measure of clustering quality.

Figures 4.7, 4.8, 4.9 and 4.10 depicts the results of summarization quality by F-measure, precision and recall. Where applicable, we adjust relevant parameters to vary the cluster granularity. As `nemo` has no parameter to tweak, only a single set of clusters can be obtained. In `mcl`, `csv`, and `mcode`, the *inflation*, *η_{mseen} cutoff*, and *node score cutoff* parameters are adjusted, respectively, to vary the cluster sizes (denoted as k in all figures). In `fuse`, the parameter k directly affects the summary granularity. Here, we use k to represent the number of clusters obtained by each method. Because most methods indirectly affects this via parameters, it may not be possible to cover the entire range of possible k values.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Observe that **FUSE** generates summary with significantly higher F-measure score compared to the graph clustering-based approaches for all values of k . In other words, **FUSE** may generate summaries at multiple levels of complexity while remaining representative of the underlying graph. Observe that, although **nemo**, **csv**, and **mcode** generate clusters with high precision, the recall scores are very low (< 0.2). This is because these two approaches identify highly cohesive subgraphs, which tend to be part of protein complexes. **csv** in particular are limited to identification of near-clique structures. Proteins in complexes belong to the same functional groups and hence the high precision. However as mentioned earlier, biological networks are not comprised solely of complexes. Consequently, majority of the underlying network was poorly represented by these approaches due to heavy bias towards complexes. Specifically, most of the clusters match the **RNA binding protein** class of proteins, leaving other groups barely represented. For instance, the **Protease inhibitor** subgraph is not well represented because of its inherent loose structure. Although the recall score of **MCL** is relatively higher as this method is known to perform very well in biological clustering applications, it is still below 0.4. Note that the **MCL** approach failed to partition the underlying network into five clusters representing the five functional groups. The **csv** approach, on the other hand, were not able to generate larger number of partitions.

Notice that these existing approaches indirectly affect the summary complexity whereas **FUSE** allows direct adjustment of summary size, which explains why summaries at any level of detail can be obtained by the latter. Figure 4.10 shows that **FUSE** generates summaries at different granularity without greatly affecting the precision and recall of the clusterings. The peak F-measure score of 0.8 is obtained in **FUSE** at $k = 5$, corresponding to the five gold standard functional groups that comprise the dataset. Observe that the recall and precision scores are equally high. As cluster

Figure 4.11: Function representativeness (precision)

granularity deviates from the underlying five functional groups, obviously the F-measure score drops.

4.6.4 Function Representativeness Comparison

The accuracy and representativeness of the function assigned to each cluster is key to generating high quality functional maps. Here, we introduce measures that quantify the representativeness of functions assigned to each clusters and compared **FUSE** to graph clustering methods in this aspect.

To obtain the functional landscape of a PPI, graph clustering methods often assign function to clusters through functional enrichment techniques. To this end, we compute the statistical significance of association of the cluster with every GO term based on the hypergeometric distribution [98]. The term with the best *p-value* is assigned as the *representative function* of the cluster. To evaluate the representativeness of this assigned function, we reuse the precision and recall measures introduced earlier with slight modification. Specifically, the *representative purity* of a cluster is defined as the proportion of nodes in the cluster that are annotated with the repre-

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.12: Function representativeness (recall)

sentative function. We also define *representative recall* for each functional group as total coverage of the functional group among the clusters that has the functional group assigned as representative function. Then, the *precision* of the representative functions is defined as the average representative purity among all clusters, and the *recall* of the representative functions is defined as the average representative recall among all functional groups.

Figures 4.11 and 4.12 depicts the representativeness of the functional summaries by different techniques. As **FUSE** is designed specifically to generate highly representative maps, each cluster is perfectly representative of the biological function assigned to it. Likewise, each function is well represented by its assigned cluster. In graph clustering methods, however, the clusters do not represent their representative function well, as indicated by the lower precision score. Hence, proteins within the clusters exhibit less functional coherence. The lower recall scores in graph clustering methods imply that only a fraction of nodes annotated with the representative function are included in the cluster. That is, **FUSE** summaries contain functional clusters that are more representative of the assigned function, and

thus provide more meaningful and interpretable higher-order functional maps of the underlying PPI. While clusters without attribute coherence may still reveal novel biological insights, assigning a function to represent such cluster could be misleading.

4.6.5 Qualitative Evaluation

Next, we qualitatively compared the summaries generated by both approaches for the DNA *S.cerevisiae* dataset. We argue that functional summaries are best evaluated qualitatively, partly because of the lack of a gold standard dataset for higher order function-function associations. We chose small and functionally specific subnetwork rather than large global networks so that qualitative comparison is feasible. To this end, we extracted the subnetwork containing DNA replication related proteins of *S.cerevisiae* network in IntAct ($n = 105$) as evaluation dataset. This dataset is obtained from the *S.cerevisiae* global network by extracting the induced subgraph whose proteins share **DNA-dependent DNA replication** function. Here, we compared our results qualitatively against the **mcl** approach. Figure 4.13A shows the **FUSE** generated functional map and Table 4.3 shows the **mcl** generated clusters. **FUSE** was able to partition the network into major components of DNA replication. Critically, **DNA polymerase complexes** (α, δ, ϵ) – key components in DNA replication – were obtained. **mcl**, on the other hand, was not able to obtain the polymerases. Deeper analysis reveals that many proteins in the *DNA replication factor C complex* cluster of Cluster+Enrich actually belong to DNA polymerases. This is a misrepresentation.

As shown in Figure 4.13B, proteins in **DNA polymerase complexes** α and ϵ and **DNA replication factor C complex** strongly interact with each other, forming a tight clique (with exception of 2 proteins). Hence, they cannot be separated via graph structure alone. In case of δ **DNA polymerase complex**, however, the situation is reversed. As it contains

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.13: Functional summarization of DNA *S.cerevisiae*. A) Summary at $k = 20$ obtained through **FUSE**. Nodes represent functional clusters. The size of a node correlates with the number of proteins that constitute the functional cluster. Edges represent associations between functional clusters; the stronger the association, the thicker the lines. B) Underlying protein interaction networks of α -DNA polymerase, ϵ -DNA polymerase, δ -DNA polymerase, and DNA replication factor C complex.

incomplete interaction data, the cluster does not appear to be densely interacting relative to other clusters. This could explain why **mCL**, which is highly dependent on structural data, did not identify the complex.

Figure 4.14: Effect of k on summary SIC.

4.6.6 Effects of User-Defined Parameters

Effect of parameter k . Recall that the user-defined parameter k controls the granularity of the summary. Intuitively, as k increases the amount of information contained within the summary as well as its complexity increase. Figure 4.14 reports the effect of k on the summaries of test datasets. As k increases, the *summary information content* (SIC), denoted by $SIC(\Theta)$, rises rapidly until it saturates to a peak value before tapering off.

$$SIC(\Theta) = \sum_{C(u) \in S_{\Theta}} -\psi^{C(u)} |V(u)| \log p_{V(u)} \quad (4.9)$$

where $p_{V(u)}$ is the probability that a protein in network is annotated with term u or its descendants. Note that summary profit cannot be used for comparing summaries with different k values because it does not make any assumption about the information content of a GO term attribute. In contrast, SIC measure is summary profit with a twist – small clusters are weighted higher than large clusters. This allows one to compare information content of summaries with different k values. Other factors being equal, a

Figure 4.15: Effect of k on summary coverage.

summary with many small clusters will contain more information than a single large cluster. The above results imply that k is useful up to a certain value, after which increasing k only increases summary complexity while providing little information gain.

Figure 4.15 plots the effect of k on coverage of the summary. Observe that except for low k values, it is relatively stable as k varies. In fact, the amount of information a summary can provide is limited by the resolution and completeness of the interaction and annotation data. This could explain why *S. cerevisiae* summaries have consistently higher coverage and information content than *D. melanogaster* summaries. The *H. sapiens* summary contains the largest number of nodes and edges, and even at $k = 600$, information content is still increasing. The smaller AD network, however, reaches a peak of information content at $k = 20$.

Effect of parameters b and d . We investigated the effect of user-defined parameters b and d on summary coverage and redundancy. We use the global *S. cerevisiae* dataset with $k = 100$. Figure 4.16 shows that increasing b or decreasing d lowers overall summary redundancy at the ex-

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.16: Effect of b and d .

pense of lower summary coverage. On the other hand, when d is increased or b is decreased, both summary redundancy and coverage increases. An intuitive explanation of this phenomenon is that more cluster overlap penalty means fewer combination of clusters to choose from, and therefore lower likelihood of finding a combination of clusters with high coverage. Both parameters allow users to control the coverage and redundancy trade-off, depending on whether it is preferable to have more coverage or less redundancy.

4.6.7 Statistical Significance

In this subsection we evaluate the statistical significance of a FUSE generated summary. Evaluation of graph clusters is not trivial because there is no analytic solution for the exact p-value of a cluster. However, an upper bound can be computed to detect if the density of a subgraph of a given size is statistically distinct compared to one that is randomly constructed. Given the graph G , we utilize the following upper bound derived from [117]

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.17: Running times of **FUSE** (in sec.).

Table 4.3: Summary of DNA *S.cerevisiae* obtained through Cluster+Enrich (9 single member clusters are excluded).

Cluster	Precision	Recall
replication fork protection complex	0.8	0.67
postreplication repair	0.8	0.8
DNA recombination	0.77	0.53
nuclear origin of replication recognition complex	0.86	1.0
GINS complex	0.43	0.75
negative regulation of cell cycle process	0.6	0.75
alcohol metabolic process	0.5	0.5
DNA replication factor C complex	0.22	1.0
rRNA metabolic process	1.0	0.5

as the p – value of a functional subgraph cluster:

$$P(R_{\rho_1} \geq (1 + \epsilon) \log n / \kappa(\rho_1, \rho)) \leq \frac{(1 - \rho)^{0.5}}{2\pi\rho^{0.5}} \frac{(1 + \epsilon) \log n}{n^{\epsilon(1+\epsilon) \log n / \kappa(\rho_1, \rho)}} \quad (4.10)$$

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Table 4.4: The p-value significance of FUSE clusters.

Cluster size	Cluster size to satisfy p-value	p-value
4	1.940572357	1.99E-06
5	2.381391052	1.83E-05
4	2.51869916	3.31E-05
4	2.51869916	3.31E-05
3	2.51869916	3.31E-05
3	2.51869916	3.31E-05
3	2.51869916	3.31E-05
3	2.51869916	3.31E-05
3	2.51869916	3.31E-05
3	2.51869916	3.31E-05
3	2.51869916	3.31E-05
7	2.740737755	7.99E-05
9	2.781159841	9.31E-05
5	2.839581494	1.16E-04
5	2.839581494	1.16E-04
5	2.839581494	1.16E-04
5	3.485484996	9.72E-04
5	3.485484996	9.72E-04
14	3.810967312	0.002456578
10	3.97437195	0.003801536
6	4.467241236	0.012844867
8	4.557278228	0.015816221

where:

$$\epsilon = \frac{r - \log n / \kappa(\rho_1, \rho)}{\log n / \kappa(\rho_1, \rho)} \quad (4.11)$$

$$\kappa(\rho_1, \rho) = \rho \log \frac{\rho}{\rho_1} + (1 - \rho) \log \frac{1 - \rho}{1 - \rho_1} \quad (4.12)$$

where ρ is the expected probability of observing an edge between two nodes, R_ρ is the size of the maximum subset of vertices that induce a ρ -dense subgraph, r is the subgraph size, and n is $|V|$.

Using the p -value above, we compute the p -value upper bound of a given FUSE cluster and the associated cluster size needed to satisfy the upper bound. In Table 4.4, we show the upper bounds p -value significance of at most 0.05 and the cluster size needed to satisfy the bound. Observe that all of the clusters we obtained from FUSE summary is at least as

large as the required size. Thus, these clusters have p – *values* that are significant. One weakness of the above formulation is that not all clusters can be associated with a meaningful p – *value* upper bound (there are bounds larger than 0.05, thus they cannot be used to meaningfully assess significance).

4.6.8 Effect of Annotation Loss

In this subsection we evaluate the effect of loss of annotations on FUSE algorithm. We observe how a summary changes when annotations are gradually removed from the PPI network. To achieve this, we first let Θ_0 be the FUSE summary generated using the full annotation dataset. Next, we remove a fraction of the annotations. Let the *annotation loss rate* be the fraction of annotations removed. For example, the annotation loss rate of 0.5 implies that half of the annotations in the PPI network has been removed. Given this measure, we compared FUSE summaries of annotation loss rate from 0.05 to 1.0 against Θ_0 generated from the human PPI network.

To measure the similarity a pair of summaries, we employed the *Jaccard index* (JI) [118] evaluation measure. Given two summaries Θ_i and Θ_j , the *Jaccard index* is defined as $J(\Theta_i, \Theta_j) = \frac{A}{A+B+C}$, where A is the number of protein pairs that is co-clustered in both Θ_i and Θ_j , B is the number of protein pairs co-clustered in Θ_i but not Θ_j , and C is the number of protein pairs co-clustered in Θ_j but not Θ_i . $J(\Theta_i, \Theta_j) \in [0, 1]$ and $J(\Theta_i, \Theta_j) = 1$ if the summaries are identical.

Figure 4.18 shows the effect of annotation loss on FUSE summaries. We observe that the Jaccard index similarity between Θ_0 and a summary with annotation loss rate gradually decreases as more annotations are removed. The gradual drop in similarity suggests that our approach is robust against loss of annotation.

Figure 4.18: Effect of annotation loss.

4.6.9 Runtime and Scalability

Figure 4.17 plots the running times of **FUSE** over the real datasets for generation of summaries ranging from $k = 3$ to $k = 600$. Observe that it increases almost linearly with k . Specifically, summarization of the yeast interaction network (the largest available network) completes within 40 minutes for $k = 600$. For practical sizes of $k = 3$ to $k = 100$, a functional summary of a PPI can be generated within few minutes. Disease networks such as AD network can be completed in less than 10 sec.

We now assess the scalability of **FUSE** with respect to network size and $|S_\Delta|$. Note that the latter feature is important as it will continue to grow as more annotation information becomes available. To assess the scalability with respect to network size, we generated synthetic networks of vertex size $|V| = 100$ to $|V| = 20000$. Note that the largest available PPI (human network) has only around 9000 vertices. For every term t , a vertex has a 2% probability of being annotated with it. The number of terms is $|S_\Delta| = 2769$. The *edge density* of the synthetic networks is such that the

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.19: Scalability of **FUSE** versus annotation size.

probability that a pair of vertices interact is 0.0025, resulting in an average of 1 million edges in a network of 20000 vertices. Summary granularity is set to $k = 50$. To measure the effect of $|S_\Delta|$ on running time, we generated synthetic networks by varying $|S_\Delta|$ ranging from $|\Delta| = 100$ to $|\Delta| = 10000$.

Figures 4.19 and 4.20 depicts the scalability of **FUSE** with respect to $|V|$ and $|S_\Delta|$. As the number of vertices increases, the execution time of **FUSE** increases in a quadratic fashion. In fact, it appears to increase almost linearly for networks with $|V| < 10000$. For larger networks, the $\psi^{C(u)}$ component and the FSG generation component take up the bulk of the execution time. Observe that in Figure 4.20, the FSG generation component takes up bulk of the computation time and is independent of $|S_\Delta|$. As $|S_\Delta|$ increases, $\psi^{C(u)}$ computation and iterative cluster selection time increases in near linear fashion, demonstrating ability of **FUSE** to handle high-dimensional annotation data.

4.7 Case Study on AD Network

In this section, we construct a low and a high resolution functional summaries of the AD network to illustrate the benefits of **FUSE** in providing a

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.20: Scalability of **FUSE** versus vertex size.

higher level functional view of the underlying PPI. A low resolution summary delineates broad functional overview of the processes related to the disease whereas a high resolution summary provides in-depth functional landscape of the disease, revealing associations between processes related to the disease. Figure 4.2 shows a low resolution summary ($k = 10$) of the AD network. It indicates that the AD network is represented by an inter-connection of several key processes, include protein phosphorylation ($B7$), cell-cell signaling ($B2$, $B3$), and microtubule-based transport and localization ($B1$, $B5$) processes.

Figure 4.1 depicts a high resolution functional summary for $k = 30$. Defective transport mechanism has major implications in AD. Consequently, several transport and cytoskeleton organization related cellular processes are represented in the summary ($A11$, $A22$, $A24$, $A26$). Disrupted transport mechanism affects, among others, synapse organization and vesicle trafficking ($A6$, $A8$, $A23$). In the literature, several lines of evidence explain disruption of transport and its related processes in AD. **Amyloid- β** plaques may lead to hyperphosphorylation of **tau proteins**, subsequently causing microtubule defects and axonal transport impairment [119]. More strik-

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

ingly, recent findings indicate that vesicle transport itself play a causative role in pathogenesis of the disease [120]. Glucose metabolic processes (*A20*) is closely linked to microtubule-based processes (*A22*, *A24*). The link between bioenergetics and transport in AD has been discussed in [121].

At the center of the summary lies protein folding and calcium ion homeostasis pathways (*A15*, *A17*). Protein misfolding is central to AD pathogenesis [122]. Misfolded amyloid- β accumulation is shown to induce calcium overload, leading to a variety of structural and functional disruption in neurons [123]. The two functional clusters are among the nodes with the highest degree in the summary. Cell fate processes that trigger or inhibit differentiation and cell fate (*A9*, *A10*, *A12*) are also linked to AD [124]. It has been suggested that Wnt signaling dysregulation, a key developmental pathway, leads to reduced synaptic plasticity and function in AD [125]. Processes such as peptide cross-linking and negative regulation of angiogenesis (*A3*, *A4*) imply vascular roles in AD pathogenesis [126].

From signaling regulation perspective, five major signaling pathways are implicated – small GTPase (*A28*), Notch (*A14*), Wnt receptor (*A18*), glutamate (*A21*), and G-protein coupled receptor signaling pathways (*A16*). Several functional clusters connect with multiple signaling pathways, indicating that signaling pathways crosstalk in AD pathogenesis. For instance, the serine/threonine kinase GSK-3 β , a potential therapeutic target, is known to be regulator of both the G-protein coupled receptor pathway and the Wnt/ β -catenin signaling pathway [127]. PS1 may be involved in regulating both Notch and Wnt pathways in AD [128].

The tight interplay of multiple pathways and processes in the aforementioned functional summary of AD network highlights the complexity of the disease. The disease remains poorly understood despite decades of research. While the summary does not suggest causal relationships, in part because of the undirected nature of the FSG, we hope that by having

Table 4.5: High-degree CC functional clusters in the *H. sapiens* summary ($k = 400$).

CC Functional Cluster	Degree
heterogeneous nuclear ribonucleoprotein complex	183
cytosolic large ribosomal subunit	161
cytosolic small ribosomal subunit	158
coated pit	158
mitochondrial nucleoid	149
chaperonin-containing T-complex	148
CRD-mediated mRNA stability complex	141
NuA4 histone acetyltransferase complex	136
actin filament	135
actomyosin	134
clathrin coat of coated pit	133
nonhomologous end joining complex	124
endocytic vesicle membrane	124
nucleosome	124
nuclear inner membrane	123

a global, big picture view of process-process interactions, researchers can better identify the causative mechanisms of AD. Most studies considered an aspect of the processes in isolation. An integrative study, however, may lead to a more consistent view of the disease that addresses distinct, often competing hypotheses.

4.8 Inferring Functional Cluster Hubs

Structural information provided by the summaries presents an opportunity to study the topology and connectivity of higher order abstractions of the underlying PPI. Here we analyzed the association patterns of functional clusters in summaries of the global *H. sapiens* PPI. To this end, we generated cellular component (CC) and biological process (BP) summaries of the human network. For each summary type, we varied the level of detail by setting k from 50 to 400.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.21: Connectivity of functional clusters in *H. sapiens* network. Functional cluster degree CDF plots for BP and CC summaries at varying cluster granularity. Plots are on a semi-log scale.

Figure 4.21 shows the frequency-degree plots of the functional clusters at different k values. At the broadest level of abstraction ($k = 50$), long-tailed degree distribution of functional clusters is not observed. As level of detail increases to $k = 400$, the smaller and more specific clusters exhibit increasingly pronounced long-tailed distribution characteristics. We note that the CDF plots on a semi-log scale form straight lines at higher k values ($k = 200$ and $k = 400$), implying exponential distribution of the cluster degrees.

In light of heavy-tailed distribution of functional cluster degrees at higher k values, we identified *functional cluster hubs* in the summary of the human network ($k = 400$) (analogous to identification of protein hubs). While Patil and Nakamura defined hub as proteins having degree of more than 6 [129], we chose a higher threshold such that they correspond to the 15 most connected functional clusters. The list of functional hubs is shown in Table 4.6.

We observed that CC cluster hubs in *S. cerevisiae* can be categorized

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Table 4.6: High-degree BP functional clusters in the *H. sapiens* summary ($k = 400$).

BP Functional Cluster	Degree
actin filament bundle assembly	208
regulation of defense response to virus by virus	206
negative regulation of catabolic process	204
peptidyl-threonine phosphorylation	200
signal complex assembly	189
positive regulation of protein complex assembly	182
regulation of nitric oxide biosynthetic process	181
glial cell development	178
cell killing	178
regulation of cytokine-mediated signaling pathway	174
protein stabilization	174
actin filament capping	170
activation of MAPKK activity	169
T cell receptor signaling pathway	164
regulation of RNA splicing	164

into several major functional groups. A significant percentage of the cluster hubs – such as cytosolic large ribosomal subunit, cytosolic small ribosomal subunit, eukaryotic translation initiation factor 4F complex, preribosome, small subunit precursor, preribosome, large subunit precursor, and polysome – are core to regulation and functioning of protein translation. It is unsurprising that these functional clusters have high degree, since every protein must be translated or regulated by these machinery. The complexity of this mechanism also suggests that it requires many processes to regulate it.

Complexes involved in chromatin remodeling and transcription, including nuclear nucleosome, Ino80 complex, replication fork protection complex, ASTRA complex, and Swr1 complex, are also highly represented. The functional cluster vacuolar proton-transporting V-type ATPase complex is known to have diverse roles and is associated with a wide array of processes [130].

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Apart from that, we also observed the existence of several ‘currency structures’ – structures that may be acted upon by proteins from multiple processes. They are generally not specific to a single biological process. We classify clusters `nuclear nucleosome`, `nuclear microtubule`, `cytoplasmic microtubule`, and `extra-cellular region` as such.

Next, we analyzed the BP functional cluster hubs. From Table 4.6, we found many translation related processes (`regulation of translational initiation`, `translational elongation`, `translational termination`, `tRNA aminoacylation for protein translation`, `negative regulation of translation`, `positive regulation of translation`, `ribosomal small subunit assembly`, `ribosomal large subunit assembly`). Chromatin assembly and remodeling processes (`nucleosome assembly` and `nucleosome disassembly`) also served as key process hubs. Finally, we found major post-translation protein modification and transport processes, such as `protein refolding`, `ATP synthesis coupled proton transport`, `co-translational protein targeting to membrane`, and `proteasome assembly`, acting as hubs.

4.9 Automatic Differential Summarization of dE-MAP networks

High-throughput mapping of genetic interaction networks of a set of genes is an important and emergent research problem [131]. The networks constructed with these methods, however, only represent a *static* “snapshot” of the genetic interaction map under a particular context or condition. Recent studies have shown that genetic interaction maps are in fact *dynamic* and *context-dependent* [132]. Consequently, there is a growing interest in studying the system-wide responses of interaction networks following environmental or condition change [133, 134]. For instance, one may be interested in elucidating the genetic interaction differences between cancer cells

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.22: The differential network that arises from two static E-MAP networks under different conditions. Red interactions – positive differential; green interactions – negative differential.

and normal cells. Specifically, some interactions may appear or disappear in the disease state, intensity of some interactions may alleviate or aggravate when in disease state compared to healthy condition, and others may remain strong irrespective of the state.

One representative method that has been recently proposed for mapping the genetic interaction responses following environment change is the dE-MAP approach [135]. In this method, two static gene interaction networks [131] for each condition are first obtained using the *epistatic miniarray profile* (E-MAP) approach [136], which constructs a quantitative genetic interaction landscape of *S. cerevisiae* by first identifying a set of genes of interest. Double mutant strains of all pairwise genes from this set of genes are then grown and their colony size measured. Genetic interaction occurs between a pair of mutant genes when one observes greater or lesser than expected colony growth rate when compared to their respective single mutant strains. When the growth rate is greater than expected, the interaction is deemed positive (alleviating); when it is lesser, it is deemed negative (aggravating). Using the two static E-MAP networks, a *differential net-*

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

work (dE-MAP network) is then computed that maps the interaction differences between the two static networks. For example, in [135], *S. cerevisiae* E-MAP networks are obtained for cells grown under two conditions: (a) cells which are treated with methyl methanesulfonate (MMS), a well known DNA-damaging agent and (b) cells which are untreated. Large-scale genetic interaction network among 418 yeast genes are quantitatively extracted using the E-MAP method under the MMS-treated condition (stressed) and untreated condition (unstressed) and the differential network that maps the genetic interaction changes due to MMS challenge is computed. Figure 4.22 depicts an example of a differential network (partial view) that is obtained from two static E-MAP networks under MMS-treated and untreated condition.

Naturally, it is important to analyze this differential network to investigate the system-wide impact of the DNA-damaging agent on the functional roles of various components. Consequently, the authors obtained physical protein-protein interactions corresponding to these genes and performed graph clustering to find protein complexes¹ enriched with differential interactions. The functional identity of each cluster is then *manually*² determined. Particularly, the authors concluded that these complexes tend to be stable across conditions and differential interactions largely lie between complexes, rather than within complexes. Unfortunately, modules constructed in this manner poorly represent the functional responses of the differential network. Hence, to find a functional response, the authors manually selected a subset of 31 genes associated with DNA Repair to test for differential interaction enrichment, concluding that DNA Repair is a pertinent functional response following MMS-treatment. However, it is time-consuming, laborious and error-prone to perform large-scale analysis of de-MAP interactomes to map all pertinent functional responses. *Here, we*

¹The topology of the differential network can be mined to identify gene clusters using techniques such as [17, 137, 138].

²A function can also be associated with each cluster by leveraging a *functional enrichment* technique [98].

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

propose a novel technique called *DiffNet* that addresses this impediment by automatically constructing a high quality differential summary of two *E-MAP* networks under environmental change. Figure 4.23 highlights some of these functional modules that are differentially effected by the DNA-damaging agent.

At first glance, the aforementioned failure of traditional graph clustering techniques to capture differential summaries in its modules may seem surprising. However, as we shall see in Section 4.11, these techniques are largely designed for static networks and are less suitable for differential networks that contain both positive and negative weights. Furthermore, since most methods rely solely on topology of the network, there is also no guarantee that each cluster corresponds well to a representative biological function response. In fact, as remarked earlier, in [135] the functional identity of each cluster following graph clustering is manually determined. Furthermore, the authors failed to assign function to a significant number of these clusters.

In fact, algorithms that perform genome-wide functional analysis of gene responses under multiple conditions have been proposed in the literature [139, 140, 141]. Particularly, these approaches perform functional analysis based on the expression levels of genes. In contrast, in our problem we focus on genome-wide functional analysis of the *gene interactions* and their responses.

Given the differential network generated from *de-MAP* interactions, *DiffNet* greedily constructs a *differential summary* comprising of a set of *skewed* and *coherent functional subgraphs*, representing significant functional responses following environment or condition change. Specifically, it leverages Gene Ontology (GO) annotations to identify these functional subgraphs, each of which represents a group of interactions corresponding to a specific biological function. A key characteristic of these functional subgraphs is that

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.23: Differential functional summary of MMS-induced/untreated yeast de-MAP network in [135]. The color of the functional modules and gene interactions indicate either positive differential (red) or negative differential (green). The thickness of the lines indicate the strength of the differential response. Gene interaction subgraphs of selected functional modules are also shown. Edges between functional modules depict differential interactions that occur between functional modules. The thickness of these edges represent the skewness of the differential interactions between a pair of functional modules. The most significant of such edges are shown.

the interactions together respond *significantly* in one direction, either positively or negatively, to the condition change. That is, unlike standard graph clustering methods, **DiffNet** is specifically designed to handle *differential interactions*, which can be positively or negatively weighted. Figure 4.24 illustrates the idea of the **DiffNet** algorithm. We shall elaborate on it in the next section.

4.10 Problem Formulation

The set of genes of interest together with their genetic interactions can be modeled as a gene-gene interaction network, denoted by $G = (V, E, w)$, where V is a set of genes selected for E-MAP study, E denotes the pairwise

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.24: Illustration of DiffNet. Red interactions are positive differential, while green interactions are negative differential. a) A functional subgraph represents interacting genes that share a specific function (*e.g.*, C1 represents gene interactions associated with DNA repair). A coherent functional subgraph has differential interactions that mostly respond in one direction. We say that a functional subgraph has high skew if the differential interaction weights have high magnitude; it has high coherence when the interactions largely respond in one direction. A functional subgraph with high coherence and skew represents a concerted, significant functional response due to the condition change. b) The DiffNet algorithm implements a greedy heuristic that selects, at each iteration, the functional subgraph with highest coherence and skew from the remaining unselected interactions. c) The output of DiffNet is a decomposition that summarizes the relevant functional responses due to condition change.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

interactions between genes, and w is a function that assigns each pairwise interaction $e \in E$ a weight that represents its interaction strength. In E-MAP studies, $w(e)$ of $e \in E$ is given by its genetic interaction score *S-score* [136]. A positive *S-score* indicates the degree of alleviating interaction between the two genes whereas a negative *S-score* indicates the degree of aggravating interaction. Therefore, $w(e)$ can be positive or negative.

Consider now two E-MAP networks $G_t = (V, E, w_t)$ and $G_c = (V, E, w_c)$ that represent two conditions: (a) the treated condition (G_t) and (b) the untreated condition (G_c). Observe that G_t and G_c share the same set of vertices and pairwise interactions. Given G_t and G_c , the differential network of G_t and G_c is a graph $G_d = (V, E, w_d)$ such that $\forall e \in E$:

$$w_d(e) = \left(1 + e^{-\frac{w_t(e) - w_c(e)}{|w_c(e)|}}\right)^{-1} - 0.5 \quad (4.13)$$

We apply the logistic function $(1 + e^{-x})^{-1}$ (shifted by 0.5 to make it an odd function) to “clip” potentially large magnitudes of differential responses. This is inspired by a similar approach used in activation functions in neural networks to bound the response of signals [142].

Intuitively, a differential network models gene interaction responses due to condition change. The differential weight $w_d(e)$ represents the normalized difference in *S-scores* between the two conditions for a pair of genes represented by e . We call $w_d(e)$ *positive differential* when $w_d(e) > 0$, and *negative differential* when $w_d(e) < 0$. A positive (*resp.* negative) differential response indicates increased alleviating (*resp.* aggravating) interaction between the two genes in treated condition compared to untreated condition. The magnitude of $w_d(e)$ reflects the strength of interaction response due to condition change. Figure A.1 shows a toy differential network of positive (red) and negative (green) differential interactions. Grey colored interactions do not respond to condition change (*i.e.*, $w_d(e) \approx 0$). The interaction between RAD52 and SIN3, for instance, has a positive differential response due to condition change.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

It is worth noting that the above definition of differential interaction w.r.t DNA damage-induced **de**-MAP network is consistent with the one in [135]. Specifically, a positive differential interaction indicate DNA damage-induced lethality, while a negative differential interaction indicate inducible epistasis or suppression. Importantly, the differential response does not distinguish, for example, one that goes from negative to positive from one that goes from positive to more positive. Although the former is arguably more interesting, the latter still is biologically significant because it indicates a significant response due to treatment.

Although we now have a model of individual gene-gene interaction responses due to condition change, it remains unclear how one *automatically* infers broader, systemic functional responses from these detailed interactions. This issue is pertinent in high-throughput experiments, which often generate thousands, even millions, of interacting genes within a single experiment. Hence we present our approach to model responses due to condition change from a functional perspective.

4.10.1 Functional Subgraphs in a Differential Network

We begin by modeling a systemic functional response by a subgraph of functionally-similar gene interactions (*i.e.*, a set of genes of a specific function and their interactions). Similar to FUSE, we introduce the notion of functional subgraphs (adapted to genes with differential interactions). We utilized the FUSE algorithm to with some modification to the problem formulation to construct **DiffNet** summaries. See the Appendix for the details of the formulation of the *the differential summarization problem*.

4.10.2 The DiffNet Algorithm

Unfortunately, the differential summarization problem defined in the preceding section is NP-hard. Hence, we describe an algorithm called **DiffNet**

that solves this problem heuristically. Here, we adopt a greedy algorithm that admits a H_k -approximation algorithm for the weighted minimum k -set cover problem [143], where $H_k = \sum_{i=1}^k \frac{1}{i}$. First, the differential network G_d is computed. Following that, **DiffNet** finds the universe of candidate functional subgraphs of G_d . The basic principle of **DiffNet** is to select, at each iteration, the functional subgraph that gives the best differential score contribution to the existing $\mathcal{S}$. At each iteration, we choose a functional subgraph that maximizes the total differential score. To achieve this, the algorithm maintains a map of interactions of G_d that is represented by currently selected functional subgraphs. For every candidate functional subgraph evaluated for selection, we evaluate its contribution to the remaining unselected interactions. The greedy algorithm then chooses the candidate subgraph that adds the highest differential score to the current summary. This process is iterated until k subgraphs have been selected. Because the penalty of choosing a remainder subgraph is always higher than any functional subgraphs, we let the remainder subgraph, if any, be the last subgraph. Given k passes and the worst case of evaluating $|E|$ edges per subgraph, the proposed algorithm has a worst case complexity of $O(k|\Delta||E|)$.

4.11 Results

The **DiffNet** algorithm is implemented in Scala. We now present the experiments to study the performance of **DiffNet** and report some of the results here. The experiments were conducted on a 1.66GHz Intel Core 2 Duo T5450 machine with 3GB memory. Unless specified otherwise, we set $k = 45$ and $\alpha = 5.0$.

Functional analysis of MMS-treated/untreated dE-MAP network.

Using the two E-MAP networks in [135], we constructed the differential functional summary associated with MMS treated/untreated genetic inter-

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.25: Functional summary of stable modules.

actions. Figure 4.23 shows the differential functional summary of the yeast genetic interactome. We observe significant positive differential functional subgraphs associated with DNA damage and DNA integrity checkpoint. The chronological cell aging genes responsible for stress-resistance – *MSN2*, *MSN4*, *RIM15* [144] – also undergo significant genetic interaction remodeling following DNA damage. This important and top-scoring functional response is not identified using manual analysis in [135]. Another type of functional modules that demonstrate significant differential following MMS treatment are pathways related to apoptosis and cell cycle, such as the G1 phase of mitotic cell cycle and cell aging modules. More interestingly, we observe significant negative differential responses in cell projection and cell wall biogenesis functions. The manual functional enrichment study conducted in [135] did not uncover the negative shift of these less obvious groups of genes.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.26: Comparison with general graph clustering algorithms.

To contrast the differential functional summary, we also constructed a summary of functional subgraphs that shows subgraphs of genes whose genetic interactions remain largely unaltered after **MMS** treatment. To this end, instead of constructing the differential network G_d , we constructed an “inverse” differential network $G_s = (V, E, w_s)$, such that $w_s = \min((w_d(e))^{-1}, \epsilon^{-1})$ where $e \in E$ and ϵ represents a pseudocount that prevents $w_s(e) \rightarrow \infty$. Observe that w_s represents the inverse of normalized S -score differences. We applied **DiffNet** on G_s to obtain a landscape of “stable” functional subgraphs, that is, functional subgraphs that are neither strongly positive differential nor strongly negative differential.

Figure 4.25 shows the functional summary of G_s following **MMS** treatment. The modules represented in this summary could be “housekeeping” processes and modules whose genetic interaction strength remain unaltered regardless of the DNA-damage challenge [135]. For instance, the composition and interaction of the subunits of the RNA polymerase enzyme, a critical module of the cell regardless of cellular context, is unlikely to change. Thus, their genetic interactions should also remain stable. One can make the same argument for preribosome.

Comparison with graph clustering algorithms. Since there is no existing technique that automatically generates differential functional summaries, we are confined to compare **DiffNet** with several representative graph clustering methods such as **MCL** [18], Affinity Propagation (**AP**) [137], and **ClusterONE** [138]. We used the dataset in [135] containing 418 genes

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

(393 with annotations). In particular, we chose the **MCL** and **ClusterONE** approaches as a recent evaluation demonstrated that both these methods outperform other graph clustering algorithms on biological networks [138]. Because **MCL** and **ClusterONE** do not accept negative edge weights, they cannot be directly applied to differential networks. To this end, we constructed two separate networks from a differential network – (a) a *positive network* containing only positive differential edges and (b) a *negative network* containing only negative differential edges. We assessed whether individually clustering both networks using general graph clustering methods, and then aggregating the clusters into one list, could provide results similar to those generated by **DiffNet**. For all approaches, we discarded clusters with fewer than 3 genes and selected the 25 best scoring clusters for cluster quality evaluation.

To quantitatively evaluate the quality of the clusters, we introduce several evaluation measures. Given a set of cluster subgraphs $\mathcal{S}$, the *average coherence* and *average skewness* are given by:

$$AvgCoherence(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{C_T \in \mathcal{S}} coherence(C_T) \quad (4.14)$$

$$AvgSkewness(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{C_T \in \mathcal{S}} skew(C_T) \quad (4.15)$$

To assess the functional relevance of each cluster, we used the annotation over-representation analysis of the clusters [145]. To this end, the *functional homogeneity* of $\mathcal{S}$ is given by :

$$FuncHomo(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{C_T=(E_T, V_T) \in \mathcal{S}} -\log(p-value(V_T)) \quad (4.16)$$

where $p-value(V_T)$ is the most significant GO term enrichment $p-value$ score of the genes in V_T .

Figure 4.26 plots the results of different approaches. Observe that **DiffNet** is superior to the clustering techniques in the following ways. First, each subgraph in **DiffNet** has a direct association with a biological function. Recall that functional subgraphs have the constraint that

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.27: Effect of parameter k on DiffNet.

every gene in a subgraph must share a specific function. With graph clustering algorithms such as MCL, each subgraph cluster may contain genes with diverging functions. In that case, it is unclear what biological function the cluster represents. This is quantified by the superior functional homogeneity score of DiffNet. Second, subgraphs in DiffNet have superior coherence compared to other methods. Traditional graph clustering methods are not designed to identify clusters of positive differential interactions and negative interactions. These methods must cluster negative and positive edges independently, and the information encoded in the mixture of positive and negative weights is lost. Third, our method is the second best performer for skewness score. This shows that despite fulfilling multiple summarization constraints, the clusters obtained have high skewness (*i.e.*, high edge weights) scores comparable to general graph clustering methods. Fourth, the ‘node-based’ decomposition in MCL do not admit overlapping genes. Consider for instance the subgraph $C3$ in Figure 4.24. If this subgraph is chosen as a cluster in MCL, then the subgraph $C4$ cannot

Figure 4.28: Effect of interaction noise.

be another cluster because of gene overlap. The ‘edge-based’ decomposition of **DiffNet**, which we argue is a more natural way of grouping interaction responses, does not suffer from this problem.

Effect of parameter k . Figures 4.27(a)-(d) show the effect of k on summary coherence, skewness, coverage and distinctiveness. We observe that k controls the trade-off of summary coverage versus distinctiveness. The higher the value of k , the greater the coverage of functional subgraphs in the summary. However, the increase in coverage reduces the quality of the clusters (lower skewness, coherence and distinctiveness) due to the fact that one must now include lower quality clusters to satisfy the coverage requirement. Note that it is unrealistic to expect the majority of differential interactions to respond significantly to condition change. Thus, full coverage of all interaction responses, especially those that respond weakly, is typically not required in a differential summary.

Effect of interaction noise. Given that interaction profiles are likely to be noisy, we evaluate the effect of interaction noise on **DiffNet** summary construction. We assume the **DiffNet** summary generated from the differential network in [135] is without interaction noise and use it as the reference summary. We then simulate the effect of noise by perturbing the interactions of the network by random rearrangement of its interactions.

The amount of perturbation is indicated by the *interaction noise rate*, which is the fraction of the original interactions that have been randomly rearranged. Figure 4.28 shows the stability of the **DiffNet** summary after interaction noise perturbation. At each noise rate, we simulate 10 perturbed network samples. We compute the Jaccard similarity of the functional subgraphs of a perturbed summary ($\mathcal{S}_1$) against the reference summary ($\mathcal{S}_2$). Specifically $JaccardSimilarity(\mathcal{S}_1, \mathcal{S}_2) = 1$ if the gene set of each functional subgraph in $\mathcal{S}_1$ and $\mathcal{S}_2$ is identical. As expected, we observe a steady decrease in similarity against the reference summary with increasing interaction noise rate.

4.11.1 Weakness of independently clustering positive and negative edges of differential network.

Figure 4.30 shows the separation of a toy differential network into a *positive network* containing only positive differential edges and a *negative network* containing only negative differential edges. When each of the positive or negative network is clustered independently, the information about the other is lost. Consequently, both networks independently show enriched positive and negative interactions. When these interactions are put together, however, the differential interactions of these genes have weak skewness and coherence due to the mixing of positive and negative interactions.

4.11.2 Effect of parameter α .

Figures 4.29(a)-(d) show the effect of α on summary coherence, skewness, coverage and distinctiveness. We observe that α directly controls the influence of summary coherence. The higher the value of α , the greater the coherence of functional subgraphs in the summary. The increased coherence, however, comes at a cost. Coverage of the summary is reduced with greater α . This is because the increase penalty for choosing incoherent

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.29: Effect of α on DiffNet.

Figure 4.30: Independently clustering positive and negative edges of differential network.

functional subgraphs reduces the exploration space during decomposition selection. Distinctiveness is also slightly increased with greater α .

4.11.3 Running time.

We generated synthetic networks by randomly adding nodes and edges to the [135] dataset network until the desired size is obtained. Figure 4.31 plots the running times of DiffNet of varying network sizes (viewed by number of nodes and edges, respectively). We observe that DiffNet scales almost linearly with the number of nodes and edges in the network. A differential network of 2500 nodes is summarized in less than 3 minutes.

CHAPTER 4. FUSE: TOWARDS MULTI-LEVEL FUNCTIONAL SUMMARIZATION OF PROTEIN INTERACTION NETWORKS

Figure 4.31: Running time of DiffNet.

Figure 4.32: Running time of DiffNet at varying network density.

This shows that DiffNet constructs a summary within a reasonable time frame.

We further evaluated the running time of DiffNet at varying network density. Figure 4.32 shows the running time on [135] dataset network from 10% density (0.1) to full density (1.0). We artificially construct networks at varying density by randomly removing network edges until the desired density is achieved. From the figure, running time of DiffNet grows almost linearly with the network density.

Figure 4.33: Effect of annotation loss on differential summary construction.

Figure 4.34: Effect of MCL inflation parameter.

4.11.4 Effect of annotation loss on differential summary construction.

As current gene annotations are likely to be incomplete, here we study the effect of gradually removing gene annotations on DiffNet summary construction.

Suppose $\mathcal{S}_0$ is a reference DiffNet summary of the [135] differential network with complete gene annotations. We then constructed DiffNet summaries of differential networks with removed annotations and observed

their similarities with the reference summary. Given two summaries $\mathcal{S}_1$ and $\mathcal{S}_2$, the similarity of the functional subgraphs between the summaries can be measured using the following:

$$JaccardIndex(\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{|\mathcal{S}_1|} \sum_{C_1 \in \mathcal{S}_1} \max_{C_2 \in \mathcal{S}_2} \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|} \quad (4.17)$$

where $JaccardIndex(\mathcal{S}_1, \mathcal{S}_2) = 1$ if the gene set of each functional subgraph in $\mathcal{S}_1$ and $\mathcal{S}_2$ is identical. We removed $n\%$ of the gene annotations from the differential network and constructed a new summary $\mathcal{S}_n$. We call $\mathcal{S}_n$ a summary of the differential network with $n\%$ *annotation loss*. Figure 4.33 shows the *JaccardIndex* similarities of summaries with varying annotation loss. We observe that annotation loss creates a summary that is increasingly different from the reference summary. The drop in *JaccardIndex* similarity is gradual, suggesting that **DiffNet** summary construction is relatively robust to annotation noise. More importantly, as annotations of genes are likely to increase with time, it will only lead to more improved performance of **DiffNet**.

Effect of MCL inflation parameter. Figures 4.34(a)-(b) show the effect of MCL inflation parameter on summary coherence and skewness. Here we use the recommended range of 1.4 to 5.0. We observe that the coherence and skewness of functional subgraphs in the summary are stable with varying inflation values. There is, however, a slight increase in coherence and a slight drop in skewness at higher inflation values.

4.12 Software Availability

Implementation of **DiffNet** can be found at the following:

- **DiffNet**: <http://www.cais.ntu.edu.sg/~assourav/DiffNet/>

4.13 Conclusions

We propose **FUSE**, a novel data-driven and generic algorithm to generating functional summaries at multiple resolutions from a PPI to provide a high level view of its functional landscape. It generates the “best” summary from both interaction and annotation data by maximizing information gain for a specific resolution. Our experimental study with real-world PPIS revealed that **FUSE** is effective and have higher accuracy compared to graph clustering techniques. It is also robust against incomplete interaction knowledge (e.g., AD network in *IntAct*). We note that the graph clustering techniques have the ability to uncover novel complexes whereas **FUSE** is designed to determine process-process, complex-complex, and process-complex associations with higher confidence. In this aspect, graph clustering and **FUSE** play complementary roles. Finally, we propose **DiffNet**, a novel data-driven algorithm that automatically constructs summaries of differential functional responses of gene interaction networks under environment or condition change. Specifically, it leverages combination of GO annotation information and underlying interaction data to greedily identify a set of functional subgraphs that are highly skewed and coherent, representing significant functional responses due to condition change.

Chapter 5

FACETS: Multi-faceted Functional Decomposition of Protein Interaction Networks

In this chapter, we propose a novel PPI decomposition algorithm called **FACETS** in order to make sense of the deluge of interaction data using Gene Ontology annotation data. **FACETS** finds not just a single functional decomposition of the PPI network, but a *multi-faceted atlas* of functional decompositions that portray alternative perspectives of the functional landscape of the underlying PPI. Each *facet* in the atlas represents a distinct interpretation of how the network can be functionally decomposed and organized. Our algorithm maximizes interpretative value of the atlas by optimizing *inter-facet orthogonality* and *intra-facet cluster modularity*. We tested our algorithm on the global networks from **IntAct**, and compared it to gold standard datasets from **MIPS** and **KEGG**. We demonstrated the performance of **FACETS**. We also performed a case study that illustrates the utility of our approach. Parts of this chapter is published in [146].

5.1 Motivation

Recall that graph clustering algorithms [17, 115, 94] discover regions of dense connectivity that represent protein complexes or functionally coherent processes. Unfortunately, these methods output *only a single optimal*

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

Figure 5.1: Illustration of multi-faceted PPI network decomposition.

functional decomposition of the PPI network. Consequently, a PPI network can only be decomposed and viewed from a single perspective, whereas in reality there are often multiple different perspectives (decompositions) associated with the functional organization of the underlying network, all of which are distinct and equally valid. We refer to each of these decompositions as a *facet* because they visualize the organization of a PPI network from a unique view, providing a distinct interpretation of the organization of the underlying network. For example, consider the toy transcriptional regulatory network depicted in Figure 5.1. A typical decomposition, based on an existing graph clustering technique (*e.g.*, MCODE in [17]), identifies dense regions of the network, which correspond to the decomposition of protein complexes as shown in *Facet 1*. However, this network can also be viewed from other different perspectives. For instance, it can be organized by the types of signaling pathways involved in it (*Facet 2*). Notice that the decomposition from this perspective is markedly different from the complex-based decomposition. Furthermore, different proteins in the network may undergo various modifications such as acetylation, phosphorylation, and ubiquitination. Hence, yet another way to decompose the network is by their modification effects as depicted in *Facet 3*. Clearly, in larger real-world networks the possibility of uncovering multiple, distinct functional decompositions is real.

At first glance, it may seem that we can tune the clustering param-

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

ters of existing graph clustering techniques in order to generate multiple facets or decompositions. Unfortunately, such tuning only generates an exponential number of *slightly perturbed* decompositions with incremental differences (see Results). In other words, such strategy does not generate functionally unique decompositions. In contrast, it is imperative to ensure that the decompositions or facets are *distinctive*, *i.e.*, they are maximally different from each other. This is because every facet should provide a fresh and informative perspective to the organization of the network, rather than providing just incremental differences with respect to other facets.

Our contribution. We propose a novel algorithm called **FACETS** that discovers an *atlas* of functionally unique decompositions from a PPI network, portraying alternative views of the functional landscape of the network (detailed in Section 5.3). Each decomposition or facet represents a distinct interpretation of how the network can be functionally decomposed and organized. Since a key objective is to obtain n unique facets that are informative and orthogonal¹, our algorithm maximizes interpretative value of the atlas by optimizing *intra-facet cluster modularity* and *inter-facet orthogonality*. *Intra-facet cluster modularity* captures the aim of decomposing a PPI network G based on a particular functional and/or structural view. For instance, based on complexes and localized structures, G can be decomposed into protein complexes. If we consider regulatory processes as a functional concept, then G can be decomposed into signaling and regulation pathways, an entirely different decomposition. *Inter-facet orthogonality*, on the other hand, demands that the n facets are structurally distinctive and functionally apart from each other. We propose a novel *objective function* that models these intuitions and **FACETS** exploit it to discover a set of distinct facets. Specifically, we exploit both the PPI graph structure and the rich functional information provided by Gene Ontology (GO) annotations

¹We use the term orthogonal to describe the idea of distinctive clusters, rather than its precise mathematical meaning.

to guide facets construction.

5.2 Related work

Multi-view clustering is a poorly studied problem in the data mining community [147]. Still, there are several works that have focused on multi-view clusterings in image and text mining domain [148]. One approach projects data into an alternative subspace [149]. Another approach generates alternative clustering through the use of *must-link* and *cannot-link* constraints [150]. In Meta-Clustering [151], a large number of clusters are generated, and clusters which are truly different are selected. All of the aforementioned approaches, however, assume data points in the *vector space* that allow the notion of metric distances in a Euclidean geometry. On the other hand, our problem demands a multi-view clustering methodology on *attributed graphs*, which requires a graph clustering paradigm on both structure and annotation. To the best of our knowledge, multi-view clustering paradigm has not been applied in clustering biological networks to identify pertinent functional modules from multiple perspectives.

Ensemble clustering methods generate an ensemble of near-optimal decompositions [152, 153, 78]. These methods have been used to increase the quality and confidence of the decomposition and understand network dynamics. The near-optimal decompositions generated, however, have no notion of the orthogonality that this work is seeking. Instead, ensemble clusterings create a large number of perturbed solutions, making them unsuitable as an atlas of functionally distinct decompositions. For instance, in [73], a small network of 32 nodes generated at least 82 permutations of clusterings.

5.3 Problem Statement

In this section, we formally introduce the multi-faceted functional decomposition problem. We begin by defining some terminology that we shall be using in the sequel. We use the network in Figure 5.1 as running example.

5.3.1 Terminology

A *facet* (*decomposition* or *view*) of G , denoted by F , is a set of functional modules $\{C_1, \dots, C_m\}$ representing a specific functional concept. Functional modules within a facet F are allowed to overlap. In the sequel, we use the terms facet, view, and decomposition interchangeably. A *functional atlas* (or atlas for brevity) of G , denoted by A , is a set of facets $\{F_1, F_2, \dots, F_n\}$ that represents distinctive functional landscapes of G . Figure 5.1 depicts an atlas of 3 facets, with each facet decomposing the network into 3 functional modules.

Similar to **FUSE**, we utilize GO annotations associated with proteins to define the multi-faceted functional decomposition problem. Given a GO directed acyclic graph (DAG) $D = (V_{go}, E_{go})$, the ordered set $\Delta = \langle \Delta_1, \Delta_2, \dots, \Delta_d \rangle$ is a topological sort of D , where Δ_i represents a single GO term. Each vertex $v \in V$ is associated with a d -dimensional *function association vector* $\Delta_v \in \{0, 1\}^d$, such that $\Delta_v = \langle \Delta_1^v, \Delta_2^v, \dots, \Delta_d^v \rangle$, $\Delta_i^v \in \{0, 1\}$ where $\Delta_i^v = 1$ if and only if the term $\Delta_i \in D$ or its descendants are associated with protein v , and $\Delta_i^v = 0$ if otherwise.

A *facet candidate bundle* $B_i = \{G_1, G_2, \dots, G_m\}$ is a set of connected subnetworks of G such that for every $G_k \in B_i$, there is a shared GO term Δ_i within every $v \in V_k$. Δ_i represents the common function of the candidate subnetwork. A facet candidate bundle B_i represents the superset of facet F_i , and it contains a large permutation of subnetworks that satisfy a particular functional concept. Typically, $|F_i| \ll |B_i|$. A *function bundle* $\omega_i = \{\Delta_1, \Delta_2, \dots, \Delta_m\}$ is the set of shared GO annotations of B_i ,

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF
PROTEIN INTERACTION NETWORKS

i.e., $\omega_i = \bigcup_{G_k \in B_i} \Delta_{G_k}$. To illustrate these concepts, consider the PPI network in Figure 5.1. Suppose that B_1 is a facet candidate bundle with $\omega_1 = \{\Delta_1, \Delta_2\}$, where Δ_1 represents the **Swr1 complex** GO term and Δ_2 the **Histone** term. In the subgraph with ‘**Swr1 complex**’ label in Facet 1, every node in that subgraph is annotated with **Swr1 complex** term. Thus, the subgraph is a valid member of B_1 . Any subgraph made up of ‘**Swr1 complex**’-labeled nodes is also a valid member of B_1 . If B_2 represents the facet candidate bundle with $\omega_2 = \{\Delta_3\}$, where Δ_3 represents **cellular component**, then the ‘**Swr1 complex**’-labeled subgraph is also a valid member of B_2 (**Swr1 complex** is a **cellular component**). Furthermore, every subgraph in Facet 1 whose nodes are labeled is a valid member of B_2 , but not necessarily a valid member of B_1 . One can see that B_i contains a set of subgraphs that shares specific functional concepts depending on the functional terms in ω_i . We define the function $f: \mathcal{P}(V_{go}) \rightarrow A$ given by $f(\omega_i) = F_i$ to make explicit the association between a functional bundle and its corresponding facet.

A *function bundle partition* $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$ is the set of function bundles that form a partition of all GO terms V_{go} , *i.e.*, $\bigcup_{\omega_i \in \Omega} \omega_i = V_{go}$. In the next section, we shall impose further constraints on facet candidate bundles and function bundles such that the shared GO terms of the subnetworks within each facet candidate bundle shares high functional commonality and the terms shares in one facet are distinct from the terms in another facet.

5.3.2 Multi-faceted Functional Decomposition Problem

The goal of multi-faceted functional decomposition problem is to identify an atlas of n distinct facets of G that maximizes *inter-facet orthogonality* and *intra-facet cluster modularity*. Each facet depicts a higher-order organization of modules of G . Recall that inter-facet functional orthogonality demands that each of the n facets is based on an orthogonal functional con-

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF
PROTEIN INTERACTION NETWORKS

cept – facets that are distinctive and functionally apart from each other. Hence, we propose two criteria that model the intra-facet functional modularity and inter-facet orthogonality of an atlas solution. Next, we propose an *objective function* that models and scores an atlas of G .

Intra-facet cluster modularity. Intra-facet cluster modularity enables us to seek clusters that are both structurally and functionally modular. Given ω_i , Ω , and G , ω -restricted decomposition procedure (denoted by g_ω) computes a decomposition of G into F_i such that F_i satisfies the following criteria:

Criterion 1. Every module $C_j \in F_i$ should be *functionally bounded* by ω_i . Let $D_{C_j} = \{\Delta_1, \Delta_2, \dots, \Delta_m\}$ be the set of shared terms in C_j , *i.e.*, for every $v \in V_c^j$, v must be annotated with every $\Delta_i \in D_{C_j}$. Then, the *functional boundedness* of module C_j by ω_i is given by $r(C_j, \omega_i) = D_{C_j} \cap \omega_i$. A cluster C_j is bounded by ω_i if $r(C_j, \omega_i) \neq \emptyset$. An ω_i -restricted decomposition of a facet draws from a restricted search space of subnetworks in G whose vertices share at least a term within ω_i . Intuitively, this means that for any subnetwork to be considered as a module, it must first be sharing a term in ω_i . Even if a subnetwork is dense, it must yield to sparser subnetwork candidates if it is not enriched with terms within ω_i . In the example of Figure 5.1, if ω_1 is terms of protein complexes, then any subgraphs enriched with complex terms is in the search space for *Facet 1*. In contrast, the modules of *Facet 2*, enriched with signaling terms, would be invalid candidates for *Facet 1* decomposition. This restricted search space is modeled by facet bundle B_i , where any valid candidate facet cluster C_j of facet F_i must belong to B_i .

Criterion 2. A facet F_i decomposes G by maximizing a clustering objective function $o(F_i)$ while satisfying the above criterion. $o(F_i)$ is determined by the specific graph clustering algorithm that is adapted for creating a facet; for generality we let this be the objective function $o(F_i)$ that has

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF
PROTEIN INTERACTION NETWORKS

to be maximized by the graph clustering algorithm. For instance, every module $C_j \in F_i$ has to be structurally dense and/or functionally coherent (*i.e.*, every node in module shares a common function), the coverage of F_i has to be high, and the amount of overlap between modules should be low. For example, modules of *Facet 2* maximize $o(F_2)$ while satisfying the ω_2 bound, despite not forming dense modules. This is because all dense modules formed are enriched with complex terms, violating the ω_2 bound.

Inter-facet orthogonality. Since we want every facet in the atlas to be functionally and structurally distinct, modules within a facet, as whole, should be structurally and functionally distinct from modules within another facet. We discuss two independent distance measure between facets: *functional orthogonality* and *structural orthogonality*.

Functional orthogonality is indirectly controllable by the function bundles attached to facets, which determines the types of allowable modules through the aforementioned restriction. By increasing inter-bundle functional orthogonality, we increase the functional distinctiveness of each facet. To impose functional orthogonality, we introduce the following constraint: for every $\omega_i, \omega_j \in \Omega$, $\omega_i \cap \omega_j = \emptyset$ and $\bigcup_{\omega_i \in \Omega} = V_{go}$. This requires that Ω actually partitions the terms of the GO DAG. The *functional distance measure* between Δ_i and Δ_j , denoted by $d(\Delta_i, \Delta_j)$, measures the functional dissimilarity between the terms. Here, $d(\Delta_i, \Delta_j)$ is simply computed as the length of the shortest path between the terms: $d_f(\Delta_i, \Delta_j) = \min_{\Delta_r \in R} |p(\Delta_r, \Delta_i)| + |p(\Delta_r, \Delta_j)|$, where R is the set of common ancestors of term Δ_i and Δ_j and $|p(i, j)|$ is the length of the shortest path from node Δ_i to Δ_j in D . The *candidate function specificity* $s(\Delta_i, C_u)$ is defined as $s(\Delta_i, C_u) = \frac{|\{\Delta_i \in \Delta_v | v \in V_c^u\}|}{|\{\Delta_i \in \Delta_v | v \in V\}|}$. $s(\Delta_i, C_u)$ measures the specificity of a shared GO term, which we will later use to weigh the contribution of the term. For instance, a cluster C_j of 5 nodes that share the **biological process** GO term in a network of 1000 **biological process** annotated

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

nodes has a low specificity value of 0.005 with respect to the term.

Likewise, we define structural orthogonality. The *structural distance measure* between two clusters C_u and C_v is defined as

$$d_s(C_u, C_v) = 1 - |E_C^u \cap E_C^v| / |\{(v_i, v_j) | v_i \in V_C^u \cap V_C^v, v_j \in V_C^u \cup V_C^v, (v_i, v_j) \in E_C^u \cup E_C^v\}| \quad (5.1)$$

$d_s(C_u, C_v)$ measures $1 -$ the ratio of the number shared edges between C_u and C_v over the number of edges incident to $V_C^u \cap V_C^v$. The distance is 0 if C_u and C_v shares all edges and 1 if C_u and C_v shares no edges.

Following that, we define $t(\Omega, A)$ as the linear combination of inter-facet functional and structural orthogonality, as follows:

$$\begin{aligned} t(\Omega, A) = & \gamma \sum_{\substack{\omega_i, \omega_j \in \Omega \\ i \neq j}} \left\{ \sum_{\substack{\Delta_j \in D_{C_j} \\ C_j \in f(\omega_j)}} \sum_{\substack{\Delta_i \in D_{C_i} \\ C_i \in f(\omega_i)}} s(\Delta_i, C_i) s(\Delta_j, C_j) \frac{d_f(\Delta_j, \Delta_i)}{|V_p^j| |V_p^i|} \right\} \\ & + (1 - \gamma) \sum_{\substack{\Delta_u \in D_{C_u}, C_u \in F_i \\ F_i \in A}} \sum_{\substack{\Delta_v \in D_{C_v}, C_v \in F_j \\ F_j \in A, i \neq j}} s(\Delta_u, C_u) s(\Delta_v, C_v) d_s(C_u, C_v) \end{aligned} \quad (5.2)$$

The parameter γ weighs the contribution of d_s against d_f , and is set to attain balanced contribution from both terms. Note that $t(\Omega, A)$ quantifies the pairwise orthogonality between two function bundles. The higher the score, the greater the orthogonality.

5.3.3 Problem Definition

The multi-faceted functional decomposition of G is defined as the problem of simultaneously constructing the atlas of decompositions $A = \{F_1, \dots, F_n\}$, and the function partition $\Omega = \{\omega_1, \dots, \omega_n\}$, such that the following objective function is maximized:

$$\begin{aligned} \max_{A, \Omega} \quad & \lambda t(\Omega, A) + (1 - \lambda) |A|^{-1} \sum_{F_i \in A} o(F_i) \\ \text{subject to} \quad & C_s \in B_i \forall C_s \in F_i, 1 \leq i \leq n \end{aligned} \quad (5.3)$$

The right half of the terms captures the cost function of decomposing G into A ; the left half, decomposing D into Ω . The parameter $\lambda \in [0, 1]$

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

Figure 5.2: Illustration of the **FACETS** algorithm. a) GO annotated PPI network is used as input. b) The set of candidate subnetworks are computed c) An initial set of modules are randomly assigned to a facet. Candidate subnetworks are then assigned to their nearest facet based on function and structure distance. d) For each facet, decomposition is performed to identify modules that are functionally contained by the facet candidate bundle. e) The candidate subnetworks are reassigned based on their distance to the new set of modules identified. Convergence is achieved when the number of terms reassigned to a different facet drops below the threshold parameter θ . Otherwise, steps d-e are repeated.

controls the weight between the two terms. Observe that one has to optimize these criteria simultaneously over the space of A and Ω . Otherwise, one may end up with a poor objective score. For instance, if $t(\Omega, A)$ is high (meaning highly orthogonal partitioning), but Ω is improperly partitioned such that one ends up with ω_i that allow only poor decompositions, then the $o(F_i)$ score would be very low. Due to the interdependence of the criteria, optimizing the aforementioned function is computationally expensive.

5.4 FACETS Algorithm

Generally, the problem of finding clusters that maximizes typical clustering objective functions that relate to graph density is known to be NP-hard [154]. Hence the **FACETS** algorithm is a heuristic implementation that attempts to find a local maximum of the objective function. Our heuristic is a step-wise iterative approach that incrementally optimizes Ω and A , one at a time. Intuitively, given an attributed PPI network (*e.g.*, Fig. 6.1a), Ω is incrementally updated by using each facet in A as functional centroids, and then using the centroids to partition D . A is updated through ω -restricted decomposition using the updated Ω . The **FACETS** algorithm consists of two phases: the *initialization* phase (Fig. 6.1b), and the *iteration* phase (Fig. 6.1c-d). We describe each of them in turn.

Initialization. This phase creates an initial set of decompositions for the second phase. We perform graph clustering on G to obtain an initial set of modules. To this end, the **FUSE** [94] algorithm is utilized. Each module of this set is then randomly associated with a facet, randomly distributing the modules over an initial set of facets. Following that, we construct *candidates subnetworks*, which use subnetworks of G that satisfy ω_i -restricted decomposition constraint. To generate candidates exhaustively is prohibitively expensive. Instead, candidates for a facet F_i are generated as follows: for every GO term $\Delta \in \omega_i$, we obtain the induced subnetwork in G whose nodes are annotated with Δ or its descendants. The subnetwork is then decomposed into connected components, each forming a candidate subnetwork G_j . Let $\Delta_j^C = \Delta$ be the term associated with this candidate. Candidates formed this way can vary greatly in resolution of the annotation that its nodes share (for example, $\Delta_j^C = \text{biological process}$), and can be highly overlapping.

Iteration. This phase – the actual optimization phase – is performed in rounds. Let i denote the i -th iteration of the algorithm. At each round, the

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF
PROTEIN INTERACTION NETWORKS

algorithm updates A and Ω in two sequential steps. To evaluate algorithm convergence, we introduce *functional reassignment* – the number of terms in Δ that is reassigned to a different function bundle after step 1 of i -th iteration. This score measures the rate of change of Ω , indicating how close the algorithm is to convergence. Observe that when Ω is fixed, the algorithm reaches a steady state. The algorithm reaches convergence and terminates when the functional reassignment at i -th iteration drops below *convergence threshold* θ , a user-defined parameter.

1) Update Ω . In this step, we assume that A is a constant and update Ω to increase $t(\Omega, A)$. For each $F_i \in A$, the enriched functional terms of the modules in F_i serve as centroids for partitioning D into orthogonal concepts; these enriched terms as whole form the centroid of ω_i , which is associated with F_i . We then reassign every candidate subnetwork to its nearest centroid to form a partition Ω . The convergence properties of such centroid-based partitioning approaches (*e.g.*, K-Means) has been well studied [155]. For every $G_j \in B_i, 1 \leq i \leq n$, we determine its *closest* centroid by considering G_j 's average functional and structural distance to functional modules within a facet. The facet that is closest to G_j is indicated by:

$$d_c(G_j, F_k) = \begin{cases} 1 & \text{if } \frac{1}{Z(F_k)} \sum_{C_i \in F_k} s(\Delta_i^C, C_i) \phi(C_i, G_j) \\ & \leq \frac{1}{Z(F_{k'})} \sum_{C_i \in F_{k'}} s(\Delta_i^C, C_i) \phi(C_i, G_j) \\ & k' \neq k, \text{ where} \\ & \phi(C_i, C_j) = \gamma \frac{d_f(\Delta_i^C, \Delta_j^C)}{|V_p^j| |V_p^i|} + (1 - \gamma) d_s(C_i, C_j) \\ & Z(F) = \sum_{C_i \in F} s(\Delta_i^C, C_i) \\ 0 & \text{otherwise} \end{cases} \quad (5.4)$$

Following that, G_j is reassigned to nearest facet candidate bundle B_k (superset of F_k) and Ω is updated based on where every $\Delta_j^C \in V_{go}$ is assigned to. Each function bundle $\omega_i \in \Omega$ represents functional terms that are most closely associated with F_i , and the decomposition of F_i in the follow-

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

ing step will be bounded by the updated ω_i . Function partitioning depends on the atlas of decompositions because not every partition of the GO DAG is capable of forming a modular decomposition of functional modules.

2) Update A . In this step, we update A to maximize the objective function while fixing Ω . To support ω_i -restricted decomposition of F_i , we propose an algorithm that employs profit maximization model (discussed below) and runs in iterations. At each iteration, we score candidate subnetworks based on a profit maximization model and greedily selects the best scoring candidate as member in F_i . An iteration runs for every $F_i \in A$ before moving to the next iteration. Every candidate considered for F_i must satisfy the ω_i -restricted decomposition constraint, *i.e.*, the candidate subnetwork must be enriched with terms in ω_i . In other words, $G_j \in B_i$.

We now describe the profit maximization model for scoring a candidate $G_j \in B_i$. Every $v \in V$ is assigned an information budget. A candidate G_j extracts, from each $v \in V_j^G$, some information revenue from the budget pool. The revenue extracted is correlated to the edge density of the subnetwork, with modular candidates giving high revenue. Each time a candidate is selected, revenue is removed from the budget pool and a cost is incurred. A penalty cost is incurred for a candidate that is structurally similar to selected clusters in other facets $F_{i'} \neq F_i$. This penalty is modeled by $cost(G_j) = \sum_{C' \in F_{i'}, i' \neq i} d_s(G_j, C')$, which utilizes the structural distance measure d_s described earlier. At each iteration, the candidate that contributes the highest information profit (revenue minus cost) is selected. To summarize, a clustering in F_i that yields high overall revenue have subgraphs with high facet modularity $o(F_i)$, while a clustering with low overall cost yields high inter-facet orthogonality $t(\Omega, A)$. Given a fixed Ω , the set of facets A with maximum overall profit maximizes the objective function. The algorithm above approximates this through greedy heuristic.

Algorithm 3 shows the steps of the **FACETS** algorithm. We illustrate the

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF
PROTEIN INTERACTION NETWORKS

Algorithm 3 Algorithm **FACETS****Input:** $G, \Delta, D, k, b, d, \beta, n, \theta$ **Output:** A

```

1:  $S = \text{FUSE}(G, \Delta, D, k, b, d, \beta)$ 
2:  $A = \{F_1, F_2, \dots, F_n\}$  where  $F_i = \emptyset$ 
3: for  $C(u) \in S$  do
4:    $F_r = F_r \cup C(u)$  where  $r$  is randomly 1 to  $n$ 
5: end for
6: while  $\text{reassignment}(\Omega_{old}, \Omega) > \theta$  do
7:    $\Omega_{old} = \Omega$ 
8:    $\Omega = \{B_1, B_2, \dots, B_n\}$  where  $B_i = \emptyset$ 
9:   for  $F_i \in A$  do
10:    for  $G_j \in B_i, B_i \in \Omega_{old}$  do
11:       $F_{min} = \text{argmin}_{F_i} d_c(F_i, G_j)$ 
12:       $B_{min} = B_{min} \cup G_j$ 
13:    end for
14:  end for
15:   $\Omega = \{B_1, B_2, \dots, B_n\}$ 
16:  for  $F_i \in A$  do
17:     $F_i = \text{FUSE}(G, \Delta, B_i, k, b, d, \beta)$ 
18:  end for
19: end while
20: for  $F_i \in A$  do
21:   for  $C(i), C(j) \in F_i$  do
22:    if  $C(i) \neq C(j)$  and  $P_i(X > oc_{C(i)C(j)}) \leq 2\beta/|S|^2$  then
23:      Add edge  $(C(i), C(j))$  to  $F$ 
24:    end if
25:  end for
26: end for

```

algorithm with the example shown in Figure 6.1. The initialization step constructs a set of initial candidate subnetworks and assigns them randomly to a facet (lines 1-5, Figure 6.1(b)-(c)). Following that, the update Ω and update A steps are performed iteratively until convergence (lines 6-19, Figure 6.1(d)-(e)). In the update Ω step, each candidate subnetwork is assigned to its nearest facet (lines 7-15, Figure 6.1(d)), while in the update A , a restricted **FUSE** profit maximization heuristic is performed to identify the best set of subnetworks that represent a facet (lines 6-19, Figure 6.1(d)). Finally, upon convergence, the network for each facet is constructed (lines 20-26, Figure 6.1(e)).

Table 5.1: Datasets used in FACETS.

Dataset	#nodes	#edges	Source
<i>H. sapiens</i>	9131	34362	IntAct [13]
<i>S. cerevisiae</i>	4768	40457	IntAct
<i>D. melanogaster</i>	3114	6472	IntAct
Human autophagy	1241	3555	IntAct

5.5 Results

5.5.1 Experiment settings

The **FACETS** algorithm is implemented in Scala ². We now present the experiments conducted to study the performance of **FACETS** and report some of the results here. All experiments were executed on a 1.66GHz Intel Core 2 Duo T5450 machine with 3GB memory. We primarily used the global human PPI network from *IntAct* [13], as well as the *yeast*, *fruit fly*, and *human autophagy* networks from *IntAct* (Table 5.1). In all experiments, we set the convergence threshold $\theta = 5$. The weight γ is set to 0.091 to balance the contribution of structure and function (equal order of magnitude). We utilize only the **Cellular Process** sub-domain of the Gene Ontology so that the facets are created not merely based on different GO domains, but created based on more subtle functional differences.

Evaluation criteria. To measure the similarity/dissimilarity between facets or decompositions, we employed the *Jaccard index* (JI) [118] evaluation measure, which is widely used to compare clusterings based on counting the agreement or disagreement of co-clustered pairs of proteins. Given two decompositions (or facets) f_1 and f_2 , the *Jaccard index* is defined as $J(f_1, f_2) = \frac{A}{A+B+C}$, where A is the number of protein pairs that is co-clustered in both f_1 and f_2 , B is the number of protein pairs co-clustered in f_1 but not f_2 , and C is the number of protein pairs co-clustered in f_2

²<http://www.scala-lang.org>

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF
PROTEIN INTERACTION NETWORKS

Table 5.2: Comparison between facets of the *H. sapiens* PPI network ($n = 6$).

Facet	#Modules	Coverage	JI score					
			Fct 1	Fct 2	Fct 3	Fct 4	Fct 5	Fct 6
1	89	294	1.0	0.014	0.065	0.0050	0.0070	0.079
2	280	1079	0.014	1.0	0.0040	0.119	0.0050	0.0070
3	106	372	0.065	0.0040	1.0	0.0010	0.0	0.013
4	94	419	0.0050	0.119	0.0010	1.0	0.0	0.0080
5	114	390	0.0070	0.0050	0.0	0.0	1.0	0.0010
6	72	306	0.079	0.0070	0.013	0.0080	0.0010	1.0

Facet	Coverage overlap					
	Fct 1	Fct 2	Fct 3	Fct 4	Fct 5	Fct 6
1	1.0	0.316	0.142	0.081	0.044	0.112
2	0.086	1.0	0.077	0.09	0.082	0.079
3	0.112	0.225	1.0	0.029	0.059	0.086
4	0.057	0.233	0.026	1.0	0.028	0.052
5	0.033	0.228	0.056	0.03	1.0	0.038
6	0.107	0.281	0.104	0.071	0.049	1.0

but not f_1 . $J(f_1, f_2)$ ranges from 0 to 1 (for identical clusterings).

5.5.2 Experiment results

Quantitative assessment. Table 5.2 shows the quantitative comparison between facets. We measure the inter-facet decomposition similarity using the JI score. The low clustering similarity scores between facets show that they are decomposed distinctively. This reflects significant organizational differences between modules of signaling pathways and protein complexes. We measured the *coverage* of a facet and the *extent* of coverage overlap between the facets. Let the coverage of a facet F_k be $Cvg(F_k) = |\bigcup_{V_c \in F_k} V_c|$. Also, let the extent of coverage overlap between F_i and F_j be $Ext(F_i, F_j) = \frac{|V_i \cap V_j|}{|V_i|}$, where $V_i = \bigcup_{V_c \in F_i} V_c$ and $V_j = \bigcup_{V_c \in F_j} V_c$. The extent of overlap between facets reaches up to 0.316. Consequently, the overlap is not insignificant, implying that the facets are not partitions of G .

Validation on real data. In this experiment, we compare the **FACETS** atlases of the global human network to gold standard functional modules. The gold standard datasets were constructed as follows: 1) **mips** – We use the set of 571 human complexes (of more than 3 proteins) from MIPS [114] to represent the decomposition of the human interactome into complexes. 2) **kegg-metabolic** – To represent decomposition into metabolic modules, we use 67 human metabolic networks from KEGG, each representing a single functional module. 3) **kegg-signaling** – We use 23 human signal transduction pathways from KEGG to represent decomposition into signaling pathways. The gold standard decompositions were chosen such that each represent a distinct functional organization of the human network. As such, we consider each gold standard dataset as a facet of the human network, and the set of these three datasets as the gold standard atlas of the human network. We then compared these datasets against the atlas of facets obtained through our algorithm and determine if there is a distinctive one-to-one mapping between our facet and a gold standard facet. We set $n = 6$ and repeated the tests fifteen times under different starting conditions to account for variability in **FACETS** output. We also compare the similarity scores against graph clustering methods, namely Markov clustering (**MCL**) [115], **MCODE** [17], **NeMo** [74], and **FUSE** [94]. These methods create a single decomposition of the human network. We removed clusters with fewer than 3 proteins. We also compare against GO term enrichment (enrich) [98], which does not utilize structural information. Following that, we measure the clustering similarities between the gold standard datasets and the decompositions obtained. Figure 5.3 shows the clustering similarities between modules in gold standard datasets and modules in facets as well as tested graph clustering methods. The Jaccard index were used to measure the agreement between pairs of decompositions. We normalize the scores so that the highest JI score obtained, within each gold standard dataset,

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

Figure 5.3: Comparison between the decomposition similarities of **FACETS**, other methods, and gold standard decompositions.

is adjusted to 1.

We consider the facet best associated with a gold standard decomposition by comparing their relative scores. The gold standard datasets were uniquely mapped to a distinct facet: **kegg-metabolic** is most similar to *Facet 3*, **kegg-signaling** is most similar to *Facet 2*, and **mips** is most similar to *Facet 1*. This unique mapping demonstrates that from a clustering perspective, the facets have significant functional orthogonality such that they are uniquely associated with different functional organization maps. *Facet 6* has poor similarity to the gold datasets, indicating a set of clusters that could be functionally distinct from these datasets.

In contrast, the tested graph clustering methods share common similarity patterns. Clusters are largely from a single dominant perspective – those of protein complexes (**mips**). We argue that objective functions based on dense connectivity tend to favor protein complex structures over other decompositions like metabolic pathways. GO term enrichment, on the other hand, generates output with little similarity to all gold standard datasets, indicating that annotations alone are unable to specifically identify important functional modules within a large PPI network. This is supported by the fact that functional analysis of large networks often involve graph clustering prior to term enrichment [115].

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

Figure 5.4: Effect of noise on **FACETS** algorithm.

Robustness. To study the robustness of **FACETS**, we tested the effect of annotation perturbations and edge deletions of the input network on **FACETS** output. Random edge deletion (*edge noise*) simulates the effect of removing false positive interactions in high-throughput interaction datasets, while annotation perturbation (*node noise*) simulates errors in curated annotations. Figures 5.4(a)-(c) show the effect of edge and node noise on **FACETS**, varying from 0% noise to 100% noise. The figures show clustering similarities (JI similarity) between the best scoring facets and gold standard datasets under increasing noise perturbations. We repeated each test fifteen times with different randomization seed. We observed that **FACETS** output quality drops gradually under increasing edge and node noise conditions. This demonstrates that the algorithm is robust to small noise perturbations. In case of edge noise, we noted that the quality of output only drops rapidly past the 0.5 noise ratio. This is desirable given that false positive rates in yeast two-hybrid and TAP experiments range between 0.35 to 0.7 [156]. MIPS clusters, which consist densely interconnected clusters, are most robust to edge noise effects. The effect of node noise is comparatively greater, but quality degradation remains gradual.

Effect of initial starting point. Given that **FACETS** belongs to the class of hill-climbing methods, the algorithm output is dependent on the initial starting point. To this end, we study the effects of multiple random

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

Figure 5.5: Effect of initial starting point versus noise on **FACETS** algorithm.

initial starting points. We compared the variability in clustering output due to starting point versus variability due to noise effects to give a sense of the magnitude of variability. We set a single FACET output as the reference output, and compared its JI similarity with outputs from different starting points and increasing noise effects. The boxplot Figures 5.5(a) and 5.5(b) show the effect of initial starting point versus noise on **FACETS**. At 0 noise rate, the variability in JI similarity is due to initial starting point. Given the fact that high throughput datasets are inherently noisy (as mentioned above), the variability due to starting point is less significant. In addition, Figures 5.4(a)-(c) show the effect of starting points with respect to gold standard datasets when one observes the similarity at 0 noise rate.

Convergence. Figures 5.6(a) and 5.6(b) show the functional reassignments after the i -th iteration. We conducted the tests on varying types of datasets with $n = 6$. We also vary the number of facets per atlas ($n = 2$ to 6) on the global human network. All tests converge in less than 9 rounds, demonstrating **FACETS**' ability to converge quickly to a solution. Larger datasets such as the human network requires more iterations to complete. The number of iterations required also tend to increase with number of facets n .

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

5.6.a: (Conv.) Datasets

5.6.b: (Conv.) n

Figure 5.6: Rate of convergence of **FACETS** algorithm .

5.7.a: $|V|$

5.7.b: n

Figure 5.7: Running time of **FACETS** algorithm.

5.5.3 Statistical Significance of FACETS clusters

We utilize the p – value bounds described in Chapter 4 to evaluate the statistical significance of FACETS clusters. Table 5.3 shows the most significant upper bound p – value scores and the cluster size needed to satisfy the bound. We note that all of the clusters we obtained from FACETS summary are at least as large as the required size needed to satisfy the upper bound. This indicates that FACETS clusters are more significant than random drawn subgraphs when assessed by their subgraph densities.

Table 5.3: The p-value significance of FACETS clusters.

Facet	Cluster size	Maximum size	p-value
0	5	1.76612927	6.49E-07
2	5	1.76612927	6.49E-07
4	5	1.76612927	6.49E-07
5	5	1.76612927	6.49E-07
3	8	1.783769936	7.36E-07
0	4	1.940572357	1.99E-06
1	4	1.940572357	1.99E-06
1	4	1.940572357	1.99E-06
1	4	1.940572357	1.99E-06
1	4	1.940572357	1.99E-06
1	4	1.940572357	1.99E-06
1	4	1.940572357	1.99E-06
2	4	1.940572357	1.99E-06
2	4	1.940572357	1.99E-06
3	4	1.940572357	1.99E-06
3	4	1.940572357	1.99E-06
4	4	1.940572357	1.99E-06
4	4	1.940572357	1.99E-06
5	4	1.940572357	1.99E-06
5	4	1.940572357	1.99E-06
5	4	1.940572357	1.99E-06
5	4	1.940572357	1.99E-06
5	4	1.940572357	1.99E-06
5	4	1.940572357	1.99E-06
1	6	2.037508859	3.44E-06
1	5	2.037508859	3.44E-06
3	6	2.037508859	3.44E-06
3	5	2.037508859	3.44E-06
5	5	2.037508859	3.44E-06
5	5	2.037508859	3.44E-06
5	11	2.152382327	6.25E-06
1	5	2.381391052	1.83E-05

5.5.4 Running time.

Figures 5.7(a) and 5.7(b) plot the running times of **FACETS** with varying network sizes $|V|$ and facet count n . Observe that the running time of **FACETS** on the largest network (human) is less than 3 minutes with $n = 11$ and less than a minute with $n = 2$.

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

Figure 5.8: Varying parameters of clustering methods.

5.5.5 Varying parameters of graph clustering methods yields delta differences

We evaluated whether graph clustering approaches can generate functionally orthogonal decompositions by varying their parameters. Figures 5.8(a)-(c) report the effect of varying parameters on the JI similarity scores between the gold standard decompositions and the clustering output of the human network. Despite varying parameters to generate different decompositions, the decompositions are still largely from a single perspective – those of protein complexes. This is indicated by highest clustering similarity to the `mips` dataset. We suggest that ignoring the clustering of protein complexes, which are dense modules, causes significant drop in the objective function score of clustering methods. Weaker clusters of other decompositions are hidden by the dominant complexes and are unlikely to be prioritized over complexes by adjusting the clustering parameters.

5.6 Case study: Human autophagy system.

To illustrate the utility of multi-faceted decomposition, we analyze the functional organization human autophagy system. Autophagy is the process where proteins and organelles are degraded [157]. Autophagosomal vesicles deliver such components to the lysosome or vacuole, where they are degraded. The autophagy system thus regulates the the expression

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

Figure 5.9: Multiple facets (subset) illustrating the functional organization of the human autophagy network under different perspectives.

of proteins, as well as removing defective components. Multiple diseases arise from the dysfunction of the autophagy system. It is thus relevant to study the organization of such system. The functional map of this system was manually constructed in [158]. The authors found many genes are significantly implicated in vesicle transport, GTPase signaling, proteolysism ubiquitination and phosphorylation.

We generated the facets of the human autophagy network ($n = 6$), and a subset of the results is shown in Figure 5.9. The automatically generated facets show the pertinent roles of vesicle transport and lipid

membrane metabolism in autophagy, which is consistent with the manually constructed map. For instance, we observe that transport role is a key facet of the autophagy system. This correlates with the finding in [158] that more than half of the proteins in the system are linked to vesicle transport. The genes implicated in vesicle function include **NEDD4**, **SQSTM1** and **FBNP1**. **NEDD4** has been implicated in endosomal protein degradation [159]. The **SQSTM1** has been previously found to be involved in recruitment of ubiquitinated cargo [159].

Additionally, the network can also be clustered from the perspective of cell cycle and apoptosis regulation modules, which is not depicted in the manual map. The **mTOR** inhibition occurs in association with **MAP1B**. Meanwhile, the **GABARAP** protein is an ortholog of the key autophagy associated protein **ATG8**, which is shown to be implicated in cell growth related signaling [158]. Other genes are found to be implicated in cell growth control, including **STK3/MST2** and **STK4/MST1** – the components of the Hippo kinase complex.

In summary, having multiple perspectives allows and explanation of the organization of a network from several angles.

5.7 Comparison with GO DAG

Here, we evaluate whether FUSE and FACETS generated summaries are superior to a baseline of simply taking GO terms of a certain level in the GO DAG. The baseline of GO Terms at Level 2, for instance, represent the GO terms that are located in level 2 of the GO DAG. We assume that by taking GO terms at a particular level and forming clusters using these terms, we could construct a set of clusters that represent a functional summary of the PPI network. To evaluate such baseline against FUSE and FACETS summaries, we consider two measures that evaluate the quality of clusters obtained. First, we use the *average cluster coherence* score to measure the

CHAPTER 5. FACETS: MULTI-FACETED FUNCTIONAL DECOMPOSITION OF PROTEIN INTERACTION NETWORKS

Table 5.4: Comparing GO terms at a particular level in the GO DAG.

Method	#Clusters	Distinctiveness	Avg. Cluster Coherence
GO Terms @ Level 2	34	0.155	0.006
GO Terms @ Level 3	82	0.130	0.012
GO Terms @ Level 4	290	0.119	0.027
GO Terms @ Level 5	534	0.106	0.023
GO Terms @ Level 6	797	0.121	0.029
FACETS	350	0.725	0.513
FUSE	150	0.835	0.359

average structural density of a cluster. To this end, it is simply the average of the *ratio association* [111] scores of clusters in a summary. The average cluster coherence score is used to evaluate the modularity of clusters in a summary. Distinctiveness (the inverse of the redundancy metric) measures the lack of cluster overlap in the summary, and is defined as:

$$distinctiveness(\Theta) = \frac{\left| \bigcup_{C(u) \in S_\Theta} V(u) \right|}{\sum_{C(u) \in S_\Theta} |V(u)|} \quad (5.5)$$

where $distinctiveness(\Theta) \in [0, 1]$. A high distinctiveness score implies that few overlap exists between clusters (*i.e.*, more interpretable summary), while a very low distinctiveness score implies that many of the clusters are significantly overlapping.

Table 5.4 presents our findings. Observe that both FUSE and FACETS summaries have significantly higher distinctiveness and average cluster coherence scores compared to the baseline. The average cluster coherence is at least 10 times greater than that obtained from the baseline (clusters of the summaries are strongly connected), while the distinctiveness is almost 5 times greater (less overlap between clusters). In general, summaries generated using our methods form modules or clusters that are much more interpretable and structurally significant.

5.8 Software Availability

Implementation of **FACETS** can be found at the following:

- **FACETS**: <http://www.cais.ntu.edu.sg/~assourav/Facets/>

5.9 Conclusion

We propose **FACETS**, a data-driven and generic algorithm for generating multi-faceted functional decompositions of a PPI network, providing multiple perspectives of the functional organization landscape of the network. Our experimental validation with real-world PPI networks demonstrates effectiveness of **FACETS** in generating functionally distinctive facets. These distinctive facets have higher relevance to real life datasets compared to single decomposition-based graph clustering techniques. We also show that our method converges rapidly to a solution with varying datasets and facet counts.

Chapter 6

DualAligner: Protein-protein Interaction Network Alignment via Dual Alignment Strategy

This chapter focuses on the final contribution of the dissertation – the *DualAligner* algorithm. DualAligner performs *dual network alignment*, in which both region-to-region alignment, where whole subgraph of one network is aligned to subgraph of another, and protein-to-protein alignment, where individual proteins in networks are aligned to one another, are performed to achieve higher accuracy network alignments. Dual network alignment is achieved in DualAligner via background information provided by a combination of GO annotation information and protein interaction network data. We demonstrate the advantage of such dual network alignment strategy compared to existing network alignment methods. We show that our results are superior in their accuracy using well-established metrics. We also show that the effects of parameters in DualAligner in controlling the quality of the alignment.

6.1 Motivation

As discussed in Chapter 2, network alignment methods are now key to understanding biology from a systems perspective, given the growing number of large scale protein-protein interaction (PPI) networks across multiple species or conditions. By aligning proteins from one network to another based on a multitude of measures, one may uncover interesting conserved regions of the PPI subnetworks. These conserved regions could explain mechanisms of function conservation and evolution beyond what is available to individual gene studies.

Recall that a range of network alignment methods have been developed that fall into two categories: *local network alignment*, which identifies locally conserved network regions and *global network alignment*, which aligns every node in the smaller network to the larger network [19, 91, 160, 85].

The local network alignment approach identifies multiple, unrelated conserved regions between the input networks, each region implying a mapping independent of others. It is particularly useful in finding known functional components (pathways, complexes) in a new species. For instance, PathBLAST [23] aligns linear pathways based on homology and interaction confidence. NetworkBLAST-M [81] finds highly conserved local regions greedily using inferred phylogeny. Another local alignment method is MaWish [82], which is modeled on evolution of protein interactions.

On the other hand, the global network alignment approach aligns every node in the smaller network to the larger network in order to find “best” overall alignment between the input networks. Particularly, it enables species-level comparisons and discovery of functional orthologs. For instance, IsoRank [20] and IsoRankN [91] identify a stationary random walk distribution to perform global network alignment. Græmlin 2.0 [83] uses a training set of alignments to learn phylogeny relationships before performing an alignment. MI-GRAAL [85] is an alignment framework that

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Figure 6.1: Illustration of DualAligner. a) GO annotated PPI networks are used as input networks. b) Function-constrained subgraphs (connected subgraphs sharing a GO function) are constructed, representing functionally coherent regions in each network. c) DualAligner computes a region-to-region alignment that best matches function-constrained subgraphs of the human network to the fly network. This involves pairwise subgraph to subgraph subalignment to identify optimally conserved subgraph pairs. Here, three pairwise alignments (f_1 , f_2 and f_3) are shown. Note that K_7 and K_8 are not chosen in the alignment – only best conserved regions are aligned. d) DualAligner computes fine-grained protein-to-protein alignment using the aligned regions in (c) as background information. Protein-to-protein alignments are *soft constrained* within subgraph-subgraph pairs – that is, any alignment between two proteins are *probabilistically* restricted within one or more subgraph-subgraph pairs in (c).

could integrate multiple similarity measures to construct a global alignment. [160] models global network alignment as a graph matching problem with hard constraints given by a set of *must-link* pairs of proteins. These *must-link* pairs are identified by sequence orthologs.

More recently, the ILP-based *Natalie* proposes a *Lagrangian relax-*

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

ation approach to solve the problem approximately [88]. PINALOG combines both the similarities of protein sequence and protein function to compute an alignment between two PPI networks [87]. To this end, the GO function similarity and sequence similarity are obtained independently and then combined using a linear combination of the independent similarities weighted by a factor θ .

Despite considerable progress made by the bioinformatics community in devising high quality network alignment strategies, state-of-the-art network alignment techniques suffer from a key drawback. Specifically, they depend on protein sequence similarity to facilitate network alignment. Unfortunately, sequence similarity is only relevant to a subset of highly conserved proteins, leaving significant network regions poorly specified by sequence homology. Furthermore, it is well-known that structural information of PPI networks suffers either from high *false positive* rate of current high-throughput experiments or from *false negatives* due to incomplete data [38]. *Consequently, in significant regions of a network, high confidence alignment of proteins is not possible.* At best, local network alignment attempts to alleviate this problem by completely ignoring low confidence mappings; global network alignment, on the other hand, pairs all proteins regardless.

To this end, we address the aforementioned limitation by taking a GO annotation-driven “*dual alignment*”¹ strategy to align a pair of PPI networks. Instead of *only* performing protein-to-protein alignments, we perform high granularity protein-to-protein alignment where confidence is high and more general functional *region-to-region* alignment where detailed protein-protein alignment cannot be ascertained, which may still yield biologically informative conclusions. Consequently, our method not only aligns highly conserved protein pairs, but also aligns *functional subgraphs* of one network to functional subgraphs of another. Informally, a *functional subgraph* is a connected component of the network whose nodes share a par-

¹here “dual” is unrelated to duality in optimization; it refers to two-step alignment.

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Figure 6.2: Hard constraint versus soft constraint.

ticular biological role or function (annotated using Gene Ontology (GO)).

The region-to-region and protein-to-protein alignments in our dual alignment strategy are not independent of each other, but inextricably linked. Specifically, region-to-region alignment sets the foundation for broad associations between functional regions. On the other hand, protein-to-protein alignment specifies detailed associations between nodes *within* each associated regions. Figure 6.1 illustrates with an example on how functionally conserved subgraphs are first aligned, followed by alignment of the underlying interaction structure. First, functional subgraphs of the networks are identified. For instance, a subgraph that shares the **transport** role is identified in the human network. Alignment between pairs of functional subgraphs is carried out and high confidence pairs are identified based on the structural and sequence similarities of their underlying subgraphs. For instance, F2 in Figure 6.1(b) depicts an alignment between the **transport**-associated subgraph and **microtubule binding**-associated subgraph, highlighting the connection between the regions. Using region-to-region alignments, we model each region-to-region pair as *soft constraints* that *probabilistically restrict* protein-to-protein alignment within the regions. Following that, protein-to-protein alignments are computed using these soft constraints as priors.

We refer to the aforementioned network alignment model as *function-*

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

*constrained network alignment*² (detailed in Section 6.2). It is worth mentioning the salient feature of our model. By integrating the rich set of functional information as soft constraints with the “hard” constraints [160] (*e.g.*, known orthologs) in the alignment framework, our dual alignment strategy not only limits the search space of “unconstrained” alignment to low knowledge regions but also guides alignment of functionally-informative regions under a reduced search space. Consequently, our approach identifies a suitable alignment from a smaller search space compared to state-of-the-art network alignment techniques, an important feature due to intractable nature of the network alignment problem. Additionally, by leveraging soft constraints, we have much lesser likelihood of generating alignments that conflict with known biological functions.

GO terms may capture functional information that arise from: 1) human curation, or 2) automatic curation using annotation algorithms (*eg*, derived from sequence information). With latter, GO annotations that arise from sequence homology provides no additional information compared to using sequence data to align networks. However, GO annotations may arise from other sources such as protein domain knowledge. Furthermore, human curated annotations provide additional functional evidence.

Compared to network based approach, our approach considers both structural and functional information in defining subgraphs. Network based methods rely purely on the assumption that densely connected proteins are likely to share similar function, and vice versa. However, there are cases where this assumption does not hold. In regulatory and signaling pathways, for example, proteins share similar function without being densely connected. An example is the MAPKKK cascade pathway. Such subgraphs can be uncovered via GO annotations but not topology alone because it is sparsely connected compared to protein complexes.

²The first step region-to-region alignment represent function constraints that restrict the second step protein-to-protein alignment.

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

To realize the function-constrained network alignment problem, we propose an algorithm called **DualAligner** that utilizes the aforementioned dual alignment strategy, where region-to-region alignments are first made followed by detailed protein-to-protein alignment. Region-to-region alignment establishes high-level functional connections between the networks, while protein-to-protein alignment specifies the detailed connections within them. We demonstrate the utility and superiority of **DualAligner** over state-of-the-art global network alignment techniques using real-world PPI networks in Section 6.3.

6.2 Problem Formulation

In this section, we formally introduce the function-constrained network alignment problem. We begin by defining some terminology that we shall be using in the sequel.

6.2.1 Terminology

Let $G = (V, E, \omega)$ be a PPI network where an edge $e \in E$ has a positive real weight ω that represents its interaction strength. Recall that the *term association vector* of $v \in V$, denoted by Δ_v , is defined as $\Delta_v = \langle I_v(\Delta_1), I_v(\Delta_2), \dots, I_v(\Delta_n) \rangle$, $I_v(\Delta_i) \in \{0, 1\}$, such that $I_v(\Delta_i) = 1$ if and only if the term Δ_i or its descendants are associated with protein v . Otherwise, $I_v(\Delta_i) = 0$. Note that Δ_v indicates GO terms that are associated with v .

A *function-constrained subgraph* $K_j = (V_j^K, E_j^K, \Delta_i)$ of G is a subgraph with the following properties: (a) it is a connected subgraph; and (b) every node $v \in V_j^K$ shares a GO annotation Δ_i , i.e. $\forall v \in V_j^K, I_v(\Delta_i) = 1$. A function-constrained subgraph (which we refer to as a *subgraph constraint* for brevity) represents a constraint on a connected region of G that shares at least one functional role, namely the role represented by the term Δ_i . For

example, in Figure 6.1, the subgraph constraint K_1 represents a connected subgraph of the human network such that every protein in it shares the **transcription elongation** annotation.

A *subgraph set* $S_i = \{K_1, K_2, \dots, K_m\}$ represents m subgraph constraints on G . Note that typically $\bigcap_{K_j \in S_i} V_j^K \neq \emptyset$ and $\bigcup_{K_j \in S_i} V_j^K \neq V$, *i.e.*, subgraphs in S_i may not form a partition on G and are highly overlapping. The overlapping nature of subgraph constraints do not indicate conflicting functions of its proteins, but rather the multi-attribute and multi-role nature of proteins. Figure 6.1(b) shows a sample subgraph set of function-constrained subgraphs for the fly and human PPI networks. Observe that the subgraph constraints may overlap (*e.g.*, K_1 and K_3).

6.2.2 Region-to-Region Alignment

Given two PPI networks G_1 and G_2 with $|V_1| \leq |V_2|$, a *region-to-region alignment* between G_1 and G_2 is an injective function $rm : \mathcal{F}_1 \rightarrow \mathcal{F}_2$ that maps $K_m \in S_1$ (a function-constrained subgraph in G_1) to $K_n \in S_2$ (another function-constrained subgraphs in G_2). Figure 6.1(c) shows a region-to-region alignment of three pairs of function-constrained subgraphs. The **transport** function-constrained subgraph (K_3) in the human network is aligned to the **microtubule binding** subgraph (K_4) in the fly network, indicating that these two regions are conserved.

To evaluate the conservation between function-constrained subgraphs during region-to-region alignment, it is necessary to consider protein-to-protein alignments within a pair of subgraphs. A protein-to-protein alignment between two subgraphs will reveal the extent of topological and functional similarities between them. To formalize this, we introduce the notion of *subalignment*. A subalignment is a subfunction $subm : V_i^s \rightarrow V_j^s$ of an alignment $Am : V_i \rightarrow V_j$ that aligns a subset $V_i^s \subset V_i$ of G_i to another subset $V_j^s \subset V_j$ of G_j . To illustrate this, consider in Figure 6.2 two toy PPI networks with nodes $V_i = \{1, 2, 3, 4, 5\}$ and $V_j = \{a, b, c, d\}$, respectively.

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Figure 6.2(a) shows a subalignment of nodes from $V_i^s = \{1, 2, 3\}$ to $V_j^s = \{b, c, d\}$. Notice that the subalignment aligns the function-constrained subgraph K_3 to K_4 . On the other hand, Figure 6.2(b) shows a subalignment from $\{1, 2, 3\}$ to $\{a, c, d\}$, which is not an alignment between two function-constrained subgraphs.

Definition 6.5 [Optimal Region-to-Region Alignment] *Given two PPI networks G_i and G_j , let $\mathbf{rm}$ be a region-to-region alignment function that aligns function-constrained subgraphs $K_m \in S_i$ of G_i to function-constrained subgraphs $K_n \in S_j$ of G_j . Let $subm_{mn}$ be the best protein-to-protein subalignment between K_m and K_n and $S(subm_{mn})$ be the score of the $subm_{mn}$. The **optimal region-to-region alignment problem** is then defined as the problem of identifying the injective $\mathbf{rm}$ that maximizes:*

$$S(\mathbf{rm}) = \sum_{K_m \in S_i, rm(K_m) \in S_j} S(subm_{mn}) \quad (6.1)$$

Here protein-to-protein subalignment $subm_{mn}$ between two subgraphs can simply be performed using any existing network alignment algorithms. The score of the alignment is then $S(subm_{mn})$. Thus, the problem can be reformulated as the problem of identifying the one-to-one mapping between function-constrained subgraph pairs that maximizes the total subalignment scores.

Figure 6.1(c) depicts a region-to-region alignment of three pairs of function-constrained subgraphs and Figure 6.1(d) shows the potential subalignments between these pairs. Consider the subalignment between the **transport** function-constrained subgraph (K_3) in the human network and the **microtubule binding** subgraph (K_4) in the fly network. The subalignment K_3 to K_4 is shown by the arrowed lines within the circled subgraph in Figure 6.1(d).

6.2.3 Function-Constrained Network Alignment Problem

Given two PPI networks G_1 and G_2 with $|V_1| \leq |V_2|$, a *protein-to-protein alignment* between G_1 and G_2 is an injective function $Am : V_1 \rightarrow V_2$ that maps each protein in the smaller network to another protein in the larger network. For each $x \in V_1$, let $y_j = Am(x_i)$ be its corresponding aligned protein in V_2 given by Am .

Intuitively, a region-to-region alignment shows the functional conservation between the regions of both networks in a coarse-grained manner. Given such a region-to-region alignment, any detailed protein-to-protein alignment between these networks must ensure that it is consistent with the aligned functional subgraph regions (from region-to-region alignment). In other words, a protein-to-protein alignment should be guided by these region-to-region alignments. In order to do that, we shall now establish and formalize several concepts that let region-to-region alignment serves as constraint to protein-to-protein alignment.

A subalignment *subm* is said to be *hard constrained* to a pair of function-constrained subgraphs $(K_u \in S_i, K_v \in S_j)$ iff $V_i^s \subseteq V_u^K$ and $V_j^s \subseteq V_v^K$ (recall that V_u^K and V_v^K are the vertices of subgraphs K_u and K_v , respectively). In this case, we refer to the pair of subgraphs as a *hard constraint*. Thus, the subalignment is strictly restricted within the subgraph regions specified by the hard constraint (K_u, K_v) . The subalignment in Figure 6.2(a) is hard constrained to (K_3, K_4) , while subalignment in Figure 6.2(b) is not.

Observe that it is computationally challenging to identify an alignment that satisfies a large number of (conflicting) hard constraints. Therefore, we introduce the notion of *probabilistic constraints* on subalignments. A *soft constraint* on a subalignment *subm* is a pair of function-constrained subgraphs (K_u, K_v) such that $\forall x \in V_i^s, P(subm(x) \in V_v^K) \geq p$ and $\forall x \in V_j^s, P(subm^{-1}(x) \in V_u^K) \geq p$. We refer to the subalignment as being *soft*

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

constrained to the (K_u, K_v) pair. A soft constraint on a subalignment restricts the alignment to the region specified by (K_u, K_v) *probabilistically*. We shall later define a *score* that is associated with the probability parameter p . Suppose the subalignment in Figure 6.2(b) is a subalignment that is soft constrained by (K_3, K_4) . For every protein in $x \in \{1, 2, 3\}$, it has probability of at least p to being mapped to a protein in $\{b, c, d\}$ (i.e., $\forall x \in V_i^s, P(\text{subm}(x) \in V_v^K) \geq p$). At the same time, for every protein in $x \in \{b, c, d\}$, it has probability of at least p to being mapped to a protein in $\{1, 2, 3\}$ (i.e., $\forall x \in V_j^s, P(\text{subm}^{-1}(x) \in V_u^K) \geq p$).

A *subgraph-subgraph pair* (or soft constraint) is a pair of subgraphs $f = (K_u, K_v)$, $K_u \in S_i$ and $K_v \in S_j$ such that there exists a subalignment subm of m that is soft constrained to f with probability $P(f)$. This subgraph pair models a single region-to-region pairing. Let $\mathcal{F} = \{f_1, f_2, \dots, f_K\}$ be the set of all subgraph-subgraph pairs, with each $f_k \in \mathcal{F}$ given a *certainty score* $s(f_k) \propto P(f_k)$. Here $s(f_k)$ reflects the likelihood that subgraph K_u is aligned to subgraph K_v , and is associated with the network *subalignment score* between K_u and K_v in terms of both sequence and topological conservation.

It is reasonable to remove the top levels of GO terms as they are likely to contribute little to the alignment. However, choosing the number of top levels to prune is not a trivial task, as the specificity of the terms do not uniformly increase with level (e.g., some terms at level 3 may be more specific than others). Instead of a hard cut-off, DualAligner implicitly weighs the effects of GO terms via subalignment scores $s(f)$. It relies on computing subalignment scores between GO term-associated subgraphs to determine its significance in functional conservation. Consider, for example, subalignment between GO term subgraphs associated with Biological process to Cellular component and Golgi vesicle transport to vacuolar transport. It is likely that the former subgraph pairs yield a poor subalignment score

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

$s(f)$ compared to the latter, which is likely to have higher percentage of conserved interactions and sequences. Because the alignment contribution of the subgraphs is weighted by $s(f)$, DualAligner only weakly consider the effect of the former pair and strongly consider the latter. Thus, it can effectively discard the contribution of non-specific terms in a data-driven manner.

We propose, with simplifying assumptions, a *constraint mixture model*, where the overall distribution is modeled by a weighted superposition of distributions under each constraint $f_k \in \mathcal{F}$. The *mixing weight* for a soft constraint f_k is given by $P(f_k) = \frac{s(f_k)}{\sum_f s(f)}$, that is, the probability that constraint f_k is restricting a subalignment of m . The probability of $x \in V_1$ assigned to $y \in V_2$ under the constraint mixture model is as follows: $P(y|x) = \sum_{f_k} P(y|x, f_k)P(f_k)$. The conditional independence assumption is made such that conditioned on f_k , the matches are independent. $P(y|x, f_k)$ is the probability that $x \in V_1$ is aligned to $y \in V_2$ given that f_k constrains alignment of this pair of protein. Let $subm_{ij}$ be a subalignment that is soft constrained to $f = (K_i, K_j)$. Let $subm_{ij}^{max}$ be the best one-to-one alignment obtained. $subm_{ij}^{max}$ can be attained using any existing network alignment algorithms. Suppose an alignment generates a scoring function $\sigma(x, y)$ for each pair of aligned proteins. Then, we let the certainty score of f be $s(f_k) = \sum_{x, subm(x)} 1 + \sigma(x, subm(x))$. Under this formulation, we note that $P(y|x, f_k) = 0$ in most matches compared to sequence similarity matrix.

Definition 6.6 [*Function-constrained Network Alignment Problem*] Given two PPI networks G_i and G_j and a region-to-region alignment $\mathbf{rm}$, let $\mathbf{Am}$ be a protein-to-protein alignment function that aligns G_i to G_j constrained by soft constraints in $\mathbf{rm}$. Let $P(y_j|x_i, f_k) = \sigma(x_i, y_j)$ if $f_k = (K_i, K_j) \in \mathbf{rm}$ and $(x_i, y_j) \in subm_{ij}^{max}$; $P(y_j|x_i, f_k) = 0$ if otherwise. The *function-constrained network alignment problem* is

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Algorithm 4 The DualAligner Algorithm

Input: PPI networks G_1 and G_2 , GO DAG D , user-defined parameter k , minimum fragment size min
Output: Alignment map A

```

1:  $S_1 \leftarrow \emptyset$ 
2:  $S_2 \leftarrow \emptyset$ 
3: for  $\Delta_i \in D, i = 1, 2, \dots, d$  do
4:    $G_1^i \leftarrow$  induced subgraph of  $G_1$  by vertices in  $\{I_v(\Delta_i) = 1 : v \in G_1\}$ 
5:    $G_2^i \leftarrow$  induced subgraph of  $G_2$  by vertices in  $\{I_v(\Delta_i) = 1 : v \in G_2\}$ 
6:    $T_i^1 \leftarrow$  partition of  $G_1^i$  into connected components, each forming a constraint subgraph  $K_i$ 
7:    $T_i^2 \leftarrow$  partition of  $G_2^i$  into connected components, each forming a constraint subgraph  $K_i$ 
8:   if  $|S_1 \cap T_i^1|/|T_i^1| \leq \beta_s$  and  $|T_i^1| \geq min$  then
9:      $S_1 \leftarrow S_1 \cup T_i^1$ 
10:  end if
11:  if  $|S_2 \cap T_i^2|/|T_i^2| \leq \beta_s$  and  $|T_i^2| \geq min$  then
12:     $S_2 \leftarrow S_2 \cup T_i^2$ 
13:  end if
14: end for
15:  $\mathcal{F} \leftarrow \emptyset$ 
16: for  $K_i \in S_1, K_j \in S_2$  do
17:    $f \leftarrow (K_i, K_j)$ 
18:    $\mathcal{F} \leftarrow \mathcal{F} \cup \{f\}$ 
19: end for
20:  $S \leftarrow$  subgraph constraints from  $\mathcal{F}$  based on highest  $s(f)$  scores
21:  $l \leftarrow 0$ 
22: while  $S \neq \emptyset$  or  $l < k$  do
23:    $f \leftarrow \max_{s(f)} S$ 
24:    $A_f \leftarrow \text{ALIGN}(K_i, K_j, \lambda_s)$  where  $(K_i, K_j) \in f$ 
25:   for  $(x, y) \in A_f$  do
26:      $P(y|x) \leftarrow P(y|x) + P(y|x, f)P(f)$ 
27:   end for
28:    $S \leftarrow S \setminus \{(K_1, K_2) \in S : \frac{|K_1 \cap K_i|}{|K_i|} \geq \gamma_s, \frac{|K_2 \cap K_j|}{|K_j|} \geq \gamma_s\}$ 
29:    $l \leftarrow l + 1$ 
30: end while
31:  $Sorted \leftarrow$  Sort  $(x, y)$  for each  $x \in V_1$  by  $P(y|x)$ 
32: while  $Sorted \neq \emptyset$  do
33:   Select best  $(x, y)$  from  $Sorted$  and add to  $A$ 
34:    $Sorted \leftarrow$  Sort  $(x, y)$  for each  $x \in V_1$  by  $P(y|x)$  for unpaired  $y$ .
35: end while
36:  $A \leftarrow \text{ALIGN}_{EXP}(G_1, G_2, A, \lambda_s)$ 
37: return  $A$ 

```

then defined as the problem of identifying the alignment function $\mathbf{Am}$ that maximizes:

$$P(\mathbf{Am}(\mathbf{x})|\mathbf{x}) = \prod_{x_i} \sum_{f_k} P(\text{Am}(x_i)|x_i, f_k)P(f_k) \quad (6.2)$$

6.2.4 The DualAligner Algorithm

We now present the *DualAligner* algorithm (Algorithm 1) that identifies an alignment that maximizes $P(\mathbf{Am}(\mathbf{x})|\mathbf{x})$ via the aforementioned dual alignment strategy. It is comprised of the following three phases.

Phase 1: Functional subgraph construction and ranking. Given two PPI networks G_1 and G_2 , the first phase of *DualAligner* identifies

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

and constructs the set of functional subgraphs S_1 and S_2 (Lines 1-14). It exhaustively identifies for each GO term the induced subgraphs of G_1 and G_2 that share that term. These subgraphs are then partitioned into a set of connected components and added to S_i (Lines 6-7). Following that, pairs of subgraph-subgraph are constructed as soft constraints, and the certainty score $s(f)$ is computed using the subalignment algorithm (discussed below). We introduce a pruning parameter β_s to remove near-duplicate functional subgraphs. While computing functional subgraphs, only non-redundant subgraphs are added to the set of functional subgraphs (Lines 8-13). In this study, we set $\beta_s = 0.9$ (remove subgraphs that share more than 90% of vertices). The user-defined parameter k is introduced to select only the top- k constraints for consideration in the next phase. These top- k constraints are selected based on their $s(f)$ scores.

Phase 2: Region-to-region alignment. In the second phase, the soft constraint pairs are ranked by their certainty scores and subalignment is performed (Lines 20-34). Given a pair of subgraph-subgraph constraint $f = (K_i, K_j)$, a subalignment constrained by (K_i, K_j) is an injective mapping of vertices from V_i^K to V_j^K . Here, we propose a subalignment algorithm to achieve this. A pair of protein $u \in V_i^K, v \in V_j^K$ is scored using the following scoring function: $s(u, v) = b(u, v) + \lambda_s \sigma(u, v)$ (Line 24). The function $b(x_i, y_j)$ measures the sequence similarity score for aligning protein u to v and is defined as $b(x_i, y_j) = (1 + e^{-bscr(x_i, y_j)})^{-1}$, where $bscr(x_i, y_j)$ is the BLAST bit-score between the two proteins. The function $\sigma(u, v)$ measures the topological score for aligning protein u to v and is defined as:

$$\sigma(u, v) = |NE \cap ME| \quad (6.3)$$

$$NE = \{(x, y) : x \in N(u), y \in N(v)\} \quad (6.4)$$

$$ME = \{(Am(x), Am^{-1}(y)) : Am(x) \in N(x), \quad (6.5)$$

$$Am^{-1}(y) \in N(v) \in E_2\} \quad (6.6)$$

where Am is function mapping currently aligned proteins and $N(x)$ are

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

the neighbors of protein x . The parameter $\lambda_s \in [0, 1]$ weighs the effect of sequence similarity versus structural similarity. Thus, $s(u, v)$ evaluates the match suitability between proteins u and v using a weighted linear combination of their BLAST bit-score similarity (representing sequence similarity) and their interaction neighborhood similarity (representing structural similarity).

The subalignment method performs a greedy seed and extension strategy similar to several existing network alignment techniques such as **MI-GRAAL**, **Græmlin** (The formal description of the subalignment algorithm is given in Algorithm 4). First, the best scoring pair of proteins (x, y) in V_i^K and V_j^K is identified as seed and added to alignment. Using this seed, an extension is performed by identifying the neighborhood of the seed, *i.e.*, the vertices in V_i^K and V_j^K which are adjacent to x and y , respectively. The pairs of proteins in this neighborhood sets are ranked according to their scoring function scores and added to the alignment. These steps are then repeated until the alignment is complete.

Observe that the pairwise subalignment step can be computationally expensive when subalignment is performed for every pair. Therefore, we introduce a pruning parameter γ_s to remove highly redundant functional subgraph-subgraph constraints (Line 28). After ranking each constraint f by their confidence score $s(f)$, we greedily perform subalignment on the highest scoring constraint and update the conditional probabilities (Lines 22-26). Following that, we prune the remaining sets of constraints to remove near duplicates, such that any remaining constraint whose overlap ratio with f is greater or equal to γ_s is removed, *i.e.*, given (K_i, K_j) , we remove $\{(K_1, K_2) \in S : \frac{|V_1^K \cap V_i^K|}{|V_i^K|} \geq \gamma_s, \frac{|V_2^K \cap V_j^K|}{|V_j^K|} \geq \gamma_s\}$ from the set S (Line 27). The procedure is then repeated until S is empty.

Phase 3: Expanded protein-to-protein alignment. In the final phase, we extend the coverage of protein-protein alignment beyond richly

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Table 6.1: Datasets.

Dataset	#nodes	#edges	Source
<i>H. sapiens</i>	9131	34362	IntAct [13]
<i>S. cerevisiae</i>	4768	40457	IntAct
<i>D. melanogaster</i>	3114	6472	IntAct

annotated regions. The strategy is to consider the top- k region-to-region subalignments as seeds, and then identify all remaining unaligned proteins and rank them pairwise by their scoring function $\sigma(u, v)$ score with reference to the seeds (Line 35). Following that, each ranked pair, starting from the highest scoring pair, are added to the alignment. Intuitively, we treat the alignment of annotated proteins in Phase 2 as high quality seeds. The expanded alignment then aligns the remaining unaligned proteins based on the topological conservation with respect to the seeds.

The worst-case time complexity of **DualAligner** is $O(\alpha(|V|) + |S|^2|V|^3)$.

6.3 Results

The **DualAligner** algorithm is implemented in Scala. We now present the experiments conducted to study the performance of **DualAligner** and report some of the results here. The experiments were conducted on a 1.66GHz Intel Core 2 Duo T5450 machine with 3GB memory. We align the PPI networks of the global human, fly, and yeast (Table 6.1).

Evaluation criteria. We use several criteria to evaluate the performance of **DualAligner**. We define the *coverage* of an alignment m between two networks G_1 and G_2 , denoted as $cov(m)$, as the fraction of protein pairs aligned by an alignment method. That is, $cov(m) = |m|/min(|V_1|, |V_2|)$, where $|m|$ is the number of protein pairs aligned in m . Observe that $cov(m) = 1$ if the alignment m is a one-to-one mapping of V_1 to V_2 .

To measure the structural similarity of an alignment m between two networks G_1 and G_2 with $cov(m) \leq 1$, we propose a modified version of the

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Figure 6.3: Selected region-to-region alignments showing highly conserved subgraphs between human and yeast networks. Note that there may not necessarily be an optimal protein-to-protein alignment between the subgraphs.

edge correctness (EC) measure [85, 160]. Given G_1 , G_2 and an alignment m between them, the EC measure is defined as:

$$E1SET = \{(x, y) : (x, y) \in E_1\} \quad (6.7)$$

$$E2SET = \{(Am^{-1}(x), Am^{-1}(y)) : (x, y) \in E_2\} \quad (6.8)$$

$$EC = \frac{|E1SET \cap E2SET|}{|E2SET|} \quad (6.9)$$

Observe that EC indicates the fraction of correctly aligned edges among the proteins that are aligned (E2SET is restricted by domain of Am). Unlike the standard edge correctness measure that is only useful for full coverage alignment, this modified measure can be used for not only global alignment results but also local alignments (*i.e.*, alignments with $cov(m) < 1.0$).

To measure the sequence similarity of an alignment m between G_1 and G_2 , we propose the *average normalized bit-score* measure as follows:

$$ANBS(G_1, G_2, m) = |m|^{-1} \sum_{(x,y) \in m} \frac{bitscr(x, y)}{(bitscr(x, x)bitscr(y, y))^{-1/2}} \quad (6.10)$$

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Figure 6.4: Performance of DualAligner (human vs yeast) protein-to-protein alignment; alignment of each method may not have the same coverage. With DualAligner, one can adjust the trade-off between alignment quality and coverage. The dashed vertical line indicate the portion of the DualAligner alignment that is aligned from region-to-region alignment.

Observe that it is simply the average normalized BLAST bit-score of the paired proteins in m . A high ANBS score implies that sequence homologs are well matched whereas low score implies that sequence homologs are not being matched.

We also use the gold standard dataset of homologous proteins from Homologene [161]. We consider the fraction of number of correctly matched Homologene protein pairs over the total number of matched and mismatched pairs. Typically, GO annotations would be used as gold standard dataset. However, since these datasets are incorporated in our algorithm, it could not be used.

Lastly, to measure the biological function quality of an alignment m ,

we utilize the *functional coherence* measure [162]:

$$\text{FC}(G_1, G_2, m) = |m|^{-1} \sum_{(x,y) \in m} \frac{|\Delta_x \cap \Delta_y|}{|\Delta_x \cup \Delta_y|} \quad (6.11)$$

where Δ_x and Δ_y are the GO terms associated with proteins x and y , respectively.

Identification of conserved regions through region-to-region alignment. We first demonstrate the importance of region-to-region alignment for network alignment. Recall that in a region-to-region alignment, pairs of subgraphs from both networks indicate conservation at a broader granularity. That is, the subgraph as whole is deemed to be functionally conserved with allowance for protein and topological differences. Here we evaluate several high scoring region-to-region alignments. A few of these conserved subgraphs are depicted in Figure 6.3. Observe that while several region-to-region alignments contains a high confidence one-to-one protein alignments, such as the conserved NAT1-ARD1-NATS acetyltransferase subgraph, other region-to-region alignments do not admit a proper one-to-one protein alignment. Furthermore, the COG complex, which plays an important role in intra-Golgi trafficking, is well conserved between yeast and human network.

The COG complex comprises eight distinct subunits, COG1 to COG8 in two lobes (COG1 to COG4 forms lobe A and COG5 to COG8 forms lobe B) [163]. Studies have shown that although yeast and mammalian COG has subunits that share no sequence homology, a similar interaction map has been reported. Although COG3 to COG8 have homologs in human and yeast, COG1, COG2 and COG7 share no homology. COG1, COG2, and COG7 are structurally unique to mammalian cells and are not homologous to their yeast counterpart [164], which imply that no individual pairing between COG1, COG2, and COG7 could exist. This partially explains the minor differences between the subgraphs and also justifying the need for region-to-region alignment in addition to individual protein-to-protein alignment. One work attempted

to speculate and understand why this is the case for COG1 and COG2 [165]. It is found that subunits of the COG complex undergo widely varying mutation rate. The authors found that COG1 and COG2 are less conserved even in vertebrates, and perhaps the more relaxed requirement or selective constraints on the function of these subunits allow significant evolution rate on these subunits compared to others. A region level alignment based on DualAligner can reveal such patterns, as it can be used to identify subunits which are conserved and subunits which are not. This could be used as predictive tool to further study the biological implications of such observations. Algorithms that rely on sequence homology would miss such patterns.

We also note that the proteasome core complex is conserved. This complex is known to be ubiquitous and highly conserved in eukaryotes. Other relationships can be inferred from well-aligned subgraphs. For instance, transport mechanisms are found to be highly conserved between human and yeast. In summary, well-aligned regions show important conservation relationships between complex-complex, complex-function, and function-process.

Global protein-to-protein alignment. Next, we compare the performance of DualAligner with the following state-of-the-art global network alignment methods: IsoRank [20], IsoRankN [91], MI-GRAAL [85], Natalie [166], PINALOG [167] and PISWAP [162]. We ran alignments between the global yeast and human networks as well as between the global fly and yeast networks. We set $min = 3$, $\lambda_s = 0.8$ and $\beta_s = 0.9$. We vary the parameter k to obtain alignments of varying coverage. For IsoRank and IsoRankN, we used the standard settings recommended by the tool with $\alpha = 0.8$. We use the default parameters for PINALOG, and for Natalie, we chose the setting with highest EC value. For MI-GRAAL, we enabled signatures, sequences and degrees cost matrices. To use PISWAP as an inde-

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Figure 6.5: Performance of DualAligner (fly vs yeast).

pendent method, we use the Hungarian method to obtain a best matching based on sequence homology as input. Our attempt to align the human and yeast global networks using PATH [160] has failed to complete within 24 hours. The scalability issues of PATH on large networks has been observed [85].

Figure 6.4 plots the results of the alignment between the global human and yeast networks. In each figure, the scores obtained using **DualAligner** are indicated by a line because we obtain multiple instances of network alignments with different coverage (by adjusting k). Observe that while **DualAligner** is not the best performer in either EC or bit-score measure (when compared to another method having the same coverage), it achieves the best balance of edge correctness and sequence similarity by being close to the best performer in each measure. **DualAligner** outperforms all meth-

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Figure 6.6: Effect of λ_s showing its role in controlling the trade-off between topology (EC) and sequence (bit-score) conservation.

ods except MI-GRAAL based on EC. It achieves relatively close EC score to MI-GRAAL. On the other hand, *DualAligner* significantly outperforms MI-GRAAL by several orders of magnitude based on average normalized bit-score. It also outperforms all methods in bit-score and functional coherence except IsoRankN (which has poor EC score). Hence, *DualAligner* provides better edge alignment quality together with more accurate matching in terms of their sequence similarity and functional coherence compared to other methods. Notice that at lower coverage, *DualAligner* even outperforms other methods in both measures. Thus, our approach can be used to obtain very high quality alignments (in both edge correctness and sequence similarity) if maximizing coverage is not a requirement. Finally, we observe that the Homologene match ratio mirrors the bit-score performance, with *DualAligner* being superior to all methods except IsoRankN and PINALOG.

Figure 6.5 plots the results of alignment between the global fly and yeast networks. Compared to other methods, *DualAligner* achieves the best balance of edge correctness and sequence similarity. Although MI-GRAAL has superior EC, the average bit-score similarity is several orders of magnitude weaker than our method. Furthermore, *DualAligner* is superior to IsoRank, IsoRankN, PISWAP and Natalie on both measures. PINALOG

CHAPTER 6. DUALALIGNER: PROTEIN-PROTEIN INTERACTION NETWORK
ALIGNMENT VIA DUAL ALIGNMENT STRATEGY

Table 6.2: Running times (in seconds).

Networks	IsoRankN	IsoRank	MI-GRAAL	DualAligner	Natalie
Human PPI-Yeast PPI	23911	3950	38849	1831	3600
Fly PPI-Yeast PPI	5725	1050	31472	1192	3600

has superior bit-score and functional coherence but lower EC. More significantly, it suffers from poor coverage. Nevertheless, we urge caution in interpreting the quality of the protein-to-protein alignments. The global fly network is highly incomplete and has large gaps in interaction data (see Table 6.1). This could also explain the low coverage of PINALOG, which rely on topology based clustering to align the networks. The rapid drop in EC and bit-score similarities at higher coverage is indicative of this issue. In this case, a broad region-to-region alignment would be more suitable than inferring detailed protein-to-protein alignment out of the incomplete interaction data.

Finally, compared to other methods, **DualAligner** is able to assign confidence scores to matched protein pairs that correlate with alignment quality (by EC, bit-score and functional coherence measures). This is demonstrated by its ability to allow users to control the trade-off between alignment coverage and quality.

Effect of λ_s . The parameter λ_s controls the effect of sequence similarity versus topology similarity in the scoring model. We study the effect of λ_s on the alignments by varying λ_s from zero to 400. Figure 6.6 depicts the results. As λ_s increases, the EC value improves while average bit-score decreases. Thus, λ_s allows a trade-off between sequence similarity and topology similarity. At low values of λ_s , more emphasis in preserving high average bit-score alignment is observed. The greater the value λ_s , the greater the effect of topology similarity. This effect, however, converges to a steady state at subsequently higher values.

Running times. Table 6.2 reports the running times of `DualAligner` compared to other methods. Observe that `DualAligner` outperforms the tested approaches. Although `IsoRankN` is designed to scale to multiple networks, `IsoRank` significantly outperforms `IsoRankN` in alignment of a pair of global PPI networks. However, the running times of `IsoRank` increase much more rapidly with the size of the networks. In comparison, `DualAligner` scales better with the size of the networks. The `Natalie` method is designed to stop after 3600s.

6.4 Software Availability

Implementation of `DualAligner` can be found at the following:

- `DualAligner`: <http://www.cais.ntu.edu.sg/~assourav/DualAligner/>

6.5 Summary

In this chapter, we propose `DualAligner`, a network alignment algorithm that performs a dual alignment strategy, in which both region-to-region alignment (*i.e.*, whole subgraph of one network is aligned to subgraph of another) and protein-to-protein alignment (*i.e.*, individual proteins in networks are aligned to one another) are performed to achieve superior quality network alignment. Specifically, global alignment is achieved in `DualAligner` via the background information provided by a combination of GO annotation information and protein interaction data. We empirically demonstrate that our proposed approach outperforms state-of-the-art global network alignment techniques and demonstrates its ability to rank regions of alignment by their alignment quality.

Chapter 7

Conclusions and Future Work

In this chapter, we summarize the contributions of this thesis. We also establish several lines of inquiry associated with our research for future work.

7.1 Summary

The contributions of our research are summarized as follows:

- In Chapter 4, we propose **FUSE**, a novel data-driven and generic algorithm to generating functional summaries at multiple resolutions from a PPI to provide a high level view of its functional landscape. **FUSE** constructs higher level functional summary that summarizes the underlying PPI network to obtain a concise, interpretable representation of the network. It generates the “best” summary from both interaction and annotation data by maximizing information gain for a specific resolution. We demonstrate the role of **FUSE** in addressing the information overload issue of analyzing large scale PPI networks. We evaluate the performance of **FUSE** on several real-world PPIs. We also compare **FUSE** to state-of-the-art graph clustering methods with GO term enrichment by constructing the biological process landscape of the PPIs. Our experimental results demonstrate that **FUSE** is highly effective in constructing higher order functional maps with superior

CHAPTER 7. CONCLUSIONS AND FUTURE WORK

accuracy and representativeness compared to these state-of-the-art graph clustering methods. Using AD network as our case study, we further demonstrate the ability of **FUSE** to quickly summarize the network and identify many different processes and complexes that regulate it. We analyze the topological features of the functional landscape of human PPI that leads us to the identification of *functional hubs* (clusters of proteins that act as hubs). Finally, we propose **DiffNet**, a summarization method based on **FUSE** that summarizes pertinent functional differences between two **E-MAP** networks under contrasting conditions.

- We propose **FACETS**, a data-driven and generic algorithm for generating multi-faceted functional summarization of a PPI network, providing multiple perspectives of the functional organization landscape of the network (Chapter 5). Each perspective (facet) in the atlas represents a distinct interpretation of how the network can be functionally summarized. Our algorithm maximizes interpretative value of the atlas by optimizing inter-facet orthogonality and intra-facet cluster modularity. We tested our algorithm on the global networks from *IntAct*, and compared it to gold standard datasets from MIPS and KEGG. We demonstrated the performance of **FACETS**. We also performed a case study that illustrates the utility of our approach. Our experimental validation with real-world PPI networks demonstrates effectiveness of **FACETS** in generating functionally distinctive facets. These distinctive facets have higher relevance to real life datasets compared to single decomposition-based graph clustering techniques. We also show that our method converges rapidly to a solution with varying datasets and facet counts.
- In this Chapter 6, we propose **DualAligner**, a network alignment algorithm that performs a dual alignment, in which both region-to-

CHAPTER 7. CONCLUSIONS AND FUTURE WORK

region alignment – where whole subgraph of one network is aligned to subgraph of another – and protein-to-protein alignment – where individual proteins in networks are aligned to one another – are performed to achieve higher accuracy network alignments. Dual network alignment global alignment is achieved in **DualAligner** via the background information provided by a combination of GO annotation information and protein interaction data. We show that our approach outperforms current methods in global protein-to-protein alignment and demonstrates its ability to rank regions of alignment by their alignment quality. The superiority of our approach is shown using well-established metrics. By adjusting the user-defined parameter λ_s , one can tweak the impact of sequence conservation versus topology conservation. We also show that the effects of parameters in **DualAligner** in controlling the quality of the alignment.

7.2 Future Work

In general, network alignment and summarization approaches have largely been limited to static PPI networks. Where previously biologists have concentrated on large scale static networks, there is now a surge of interest in constructing large scale quantitative “-omics” data, including quantitative PPI and gene interaction networks. In light of this, a common theme in our proposed future work is extending the notion of network summarization and alignment to more sophisticated quantitative models. We begin by briefly discussing the current limitations of our work. Following that, we suggest several potential future work for addressing these limitations.

7.2.1 Extending Beyond Functional Information

As the proposed algorithms rely on functional information, one limitation of our algorithms (**FUSE**, **FACETS** and **DiffNet**) is their inability to summarize

regions without functional information. This could be important when there is a species network with many proteins of unknown functions. To address this, one future work is to develop summarization algorithms with the ability to extend to such regions.

7.2.2 Quantitative Network Summarization

In Chapter 4, we propose **FUSE**, a novel algorithm for summarizing PPI networks. PPI networks are static and limited in its power to model the complex behavior of the biological system. In fact, biological systems cannot be accurately modeled as static networks; they are dynamic and respond to both environmental and genetic factors. Therefore, quantitative models that can incorporate the dynamic properties of biological systems are increasingly important. Among existing quantitative models are *ordinary differential equations* (ODEs) and *partial differential equations* (PDEs) models. To this end, we proposed the **DiffNet** method as a first step towards quantitative network summarization (Chapter 4). One no longer treat PPI networks as static models, but dynamic models that change under contrasting conditions. The **DiffNet** approach aims to capture and summarize these dynamic regions. However, **DiffNet** is limited to binary snapshots of the network and will not easily extend to systems that model a continuum of states (*e.g.*, ODE and PDE models). As part of future work, there is opportunity to extend the notion of quantitative network summarization to more powerful ODE and PDE quantitative models. As far as we know, no such work exists. The complexity associated with understanding ODE and PDE models is well known, and extending network summarization to such quantitative models may help researchers better visualize their dynamical behavior in a multi-perspective manner. This is a challenging problem that requires careful investigation.

7.2.3 Quantitative Network Alignment

Similar to earlier proposal, a future goal in network alignment is supporting multi-resolution alignment of more powerful quantitative models. In fact, there is a dearth of research in the alignment between ODE or PDE models. With increasingly sophisticated experimental methods for constructing ODE and PDE models, we foresee the opportunity – and importance – of aligning ODE and PDE networks as future work. A careful study has to be conducted to understand the subtleties of aligning dynamic models.

7.2.4 Scalable Multi-resolution Network Alignment

In Chapter 6, we propose the concept of dual alignment in biological network alignment. Instead of simply constructing a fine detailed alignment between a pair of networks, we first align regions of one network to another in a coarse detailed manner. Such alignment is guided by existing knowledge in protein annotations. The coarse region-to-region alignment then serves as background information to improve the quality of the subsequent protein-to-protein alignment. The method proposed is limited to aligning pairs of biological networks. It is also limited to two levels of detail. Hence, we plan to design a annotation-guided, multi-resolution alignment approach that scales to multiple networks as well as alignment at multiple resolutions.

7.3 Conclusions

In this thesis, we have presented a framework for multi-resolution *and* multi-perspective summarization and alignment of biological models. Core to our proposed methods is the utilization of existing annotation knowledge to guide multi-resolution and multi-perspective summarization and alignment. Our research is applied to real-world biological datasets and compared to existing state-of-the-art methods. We demonstrated the strengths

CHAPTER 7. CONCLUSIONS AND FUTURE WORK

of our approaches in allowing biological researchers more power to understand system-wide models in a multi-level manner. The contributions of our research lays out paths for potential future work.

Appendix A

Differential Functional Summarization

In this section, we redefine concepts in **FUSE** to cater specifically to differential functional summarization of an **E-MAP** gene interaction network under two conditions. . Every gene $v \in V$ is annotated with zero or more biological functions in Δ . Then a *functional subgraph*, denoted by $C_T = (V_T, E_T)$, is a subgraph of G_d such that: (a) C_T is a subgraph of G_d induced by V_T , and (b) every gene $v \in V_T$ shares a function $T \in \Delta$. For instance, the subgraph $C1$ in Figure 4.24(a) is a functional subgraph of genes sharing the **DNA repair** function. One can see that a functional subgraph models the interaction responses of genes with a specific function as a whole.

We evaluate each functional subgraph C_T with the *skewness* and *coherence* measures. We say that a functional subgraph is *skewed* if its interactions significantly respond to condition change (*i.e.*, the interactions in the subgraph are significantly positive or negative differential). Analogous to individual gene interactions, we call a subgraph $C_T = (V_T, E_T)$ *positively skewed* if the sum of its edge weights, defined as $skew(C_T) = \sum_{e \in E_T} w_d(e)$, is greater than 0; it is *negatively skewed* if the sum of its edge weights is less than 0, *i.e.*, $skew(C_T) < 0$. The greater the value of $skew(C_T)$, the more the interactions of C_T respond to condition change.

We say that a functional subgraph is *coherent* if its interactions are

CHAPTER A. DIFFERENTIAL FUNCTIONAL SUMMARIZATION

largely skewed in one direction (either positive or negative differential). Figure 4.24(a) depicts the coherence of subgraphs of the toy network in Figure A.1. Consider the subgraph representing **DNA repair** function. It is coherent because it consists of interactions that are skewed towards positive differential in tandem. Intuitively, this would mean that the **DNA repair** function, as a whole, has increased alleviating response due to the condition change. Meanwhile, the subgraph representing **transport** has a mix of positive and negative differential interactions. There is no clear indication whether the **transport** function is positively or negatively affected by the condition change. We now formally define the notion of *subgraph coherence*. Given a subgraph C_T , $\text{coherence}(C_T) \in [0, 1]$ is given by:

$$\text{coherence}(C_T) = \frac{\max(|\{e : w_d(e) > 0\}|, |\{e : w_d(e) < 0\}|)}{|E_T|} \quad (\text{A.1})$$

The greater the value of $\text{coherence}(C_T)$, the more coherent is the subgraph. If $\text{coherence}(C_T) = 1$ then it indicates that all interactions are exclusively positive differential or exclusively negative differential.

Figure 4.24(a) depicts the skewness and coherence of several functional subgraphs. Each bar graph associated with a functional subgraph depicts the differential weight w_d values of the interactions in the subgraph. A high coherence and high skew subgraph has interactions with large w_d values in one direction. On the other hand, a low coherence and low skew subgraph has low w_d values in diverging directions. Consider the following two functional subgraphs: the subgraph of genes sharing the **DNA repair** function (**RAD5**, **RAD52**, **SIN3**, **ASH1**), and subgraph of genes sharing the **transport** function (**MSN1**, **ASH1**, **MRC1**, **PPH3**, **PSY4**, **PSY2**). Observe that interactions in the former are positive differential and skewed in one coherent direction, while the latter is not. We are more interested in the former type of subgraphs because it represents a concerted and significant functional response due to the condition change. Generally, functional subgraphs that are high skew and high coherence are informative and represent significant

CHAPTER A. DIFFERENTIAL FUNCTIONAL SUMMARIZATION

functional responses due to condition change. On the other hand, a subgraph with both low coherence and skew represent function that remain relatively unchanged.

From a statistical point of view, a module constructed from randomly drawn interactions will have a coherence score equal to the mean of a hypergeometric distribution centered around 0 (zero coherence). A high coherence module represents a module that is highly unexpected based on this distribution, thus representing a statistically significant module. Meanwhile, a low coherence module is no more significant than any module with randomly drawn interactions. Biologically, analogous to functional enrichment in gene lists, the statistical significance of high entropy modules means that the function associated with such module exhibit statistically significant interaction response patterns compared to a random function.

Based on the above observation, if one can decompose G_d into a set of highly coherent and skewed functional subgraphs, denoted by $\mathcal{S} = \{C_{T^1}, C_{T^2}, \dots, C_{T^k}\}$, then one can meaningfully obtain a summary representing positive and negative functional responses of G_d due to condition change. We shall later describe how one quantifies the decomposition of G_d based on the coherence and skewness of its functional subgraphs. Consider the decomposition depicted in Figures 4.24(b)-(c). The network of differential interactions is summarized into a set of functional subgraphs representing the following functional responses – **DNA repair** (positive), **response to radiation** (positive), **DNA integrity checkpoint** (negative) and **pseudohyphal growth** (negative). Each subgraph is coherent and skewed towards either positive or negative differential response.

At this point, it remains unclear how to optimally decompose G_d into a set of coherent and skewed functional subgraphs. To contrast with the previous example, suppose we decompose G_d into $\mathcal{S} = \{\text{transport (MSN1, ASH1, MRC1, PPH3, PSY4, PSY2)}, \text{response to radiation (MRC1, PPH3,}$

Figure A.1: A toy differential network of gene interactions.

$\text{PSY4}, \text{PSY2})\}$. This decomposition poorly summarizes the network in Figure A.1 because a significant portion of differential interactions are not captured by the subgraphs in $\mathcal{S}$. The `transport` subgraph also has low coherence.

A.0.1 Differential Summarization Problem

Given the existence of potentially many possible decompositions of G_d , the problem of *differential summarization* is to identify the *best* decomposition that represents the functional responses in G_d . Suppose we have a set containing all possible functional subgraphs of G_d . Let us denote this set by the universe $\mathcal{E}$. Clearly, some subgraphs will represent meaningful functional responses, while others will be unaffected by the condition change. One would like to choose a subset of $\mathcal{E}$ representing functional responses in G_d that are significantly affected by the condition change. To do this, we must first identify *summarization objectives* that assess the quality of a decomposition of G_d . We argue that a good decomposition of G_d should have the following desirable summary objectives:

- **Subgraph Coherence and Skewness.** A decomposition $\mathcal{S}$ should comprise of functional subgraphs that are significantly coherent and skewed. Recall that our goal is to identify functional regions that significantly respond, either positively or negatively, to condition

CHAPTER A. DIFFERENTIAL FUNCTIONAL SUMMARIZATION

change. This directly correlates to having coherent and skewed functional subgraphs, and finding $\mathcal{S}$ that maximizes coherence and skewness of its functional subgraphs is desirable. The *differential score* of C_T combines the skewness and coherence of the subgraph as follows:

$$\text{differential}(C_T) = \text{coherence}(C_T)^\alpha \times \text{skew}(C_T) \quad (\text{A.2})$$

where $\alpha \geq 0$ is a parameter controlling the influence of coherence on the differential score. Note that $0 \leq \text{coherence}(C_T)^\alpha \leq 1$.

- **Edge Coverage.** A good decomposition of G_d should convey key information regarding functional regions affected by condition change. It is natural to prefer a decomposition that covers as much differential interactions in G_d as possible. We introduce the *edge coverage* measure that reflects how well $\mathcal{S}$ represents the differential interactions of G_d . Formally, the *edge coverage* of $\mathcal{S}$ can be expressed as:

$$\text{coverage}(\mathcal{S}) = \frac{|\bigcup_{C_i \in \mathcal{S}} E_i|}{|E|} \quad (\text{A.3})$$

Intuitively, it indicates the percentage of interactions in G_d that is represented by the subgraphs in $\mathcal{S}$. The wider the coverage, the more representative is the decomposition of the interactions in G_d .

- **Distinctiveness.** Intuitively, two functional subgraphs having disjoint differential interactions is more informative than two redundant subgraphs with identical interactions. Thus, one prefers a decomposition which cleanly partitions G_d into distinctive sets of interactions. We quantify this objective with the *distinctiveness* measure. It quantifies redundancy of functional subgraphs, such that the greater the redundancy, the lower the distinctiveness value. Hence, distinctiveness of $\mathcal{S}$ is 1 if its subgraphs are mutually disjoint. Formally, it is defined as:

$$\text{distinctiveness}(\mathcal{S}) = \frac{|\bigcup_{C_i \in \mathcal{S}} E_i|}{\sum_{C_i \in \mathcal{S}} |E_i|} \quad (\text{A.4})$$

CHAPTER A. DIFFERENTIAL FUNCTIONAL SUMMARIZATION

We introduce an optimization model that selects functional subgraphs to maximally cover the set of differential interactions of G_d to maximize the above objective scores. This optimization model can be posed as a *weighted k -set cover* problem [143] of choosing a subset $\mathcal{S} \subseteq \mathcal{E}$ and a set of *remainder subgraphs* $\mathcal{R}$ with cardinality constraint k that minimizes the reciprocal of $\text{differential}(\mathcal{S})$. A remainder subgraph $R = (V_R, E_R) \in \mathcal{R}$ is a subgraph of G that is not part of the summary (*i.e.*, $R \cap C_T = \emptyset$ for all $C_T \in \mathcal{S}$). We shall later introduce a penalty for having remainder subgraphs.

Definition A.7 [Differential summarization problem]. Let G_d be the differential network of two gene interaction networks, G_c and G_t , under different conditions. Let $U = \bigcup_{C_T \in \mathcal{E}} E_T$ be the universe of differential interactions in G_d where $\mathcal{E}$ is a set of all possible functional subgraphs C_T . The **differential summarization problem** is to identify the differential decomposition $\mathcal{S}$ of functional subgraphs and $\mathcal{R}$ of remainder subgraphs (representing unselected interactions) by solving the following optimization problem:

$$\arg \min_{\mathcal{S} \cup \mathcal{R}} f(\mathcal{S} \cup \mathcal{R}) = \arg \min_{\mathcal{S}} \sum_{C_T \in \mathcal{S}} \text{differential}^{-1}(C_T) + \sum_{R \in \mathcal{R}} r(R) \quad (\text{A.5})$$

$$\text{subject to} \quad E = \bigcup_{C_T \in \mathcal{S}} E_T \cup \bigcup_{R \in \mathcal{R}} E_R \quad (\text{A.6})$$

$$|\mathcal{S}| + |\mathcal{R}| \leq k \quad (\text{A.7})$$

where the $\text{differential}^{-1}(C_T)$ – the reciprocal of the coherence and skewness of C_T – is the cost associated with each functional subgraph $C_T \in \mathcal{S}$, and $r(R) = (|E_R| + 1) \max_{C_T \in \mathcal{E}} \text{differential}^{-1}(C_T)$ captures the penalty for not covering the edges of the network.

It can be proven that there is at most one remainder subgraph that can be selected, which is disjoint from all functional subgraphs in $\mathcal{S}$.

A.0.2 Proof on Remainder Subgraphs

We prove that there is at most a single remainder subgraph R , and the remainder subgraph does not overlap with $\mathcal{S}$.

Theorem A.2 *Suppose $\mathcal{S}_0 \cup \mathcal{R}_0$ is an optimal solution. Then $|\mathcal{R}_0| \leq 1$.*

Proof: We begin by assuming the contradiction that $|\mathcal{R}_0| > 1$. $\mathcal{R}_0$ covers $\bigcup_{R \in \mathcal{R}_0} V_R$ with cost

$$|\mathcal{R}_0| \max_{C_T \in \mathcal{E}} \text{differential}^{-1}(C_T) + (\max_{C_T \in \mathcal{E}} \text{differential}^{-1}(C_T)) \sum_{R \in \mathcal{R}_0} |V_R|.$$

We can replace $\mathcal{R}_0$ with a single remainder subgraph with a lower cost. Let $\mathcal{R}' = \{\bigcup_{R \in \mathcal{R}_0} V_R\}$. The single remainder subgraph $\mathcal{R}'$ covers the same set of vertices with lower cost and set cover cardinality.

Theorem A.3 *Suppose $\mathcal{S}_0 \cup \mathcal{R}_0$ is an optimal solution. It holds that $\bigcup_{C_T \in \mathcal{S}_0} V_T \cap \bigcup_{R \in \mathcal{R}_0} V_R = \emptyset$.*

Proof: Assume by contradiction that $\bigcup_{C_T \in \mathcal{S}_0} V_T \cap \bigcup_{R \in \mathcal{R}_0} V_R \neq \emptyset$. Let $\mathcal{R}' = \{\bigcup_{R \in \mathcal{R}_0} V_R \setminus \bigcup_{C_T \in \mathcal{S}_0} V_T\}$. $\mathcal{S}_0 \cup \mathcal{R}'$ covers the same set of vertices with lower cost.

Because of $r(R)$, the formulation penalizes a summary that provides low interaction coverage. Also, observe that in principle the above cost function penalizes functional subgraphs with low coherence or skewness scores. The decomposition $\mathcal{S}$ summarizes the key functional responses representing the differences between G_c and G_t . The cardinality constraint k controls the distinctiveness and coverage of the decomposition.

CHAPTER A. DIFFERENTIAL FUNCTIONAL SUMMARIZATION

Algorithm 5 DiffNet.

Input: $G_t = (V, E, w_t)$, $G_c = (V, E, w_c)$, Δ , k **Output:** $\mathcal{S}$

```

1: Let  $p_{\max} = 0$ 
2: for  $e \in E$  do
3:    $w_d(e) = (1 + e^{-\frac{w_t(e) - w_c(e)}{|w_c(e)|}}) - 0.5$ 
4: end for
5: Let  $G_d = (V, E, w_d)$ 
6: Let  $\mathcal{E} = \emptyset$ 
7: for  $T \in \Delta$  do
8:    $\mathcal{E} \leftarrow \mathcal{E} \cup \{C_T\}$ 
9: end for
10: Let  $\mathcal{S} = \emptyset$ 
11: repeat
12:    $\text{mincost} \leftarrow \infty$ 
13:    $\text{best} \leftarrow \emptyset$ 
14:   for all  $C_T = (V_T, E_T) \in \mathcal{E} \setminus \mathcal{S}$  do
15:      $\text{SelectedEdges} \leftarrow \bigcup_{C \in \mathcal{S}} E$ 
16:      $n \leftarrow |E_T \setminus \text{SelectedEdges}|$ 
17:      $f \leftarrow \text{differential}^{-1}(C_T)/n$ 
18:     if  $f < \text{mincost}$  and  $n > 0$  then
19:        $\text{mincost} \leftarrow f$ 
20:        $\text{best} \leftarrow \{C_T\}$ 
21:     end if
22:   end for
23:    $\mathcal{S} \leftarrow \mathcal{S} \cup \text{best}$ 
24: until  $|\mathcal{S}| > k$ 
25: return  $\mathcal{S}$ 

```

Appendix B

List of Publications

- **B.S. Seah**, S.S. Bhowmick, C.F. Dewey Jr., H. Yu. FUSE: towards multi-level functional summarization of protein interaction networks. In *Proceedings of 2nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pp. 2-11, 2011.
- **B.S. Seah**, S.S. Bhowmick, C.F. Dewey Jr., H. Yu. FUSE: a profit maximization approach for functional summarization of biological networks. *BMC Bioinformatics*, vol. 13(Suppl 3), p. S10, 2012.
- **B.S. Seah**, S.S. Bhowmick, C.F. Dewey Jr., H. Yu. FUSE: a system for data-driven multi-level functional summarization of protein interaction networks. In *Proceedings of 2nd ACM SIGHIT International Health Informatics Symposium*, pp. 847-850, 2012.
- **B.S. Seah**, S.S. Bhowmick, C.F. Dewey Jr. FACETS: multi-faceted functional decomposition of protein interaction networks. *Bioinformatics*, vol. 28(20), pp. 2624-2631, 2012.

CHAPTER B. LIST OF PUBLICATIONS

- **B.S. Seah**, S.S. Bhowmick, C.F. Dewey Jr. DualAligner: protein-protein interaction network alignment via dual alignment strategy. *Under revision in Bioinformatics Journal.*

References

- [1] R. T. Peterson, “Chemical biology and the limits of reductionism.,” *Nature chemical biology*, vol. 4, pp. 635–8, Nov. 2008.
- [2] A. C. Ahn, M. Tewari, C.-S. Poon, and R. S. Phillips, “The limits of reductionism in medicine: could systems biology offer an alternative?,” *PLoS medicine*, vol. 3, p. e208, May 2006.
- [3] C. J. Jeffery, “Moonlighting proteins: old proteins learning new tricks.,” *Trends in genetics : TIG*, vol. 19, pp. 415–7, Aug. 2003.
- [4] C. R. Smith, A. Dolezal, D. Eliyahu, C. T. Holbrook, and J. Gadau, “Ants (Formicidae): models for social complexity.,” *Cold Spring Harbor protocols*, vol. 2009, p. pdb.emo125, July 2009.
- [5] T. Ideker, T. Galitski, and L. Hood, “A new approach to decoding life: systems biology.,” *Annual review of genomics and human genetics*, vol. 2, pp. 343–72, Jan. 2001.
- [6] U. Sauer, M. Heinemann, and N. Zamboni, “Genetics. Getting closer to the whole picture.,” *Science (New York, N.Y.)*, vol. 316, pp. 550–1, Apr. 2007.
- [7] U. Alon, “Biological networks: the tinkerer as an engineer.,” *Science (New York, N.Y.)*, vol. 301, pp. 1866–7, Sept. 2003.

REFERENCES

- [8] D. Dotan-Cohen, S. Letovsky, A. A. Melkman, and S. Kasif, “Biological process linkage networks,” *PloS one*, vol. 4, p. e5313, Jan. 2009.
- [9] M. Dreze, “Evidence for network evolution in an Arabidopsis interactome map,” *Science (New York, N.Y.)*, vol. 333, pp. 601–7, July 2011.
- [10] A. Barabasi and R. Albert, “Emergence of scaling in random networks,” *Science (New York, N.Y.)*, vol. 286, pp. 509–12, Oct. 1999.
- [11] D. Ekman, S. Light, A. K. Björklund, and A. Elofsson, “What properties characterize the hub proteins of the protein-protein interaction network of *Saccharomyces cerevisiae*?,” *Genome biology*, vol. 7, p. R45, Jan. 2006.
- [12] T. Yamada and P. Bork, “Evolution of biomolecular networks: lessons from metabolic and protein interactions,” *Nature reviews. Molecular cell biology*, vol. 10, pp. 791–803, Nov. 2009.
- [13] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, R. Huntley, C. Kohler, J. Khadake, C. Leroy, A. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, B. Roechert, D. Thornycroft, Y. Zhang, R. Apweiler, and H. Hermjakob, “IntAct—open source resource for molecular interaction data,” *Nucleic acids research*, vol. 35, pp. D561–5, Jan. 2007.
- [14] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Roth-

REFERENCES

- berg, “A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*,” *Nature*, vol. 403, pp. 623–7, Feb. 2000.
- [15] P. Braun, M. Tasan, M. Dreze, M. Barrios-Rodiles, I. Lemmens, H. Yu, J. M. Sahalie, R. R. Murray, L. Roncari, A.-S. de Smet, K. Venkatesan, J.-F. Rual, J. Vandenhoute, M. E. Cusick, T. Pawson, D. E. Hill, J. Tavernier, J. L. Wrana, F. P. Roth, and M. Vidal, “An experimentally derived confidence score for binary protein-protein interactions,” *Nature methods*, vol. 6, pp. 91–7, Jan. 2009.
- [16] I. Iossifov, M. Krauthammer, C. Friedman, V. Hatzivassiloglou, J. S. Bader, K. P. White, and A. Rzhetsky, “Probabilistic inference of molecular networks from noisy data sources,” *Bioinformatics (Oxford, England)*, vol. 20, pp. 1205–13, May 2004.
- [17] G. D. Bader and C. W. V. Hogue, “An automated method for finding molecular complexes in large protein interaction networks,” *BMC bioinformatics*, vol. 4, p. 2, Jan. 2003.
- [18] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, “An efficient algorithm for large-scale detection of protein families,” *Nucleic acids research*, vol. 30, pp. 1575–84, Apr. 2002.
- [19] M. Kalaev, M. Smoot, T. Ideker, and R. Sharan, “NetworkBLAST: comparative analysis of protein networks,” *Bioinformatics (Oxford, England)*, vol. 24, pp. 594–6, Feb. 2008.
- [20] R. Singh, J. Xu, and B. Berger, “Global alignment of multiple protein interaction networks with application to functional orthology detection,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, pp. 12763–8, Sept. 2008.

REFERENCES

- [21] R. Sharan and T. Ideker, “Modeling cellular machinery through biological network comparison,” *Nature biotechnology*, vol. 24, pp. 427–33, Apr. 2006.
- [22] N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis, T. Punna, J. M. Peregrín-Alvarez, M. Shales, X. Zhang, M. Davey, M. D. Robinson, A. Paccanaro, J. E. Bray, A. Sheung, B. Beattie, D. P. Richards, V. Candien, A. Lalev, F. Mena, P. Wong, A. Starostine, M. M. Canete, J. Vlasblom, S. Wu, C. Orsi, S. R. Collins, S. Chandran, R. Haw, J. J. Rilstone, K. Gandi, N. J. Thompson, G. Musso, P. St Onge, S. Ghanny, M. H. Y. Lam, G. Butland, A. M. Altaf-Ul, S. Kanaya, A. Shilatifard, E. O’Shea, J. S. Weissman, C. J. Ingles, T. R. Hughes, J. Parkinson, M. Gerstein, S. J. Wodak, A. Emili, and J. F. Greenblatt, “Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*,” *Nature*, vol. 440, pp. 637–43, Mar. 2006.
- [23] B. P. Kelley, B. Yuan, F. Lewitter, R. Sharan, B. R. Stockwell, and T. Ideker, “PathBLAST: a tool for alignment of protein interaction networks,” *Nucleic acids research*, vol. 32, pp. W83–8, July 2004.
- [24] J. Dutkowski and J. Tiuryn, “Identification of functional modules from conserved ancestral protein-protein interactions,” *Bioinformatics (Oxford, England)*, vol. 23, pp. i149–58, July 2007.
- [25] M. A. Harris, J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G. M. Rubin, J. A. Blake, C. Bult, M. Dolan, H. Drabkin, J. T. Eppig, D. P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J. M. Cherry, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. Engel, D. G. Fisk, J. E. Hirschman, E. L. Hong, R. S. Nash, A. Sethuraman, C. L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Berardini, S. Mundodi,

REFERENCES

- S. Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dimmer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E. M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, N. de la Cruz, P. Tonellato, P. Jaiswal, T. Seigfried, and R. White, "The Gene Ontology (GO) database and informatics resource.," *Nucleic acids research*, vol. 32, pp. D258–61, Jan. 2004.
- [26] J. Lodish, H.F., Baltimore, D., Berk, A., Darnell, *Molecular cell biology*. WH Freeman New York, NY, 1995.
- [27] G. Scatchard, "THE ATTRACTIONS OF PROTEINS FOR SMALL MOLECULES AND IONS," *Annals of the New York Academy of Sciences*, vol. 51, pp. 660–672, May 1949.
- [28] T. S. Young and P. G. Schultz, "Beyond the canonical 20 amino acids: expanding the genetic lexicon.," *The Journal of biological chemistry*, vol. 285, pp. 11039–44, Apr. 2010.
- [29] F. Crick, "Central dogma of molecular biology.," *Nature*, vol. 227, pp. 561–3, Aug. 1970.
- [30] I. M. A. Nooren and J. M. Thornton, "Diversity of protein-protein interactions.," *The EMBO journal*, vol. 22, pp. 3486–92, July 2003.
- [31] V. Marshansky and M. Futai, "The V-type H⁺-ATPase in vesicular trafficking: targeting, regulation and function.," *Current opinion in cell biology*, vol. 20, pp. 415–26, Aug. 2008.
- [32] W. H. Landschulz, P. F. Johnson, and S. L. McKnight, "The leucine zipper: a hypothetical structure common to a new class of DNA binding proteins.," *Science (New York, N.Y.)*, vol. 240, pp. 1759–64, June 1988.
- [33] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki, "A comprehensive two-hybrid analysis to explore the yeast protein

REFERENCES

- interactome.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, pp. 4569–74, Apr. 2001.
- [34] L. A. Huber, “Is proteomics heading in the wrong direction?,” *Nature reviews. Molecular cell biology*, vol. 4, pp. 74–80, Jan. 2003.
- [35] O. Puig, F. Caspary, G. Rigaut, B. Rutz, E. Bouveret, E. Bragado-
Nilsson, M. Wilm, and B. Séraphin, “The tandem affinity purification (TAP) method: a general procedure of protein complex purification.,” *Methods (San Diego, Calif.)*, vol. 24, pp. 218–29, July 2001.
- [36] H. Schägger, “Tricine-SDS-PAGE.,” *Nature protocols*, vol. 1, pp. 16–22, Jan. 2006.
- [37] C.-D. Hu, Y. Chinenov, and T. K. Kerppola, “Visualization of interactions among bZIP and Rel family proteins in living cells using bimolecular fluorescence complementation.,” *Molecular cell*, vol. 9, pp. 789–98, Apr. 2002.
- [38] H. Huang and J. S. Bader, “Precision and recall estimates for two-hybrid screens.,” *Bioinformatics (Oxford, England)*, vol. 25, pp. 372–8, Mar. 2009.
- [39] S. Peri, J. D. Navarro, R. Amanchy, T. Z. Kristiansen, C. K. Jonnalagadda, V. Surendranath, V. Niranjana, B. Muthusamy, T. K. B. Gandhi, M. Gronborg, N. Ibarrola, N. Deshpande, K. Shanker, H. N. Shivashankar, B. P. Rashmi, M. A. Ramya, Z. Zhao, K. N. Chandrika, N. Padma, H. C. Harsha, A. J. Yatish, M. P. Kavitha, M. Menezes, D. R. Choudhury, S. Suresh, N. Ghosh, R. Saravana, S. Chandran, S. Krishna, M. Joy, S. K. Anand, V. Madavan, A. Joseph, G. W. Wong, W. P. Schiemann, S. N. Constantinescu, L. Huang, R. Khosravi-Far, H. Steen, M. Tewari, S. Ghaffari, G. C. Blobel, C. V. Dang, J. G. N. Garcia, J. Pevsner, O. N. Jensen, P. Roepstorff,

REFERENCES

- K. S. Deshpande, A. M. Chinnaiyan, A. Hamosh, A. Chakravarti, and A. Pandey, “Development of human protein reference database as an initial platform for approaching systems biology in humans.,” *Genome research*, vol. 13, pp. 2363–71, Oct. 2003.
- [40] C. Stark, B.-J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers, “BioGRID: a general repository for interaction datasets.,” *Nucleic acids research*, vol. 34, pp. D535–9, Jan. 2006.
- [41] I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg, “DIP: the database of interacting proteins.,” *Nucleic acids research*, vol. 28, pp. 289–91, Jan. 2000.
- [42] M. Kanehisa and S. Goto, “KEGG: kyoto encyclopedia of genes and genomes.,” *Nucleic acids research*, vol. 28, pp. 27–30, Jan. 2000.
- [43] G. D. Bader, I. Donaldson, C. Wolting, B. F. Ouellette, T. Pawson, and C. W. Hogue, “BIND—The Biomolecular Interaction Network Database.,” *Nucleic acids research*, vol. 29, pp. 242–5, Jan. 2001.
- [44] P. Pagel, S. Kovac, M. Oesterheld, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, P. Mark, V. Stümpflen, H.-W. Mewes, A. Ruepp, and D. Frishman, “The MIPS mammalian protein-protein interaction database.,” *Bioinformatics (Oxford, England)*, vol. 21, pp. 832–4, Mar. 2005.
- [45] D. Szklarczyk, A. Franceschini, M. Kuhn, M. Simonovic, A. Roth, P. Minguéz, T. Doerks, M. Stark, J. Muller, P. Bork, L. J. Jensen, and C. von Mering, “The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored.,” *Nucleic acids research*, vol. 39, pp. D561–8, Jan. 2011.
- [46] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D’Eustachio, E. Schmidt, B. de Bono, B. Jassal, G. R. Gopinath, G. R. Wu, L. Matthews,

REFERENCES

- S. Lewis, E. Birney, and L. Stein, “Reactome: a knowledgebase of biological pathways.,” *Nucleic acids research*, vol. 33, pp. D428–32, Jan. 2005.
- [47] P. D. Karp, C. A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahrén, S. Tsoka, N. Darzentas, V. Kunin, and N. López-Bigas, “Expansion of the BioCyc collection of pathway/genome databases to 160 genomes.,” *Nucleic acids research*, vol. 33, pp. 6083–9, Jan. 2005.
- [48] Biocarta, “BioCarta.”
- [49] F. Gnäd, S. Ren, J. Cox, J. V. Olsen, B. Macek, M. Orosi, and M. Mann, “PHOSIDA (phosphorylation site database): management, structural and evolutionary investigation, and prediction of phosphosites.,” *Genome biology*, vol. 8, p. R250, Jan. 2007.
- [50] H. Dinkel, C. Chica, A. Via, C. M. Gould, L. J. Jensen, T. J. Gibson, and F. Diella, “Phospho.ELM: a database of phosphorylation sites—update 2011.,” *Nucleic acids research*, vol. 39, pp. D261–7, Jan. 2011.
- [51] B. Raghavachari, A. Tasneem, T. M. Przytycka, and R. Jothi, “DOMINE: a database of protein domain interactions.,” *Nucleic acids research*, vol. 36, pp. D656–61, Jan. 2008.
- [52] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, “Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.,” *Nature genetics*, vol. 25, pp. 25–9, May 2000.
- [53] E. Camon, M. Magrane, D. Barrell, V. Lee, E. Dimmer, J. Maslen, D. Binns, N. Harte, R. Lopez, and R. Apweiler, “The Gene Ontology

REFERENCES

- Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology,” *Nucleic acids research*, vol. 32, pp. D262–6, Jan. 2004.
- [54] A. Hamosh, A. F. Scott, J. S. Amberger, C. A. Bocchini, and V. A. McKusick, “Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders,” *Nucleic acids research*, vol. 33, pp. D514–7, Jan. 2005.
- [55] D. W. Huang, B. T. Sherman, and R. A. Lempicki, “Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources,” *Nature protocols*, vol. 4, pp. 44–57, Jan. 2009.
- [56] T. Beissbarth and T. P. Speed, “GOstat: find statistically overrepresented Gene Ontologies within a group of genes,” *Bioinformatics (Oxford, England)*, vol. 20, pp. 1464–5, June 2004.
- [57] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, “Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 15545–50, Oct. 2005.
- [58] S. W. Doniger, N. Salomonis, K. D. Dahlquist, K. Vranizan, S. C. Lawlor, and B. R. Conklin, “MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data,” *Genome biology*, vol. 4, p. R7, Jan. 2003.
- [59] A. D. King, N. Przulj, and I. Jurisica, “Protein complex prediction via cost-based clustering,” *Bioinformatics (Oxford, England)*, vol. 20, pp. 3013–20, Nov. 2004.

REFERENCES

- [60] T. Nepusz, H. Yu, and A. Paccanaro, “Detecting overlapping protein complexes in protein-protein interaction networks.,” *Nature methods*, vol. 9, pp. 471–2, May 2012.
- [61] P. Jiang and M. Singh, “SPICi: a fast clustering algorithm for large biological networks.,” *Bioinformatics (Oxford, England)*, vol. 26, pp. 1105–11, Apr. 2010.
- [62] V. Spirin and L. A. Mirny, “Protein complexes and functional modules in molecular networks.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, pp. 12123–8, Oct. 2003.
- [63] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society.,” *Nature*, vol. 435, pp. 814–8, June 2005.
- [64] G. Liu, L. Wong, and H. N. Chua, “Complex discovery from weighted PPI networks.,” *Bioinformatics (Oxford, England)*, vol. 25, pp. 1891–7, Aug. 2009.
- [65] E. Georgii, S. Dietmann, T. Uno, P. Pagel, and K. Tsuda, “Enumeration of condition-dependent dense modules in protein interaction networks.,” *Bioinformatics (Oxford, England)*, vol. 25, pp. 933–40, Apr. 2009.
- [66] B. J. Frey and D. Dueck, “Clustering by passing messages between data points.,” *Science (New York, N.Y.)*, vol. 315, pp. 972–6, Feb. 2007.
- [67] D. A. Spielman and S.-H. Teng, “A Local Clustering Algorithm for Massive Graphs and its Application to Nearly-Linear Time Graph Partitioning,” Sept. 2008.

REFERENCES

- [68] K. Macropol, T. Can, and A. K. Singh, “RRW: repeated random walks on genome-scale protein networks for local cluster discovery,” *BMC bioinformatics*, vol. 10, p. 283, Jan. 2009.
- [69] V. Satuluri, S. Parthasarathy, and D. Ucar, “Markov clustering of protein interaction networks with improved balance and scalability,” in *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology - BCB '10*, (New York, New York, USA), p. 247, ACM Press, 2010.
- [70] Y.-K. Shih and S. Parthasarathy, “Identifying functional modules in interaction networks through overlapping Markov clustering,” *Bioinformatics (Oxford, England)*, vol. 28, pp. i473–i479, Sept. 2012.
- [71] G. Karypis and V. Kumar, “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,” *SIAM Journal on Scientific Computing*, vol. 20, pp. 359–392, Jan. 1998.
- [72] D. Dotan-Cohen, A. A. Melkman, and S. Kasif, “Hierarchical tree snipping: clustering guided by prior knowledge,” *Bioinformatics (Oxford, England)*, vol. 23, pp. 3335–42, Dec. 2007.
- [73] S. Navlakha, J. White, N. Nagarajan, M. Pop, and C. Kingsford, “Finding biologically accurate clusterings in hierarchical tree decompositions using the variation of information,” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 17, pp. 503–16, Mar. 2010.
- [74] C. G. Rivera, R. Vakil, and J. S. Bader, “NeMo: Network Module identification in Cytoscape,” *BMC bioinformatics*, vol. 11 Suppl 1, p. S61, Jan. 2010.

REFERENCES

- [75] Y. Park and J. S. Bader, “Resolving the structure of interactomes with hierarchical agglomerative clustering,” *BMC bioinformatics*, vol. 12 Suppl 1, p. S44, Jan. 2011.
- [76] W.-M. Song, T. Di Matteo, and T. Aste, “Hierarchical information clustering by means of topologically embedded graphs,” *PloS one*, vol. 7, p. e31929, Jan. 2012.
- [77] S. Asur, D. Ucar, and S. Parthasarathy, “An ensemble framework for clustering protein-protein interaction networks,” *Bioinformatics (Oxford, England)*, vol. 23, pp. i29–40, July 2007.
- [78] C. Kingsford and S. Navlakha, “Exploring biological network dynamics with ensembles of graph partitions,” in *Pac Symp Biocomput.*, pp. 166–77, 2010.
- [79] J. Vlasblom and S. J. Wodak, “Markov clustering versus affinity propagation for the partitioning of protein interaction graphs,” *BMC bioinformatics*, vol. 10, p. 99, Jan. 2009.
- [80] Kernighan B.W. and Lin S., “An efficient heuristic procedure for partitioning graphs,” *Bell Syst. Techn. J.*, vol. 49, no. 2, pp. 291–307, 1970.
- [81] M. Kalaev, V. Bafna, and R. Sharan, “Fast and accurate alignment of multiple protein networks,” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 16, pp. 989–99, Aug. 2009.
- [82] M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama, “Pairwise alignment of protein interaction networks,” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 13, pp. 182–99, Mar. 2006.

REFERENCES

- [83] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou, “Graemlin: general and robust alignment of multiple large interaction networks.,” *Genome research*, vol. 16, pp. 1169–81, Sept. 2006.
- [84] J. Dutkowski and J. Tiuryn, “Identification of functional modules from conserved ancestral protein-protein interactions.,” *Bioinformatics (Oxford, England)*, vol. 23, pp. i149–58, July 2007.
- [85] O. Kuchaiev and N. Przulj, “Integrative network alignment reveals large regions of global network similarity in yeast and human.,” *Bioinformatics (Oxford, England)*, vol. 27, pp. 1390–6, May 2011.
- [86] V. Memišević and N. Przulj, “C-GRAAL: common-neighbors-based global GRAPh ALignment of biological networks.,” *Integrative biology : quantitative biosciences from nano to macro*, vol. 4, pp. 734–43, July 2012.
- [87] H. T. T. Phan and M. J. E. Sternberg, “PINALOG: a novel approach to align protein interaction networks—implications for complex detection and function prediction.,” *Bioinformatics (Oxford, England)*, vol. 28, pp. 1239–45, May 2012.
- [88] R. A. Pache and P. Aloy, “A novel framework for the comparative analysis of biological networks.,” *PloS one*, vol. 7, p. e31220, Jan. 2012.
- [89] A. E. Aladag and C. Erten, “SPINAL: scalable protein interaction network alignment.,” *Bioinformatics (Oxford, England)*, Mar. 2013.
- [90] R. Y. Pinter, O. Rokhlenko, E. Yeger-Lotem, and M. Ziv-Ukelson, “Alignment of metabolic pathways.,” *Bioinformatics (Oxford, England)*, vol. 21, pp. 3401–8, Aug. 2005.

REFERENCES

- [91] C.-S. Liao, K. Lu, M. Baym, R. Singh, and B. Berger, “IsoRankN: spectral methods for global alignment of multiple protein networks.,” *Bioinformatics (Oxford, England)*, vol. 25, pp. i253–8, June 2009.
- [92] J. Berg and M. Lässig, “Cross-species analysis of biological networks by Bayesian alignment.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, pp. 10967–72, July 2006.
- [93] G. W. Klau, “A new graph-based method for pairwise global network alignment.,” *BMC bioinformatics*, vol. 10 Suppl 1, p. S59, Jan. 2009.
- [94] B.-S. Seah, S. S. Bhowmick, C. F. Dewey, and H. Yu, “FUSE: a profit maximization approach for functional summarization of biological networks.,” *BMC bioinformatics*, vol. 13 Suppl 3, p. S10, Jan. 2012.
- [95] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, “Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.,” *Nature genetics*, vol. 25, pp. 25–9, May 2000.
- [96] C. H. Wu, R. Apweiler, A. Bairoch, D. a. Natale, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, R. Mazumder, C. O’Donovan, N. Redaschi, and B. Suzek, “The Universal Protein Resource (UniProt): an expanding universe of protein information.,” *Nucleic acids research*, vol. 34, pp. D187–91, Jan. 2006.
- [97] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, a. Bridge, C. Derow, E. Dimmer, M. Feuermann, a. Friedrichsen, R. Huntley,

REFERENCES

- C. Kohler, J. Khadake, C. Leroy, a. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, B. Roechert, D. Thorneycroft, Y. Zhang, R. Apweiler, and H. Hermjakob, “IntAct–open source resource for molecular interaction data.,” *Nucleic acids research*, vol. 35, pp. D561–5, Jan. 2007.
- [98] E. I. Boyle, S. Weng, J. Gollub, H. Jin, D. Botstein, J. M. Cherry, and G. Sherlock, “GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes.,” *Bioinformatics (Oxford, England)*, vol. 20, pp. 3710–5, Dec. 2004.
- [99] I. Dhillon, Y. Guan, and B. Kulis, “A fast kernel-based multilevel algorithm for graph clustering,” in *Proceedings of the eleventh ACM SIGKDD international conference on knowledge discovery in data mining*, p. 634, ACM, 2005.
- [100] Y. Zhou, H. Cheng, and J. Yu, “Graph clustering based on structural/attribute similarities,” *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 718–729, 2009.
- [101] N. Wang, S. Parthasarathy, K.-L. Tan, and A. K. H. Tung, “CSV,” in *Proceedings of the 2008 ACM SIGMOD international conference on management of data - SIGMOD ’08*, (New York, New York, USA), p. 445, ACM Press, 2008.
- [102] A.-C. Gavin, M. Bösche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A.-M. Michon, C.-M. Cruciat, M. Remor, C. Höfert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M.-A. Heurtier, R. R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, and

REFERENCES

- G. Superti-Furga, “Functional organization of the yeast proteome by systematic analysis of protein complexes,” *Nature*, vol. 415, pp. 141–7, Jan. 2002.
- [103] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, pp. 465–471, Sept. 1978.
- [104] C. Huttenhower, E. M. Haley, M. A. Hibbs, V. Dumeaux, D. R. Barrett, H. A. Collier, and O. G. Troyanskaya, “Exploring the human genome with functional maps Exploring the human genome with functional maps,” *Genome Research*, pp. 1093–1106, 2009.
- [105] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, pp. 814–8, June 2005.
- [106] B. Adamcsek, G. Palla, I. J. Farkas, I. Derényi, and T. Vicsek, “CFinder: locating cliques and overlapping modules in biological networks,” *Bioinformatics (Oxford, England)*, vol. 22, pp. 1021–3, Apr. 2006.
- [107] Y. Tian, R. a. Hankins, and J. M. Patel, “Efficient aggregation for graph summarization,” in *Proceedings of the 2008 ACM SIGMOD international conference on management of data - SIGMOD '08*, (New York, New York, USA), p. 567, ACM Press, 2008.
- [108] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, “A model-based approach to attributed graph clustering,” in *Proceedings of the 2012 international conference on management of Data - SIGMOD '12*, (New York, New York, USA), p. 505, ACM Press, 2012.
- [109] S. Berchtold, C. Böhm, D. A. Keim, and H.-P. Kriegel, “A cost model for nearest neighbor search in high-dimensional data space,” *Symposium on Principles of Database Systems*, 1997.

REFERENCES

- [110] H. Kriegel, P. Kroger, M. Renz, and S. Wurst, *A Generic Framework for Efficient Subspace Clustering of High-Dimensional Data*. No. Icdm, IEEE, 2005.
- [111] P. K. Chan, M. D. F. Schlag, and J. Y. Zien, *Spectral K -way ratio-cut partitioning and clustering*. New York, New York, USA: ACM Press, 1993.
- [112] S. Khuller, A. Moss, and J. S. Naor, “The budgeted maximum coverage problem,” *Information Processing Letters*, vol. 70, no. 1, 1999.
- [113] T. S. Keshava Prasad, R. Goel, K. Kandasamy, S. Keerthikumar, S. Kumar, S. Mathivanan, D. Telikicherla, R. Raju, B. Shafreen, A. Venugopal, L. Balakrishnan, A. Marimuthu, S. Banerjee, D. S. Somanathan, A. Sebastian, S. Rani, S. Ray, C. J. Harrys Kishore, S. Kanth, M. Ahmed, M. K. Kashyap, R. Mohmood, Y. L. Ramachandra, V. Krishna, B. A. Rahiman, S. Mohan, P. Ranganathan, S. Ramabadran, R. Chaerkady, and A. Pandey, “Human Protein Reference Database–2009 update,” *Nucleic acids research*, vol. 37, pp. D767–72, Jan. 2009.
- [114] H. W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Münsterkötter, S. Rudd, and B. Weil, “MIPS: a database for genomes and protein sequences,” *Nucleic acids research*, vol. 30, pp. 31–4, Jan. 2002.
- [115] N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis, T. Punna, J. M. Peregrín-Alvarez, M. Shales, X. Zhang, M. Davey, M. D. Robinson, A. Pacanaro, J. E. Bray, A. Sheung, B. Beattie, D. P. Richards, V. Canadian, A. Lalev, F. Mena, P. Wong, A. Starostine, M. M. Canete, J. Vlasblom, S. Wu, C. Orsi, S. R. Collins, S. Chandran, R. Haw, J. J. Rilstone, K. Gandi, N. J. Thompson, G. Musso, P. St Onge,

REFERENCES

- S. Ghanny, M. H. Y. Lam, G. Butland, A. M. Altaf-Ul, S. Kanaya, A. Shilatifard, E. O'Shea, J. S. Weissman, C. J. Ingles, T. R. Hughes, J. Parkinson, M. Gerstein, S. J. Wodak, A. Emili, and J. F. Greenblatt, "Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*," *Nature*, vol. 440, pp. 637–43, Mar. 2006.
- [116] D. Crabtree, P. Andreae, and X. Gao, "QC4: a clustering evaluation method," in *Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'07)*, pp. 59–70, 2007.
- [117] M. Koyutürk, W. Szpankowski, and A. Grama, "Assessing significance of connectivity and conservation in protein interaction networks," *Journal of computational biology*, vol. 14, no. 6, pp. 747–64, 2007.
- [118] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," in *Biocomputing 2002 - Proceedings of the Pacific Symposium*, (Singapore), pp. 6–17, World Scientific Publishing Co. Pte. Ltd., 2001.
- [119] K. J. De Vos, A. J. Grierson, S. Ackerley, and C. C. J. Miller, "Role of axonal transport in neurodegenerative diseases," *Annual review of neuroscience*, vol. 31, pp. 151–73, Jan. 2008.
- [120] D. J. Owen and B. M. Collins, "Vesicle transport: a new player in APP trafficking," *Current biology : CB*, vol. 20, pp. R413–5, May 2010.
- [121] M. T. Lin and M. F. Beal, "Mitochondrial dysfunction and oxidative stress in neurodegenerative diseases," *Nature*, vol. 443, pp. 787–95, Oct. 2006.

REFERENCES

- [122] D. J. Selkoe, “Folding proteins in fatal ways,” *Nature*, vol. 426, pp. 900–4, Dec. 2003.
- [123] K. V. Kuchibhotla, S. T. Goldman, C. R. Lattarulo, H.-Y. Wu, B. T. Hyman, and B. J. Bacskai, “Abeta plaques lead to aberrant regulation of calcium homeostasis in vivo resulting in structural and functional disruption of neuronal networks,” *Neuron*, vol. 59, pp. 214–25, July 2008.
- [124] K. Herrup and Y. Yang, “Cell cycle regulation in the postmitotic neuron: oxymoron or new biology?,” *Nature reviews. Neuroscience*, vol. 8, pp. 368–78, May 2007.
- [125] R. a. C. M. Boonen, P. van Tijn, and D. Zivkovic, “Wnt signaling in Alzheimer’s disease: up or down, that is the question,” *Ageing research reviews*, vol. 8, pp. 71–82, Apr. 2009.
- [126] B. V. Zlokovic, “Neurovascular mechanisms of Alzheimer’s neurodegeneration,” *Trends in neurosciences*, vol. 28, pp. 202–8, Apr. 2005.
- [127] B. W. Doble, “GSK-3: tricks of the trade for a multi-tasking kinase,” *Journal of Cell Science*, vol. 116, pp. 1175–1186, Apr. 2003.
- [128] B. De Strooper and W. Annaert, “Where Notch and Wnt signaling meet. The presenilin hub,” *The Journal of cell biology*, vol. 152, pp. F17–20, Feb. 2001.
- [129] A. Patil and H. Nakamura, “Disordered domains and high surface charge confer hubs with the ability to interact with multiple proteins in interaction networks,” *FEBS letters*, vol. 580, pp. 2041–5, Apr. 2006.
- [130] N. Nelson, N. Perzov, a. Cohen, K. Hagai, V. Padler, and H. Nelson, “The cellular biology of proton-motive force generation by V-

REFERENCES

- ATPases.,” *The Journal of experimental biology*, vol. 203, pp. 89–95, Jan. 2000.
- [131] S. R. Collins, M. Schuldiner, N. J. Krogan, and J. S. Weissman, “A strategy for extracting and analyzing large-scale quantitative epistatic interaction data.,” *Genome biology*, vol. 7, p. R63, Jan. 2006.
- [132] R. P. St Onge, R. Mani, J. Oh, M. Proctor, E. Fung, R. W. Davis, C. Nislow, F. P. Roth, and G. Giaever, “Systematic pathway analysis using high-resolution fitness profiling of combinatorial gene deletions.,” *Nature genetics*, vol. 39, pp. 199–206, Feb. 2007.
- [133] T. M. Przytycka, M. Singh, and D. K. Slonim, “Toward the dynamic interactome: it’s about time.,” *Briefings in bioinformatics*, vol. 11, pp. 15–29, Jan. 2010.
- [134] T. Ideker and N. J. Krogan, “Differential network biology.,” *Molecular systems biology*, vol. 8, p. 565, Jan. 2012.
- [135] S. Bandyopadhyay, M. Mehta, D. Kuo, M.-K. Sung, R. Chuang, E. J. Jaehnig, B. Bodenmiller, K. Licon, W. Copeland, M. Shales, D. Fiedler, J. Dutkowski, A. Guénolé, H. van Attikum, K. M. Shokat, R. D. Kolodner, W.-K. Huh, R. Aebersold, M.-C. Keogh, N. J. Krogan, and T. Ideker, “Rewiring of genetic networks in response to DNA damage.,” *Science (New York, N.Y.)*, vol. 330, pp. 1385–9, Dec. 2010.
- [136] M. Schuldiner, S. R. Collins, J. S. Weissman, and N. J. Krogan, “Quantitative genetic analysis in *Saccharomyces cerevisiae* using epistatic miniarray profiles (E-MAPs) and its application to chromatin functions.,” *Methods (San Diego, Calif.)*, vol. 40, pp. 344–52, Dec. 2006.

REFERENCES

- [137] B. J. Frey and D. Dueck, “Clustering by passing messages between data points.,” *Science (New York, N.Y.)*, vol. 315, pp. 972–6, Feb. 2007.
- [138] T. Nepusz, H. Yu, and A. Paccanaro, “Detecting overlapping protein complexes in protein-protein interaction networks.,” *Nature methods*, vol. 9, pp. 471–2, May 2012.
- [139] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov, “Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 15545–50, Oct. 2005.
- [140] A. Subramanian, H. Kuehn, J. Gould, P. Tamayo, and J. P. Mesirov, “GSEA-P: a desktop application for Gene Set Enrichment Analysis.,” *Bioinformatics (Oxford, England)*, vol. 23, pp. 3251–3, Dec. 2007.
- [141] J. Gillis, M. Mistry, and P. Pavlidis, “Gene function analysis in complex data sets using ErmineJ.,” *Nature protocols*, vol. 5, pp. 1148–59, June 2010.
- [142] A. Jain and K. Mohiuddin, “Artificial neural networks: a tutorial,” *Computer*, vol. 29, pp. 31–44, Mar. 1996.
- [143] V. Chvatal, “A Greedy Heuristic for the Set-Covering Problem,” *Mathematics of Operations Research*, vol. 4, pp. 233–235, Aug. 1979.
- [144] P. Fabrizio, F. Pozza, S. D. Pletcher, C. M. Gendron, and V. D. Longo, “Regulation of longevity and stress resistance by Sch9 in yeast.,” *Science (New York, N.Y.)*, vol. 292, pp. 288–90, Apr. 2001.

REFERENCES

- [145] B. Zhang, B.-H. Park, T. Karpinets, and N. F. Samatova, “From pull-down data to protein interaction networks and complexes with biological relevance.,” *Bioinformatics (Oxford, England)*, vol. 24, pp. 979–86, Apr. 2008.
- [146] B.-S. Seah, S. S. Bhowmick, and C. F. Dewey, “FACETS: multifaceted functional decomposition of protein interaction networks.,” *Bioinformatics (Oxford, England)*, vol. 28, pp. 2624–31, Oct. 2012.
- [147] Z. Qi and I. Davidson, “A principled and flexible framework for finding alternative clusterings,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, (New York, New York, USA), p. 717, ACM Press, 2009.
- [148] D. Niu, J.G. Dy, and M. I. Jordan, “Multiple Non-Redundant Spectral Clustering Views,” in *Proceeding of the 27th International Conference on Machine Learning - ICML '10*, Haifa, Israel, 2010.
- [149] Y. Cui, X. Z. Fern, and J. G. Dy, *Non-redundant Multi-view Clustering via Orthogonalization*, vol. 3. IEEE, Oct. 2007.
- [150] C. C. Kiri Wagstaff, “Clustering with Instance-level Constraints,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.
- [151] N. N. Rich Caruana, Mohamed Elhawary, “Meta clustering,” in *IEEE International Conference on Data Mining*, 2006.
- [152] G. Agarwal and D. Kempe, “Modularity-maximizing graph communities via mathematical programming,” *The European Physical Journal B*, vol. 66, pp. 409–418, Nov. 2008.
- [153] C. Massen and J. Doye, “Identifying communities within energy landscapes,” *Physical Review E*, vol. 71, p. 046101, Apr. 2005.

REFERENCES

- [154] A. Jagota, “Approximating maximum clique with a Hopfield network,” *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 6, pp. 724–35, Jan. 1995.
- [155] Y. Botton, L., Bengio, “Convergence properties of the k-means algorithms,” in *In Advances in Neural Information Processing Systems* 7, 1994.
- [156] G. T. Hart, A. K. Ramani, and E. M. Marcotte, “How complete are current yeast and human protein-interaction networks?,” *Genome biology*, vol. 7, p. 120, Jan. 2006.
- [157] N. Mizushima, B. Levine, A. M. Cuervo, and D. J. Klionsky, “Autophagy fights disease through cellular self-digestion,” *Nature*, vol. 451, pp. 1069–75, Feb. 2008.
- [158] C. Behrends, M. E. Sowa, S. P. Gygi, and J. W. Harper, “Network organization of the human autophagy system,” *Nature*, vol. 466, pp. 68–76, July 2010.
- [159] I. Novak, V. Kirkin, D. G. McEwan, J. Zhang, P. Wild, A. Rozenknop, V. Rogov, F. Löhr, D. Popovic, A. Occhipinti, A. S. Reichert, J. Terzic, V. Dötsch, P. A. Ney, and I. Dikic, “Nix is a selective autophagy receptor for mitochondrial clearance,” *EMBO reports*, vol. 11, pp. 45–51, Jan. 2010.
- [160] M. Zaslavskiy, F. Bach, and J.-P. Vert, “Global alignment of protein-protein interaction networks by graph matching methods,” *Bioinformatics (Oxford, England)*, vol. 25, pp. i259–67, June 2009.
- [161] D. L. Wheeler, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, L. Y. Geer, W. Helmberg, Y. Kapustin, D. L. Kenton, O. Khovayko, D. J. Lipman, T. L. Madden, D. R. Maglott, J. Ostell, K. D.

REFERENCES

- Pruitt, G. D. Schuler, L. M. Schriml, E. Sequeira, S. T. Sherry, K. Sirotkin, A. Souvorov, G. Starchenko, T. O. Suzek, R. Tatusov, T. A. Tatusova, L. Wagner, and E. Yaschenko, “Database resources of the National Center for Biotechnology Information.,” *Nucleic acids research*, vol. 34, pp. D173–80, Jan. 2006.
- [162] L. Chindelevitch, C.-Y. Ma, C.-S. Liao, and B. Berger, “Optimizing a global alignment of protein interaction networks,” *Bioinformatics (Oxford, England)*, vol. 29, pp. 2765–73, Nov. 2013.
- [163] D. M. Walter, K. S. Paul, and M. G. Waters, “Purification and characterization of a novel 13 S hetero-oligomeric protein complex that stimulates in vitro Golgi transport.,” *The Journal of biological chemistry*, vol. 273, pp. 29565–76, Nov. 1998.
- [164] D. Ungar, T. Oka, E. E. Brittle, E. Vasile, V. V. Lupashin, J. E. Chatterton, J. E. Heuser, M. Krieger, and M. G. Waters, “Characterization of a mammalian Golgi-localized protein complex, COG, that is required for normal Golgi morphology and function.,” *The Journal of cell biology*, vol. 157, pp. 405–15, Apr. 2002.
- [165] R. Quental, L. Azevedo, R. Matthiesen, and A. Amorim, “Comparative analyses of the Conserved Oligomeric Golgi (COG) complex in vertebrates.,” *BMC evolutionary biology*, vol. 10, p. 212, Jan. 2010.
- [166] G. W. Klau, “A new graph-based method for pairwise global network alignment.,” *BMC bioinformatics*, vol. 10 Suppl 1, p. S59, Jan. 2009.
- [167] H. T. T. Phan and M. J. E. Sternberg, “PINALOG: a novel approach to align protein interaction networks—implications for complex detection and function prediction.,” *Bioinformatics (Oxford, England)*, vol. 28, pp. 1239–45, May 2012.